Handling User Cold Start Problem In Recommender Systems Using Fuzzy Clustering

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Computer Science and Engineering

Submitted By

Sugandha Gupta

(Roll No. - 801432028)

Under the supervision of

Dr. Shivani Goel

Assistant Professor (CSED)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

June 2016

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "***Handling User Cold Start Problem In Recommender Systems Using Fuzzy Clustering***," in partial fulfillment of the requirements for the award of degree of Master of Engineering in ***Computer Science and Engineering*** submitted in Computer Science and Engineering Department of Thapar University, Patiala is an authentic record of my own work carried out under the supervision of ***Dr. Shivani Goel*** and refers other researchers' work duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Sugandha Gupta

Sighature:

Sugandha Gupta
801432028
ME CSE

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Shivani Goel.

Dr. Shivani Goel
Assistant Professor,
CSED

Countersigned by

Dr. Maninder Singh
Head
Computer Science and Engineering Department
Thapar University
Patiala

Dr. S.S. Bhatia
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

No volume of words are sufficient to express gratitude towards my guide **Dr. Shivani Goel**, Computer Science & Engineering Department, Thapar University, Patiala, who has been very considerate and cooperative and guided whole-heartedly. She has been a continuous source of motivation, encouragement and kind support for preparing this thesis report. It's all because of my guide that I was able to explore such a vast and new topic for me in a limited time. She shared with me all the innovative ideas to do the research in a right direction and in an organized manner.

I am also thankful to **Dr. S. S. Bhatia**, Dean of Academic Affairs, **Dr. Maninder Singh**, Head of Computer Science & Engineering Department and **Dr. Ashutosh Mishra**, P.G. Coordinator, for providing all the facilities and harmonious environment for learning.

I would also like to thank my colleagues for extending all kind of help and cooperation during thesis.

Most importantly, I would like to thank my parents, my family members and the almighty for showing me the way out of darkness and for giving me strength and intelligence to carry out this research work. This work would not have been possible without their support and patience and for being with me throughout my journey of exploration.

Sugandha Gupta
801432028

ABSTRACT

The use of recommender systems has grown immensely in the recent years because the count of people using internet has grown at an enormous rate. Different websites have been successful in implementing recommender systems. Various techniques like collaborative filtering, content based filtering, knowledge based filtering etc have made recommendations easy and reliable. Yet these techniques face various challenges like cold start problem, scalability and sparsity issues. Cold start problem occurs when there is no sufficient rating information for a new user who enters the system. Thus no recommendations can be made for this user. Different approaches like k-means clustering, hierarchical clustering etc have been used to cluster users so that a new user could get recommendations based on this neighborhood. But k-means clustering fails when dataset size is huge, thus giving lower accuracy. In this thesis, a novel approach is introduced which implements fuzzy c-means clustering algorithm using RStudio to address user cold start problem. A comparison is made between fuzzy c-means clustering approach and the traditional k-means clustering approach based on different sizes of the user dataset. It is proved that as the number of users increase, fuzzy c-means clustering turns out to be a successful and a more accurate technique than k-means approach to generate better quality recommendations for the new user. Thus, solving user cold start problem.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

1.1 Recommender system

Recommender engines fall under the sub category of information filtering systems that aim to forecast preferences or ratings given to the item by the user. Technology has been influencing everybody's life. Therefore, recommender engines are now-a-days an integral portion of e-commerce sites which help in recommending items or products of interest to people all around the world. The major assistances of having a recommender systems are customer retention, information retrieval, personalization and many more. Also these systems can be used for recommendations of products such as music, books, restaurant, TV shows and movies. Presently these are used successfully in commercial websites such as Movielens, Amazon, MovieFinder, ebay, Linkedln, Jinni, Facebook and Myspace.

Basically, recommender systems compare the profile of a user to basic characteristics and try to generate ratings given by a user to items they hadn't well thought-out by the user. There are mainly two kinds of data groups or ratings under which the user profiles can be classified:

1. Explicit data / Explicit ratings: prompts a user to rate a given item on scale 1 to 5 and then group them accordingly. Thus asking the user to select the best item out of the given two and then forming a list of selected items he/she likes and finally ranking the collected items from most favorite to the least one [1].

2. Implicit data/Implicit ratings: works by perceiving the items that a user view, analyzing user's shopping trends and viewing times. Side by side, it maintains its records and then finally discovers similar likes and dislikes for the users [1].

1.2 Recommender system categories

Recommender engines are categorized into three basic categories based on the way recommendations are generated:

- Collaborative filtering
- Content-Based filtering
- Hybrid filtering

1.2.1 Collaborative filtering (CF) based recommender systems

This recommendation technique is also known as social filtering. The information is filtered by using the recommendations from different people as shown in figure 1. It works on the notion that people who agree with the evaluation of certain items or similar tastes or references in the past would agree in the future too. Thus, building a group or a neighborhood [2].

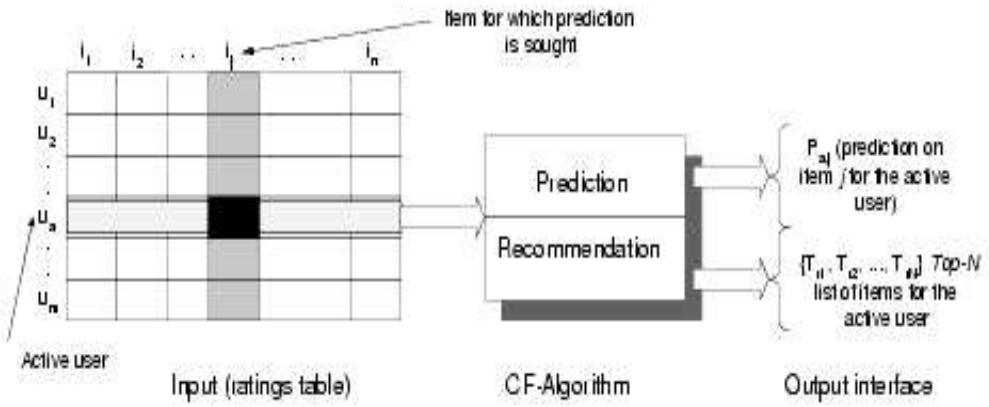


Figure 1 Collaborative filtering process [3]

Therefore, a user is given recommendations for the items that were not rated by him/her before, but users in his/her neighborhood already rated it positively. Figure 2 given below illustrates how the movies are being rated by all three users positively and with similar marks signifying that they are having identical taste and hence build a neighborhood. The movie "TRON: Legacy" is not being rated by user A, which perhaps mean that he hasn't watched the movie yet. Therefore, he will get the movie recommended as the item was positively rated by the other users. In contrast to comparatively simple recommender systems where recommendations are based on the most rated item or the most popular item methods, collaborative recommender

systems consider the taste of the user which can be consistent or can change slowly [4].

Movies / Users	Titanic	Gladiator	Black Swan	The Fighter	TRON: Legacy
A	8	7	9	10	-
B	9	7	9	9	10
C	9	8	9	8	9

Figure 2 Collaborative filtering example [4]

Further collaborative filtering is classified into two categories:

- Memory-based collaborative filtering techniques
  - User-based Approach
  - Item-based Approach
- Model-based collaborative filtering techniques

1.2.1.1 Memory-based collaborative filtering technique:

Memory based collaborative filtering technique is classified into two approaches namely user-based approach and item-based approach.

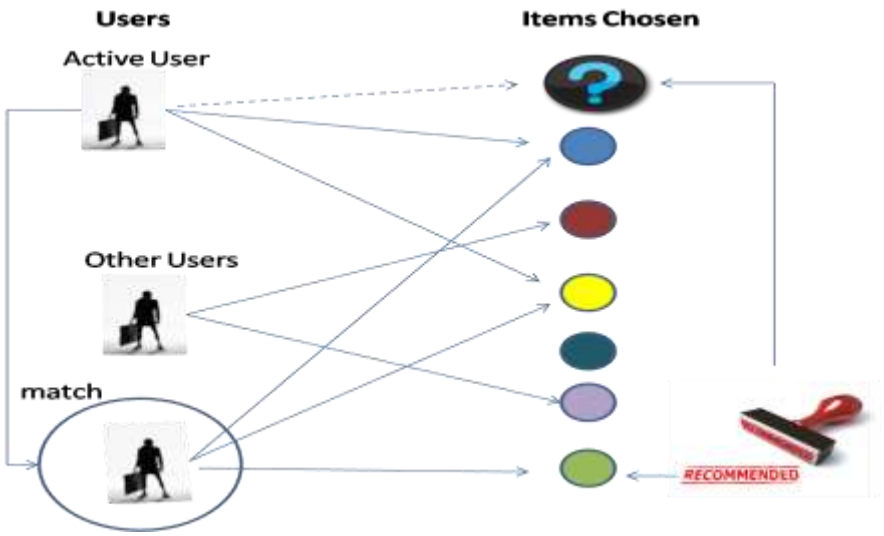


Figure 3 Memory based collaborative filtering example

- **User-based approach:**

  In this approach, the users are the main leaders to perform the associated task. If a certain majority of people share similar tastes then they are clustered into a single group.

  Therefore, recommendations to the user are given on the basis of the evaluation of items by other users from similar cluster with which he/she shares identical taste or preferences. Hence, the item will be recommended to the user if it was effectively rated by the community. Thus the items which were rated beforehand by the user beforehand plays a crucial role in user-based approach in locating the cluster that shares appreciations with him [4].

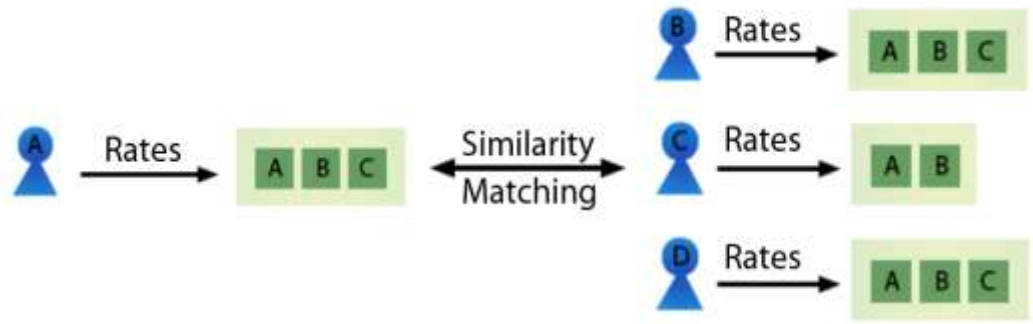


Figure 4 User-based collaborative filtering recommender system [4]

Thus, the approach is based upon finding neighbors of an active user or new user to predict his/her preferences on a new item. Here, nearest-neighbor based collaborative filtering technique has been used to predict preferences for the active user.

Different kind of aggregate analysis can be performed on similar user's data to generate relevant recommendations, in certain priority order, if required.

Similarity computed can be item-based or user-based. There are several techniques to find the similarity or distance between users or items. For example, Euclidean distance, Minkowski distance, Pearson correlation and Cosine-similarity metrics.

i. **Euclidean distance:** For two data points P = $(p_1, p_2, \ldots\ldots, p_n)$ and Q = $(q_1, q_2, \ldots\ldots\ldots, q_n)$, the Euclidean distance is defined as:

$$\mathrm{d}(\mathbf{p},\mathbf{q}) = \mathrm{d}(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

---------(1)

ii. **Minkowski distance:** For two data points, X = (x1, x2, . . . , xn) and Y =(y1, y2, . . . , yn), the popular Minkowski distance is defined as [6]

$$d(X,Y) = \sqrt[q]{\sum_{i=1}^{n} |x_i - y_i|^q},$$

----------(2)

where n denotes the number of the objects and x_i , y_i give the values of the i^{th} dimension for the data points X and Y respectively . Here q is a positive integer.

iii. **Pearson Correlation Similarity Measure:** Pearson correlation is used to find the degree to which two given variables are linearly related with each other. For user-based similarity, the Pearson correlation for two users u and v is defined as:

$$w_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}},$$

--------(3)

where i ∈ I denotes summations over items u and v both have rated. Also, $\bar{r}_u$ and $\bar{r}_v$ are the average rating of those items that both u and v have rated.

Similarly, for item-based similarity, $w_{i,j}$ defines the set of users u ∈ U who have cast rating for both items i and j:

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}},$$

----------(4)

where $r_{u,i}$ is the rating of user u on item i, $\bar{r}_i$ is the average rating cast by the user for the i^{th} item. [5]

iv. **Vector cosine-based similarity:** Vector cosine-based similarity for the items i and j is given by:

$$w_{i,j} = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \bullet \vec{j}}{\|\vec{i}\| * \|\vec{j}\|},$$

-------(5)

where "•" is the dot-product between the two vectors. To obtain the required similarity calculation for n items, n × n similarity matrix is calculated. Say, if there are two vectors say, vector $\overrightarrow{A}$ = { x_1, y_1 } and vector $\overrightarrow{B}$ = { x_2 , y_2 } then vector cosine similarity for the two vectors $\overrightarrow{A}$ and $\overrightarrow{B}$ is explained as:

$$w_{A,B} = \cos(\vec{A}, \vec{B}) = \frac{\vec{A} \bullet \vec{B}}{\|\vec{A}\| * \|\vec{B}\|} = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2}\sqrt{x_2^2 + y_2^2}}.$$

---------(6)

So, the conclusion is, a choice is to be made among all similarity measures. The point to remember at this step is:

a. if the data is subject to grade-inflation (i.e. different users may be using different scales) then use pearson correlation coefficient [6].

b. if data is dense (i.e. if almost all attributes have non-zero values) and the magnitude of the attribute value is important, use distance measures such as Euclidean distance.

c. if the data is sparse consider using Cosine-similarity [6].

- **Item-based approach:**

Item-based approach works on the notion that taste of the users remains more or less consistent or change very slightly. Thus, identical items develop

neighborhoods on the basis of appreciations of the users. Finally, the system generates the recommendations according to user preference with items in the neighborhood.

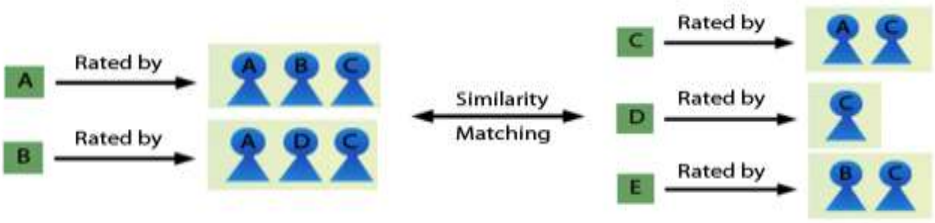


Figure 5 Item-based collaborative filtering recommender system [4]

1.2.1.2 Model-based collaborative filtering technique:

In model-based collaborative filtering approach, based on the training data the system learns to recognize complex patterns and then make intelligent predictions for test data or real world data for collaborative filtering systems based on the models learned by the system. Sometimes complete dataset is taken as training data and sometimes the dataset is spilt in a particular ratio to train the model and for testing purpose. Bayesian models, clustering models are examples of model-based CF [7].

1.2.2 Content filtering based recommender systems :

This recommendation technique is also known to as cognitive filtering. It recommends items similar to those items which the user liked previously. Each item's content is symbolized by set of terms, usually the words that appear in a document. These terms represent the profiles of users, which are made after analyzing the contents of items seen by the user [8].

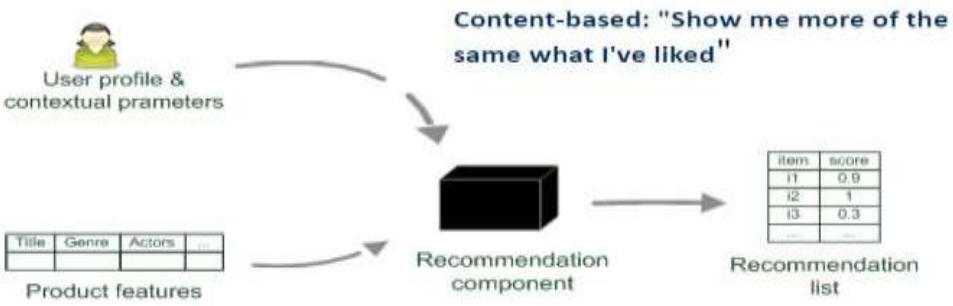


Figure 6 Content-based recommender systems [9]

1.2.3 Hybrid filtering based recommender systems :

These use integration of two techniques i.e. content-based filtering along with collaborative filtering which could be more effective in some cases. Therefore, no single recommender system approach is found to be efficient enough to generate relevant and accurate recommendation preferences. So, a hybrid recommender system came into existence to overcome the limitations of traditional recommendation approaches mentioned above. These systems are based upon combining collaborative filtering approach with content-based approach or collaborative filtering and demographic characteristics based recommendation approach.

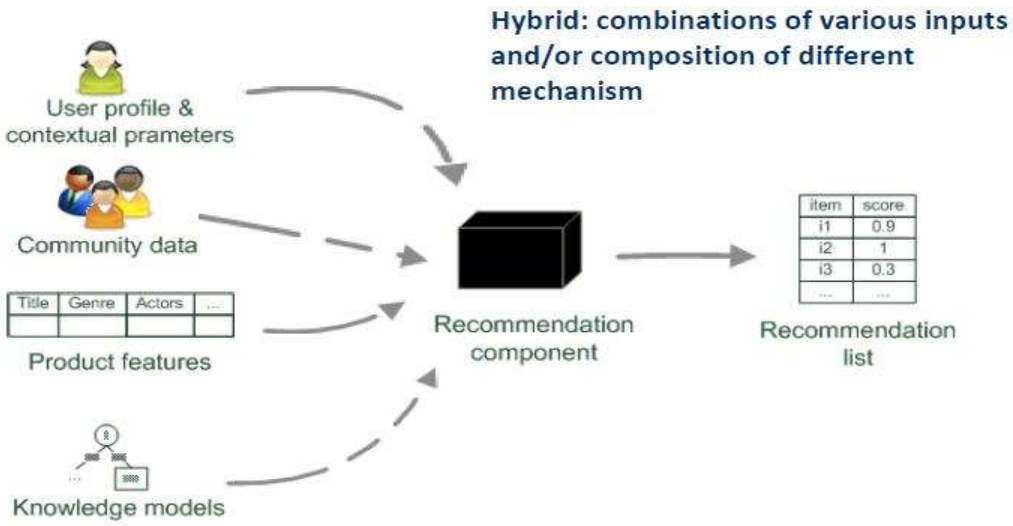


Figure 7 Hybrid recommender systems [9]

1.3 Recommendation techniques shortcomings

Recommender systems have been used extensively for generating recommendations using various recommendation techniques in multifarious application domains and this has brought into notice many challenges. Research areas are emphasizing on solving the issues mentioned below:

1.3.1 Cold start problem

Cold start problem, also known as new user problem or new item problem, is a special kind of sparsity problem. It occurs if a user or an item has no ratings. Due to the absence of purchase history or rating information of a new user or item, it becomes a challenging task for recommender engines to give suitable recommendations for new

users as well as items. This problem can be solved using clustering methods, for example k-means, fuzzy c-means, etc.

1.3.2 Scalability

With the increasing number of users and items, there are a number of scalability problems in traditional CF algorithms. Say, for a huge customer number and a huge number of items in the catalog, a CF algorithm with O(n) complexity becomes extremely big. A high amount of scalability is required in the CF algorithm to meet the online demands and requirements of these millions of users, regardless of what they have purchased or rated previously.

Singular value decomposition (SVD) is a dimensionality reduction approach which helps resolve the problem of scalability in recommender systems. Although quality of recommendations is improved, the cost involved in matrix factorization is high. The incremental SVD CF algorithm perform SVD such that new users rate items and are added to the database and a folding-in projection technique builds a system without performing any computations from the beginning. Thus it adds scalability to the recommender system.

1.3.3 Data sparsity

A large number of product sets are evaluated by recommender systems for commercial use. The user item rating matrix is generally sparse and performance of CF system is affected. With the hurdle of data sparsity, a number of problems like cold start arise when a new user is entered into the database. The cold start problem could be the new user problem or new item problem. It occurs because there isn't enough information to look for users with a taste alike to the new user. A new user would not have any recent history of ratings and purchases. So it would be tough to recommend new items to him.

If the ratings by a number of users are too small in comparison with the number of items, it leads to reduction in coverage problem, leading to difficulty in generation of recommendations. Another problem known as neighbor transitivity arises when there are sparse databases and users with similar tastes are difficult to get identified as they might not have rated the same items. As recommenders compare user pairs for predictions, this might reduce their effectiveness.

To get rid of sparse matrix problems, dimensionality reduction techniques like SVD remove the users and items which are not significant and also lower the number of dimensions in the user-item matrix by using latent features. Latent features are defined with respect to the item and user by representing them with vectors. The vector associated with the item defines the amount of feature that the item has whereas the vector related to the user defines the amount of interest a user has in that feature.

1.3.4 Shilling attacks

It is possible that users might give biased recommendations in favor of their own possessions and negative ones for their competitor's products. This kind of phenomena should be prevented in CF systems. Shilling attack models have been applied to recommender systems and their effectiveness has been analyzed. Lam and Riedl showed that item based CF algorithm was less affected as much as the user CF algorithm was. Thus they proposed a novel method to detect shilling attacks in recommender systems. A better approach to deal with shilling attacks would be to remove global effects when data normalization is performed in neighbor based CF. The residual of these effects could be used to select neighbors.

1.3.5 Gray sheep and Black sheep problem

Gray sheep refers to those users who don't benefit from collaborative filtering system as their views do not constantly agree or disagree with any cluster of people. In contrast, black sheep are the group of people having entirely opposite views as their idiosyncratic tastes make recommendations nearly impossible.

1.4 Evaluation metrics of recommender systems

The different type of CF based recommender systems use different metrics for evaluating their performance and hence provide information regarding the quality of recommender systems. Evaluation metrics plays a very important role in machine learning task. There are different metrics for the tasks of classification, regression, ranking, clustering, etc. Classification, regression, and ranking are examples of supervised learning, which constitute the majority of machine learning applications. Diversity in recommendation preferences is essential for usefulness and user satisfaction of recommender system [10].

Classification is about predicting class labels given input data. In *binary classification*, there are two possible output classes. In *multi-class classification*, there are more than two possible classes.

Metrics used in recommender systems can be categorized into:

- Predictive accuracy metrics, for example, Mean Absolute Error (MAE). It measures the extent to which a system predicts user ratings.
- Classification accuracy metrics gauge how well a system classifies items correctly. For example, Precision, Recall, F1-measure and ROC Sensitivity
- Coverage metrics find the percentage of items for which recommendations are generated by the system.
- Confidence metrics measures the certainty of system about the accuracy of the recommendations.
- Rank accuracy metrics such as Pearson's product-moment correlation, Kendall's Tau, Mean Average Precision (MAP), half-life utility and normalized-distance based performance metric(NDPM).

1.4.1 Mean absolute error (MAE) and Normalized mean absolute error (NMAE) :

Mean Absolute Error (MAE) computes the average of the absolute difference between the predicted ratings and actual ratings [7]

$$MAE = \frac{\sum_{\{i,j\}} \left| p_{i,j} - r_{i,j} \right|}{n}, \qquad \text{----------(7)}$$

Here n denotes the total no. of ratings for all users , $p_{i,j}$ gives the predicted rating for user i on item j and $r_{i,j}$ denotes the actual rating. The prediction is better if MAE is lower.

Normalized Mean Absolute Error (NMAE) normalizes MAE for describing errors as full scale percentages.

$$NMAE = \frac{MAE}{r_{\max} - r_{\min}}, \qquad \text{----------(8)}$$

Here, r_{max} and r_{min} are the upper and lower bounds of the ratings and different recommender systems make use of various numerical ratings as in the case of MovieLens database, ratings for movies are chosen from 1-5 scale. The lowest rating is 1 and 5 is considered as highest rating. These ratings are a primary source of predicting preferences, if available in dataset.

1.4.2 Root mean square error (RMSE)

It is a very popular performance measurement metric for recommender systems. It is expressed as:

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

----------(9)

RMSE magnifies the participation of absolute errors between the actual and predicted values [7].

1.4.3 Accuracy

Accuracy simply measures how often the classifier makes the correct prediction. It's the ratio between the number of correct predictions and the total number of predictions (the number of test data points) [11].

Accuracy = no. of correct predictions/no. of total data points ----------(10)

A variation of accuracy is the average pre-class accuracy- the average of the accuracy for each class. Accuracy is an example of a micro-average and average per-class accuracy is macro-average.

1.4.4 Precision-Recall-F Score

Precision and Recall metrics are usually used together. Precision means out of the item that the ranker/classifier predicted to be relevant, how many are truly relevant? On the other hand, recall measures out of all the items that are truly relevant, how many are found by the ranker/classifier?

Precision = no. of correct answers / no. of total items returned by Ranker -----(11)

Recall = no. of correct answers / no. of total items ----------(12)

F-score is the harmonic mean between Precision and Recall. If Precision is high then Recall is low or vice-versa. F-measure is also referred to as the F1 measure in recommender system evaluation terminology. It is defined as:

$$F\text{-}measure = \frac{2\,Precision\,Recall}{Precision + Recall} = \frac{2}{1/Precision + 1/Recall}$$

----------(13)

1.5 Structure of thesis

Chapter 2: Literature review: This chapter details the work done by the researchers in this area. The results of survey of literature are represented in tabular form also.

Chapter 3: Problem statement: In this chapter objective of the proposed work is given.

Chapter 4: Proposed Methodology: This chapter summarizes the proposed solution along with the implementation details

Chapter 5: Experimental results: This chapter contains the description of the results achieved against experiments conducted on MovieLens dataset and observations of the experimental analysis of proposed system.

Chapter 6: Conclusion and future scope: All the work done in thesis is summarized in this chapter and it also contains the scope for future research work in same or different problem domain.

Chapter 2

Literature Review

2.1 Evolution of recommender systems

Recommender systems have recently become immensely important in the world of internet. The count of people using internet for various reasons is growing at an overwhelming rate. Recommender systems are becoming an efficient tool to deal with the problem of infobesity. The concept of online recommender system has gained popularity due to online availability of goods and services and the role of recommender system becomes more important when one kind of product or services are being offered by so many online service providers. The recommender system uses several approaches for providing best results in the form of recommendations for an individual who is looking for preferences from an online recommender system. Hence, at present different websites implement recommender systems using different techniques like content-based filtering, collaborative filtering, knowledge-based filtering or hybrid filtering which make use of combination of traditional approaches as explained below:

1. **Content-based filtering:** This method finds preferences of current user about new item with the help of rating history of current user related to previously used items. Similarity between items can be found by measuring the similarity in properties of these items. So, in this type of filtering method there is no dependency on rating records of other users in order to generate preferences for current user. Content-Based systems work on properties of items [12].
2. **Collaborative filtering:** Collaborative-Filtering systems work on the user-item relationship. This method finds user-similarities using m x n rating matrix with ratings data given by various users corresponding to same set of items in the online store [12].
3. **Knowledge-based filtering:** Knowledge based recommender engine requires a knowledge structure to make inference about the user needs and preferences. Such kind of recommender systems have knowledge about what kind of items are liked by a user. So, a relationship can be established between user needs and relevant recommendation for that user [13].

Hence a tradeoff between different approaches is given in the table 1.

Table 1 Tradeoff between recommendation approaches

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

2.2 Overview of collaborative filtering (CF) technique

Recommender systems using CF are called as collaborative filtering based recommender systems as in these systems the information is filtered by using the recommendation from different people. It works on the notion that in a neighborhood people who have same evaluation of certain items or similar tastes or references in the past would agree in the future too [2]. Collaborative filtering techniques are classified into two approaches i.e. memory based approach and model based approach. These can be compared using different parameters including scalability, accuracy, memory consumption and time complexity for offline and online collaborative filtering recommender systems.

Different collaborative filtering techniques are compared according to different parameters is given in the table 2 where m gives number of users while n denotes number of items [14].

Table 2 Comparison of Collaborative Filtering Techniques

Parameters		Algorithms		
		Memory based	Model based	Hybrid Recommender
Scalability		Low	High	Very High
Accuracy		Low	High	Very High
Memory Consumption		Low	High	Low
Time Complexity	Offline	-	O(m)	O(m)
	Online	O(mn)	O(mn)	O(mn)

Collaborative filtering techniques can be categorized further on the basis of the representative techniques they use along with their advantages and disadvantages as given below [7]:

i. **Memory-based CF:**

The representative techniques include neighbor-based CF (item-based/ user-based CF algorithms with pearson/ vector cosine correlation) and item-based/ user-based top-N recommendations.

Advantages:

- Implementation is easy.
- Scalable with co-rated items**.**
- Easy to add new data.
- Content of the items need not be considered.

Disadvantages:

- Need human ratings.
- If data is sparse, performance is low.
- Difficult to recommend in case of new users and items.
- Very less scalable if big datasets are employed.

ii. **Model-based CF:**

The representative techniques for model-based CF include bayesian belief nets CF, clustering CF, latent semantic CF and CF using dimensionality reduction techniques like SVD and PCA.

Advantages:

- Resolve problems like sparsity and scalability
- Performance of prediction is better.

Disadvantages:

- Building the model is costly.
- Trade-off exists for prediction performance and scalability.
- In dimensionality reduction techniques, useful information is lost.

iii. **Hybrid recommenders:**

Hybrid recommenders can be built using various techniques such as content-based CF recommender (for example, Fab), content-boosted CF and hybrid CF combining memory-based and model-based CF algorithms (for example, personality diagnosis).

Advantages:

- Drawbacks of CF, content-based and other recommenders are overcome.
- Prediction performance improves.
- Deals with sparsity and gray sheep issues.

Disadvantages:

- Too expensive and increased complexity.
- Requires external information, generally unavailable.

The CF based recommender systems get affected by the problem of cold start as discussed in next section.

2.3 Cold start problem and its solutions using various approaches

Cold start problem, also known as new user problem or new item problem, is a special kind of sparsity problem. It occurs if a user or item has no ratings. Due to the absence of purchase history or rating information of a new user or item, it becomes a

challenging task for recommender engines to give suitable recommendations for new users as well as items.

Methods like character capture and clustering are used to handle user cold start problem i.e. generating recommendations for the new users who do not have ratings on items [15]. Vector cosine method is used to compute user's similarity matrix and top N recommendations for each group are produced by computing the average rating of each item for which rating is done by greater than a user as follows:

$$X_i = \{x_{ij}\_AverageRating \in AverageRating\_TopN\} \quad \text{----------(14)}$$

where X_i is the recommendation of group i, $x_{ij}\_AverageRating$ denotes item's average rating, $x_{ij}.AverageRating\_TopN$ gives the item collection for which average rating is one among the top N list:

$$Y = \{X_i | u \in Group_i\_User\} \quad \text{----------(15)}$$

where Y is the recommendation for a new user u who enters the system, $Group_i\_User$ symbolizes the users belonging to group i.

Experimentation has been done on MovieLens dataset from which 100 users and their rating has been selected. Clustering is done on the basis of user demographic data which include gender, age and occupation. For performance evaluation, the dataset is randomly portioned as training and testing data with two ratios: r = 0.6 and r = 0.8.

A trust between users is calculated using probabilistic neural network based on the rating matrix by Devi et al. [16]. Then sparse rating matrix is smoothened for items not yet rated, using the calculated trust values. Then a smoothened matrix is used to find trust for online users and recommendations are made. The experiments are done using MovieLens dataset and the system performance is evaluated using different evaluation metrics like mean absolute error, precision and recall.

Association rules and clustering techniques are used to resolve cold start problem in recommender systems as in [17] [18]. This can be done by expanding user profiles (U) which means that it contains more ratings. Using the taxonomy driven user

profiles that already exist, a transactional dataset is constructed using which frequent patterns are mined. From the patterns, association rules can be derived among topics that look interesting to the user. Now using the rule set and the user profiles (U), for each user profile u (p_x) all topics (t) are extracted and recommendations are generated in the form of all combinations possible from the set of topics. Weights are assigned for every topic using the weight of the topics in the antecedent of the rule. It helps in resolving new user problem to some extent.

Further for clustering, fuzzy k-means algorithm is used on the group items and pearson correlation based similarity and adjusted cosine based similarity is used to compute similarity matrix. Finally, a linear combination of results is formed. This experiment is performed on MovieLens dataset from GroupLens. The data is preprocessed using Weka software and this data is stored in MS Access database after preprocessing [18]. Furthermore, improvement in performance is obtained using non-redundant rule sets as non-redundant rules do not lead to any information loss and are as informative as redundant rules.

Chameleon based recommender system i.e. a hierarchical clustering algorithm is proposed by Gupta and Patil in which clusters are formed out of user or item specific information using chameleon hierarchical clustering algorithm [19]. To predict the rating of a particular item, voting system is used as voting system is less expensive in terms of computation as compared to similarity algorithms. Hence, it is much better for e-commerce applications. Hierarchical clustering algorithm is explained in two phases where phase 1 is used to construct initial graphs and then divided into random number of partitions. In phase 2, grouping of partitions is done till natural number of clusters is formed.

Further, the clusters are formed using two parameters i.e. Relative Closeness (RC) and Relative Interconnectivity (RI). Relative Closeness is defined as the absolute closeness between clusters say C_i and C_j normalized with respect to internal closeness in C_i and C_j as follows:

$$RC(C_i, C_j) = \frac{\bar{S}_{EC_{\{C_i, C_j\}}}}{\frac{|C_i|}{|C_i| + |C_j|} \bar{S}_{ECC_i} + \frac{|C_i|}{|C_i| + |C_j|} \bar{S}_{ECC_j}}$$

----------(16)

where $\bar{S}_{EC_{\{C_i,C_j\}}}$ denotes the average edge weight connecting C_i and C_j normalized with regard to average weight of edge cut in the cluster in C_i and C_j. $|C_i|$ gives the number of edges with starting and ending vertices within cluster in C_i.

Relative Interconnectivity (RI) is described as the connectivity between two clusters C_i and C_j normalized with regard to internal interconnectivity between clusters C_i and C_j as follows:

$$RI(C_i, C_j) = \frac{|EC_{\{C_i, C_j\}}|}{\frac{1}{2}(|EC_{C_i}| + |EC_{C_j}|)}$$

----------(17)

The test is performed using movie rating dataset. Similarity is computed using Euclidean distance method and the system is evaluated using mean absolute error metric.

A novel approach is proposed by Sun et al. using a subset of movie ratings collected from the MovieLens recommender with 100,000 ratings obtained from 943 users and 1,682 movies. All users have rated atleast 20 items, to handle new item cold start problem by using both the ratings and the content information [20]. At first the items are clustered based on the present item-user rating matrix and then using clustering results and content information, decision tree is built to associate the new items and the ones that exist. In future, if the direction of the item-user matrix is inverted and decision tree is built using user content information then this model can be used to handle user side cold start problem.

A new hybrid recommender system is kept forward by Basiri et al. that deal with improving the performance in case of new user cold-start condition when users have given none or few ratings [21]. In this five classification strategies are used and then

results of these classifiers are merged by Optimistic exponential type of ordered weighted averaging (OWA) operator. First approach utilizes user rating history for similarity computation. Second approach focuses on items attribute based similarity computation, which are already purchased by the user. Third approach makes use of demographic information contained in user profiles and previous ratings. Fourth approach is purely based upon the user demographic information. Finally, fifth approach applies both i.e. items attribute information of past purchases and each customer's rating history. If recommender's final output is in the form of predicted label of item x for the user y is 1, x will recommend to y, else it won't.

OWA operator (by Yager) of dimension n is a mapping such as:

$F: R^n \rightarrow R$ and is given by

$$OWA(a_1, a_2, \ldots\ldots, a_n) = \sum_{i=1}^{n} w_i b_i$$

where b_i is the i^{th} largest element among a_i's. The weights are non-negative ($w_i >= 0$) and sum of these equals one $(\sum_{i=1}^{n} w_i = 1)$.

The aggregation performed by OWA operator is always between maximum and minimum. A degree of maxness (initially called orness) was introduced as follows:

$$Maxness(w_1, w_2, \ldots\ldots, w_n) = \frac{\sum_{i=1}^{n}(n-i)w_i}{n-1} \quad \text{----------(18)}$$

A simple class of OWA operators as exponential class was used to generate OWA weights fulfilling a given degree of maxness. The optimistic and pessimistic exponential OWA operators were correspondingly formulated as follows:

Optimistic:

$$w_i = \alpha \times (1-\alpha)^{i-1}, \forall\, i \neq n;\ w_n = (1-\alpha)^{n-1} \quad \text{----------(19)}$$

Pessimistic:

$$w_1 = \alpha^{n-1};\ w_i = (1-\alpha) \times \alpha^{n-i}, i \neq 1 \quad \text{----------(20)}$$

where parameter α belongs to the unit interval [0 1] and is related to orness value concerning n. The proposed hybrid approach has shown improvement in the prediction accuracy of the existing recommender engines in the cold-start (new user)

condition. This approach can further be improved by providing dynamicity to α which is used as a static parameter in this research study. Further the same approach can be tested for solving other problems in recommender system.

A hybrid approach for enhancing correlation is proposed by Dang et al. in order to address cold-start problem in recommender systems [22]. The data is clustered using user's demographic information (that includes gender, age, country, etc) which is taken from MovieLens dataset. Demographic filtering works on the principle that individuals with similar personal attributes also have same common preferences. Therefore, this hybrid recommender system combines demographic filtering approach and collaborative filtering approach in which neighbor formation is done using pearson correlation metric. Further after combining the results of demographic filtering approach and collaborative filtering approach, perceptron learning neural network approach is applied to find out the new rating predictions from the already existing ones as shown in the figure 7. Hence, correlation evaluation metric is applied to test the new system and it shows that the combination of demographic filtering approach and collaborative filtering approach enhances the accuracy of this recommender system. Further the system can be expanded by using different methods like content-based and knowledge-based.

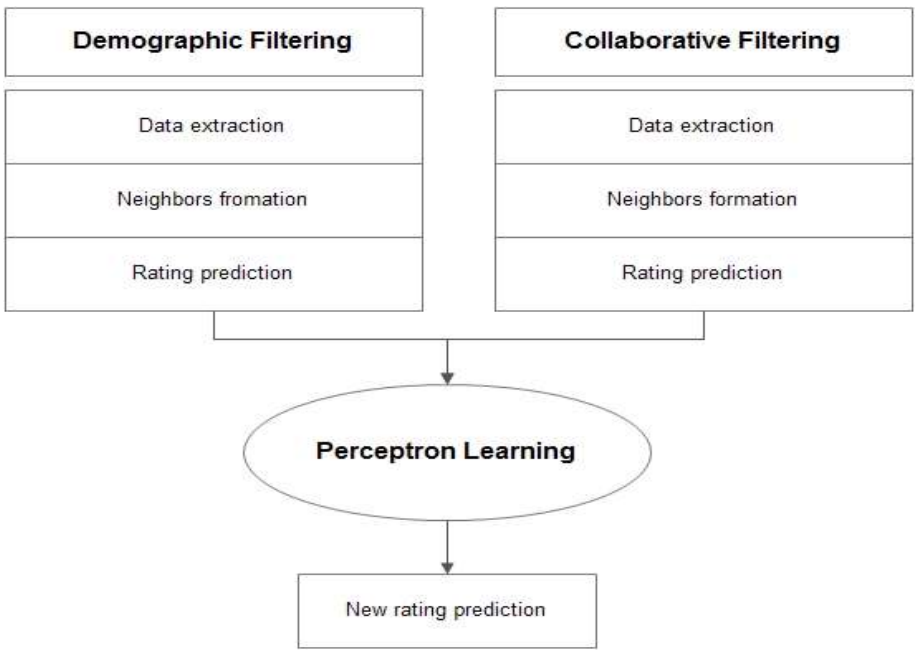


Figure 8 Combination of demographic filtering and collaborative filtering using perceptron learning

A fuzzy geographical clustering method is proposed by Son et al. to deal with cold start issue in recommender engines [23]. Thus, it combines some of the latest result of fuzzy geographically clustering and collaborative filtering and proved that this system gives better accuracy as compared to the other relevant systems.

A hybrid approach is proposed by Verma et al. that is capable of dealing with sparsity and scalability issues [24]. This approach combines collaborative filtering and fuzzy c-means clustering algorithms. The approach is further divided into two phases. In phase 1 items are clustered based on item's profile and in phase 2 item-based collaborative filtering algorithm is done on each cluster in order to predict rating for each cluster. To reduce cold start problem for new user, ratings are given for a set number of items to get recommendations. The experimentation is performed on MovieLens dataset consisting of 1,00,000 ratings(1-5) from 943 users on 1682 movies and similarity matrix is calculated using pearson correlation metric. Also dataset includes genre of each movie (for example: comedy, action, etc). In total there are 19 genres on the basis of which clustering is done. Since it is a memory based approach so time is not wasted in training the model.

Another combinatorial approach is carried out to develop a collaborative filtering based recommender system by combining fuzzy c-means clustering technique and genetic algorithm based weighted similarity measure [25]. Fuzzy c-means clustering helps in clustering the datasets on the basis of the ratings given by the users in the dataset. Then these cluster values are passed on to the genetic algorithm based weighted similarity measure to find the similarity between the clustered values and thus obtain the optimal similarity metrics. Also the quality measures of the recommender systems are computed in order to show an improvement in the results quality on the basis of the number of identical values retrieved with respect to the number of iteration runs performed by an algorithm. The basic outline of the architecture is shown in figure 8. This proposed recommender system can be used in all collaborative filtering systems and it does not even require any hybrid model to get implemented with.

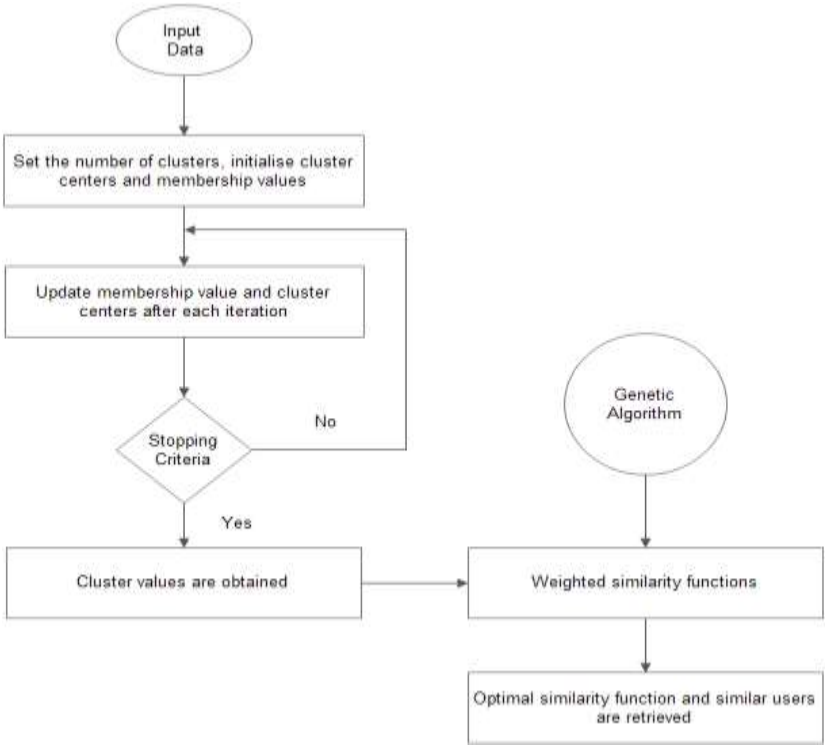


Figure 9 Outline of the combinatorial approach [25]

Chapter 3

Problem Statement

3.1 Introduction

Recommender engines have become highly popular among Internet users due to a vast set of available choices. Collaborative-filtering (CF) technique is a popular technique in recommender systems to suggest items as per the user's taste. Recommender systems face various obstacles such as sparsity problem, cold start problem and scalability issues. Cold start problem arises when there is no sufficient information for the user who has recently logon into the system and no proper recommendations can be made. Owing to no purchase history or rating information related to a new user or item, it becomes a challenging task for the recommender engines to produce quality recommendations for new users or items. Many techniques like k-means, fuzzy c-means, etc have been applied to address cold start problem by clustering similar users. But there is a need to handle this problem with better accuracy. In this thesis, fuzzy c-means clustering has been used as a comparison method to cluster different sets of users. It is found that its accuracy is better than that of k-means clustering as the number users increase in a data set. Thus fuzzy c-means approach is more accurate in terms of clustering users and subsequently generating a better quality of recommendations for a new user who enters the system.

3.2 Objectives

The main focus of this work is to apply fuzzy clustering technique to address cold start problem in the existing system using demographic information. The main objectives achieved through the work are:

- To achieve appropriate clustering of instances using fuzzy clustering in an unsupervised machine learning scenario.
- To compare accuracies of fuzzy c means and k means clustering.
- To use demographic attributes like age, gender and occupation to improve the accuracy of clustering.
- To generate predictions for a new user who has just arrived in the system when no rating data is available.

- To predict top-N recommendations for a new user based on the cluster he is placed in.

Chapter 4

Proposed Methodology and Technologies Used

4.1 Introduction to methodology

Collaborative filtering recommender systems give best results when the user-item matrix is extensive and the dataset has high matching information according to the new user. The work done is based upon exploiting user's demographic data for finding similarity between the already existing user and the new user. Demographic information includes different user features like gender, occupation, religion, age, zip-code, race, locality, hobbies, marital-status and many more.

The main data mining technique used here is fuzzy c-means clustering algorithm for finding user similarity based upon user's demographic characteristics like age, gender and occupation and then evaluating the clusters performance accuracy in comparison to k-means clustering algorithm. It is a form of unsupervised machine learning. The proposed approach is explained in the following two steps:

Step-1: Offline analysis of data is done using R tool for building a model for collaborative filtering using appropriate demographic attributes.

Step-2: New user data is fetched. New users are clustered into different groups or clusters by applying fuzzy c-means clustering approach based on their features (i.e. gender and age) which they will enter at the time of login into the system. Thus, the cluster's recommendation is obtained to which the user belongs to by using the preprocessed data which will be stored in MySQL database for generating the recommendations. Therefore, the user gets the appropriate recommendations as it is assumed that the users with identical tastes share same clusters and therefore tend to rate in the similar fashion.

Clustering methods are of three types: partitioning methods, density-based methods, and hierarchical methods. Partitioning method mostly used is k-means clustering where each object or data point is a part of only one cluster or group. Density-based clustering methods are used for searching dense clusters of objects which are

demarcated by sparse regions signifying noise, for example, *DBSCAN* and *OPTICS*[6]. Hierarchical clustering methods, decompose a set of data objects in a particular hierarchy based upon some criteria e.g. *BIRCH* [7].

4.2 Collaborative filtering using clustering algorithm

In collaborative filtering systems information is filtered by comparing tastes of a user with other users. It is based on the notion that people who have similar tastes or preferences in the past are likely to agree in the future again. Clustering involves assigning data points to clusters or groups or homogeneous classes. The items in the same group or cluster are assumed to be as identical as possible while items belonging to different clusters are as different as possible. Similarity measures like distance, connectivity and intensity are used to identify the clusters. Different similarity measures are chosen according to the requirement of the data or the application. Further clustering can be classified into two types i.e. hard clustering and soft clustering [28].

- **Hard Clustering:** Hard clustering also known as exclusive clustering is the one in which each data point or object belongs to one cluster only or to none of the clusters. K-means clustering is a very important hard clustering technique.
- **Soft Clustering:** In soft clustering each data point or object belongs to each group to a certain extent i.e. each data point can be associated with more than one cluster or group. Soft clustering is also known as overlapping clustering. In case of soft clustering techniques, fuzzy sets are used to cluster data, so that each point may belong to two or more clusters with different degrees of membership. In this case, data will be associated to an appropriate membership value. Fuzzy c-means (FCM) clustering is a very popular soft clustering technique [29].

4.2.1 K-means Clustering

K-means clustering algorithm is one of the popular hard clustering algorithms. It partitions (or clusters) N data points into K disjoint subsets or clusters S_j with N_j data points so that the inter-cluster similarity obtained in the result is low while offering a high intra-cluster similarity. The algorithm aims at minimizing the

objective function i.e. squared error function in this case. The objective function is as follows [31]:

$$J = \sum_{j=1}^{K} \sum_{n \in S_j} |x_n - \mu_j|^2 \qquad \text{----------(21)}$$

where x_n is a vector that signifies the n^{th} data point and μ_j is the geometric centroid of the data points in S_j. In general, the algorithm does not acquire a global minimum *J* over the assignments. In fact, since the algorithm uses discrete assignment rather than using continuous parameters, the "minimum" it reaches can't be called a local minimum in a proper manner. Despite these drawbacks, the algorithm is used rather frequently as it is easy to implement.

The algorithm consists of a re-estimation procedure as given. Initially, *K* sets are available and data points are randomly assigned. For step 1, the centroid for each set is computed. In step 2, each point is given to the cluster whose centroid is nearest to that point. Alternation of these steps is done until stopping criteria is met, i.e. no further changes in the data point assignments are needed.

A global minimum, also known as an absolute minimum, is the minimum value of a set, function, etc., over its entire range as a whole. It is not possible to design an algorithm resulting in a global minimum for any arbitrary function. A local minimum, also called a relative minimum, is a minimum within a neighborhood, not necessarily (but may be) a global minimum.

K-means, proposed by MacQueen is an example of most-commonly used partitioning method due to two major advantages i.e. relative efficiency and ease of implementation. Also, k-means algorithm is very sensitive to initially selected random cluster center due to which the algorithm can be executed many times in order to eliminate this result. K-means is a simple algorithm being used in many problem domains and it is a good approach when data points are randomly generated.

4.2.2 Fuzzy c-means clustering

Fuzzy clustering or soft clustering is a type of clustering in which each data point can

be linked to more than one cluster or group. This technique was presented by Jim Bezdek in 1981 as it worked better than initial clustering techniques [28]. As compared to the existing k-means technique, fuzzy c-means clustering approach works by providing membership value to every piece of data or data point equivalent to every cluster center based on the distance from data point to its cluster center. This technique allows one data point to be a part to two clusters or greater. Therefore, if the data is nearer to cluster center, its association with specific cluster center is more.

As shown in the algorithm there are two important parameters a_{ij} and b_j. a_{ij} represents association between i^{th} data point and the j^{th} cluster center and b_j represents the j^{th} cluster center [30].

Algorithm:

Require: User demographic data D=$\{d_1, d_2, d_3, \ldots\ldots, d_n\}$, Set of cluster center C=$\{c_1, c_2, c_3, \ldots\ldots\ldots, c_m\}$. Assume fuzziness index 'f' $(1 \leq f \leq \infty)$ and Euclidean distance between the i^{th} data point and the j^{th} cluster center $\|d_i - b_j\|^2$

Ensure: New user groups: User Cluster

1. Arbitrarily select 'm' cluster centers.

2. Calculate fuzzy association between the data points and the cluster centers $'a_{ij}'$ using:

$$a_{ij} = 1 \Big/ \sum_{k=1}^{m} \left(\frac{e_{ij}}{e_{ik}}\right)^{\left(2/f-1\right)}$$

3. Compute fuzzy centers of the clusters $'b_j'$ using:

$$b_j = \left(\sum_{i=1}^{n} (a_{ij})^f d_i\right) / \left(\sum_{i=1}^{n} (a_{ij})^f\right.$$

4. Repeat steps [2] and [3] till the objective function value O is minimized :

$O(A, B) = \sum_{i=1}^{n} \sum_{j=1}^{m} (a_{ij})^f ||d_i - b_j||^2$

In fuzzy c-means clustering algorithm, data are bound to each cluster by means of a membership function, which represents the fuzzy behavior of this algorithm.

4.3 Architecture for proposed approach

Here a sketch of the planned technique for addressing the user cold start problem is explained.

The approach is to combine the two techniques in serial order one after the other. At first fuzzy c-means clustering technique is applied on different attributes of user's demographic data (i.e. gender and age). Secondly, using the clustered data after applying fuzzy c-means clustering algorithm, MySQL database is used in which the preprocessed data is stored for generating recommendations. Thus, aiming at new user recommendations. The proposed system architecture is given in figure 10.

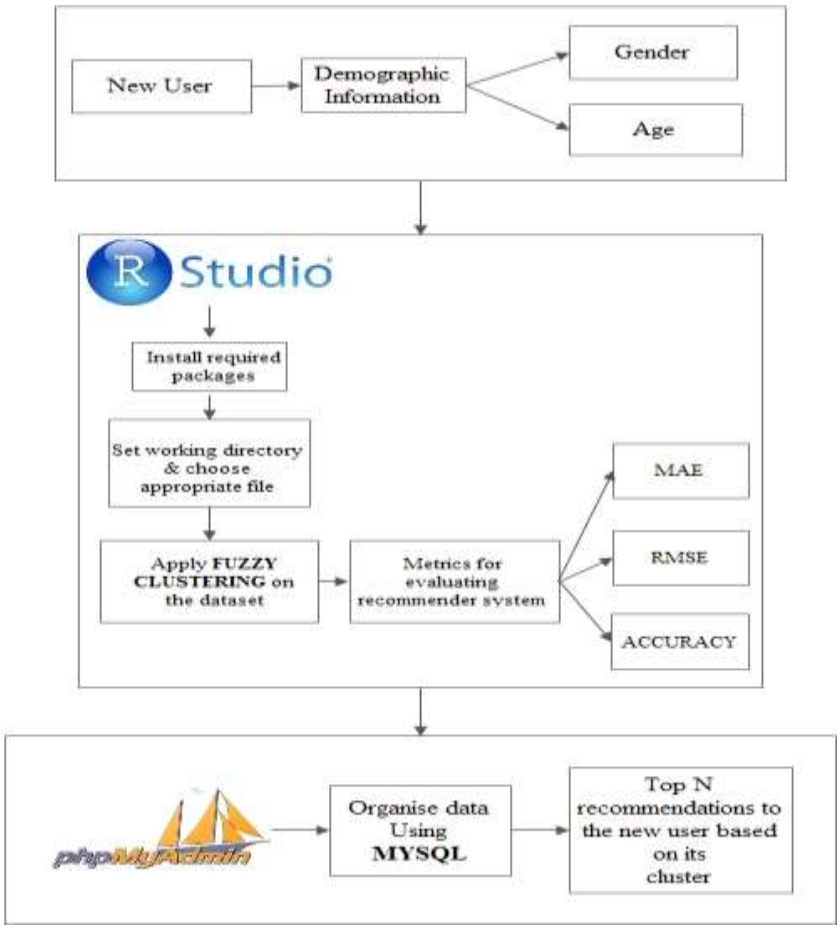


Figure 10 Architecture of the Collaborative Filtering recommender system using fuzzy clustering technique

4.4 Technologies used

The two main technologies used in this thesis work are RStudio for data mining and MySQL for generating recommendations.

4.4.1 RStudio

RStudio is an integrated development environment (IDE) for R, a programming language used in statistical computing. It was founded by JJ Allaire, who created programming language ColdFusion [31]. Work on RStudio started near about December 2010. RStudio is written in C++ programming language. RStudio comes in two editions:

1. RStudio Desktop, run locally as a regular desktop application. Prepackaged distributions can be obtained for Windows, OS X, and Linux.
2. RStudio Server, which allows a web server to access RStudio while it runs on a remote Linux server.

RStudio is preferred for doing offline analysis in this thesis work because of following key features [31]:

- It is available in open source and runs locally on the desktop (Windows, OS X, and Linux) or in a browser linked to RStudio Server or RStudio Server Pro (Debian, Ubuntu, Red Hat Linux, CentOS, openSUSE and SLES).
- It is obtainable free of cost.
- It is platform-independent and easily installable on all operating systems like Windows, Mac and Linux etc.
- Lots of learning resources including video tutorials are freely available online.

4.4.2 MySQL

MySQL is used for running queries for generating recommendations for new user in a system. The MovieLens dataset is created which uses this database as it is easily managed using GUI based phpMyAdmin environment.

It is an open-source database and XAMPP installation is needed before running any query in it. The learning of this is quite easy due to availability of many online resources which proved to be a great source of help in completing experimentation work. It can be easily integrated with web development scripting languages like PHP

for producing recommendation results in a presentable and organized manner. It provides powerful features like database creation, consistency and integrity along with security features to protect hacking of confidential data of online users of recommender system.

Chapter 5

Experimental Results

5.1 Introduction to dataset

The dataset which is used for experimentation contains 1,000,209 ratings from 6040 users for 3900 different movies. It describes a 5-star rating activity by the users for the movies of different genres from Movielens (http://movielens.org) [32], a movie recommendation service.

Users were selected at random for inclusion. All selected users had rated at least 1 movie. The data for experimentation is taken from three files 'user.csv', 'movie.csv' and 'rating.csv'. The dataset files are written as [comma-separated values] files with a single header row.

Movie ids are consistent between `rating.csv`, `movie.csv`, and `user.csv` (i.e., the same id refers to the same movies across these three data files). For experimentation a sample of 6040 users is taken. This dataset consists of the following files:

5.1.1 Data files details of the dataset

1. **Ratings data file structure (rating.csv)**

   This file contains information in the format as given below:

   userid, movieid, rating

   The records are arranged in the following order i.e. userid is followed by movieid which is further followed by rating. Ratings are given on a scale of 1 to 5.

2. **Movies data file structure (movie.csv)**

   Movies information is contained in the file 'movie.csv'. The file is ordered in the following format:

   movieid, title, genre

   Genres are separated by pipes and are represented as:

   Action| Adventure| Animation| Children's| Comedy| Crime| Documentary| Drama| Fantasy| Film-Noir| Horror| Musical| Mystery| Romance| Sci-Fi|

Thriller| War| Western| (no genres listed)). Errors may exist as movies are entered manually.

3. **Users data file structure (user.csv)**

   User information is in the file 'user.csv' and is in the following format:

   userid, gender, age, occupation

   This .csv file is used for experimentation using RStudio software. All demographic information is given by users and its detailed description is given below:

   - User gender is denoted as **'M'** for male and **'F'** for female.
   - Age is chosen from varied ranges.
   - Occupation of users could range from:

   0: "other" 7: "sales/marketing"
   1: "academic/educator" 8: "scientist"
   2: "artist" 9: "self-employed"
   3: "clerical/admin" 10: "technician/engineer"
   4: "college/grad student" 11: "tradesman/craftsman"
   5: "customer" 12: "unemployed"
   6: "doctor/health care" 13: "retired"

The experiment focuses on finding how clustering results can be achieved well based upon appropriate choice of attributes using fuzzy c-means clustering and k-means clustering approach and further cluster's performance is validated on the basis of the accuracy for different sizes of the dataset. After experimentation in RStudio software for data mining, a model is proposed for further testing and for retrieving recommendation results using MySQL.

5.2 Experiment I: Part A

The tools used for experiment are RStudio and MySQL. Using the R code of traditional k-means clustering approach the first part of the experiment is started i.e. user dataset 'user.csv' is clustered on the basis of demographic information of the user (i.e. gender and age) using three clusters. The user dataset is clustered by changing the number of users and hence the MAE, RMSE and accuracy of the clusters being formed is recorded. Figure 11(a) and figure 11(b) R code and summary results of k-means clustering approach applied to dataset of 6040 users.

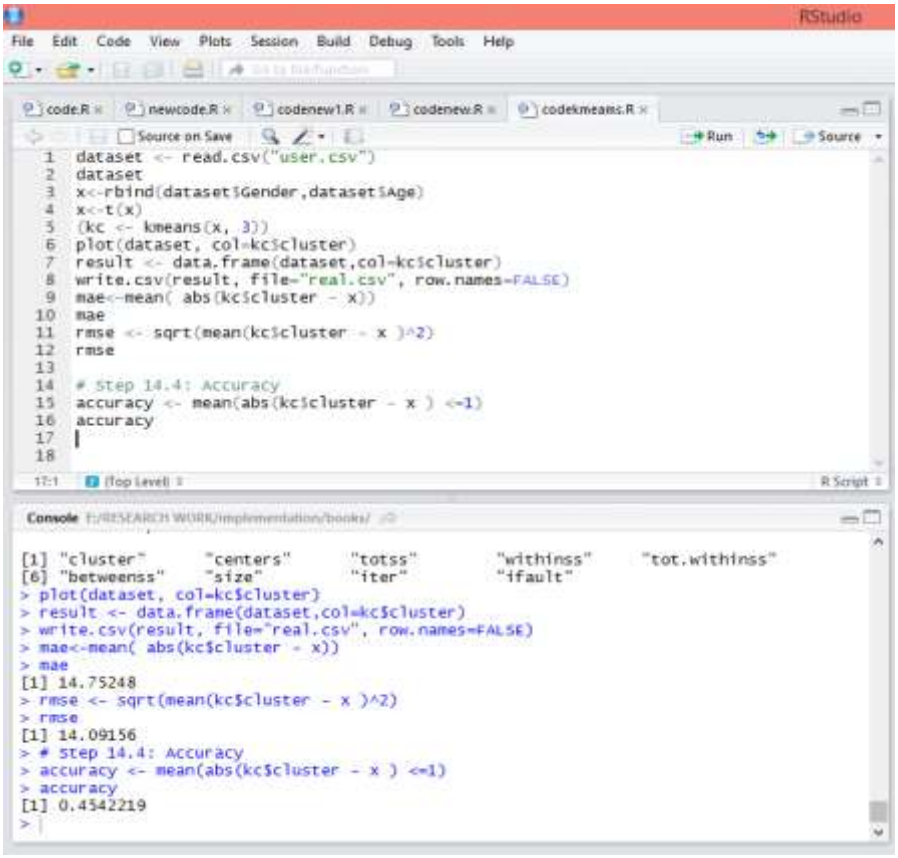


Figure 11(a) R code for k-means clustering for 6040 users

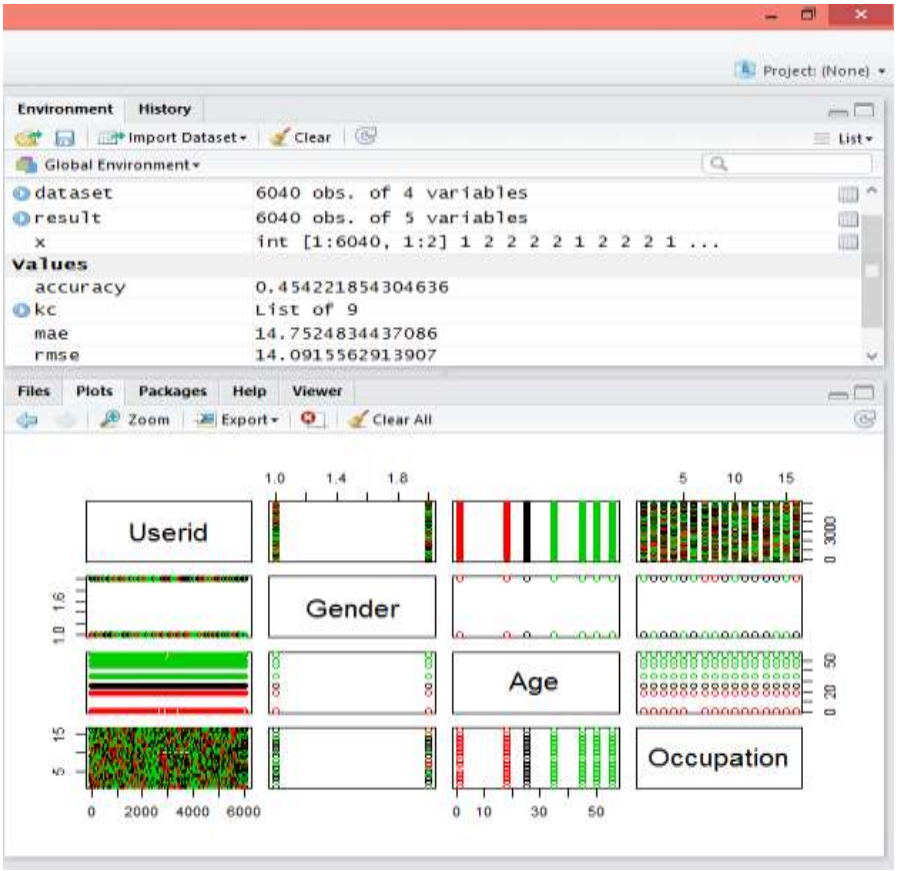


Figure 11(b) Summary of experimental results shown in 11(a)

5.3 Experiment I: Part B

After applying k-means approach then R code of fuzzy c-means approach is executed by varying number of users on the basis of their gender and age using three clusters. For running code of fuzzy c-means approach "e1071" package and its library need to be installed. Figure 12(a) and figure 12(b) R code and summary results of fuzzy c-means clustering approach applied to dataset of 1000 users.

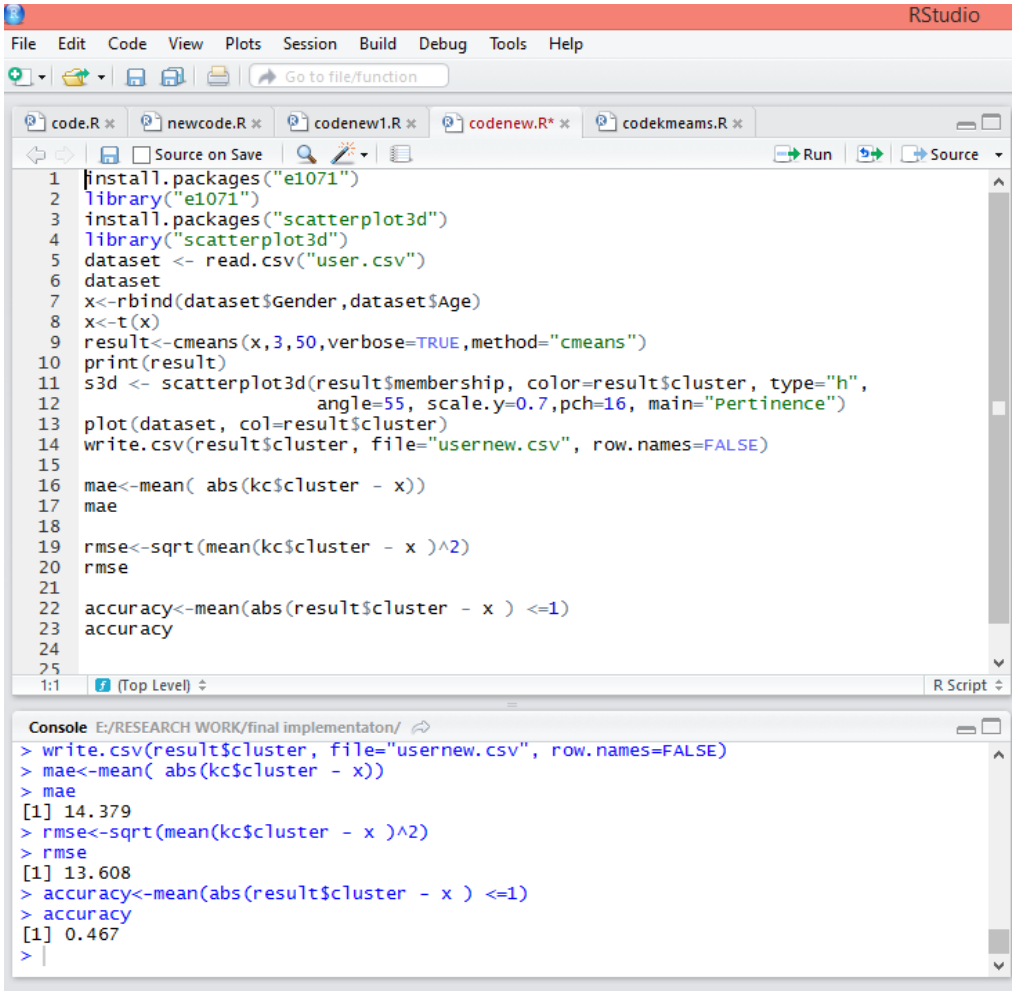


Figure 12(a) R code for fuzzy c-means clustering for 1000 users

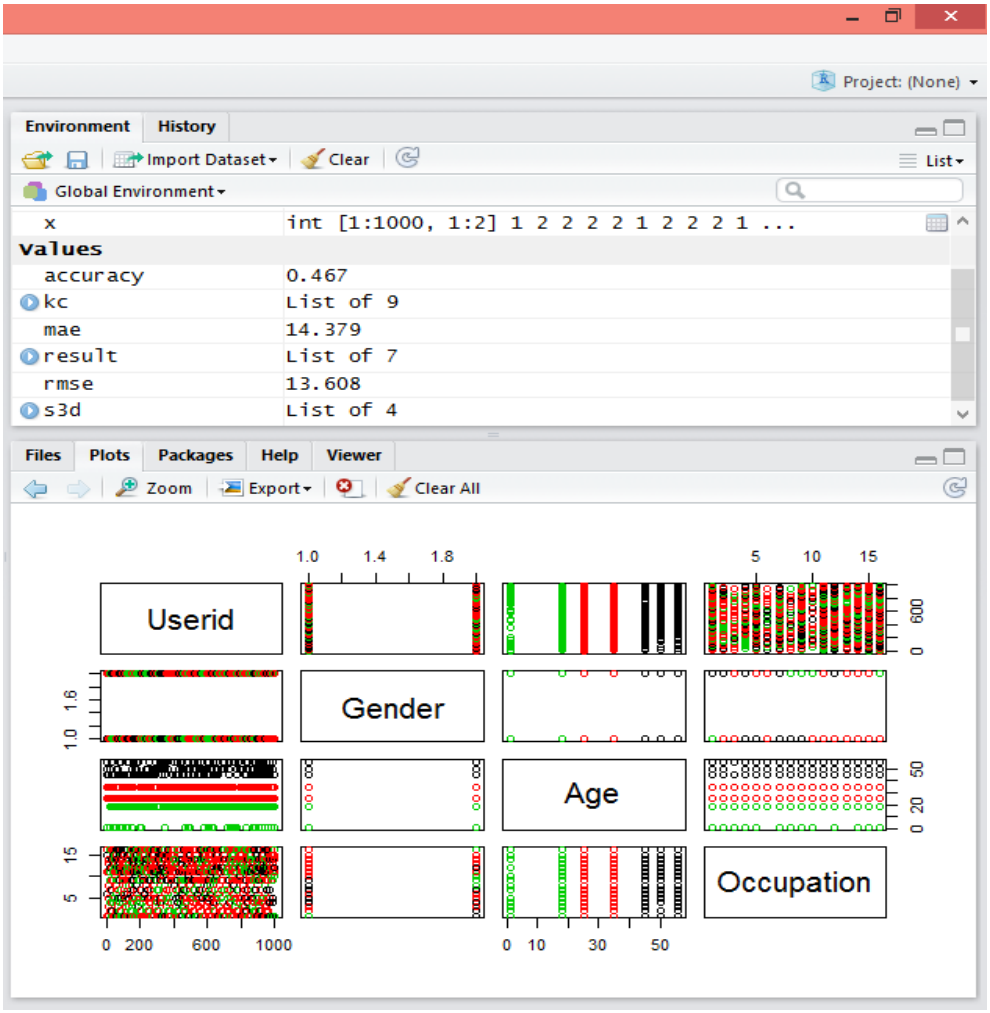


Figure 12(b) Summary of experimental results shown in 12(a)

5.4 Experiment II: Comparing k-means clustering approach and fuzzy c-means clustering approach on the basis of evaluation metrics

Both k-means clustering approach and fuzzy c-means clustering approach is applied on the user dataset by varying the size of the dataset (i.e. the number of users in the dataset). Therefore, mean absolute error (MAE), root mean square error (RMSE) and the accuracy of the clusters being formed has been calculated for both the approaches. Thus, it has been shown that fuzzy c-means clustering approach is more accurate for larger size of the dataset as compared to k-means clustering approach. Table 3 below shows MAE, RMSE and accuracy values for different number of users for k-means approach and fuzzy c-means approach. Also, the graphs have been plotted for these values of MAE, RMSE and accuracy which are shown in figures 17, 18 and 19 respectively.

Table 3 Comparison statistics of both clustering approaches on user.csv dataset

Number of users	MAE		RMSE		Accuracy	
	k-means	Fuzzy c-means	k-means	Fuzzy c-means	k-means	Fuzzy c-means
10	15.25	15.55	23.80	23.51	0.5	0.45
100	13.965	14.025	21.541	21.594	0.485	0.45
500	14.407	14.437	21.90	21.906	0.478	0.457
1000	14.611	14.417	21.970	21.898	0.448	0.467
2500	14.706	14.660	22.647	22.294	0.4828	0.4858
5000	14.839	14.729	22.420	22.227	0.4703	0.4912
6000	15.060	14.618	22.151	22.033	0.4542	0.4822

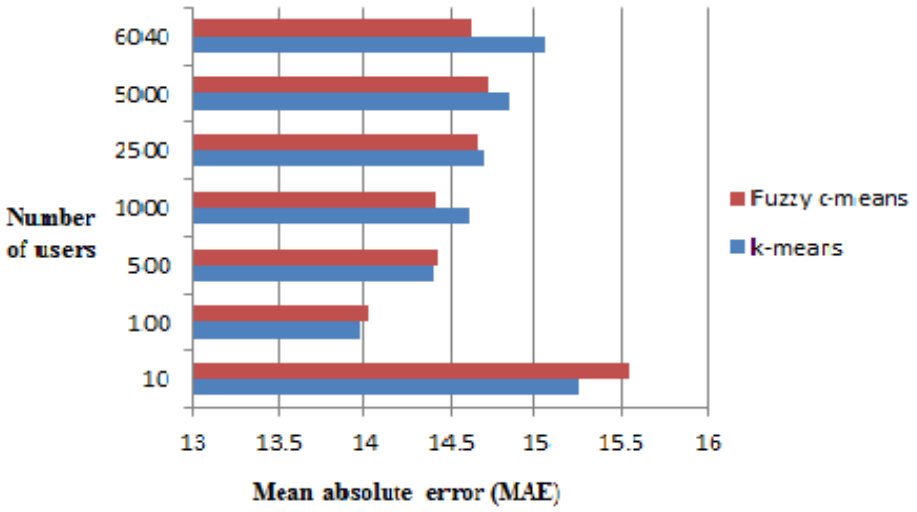


Figure 13 MAE comparison of c-means and k-means technique

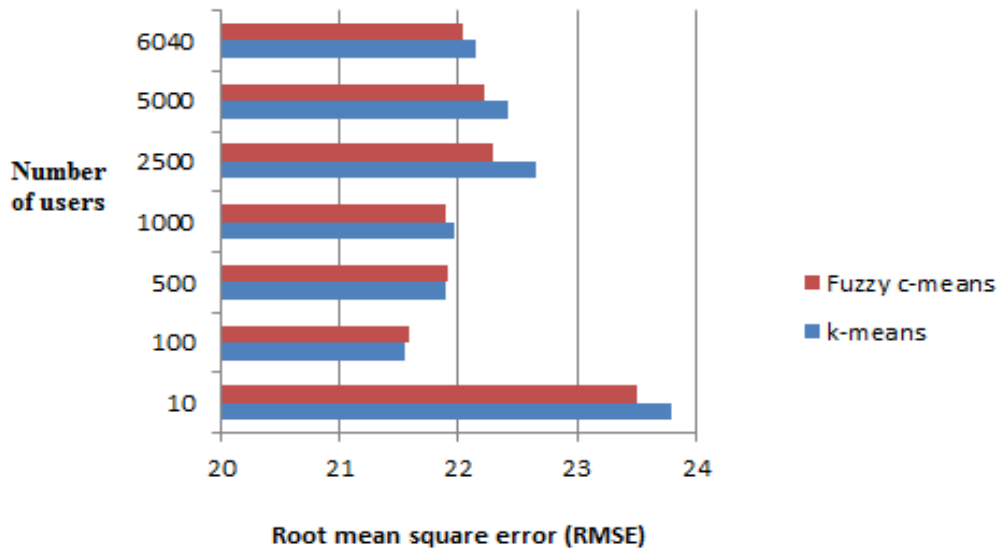


Figure 14 RMSE comparison of c-means and k-means technique

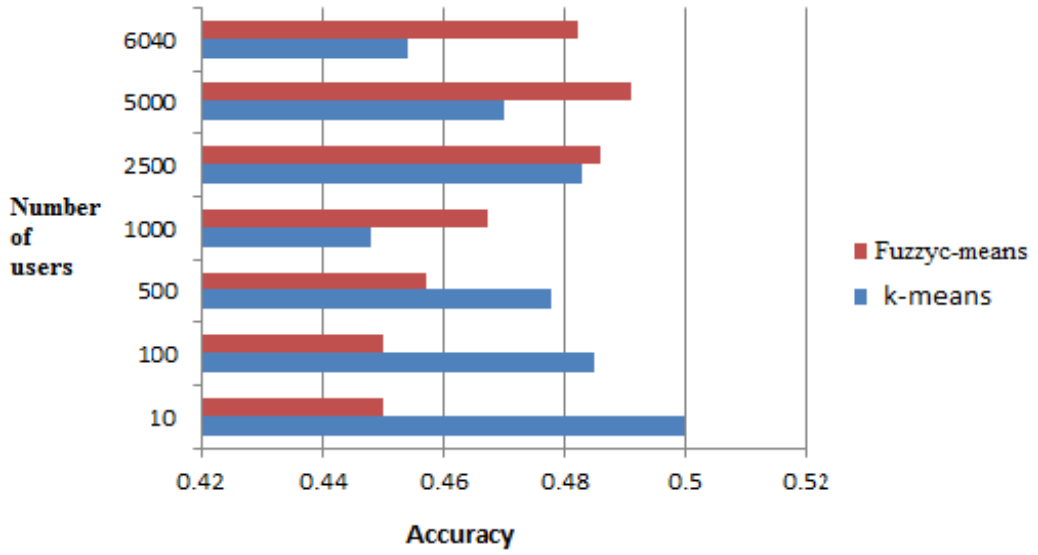


Figure 15 Accuracy comparison of c-means and k-means technique

5.5 EXPERIMENT III: Generating recommendations using MySQL

In the above sections, users are clustered into different groups or clusters by applying fuzzy c-means clustering approach based on their features (i.e. gender and age) which they will enter at the time of login into the system. Thus, the cluster's recommendations are obtained to which the user belongs to by using the preprocessed data which will be stored in MySQL database for generating the recommendations. Therefore, the user gets the appropriate recommendations. The sample dataset

consisted of 500 records from user.csv and saved as userphp.csv but movie movie.csv is taken as it is with 3900 movies. The rating.csv file contains user rating records corresponding to movie ratings given by 500 users on 3900 movies and rest of the records are not considered. The tables under experimentation are imported using phpMyAdmin and under a new MySQL database moviedb.

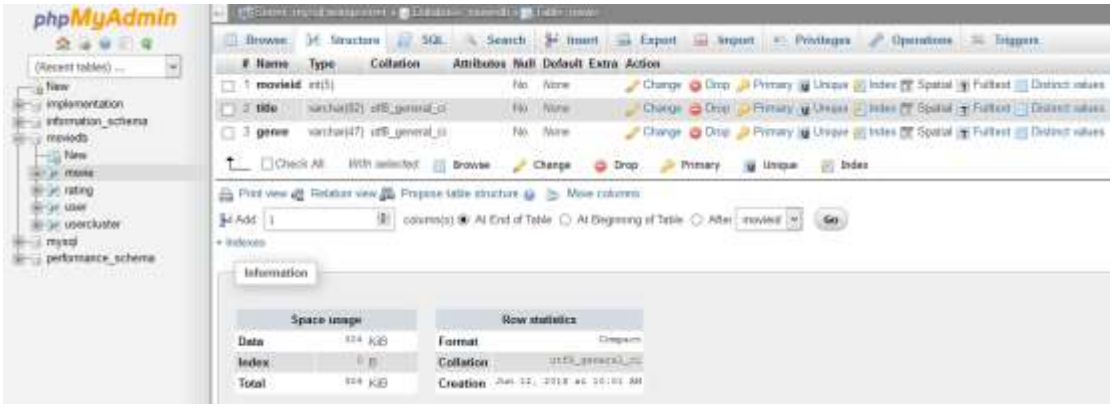


Figure 16 Schema of movie data table

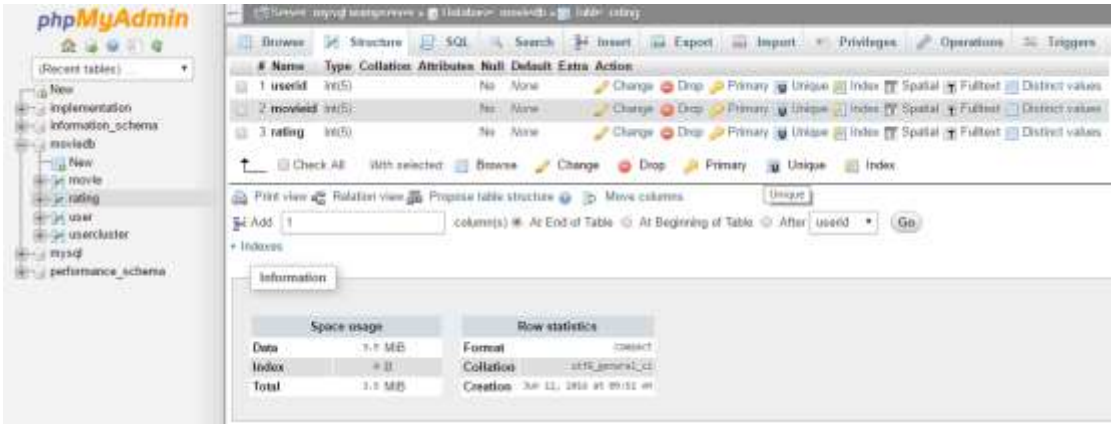


Figure 17 Schema of rating data table

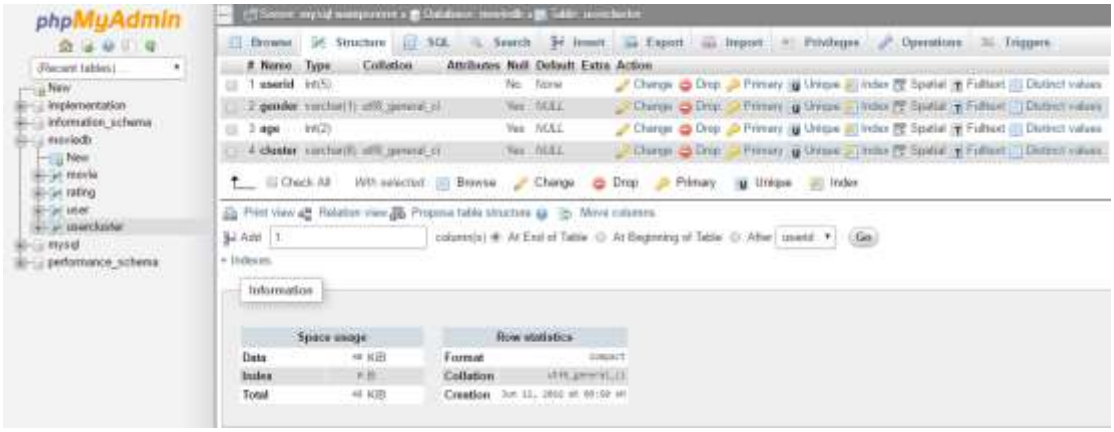


Figure 18 Schema of usercluster data table

Rating table is having two foreign keys i.e. userid which refers to userid of usercluster table and movieid which refer to movieid of movie table.

Query: To find out the user rating based movie recommendation

SQL Query:

Select rating.movieid, movie.title, rating.rating, COUNT (rating.userid) AS count from JOIN movie ON ratin.movieid = movie.movieid where rating.rating= '3' GROUP BY rating.movieid ORDER BY count DESC limit 0, 20

movieid	title	rating	count
648	Mission: Impossible (1996)	3	65
480	Jurassic Park (1993)	3	61
1580	Men in Black (1997)	3	56
2628	Star Wars: Episode I - The Phantom Menace (1999)	3	54
1544	Lost World: Jurassic Park (1997)	3	52
3755	The Perfect Storm (2000)	3	51
1210	Star Wars: Episode VI - Return of the Jedi (1983)	3	48
377	Speed (1994)	3	47
2054	Honey, I Shrunk the Kids (1989)	3	46
367	The Mask (1994)	3	45
2174	Beetlejuice (1988)	3	43
2916	Total Recall (1990)	3	43
380	True Lies (1994)	3	43
1370	Die Hard 2 (1990)	3	42
3793	X-Men (2000)	3	40
1377	Batman Returns (1992)	3	40
3623	Mission: Impossible 2 (2000)	3	40
1265	Groundhog Day (1993)	3	39
2987	Who Framed Roger Rabbit? (1988)	3	39

Figure 19 Query result showing count of users who rated '3' to a particular movie

Query: To find out the user rating based movie recommendation after filtering the data on the basis of cluster information for the new user in the system whose rating record is not available.

SQL Query:

Select rating.movieid, rating.rating, movie.title, count (rating.userid) AS count from rating LEFT JOIN usercluster ON rating.userid = usercluster.userid LEFT JOIN

movie ON rating.movieid = movie.movieid where rating.rating = '3' AND rating.userid IN (select userid from usercluster where cluster = '2') GROUP BY rating.movieid ORDER BY count DESC limit 0, 20

movieid	rating	title	count
1375	3	Star Trek III: The Search for Spock (1984)	13
1210	3	Star Wars: Episode VI - Return of the Jedi (1983)	11
1580	3	Men in Black (1997)	11
648	3	Mission: Impossible (1996)	11
377	3	Speed (1994)	11
1127	3	Abyss (1989)	11
2858	3	American Beauty (1999)	10
2174	3	Beetlejuice (1988)	10
296	3	Pulp Fiction (1994)	10
380	3	True Lies (1994)	10
2916	3	Total Recall (1990)	10
1270	3	Back to the Future (1985)	10
2628	3	Star Wars: Episode I - The Phantom Menace (1999)	10
3698	3	The Running Man (1987)	9
3751	3	Chicken Run (2000)	9
480	3	Jurassic Park (1993)	9
457	3	The Fugitive (1993)	8
1584	3	Contact (1997)	8
2640	3	Superman (1978)	8

Figure 20 Query result showing count of users who rated '3' to a particular movie in a cluster to which a new user belongs

5.6 Top-N Recommendations

There is a difference in recommendation preferences given for new user whose details

are suppose:

userid	age	gender	occupation
501	M	25	other

The new user belongs to cluster 3 as per fuzzy c-means clustering algorithm based results using RStudio on user.csv dataset.

Table 4 Comparison between movie recommendations based on query results 1 and 2

Movieid from first query (on the basis of user information)	Movieid from second query (on the basis of clusters formed using demographic information)
648	1375
480	1210
1580	1580
2628	648
1544	377

Now consider recommendations are provided gender specific i.e. for male or female for the new user with specific details given below:

SQL QUERY:

Select rating.movieid, rating.rating, COUNT (rating.userid) AS male_count From rating JOIN usercluster ON rating.userid= usercluster.userid where rating.rating= '3' AND rating.userid IN (select userid from usercluster where(cluster= '3' AND gender= 'M')) group by rating.movieid order by male_count DESC limit 0, 30

movieid	rating	male_count
1544	3	34
1580	3	33
648	3	32
480	3	30
3755	3	29
367	3	28
1370	3	27
2054	3	26
2628	3	25
1573	3	25
1377	3	25
1527	3	24
377	3	24
3623	3	24
2002	3	23
2174	3	22
21	3	22
2706	3	22
500	3	21

Figure 21 Recommendations for new user

Hence, the proposed system recommends movie choices on the basis of the user rating information as shown in figure 21. Further, other relevant attributes specific to problem domain can be considered for improving the quality and relevance of recommendations as per new user in the system.

Chapter 6

Conclusion and Future Scope

The proposed approach deals with cold start problem by applying fuzzy c-means clustering and generating Top N recommendations for a user who enters the system. In a user item rating matrix with m users and n items, the order of complexity in giving recommendations would be O(mn). The number of ratings is fairly high as compared to the number of users. In the proposed method, the dataset is grouped into clusters by analyzing a group of demographic attributes (i.e. gender and age) and providing a neighborhood of similar users on this basis. Also, fuzzy c-means technique works better than the traditional k-means clustering approach when the set of users is large. Thus accuracy is improved by using fuzzy c-means with regard to clustering and subsequent recommendation generation. These records constitute neighborhood and as a consequence the search space for recommendation generation has reduced to great extent.

Top N recommendations are generated by taking into account the frequency of users who have rated a particular movie with a particular score i.e. highest score of 5. Rating distribution and aggregate analysis like highest rating frequency or average rating for a movie also decide the quality of recommendations. This is done using phpMyAdmin and MySQL.

As future work, more clustering techniques can be exploited, which provide better accuracy for generating recommendations. The demographic attribute collection of users can be expanded like including the cultural background or likes/dislikes of a user. Such kind of personality diagnosis can help to improve recommendation generation for people of similar preferences.

Other problem domains like new item cold start problem could be subject to this methodology. The dataset could be extended to real time and social tagging using social networks could be done to collect more information from users.

References

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

[2] http://recommender-systems.org/collaborative-filtering/

[3] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms”, in Proceedings of the 10th International Conference on World Wide Web, WWW ’01, New York, NY, USA: ACM, pp. 285–295, 2001.

[4] D. Asanov, “Algorithms and Methods in Recommender Systems”, Berlin Institute of Technology, Berlin, Germany, 2011.

[5] S.O. Ojo, S. M. Ngwira, K. Zuva and T. Zuva, “A Survey of Recommender Systems Techniques, Challenges and Evaluation Metrics”, International Journal of Emerging Technology and Advanced Engineering, vol. 2, no. 11, pp. 1-5, Nov 2012.

[6] Ron Zacharski. A Programmer Guide to Data Mining: The Ancient Art of the Numerati, GuideToDataMining.com, 2015.

[7] X. Su and T.M. Khoshgoftaar, “A Survey of Collaborative Filtering Techniques”, Advances in Artificial Intelligence, vol. 2009, Aug 2009.

[8] http://recommender-systems.org/content-based-filtering/

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

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

[11] How to Evaluate Machine Learning Models: Classification Metrics. [Online]. http://blog.dato.com/how-to-evaluate-machine-learning-models-part-2a-classification-metrics

[12] G. Adomavicius and A. Tuzhilin, “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.

[13] Akshita and Smita, “Recommender System: Review”, International Journal of Computer Applications, vol. 71, no. 24, pp. 38-42, June 2013.

[14] U. Shinde and R. Shedge, “Comparative Analysis of Collaborative Filtering Technique”, IOSR Journal of Computer Engineering, pp.77-82, Mar.-Apr. 2013

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

[16] M. K. K. Devi, R. T. Samy, S. V. Kumar, and P. Venkatesh, “Probabilistic neural network approach to alleviate sparsity and cold start problems in collaborative recommender systems”, in IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–4, Dec 2010.

[17] G. Shaw, Y. Xu, and S. Geva, “Using association rules to solve the cold-start problem in recommender systems”, Lecture Notes in Computer Science, vol. 6118, pp. 340-347, June 2010.

[18] H.Sobhanam and A. K. Mariappan, “Addressing cold start problem in recommender systems using association rules and clustering technique”, in International Conference on Computer Communication and Informatics (ICCCI), 2013, pp. 1–5, Jan 2013.

[19] U. Gupta and N. Patil, “Recommender system based on hierarchical clustering algorithm chameleon”, in IEEE International Advance Computing Conference (IACC), pp. 1006–1010, June 2015.

[20] D. Sun, Z. Luo, and F. Zhang, “A novel approach for collaborative filtering to alleviate the new item cold-start problem”, in 11th International Symposium on Communications and Information Technologies (ISCIT), pp. 402–406, Oct 2011.

[21] J. Basiri, A. Shakery, B. Moshiri, and M. Z. Hayat, “Alleviating the cold start problem of recommender systems using a new hybrid approach”, in 5th International Symposium on Telecommunications (IST), pp. 962–967, Dec 2010.

[22] T. T. Dang, T. H. Duong, and H. S. Nguyen, “A hybrid framework for enhancing correlation to solve cold-start problem in recommender systems”, in Seventh IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), pp. 1–5, Dec 2014.

[23] L. H. Son, K. M. Cuong, N. T. H. Minh, and N. V. Canh, “An application of fuzzy geographically clustering for solving the cold-start problem in recommender

systems", in International Conference of Soft Computing and Pattern Recognition (SoCPaR), pp. 44–49, Dec 2013.

[24] S.K. Verma, N. Mittal and B Agarwal , "Hybrid Recommender System based on Fuzzy Clustering and Collaborative Filtering", 4th International Conference on Computer and Communication Technology, 2013.

[25] H. Shivhare, A. Gupta and S. Sharma, "Recommender system using fuzzy c-means clustering and genetic algorithm based weighted similarity measure", IEEE International Conference on Computer, Communication and Control, ICA-2015.

[26] C. Budayan, I. Dikmen, and M. T. Birgonul, "Comparing the performance of traditional cluster analysis, self-organizing maps and fuzzy c-means method for strategic grouping", Expert Systems with Applications, vol. 36, no. 9, pp. 11772 – 11781, 20019.

[27] Q. Shambour and J. Lu, "A hybrid multi-criteria semantic-enhanced collaborative filtering approach for personalized recommendations," in International conference onWeb Intelligence and Intelligent Agent Technology (WI-IAT), vol. 1, pp. 71–78 , Aug 2011.

[28] D.J. Bohra and A.K. Gupta, "A Comparative study Between Fuzzy Clustering Algorithm and Hard Clustering Algorithm", International Journal of Computer Trends and Technology (IJCTT), vol. 10, no. 2, pp. 108-113, April 2014.

[29] https://en.wikipedia.org/wiki/Cluster_analysis

[30] https://sites.google.com/site/dataclusteringalgorithms/fuzzy-c-means-clustering-algorithm

[31] https://en.wikipedia.org/wiki/RStudio

[32] www.grouplens.org. [Online]. http://grouplens.org/datasets/movielens/

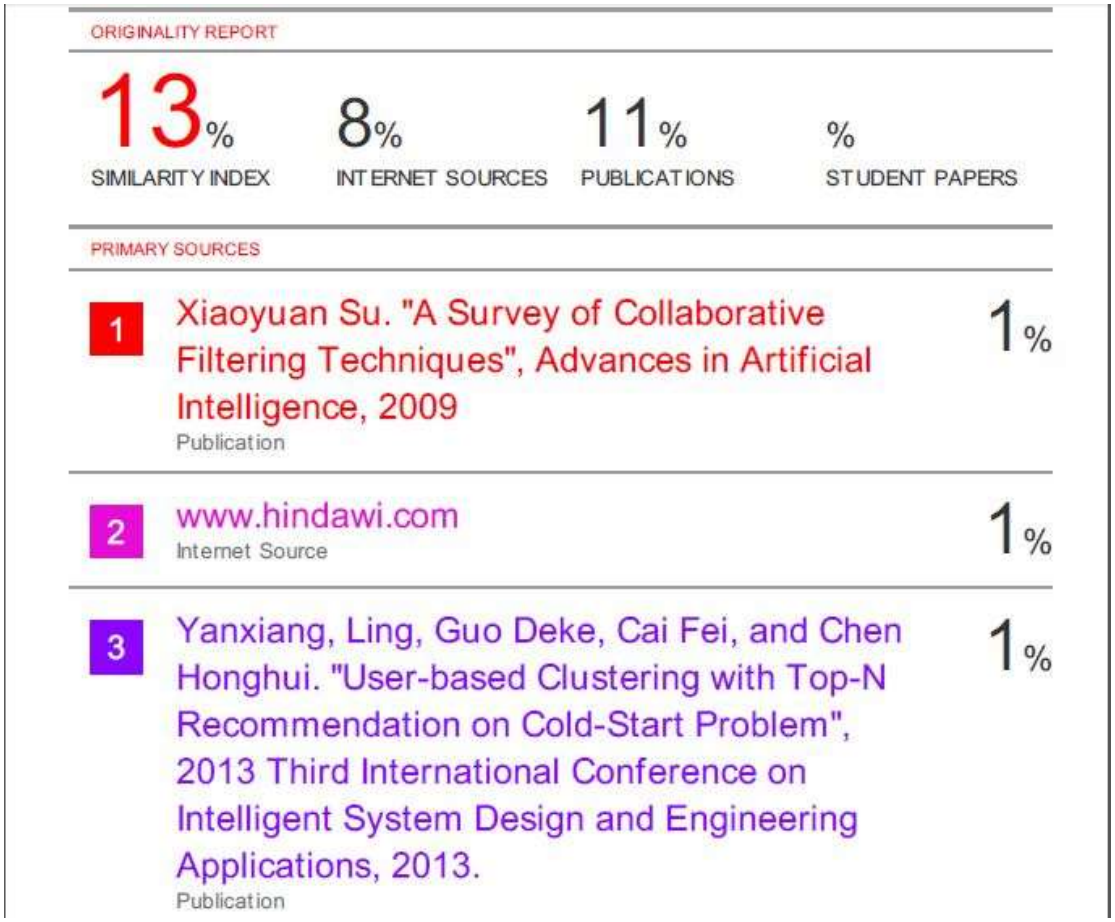
List of Publications

- Sugandha Gupta, Shivani Goel, "Handling User Cold Start Problem In Recommender Systems Using Fuzzy Clustering," in International Conference on ICT for Sustainable Development (ICT4SD 2016). [Accepted] to be held from July 1-2 at Vivanta Hotels and Resorts, Panaji, Goa.

Video Link

https://www.youtube.com/channel/UC8zCZWNKVo62Jy3u4HRn1iA

Plagiarism Report



ORIGINALITY REPORT

13%	8%	11%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Xiaoyuan Su. "A Survey of Collaborative Filtering Techniques", Advances in Artificial Intelligence, 2009 Publication	1%
2	www.hindawi.com Internet Source	1%
3	Yanxiang, Ling, Guo Deke, Cai Fei, and Chen Honghui. "User-based Clustering with Top-N Recommendation on Cold-Start Problem", 2013 Third International Conference on Intelligent System Design and Engineering Applications, 2013. Publication	1%