

**RECOMMENDATION BASED APPROACH USING FCM**  
**IN**  
**E-COMMERCE SYSTEM**

MAJOR PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE AWARD OF DEGREE OF

Master of Technology

In

Information Systems

Submitted By:

**HITESH NIRWAN**

**(2K12/ISY/12)**

Under the Guidance of

**Prof. O. P. Verma**

(Head, Department of IT)



DEPARTMENT OF INFORMATION TECHNOLOGY  
DELHI TECHNOLOGICAL UNIVERSITY  
(2012-2014)

# CERTIFICATE

---

This is to certify that **Mr. Hitesh Nirwan (2K12/ISY/12)** has carried out the major project titled **“Hybrid Recommendation System in E-Commerce using FCM”** as a partial requirement for the award of Master of Technology degree in Information Systems by Delhi Technological University.

The major project is a bonafide piece of work carried out and completed under my supervision and guidance during the academic session **2012-2014**.

The matter contained in this report has not been submitted elsewhere for the award of any other degree.

(Project Guide)

**Prof. O.P.Verma**

Head, Department of Information Technology

Delhi Technological University

Bawana Road, Delhi-110042

# ACKNOWLEDGEMENT

---

I express my gratitude to my major project guide **HOD Dr. O.P.Verma, IT Dept.** Delhi Technological University, for the valuable support and guidance he provided in making this project. It is my pleasure to record my sincere thanks to my respected guide for his constructive criticism and insight without which the project would not have shaped as it has.

His suggestion and advice proved very valuable throughout. I humbly extend my words of gratitude to other faculty members of this department for providing their valuable help and time whenever it was required. I am thankful to all teaching or non-teaching staff of DTU and my fellows, who have helped me in the completion of this thesis report.

Hitesh Nirwan

Roll No 2K12/ISY/12

M.Tech (Information Systems)

Department of IT

E-mail: hiteshnirwan@gmail.com

# ABSTRACT

---

Aim of recommendation system is to predict the level of preferences of users towards some items, with the purpose of suggesting ones that they may like, among the sets of items they have not visited yet. In other words the aim of recommendation system is to improve the user experience by suggesting items the user may purchase. The recommendation system targets the personalization aspects of each user. It provides the items in the recommendation list according to particular customers and improve the user experience and for this it exploits the user specific interest and the preferences by his profile. Recommendation system applies in E-Commerce, Web search and the advertising fields. The main aim of recommendation is to increase the sale in E-Commerce System. Recent researches have proved that a good recommendation system always increases the sale by good amount of percentage.

Recommendation System also suffers from some problems. It depends upon the following factors:

- If the user request is not clear and result to multiple interpretations then this can result to the collection the responses for each of those meanings, and then deciding which result we can present to the user.
- If the retrieved results are very similar to each other and if one of them is relevant then we have to present all of them, this can present lots of unnecessary data
- The profiles, needs, and activities of different users can be very different and that's why we have to make a system that can response to all type of queries submitted to that.

We can solve these kinds of problems by classification and clustering techniques and for this purpose we use the clustering approach to perform the classification of user profile, interests and the items. It performs the partitioning of the data from the large data set to produce the short representation of the recommendation system's behaviour.

Fuzzy c-means (FCM) is a data clustering technique in which we divide the dataset into n number of clusters and each data point in the dataset belongs to every cluster in the group of clusters to a certain degree of membership. If a data point lies close (near) to the centre of the cluster, then it will have a high degree of membership to that cluster and another data point which lies away from the centre of the cluster would have the low degree of membership to that cluster. It allows belongingness of one data point to more than one cluster.

# LIST OF FIGURES

---

<b><u>Fig No</u></b>	<b><u>Title</u></b>	<b><u>Pg. No</u></b>
1.1	Earlier Recommendation System	3
1.2	High Level Architecture of A Content-Based Recommender	6
1.3	A Schematic Overview of Collaborative Filtering	8
2.1	Example of Hierarchical Clustering	16
2.2	Initialize Representatives (“Means”)	18
2.3	Assign To Nearest Representatives	19
2.4	Re-Estimate Means	19
3.1	Data Point Distributed On an Axis	24
3.2	Membership Function Diagram in K-Means	24
3.3	Membership Function Diagram in FCM	25
3.4	FCM Clustering Example	28
3.5	Membership Function Matrix of Hard and Fuzzy C-Mean Clustering	29
3.6	Hard Clustering	30
3.7	Fuzzy Clustering	30
4.1	Content-Based Approach	34
4.2	Collaborative Filtering Process	36
4.3	Neighborhood Formation from Clustered Partitions	37
4.4	Proposed Hybrid Recommendation System	39
5.1	Login System	42
5.2	Registration Page	43
5.3	Search Example 1	43
5.4	Search Example 2	44

5.5	Search Example 3	44
5.6	Search Example 4	44
5.7	Search Example 5	45
5.8	Search Example 6	45
5.9	Previous Search Output	46
5.10	Item Detail and Rating System	47
5.11	Search Example 7	48
5.12	Search Example 8	48

# TABLE OF CONTENTS

---

Certificate.....	ii
Acknowledgement.....	iii
Abstract.....	iv
List of figures.....	v
<b>Chapter 1: Introduction to Recommendation System.....</b>	<b>1</b>
1.1 Motivation.....	1
1.2 Personalization in Context.....	2
1.3 Evaluation.....	10
1.4 Literature Review.....	11
<b>Chapter 2: Clustering.....</b>	<b>13</b>
2.1 Introduction.....	13
2.2 Clustering Types.....	14
<b>Chapter 3: Fuzzy C-Mean.....</b>	<b>22</b>
3.1 Introduction.....	22
3.2 Comparison.....	29
<b>Chapter 4: Proposed Methodology.....</b>	<b>33</b>
4.1 Content Based Approach.....	33
4.2 Collaborative Based Approach.....	35

4.3 Knowledge Based Approach.....	38
4.4 Proposed Hybrid Recommendation System.....	39
<b>Chapter 5: Result &amp; Analysis.....</b>	<b>42</b>
5.1 Different Approaches Output.....	42
5.2 Conclusion & Future Work.....	49
<b>References.....</b>	<b>50</b>



# Chapter 1

## Introduction to Recommendation System

---

### 1.1 Motivation

In today's world more and more people are moving towards E-Commerce system to purchase the items over the internet. It is because; first it provides them the ease to buy the items by sitting in their home and second allows them to compare the same and different products in no time. For example they can compare the features and price of mobile phone of two different companies. This helps them in purchasing the item with more value to them.

So the user comes back to the same E-Commerce website based upon its experience and the experience is made by two things, price of different items and a good recommendation system. It helps the E-commerce website owner to do one to one marketing.

Recommendation, suggesting items that, they may like, among a set of items about those they have not think yet. To compete with the other companies in the E-Commerce world we would need a good recommendation system. It would not only increase the cross sales rather also increase the company's reputation in the user mind by suggesting good relevant item. Only those companies can survive in the today's world which not only provide item with less price but also suggesting more and more relevant item, so the user can also purchase them.

In the earlier time the recommendation system provided the same recommendation list to all users. Individual recommendation was not there in the beginning. Then as the time progressed, competition increased and with the increment in completion, personalization aspect came into the E-Commerce world. The E-Commerce websites started to focus on the individual needs, to increase the sales. Personalization is the major revolutionary factor in the E-Commerce world. It increased the profit of E-Commerce websites very much.

The E-Commerce is in such a state where it has millions of products to sale. The user can not be able to go through all the items and check it. In this case recommendation system helps. Aim of the recommendation is "calculate level of likeliness of the customer towards some objects", with the purpose of suggesting the ones that they may like, among a set of items that they have not visited yet and purchase those things also.

E-Commerce sites try to provide personal experience to each user and give the real time update to the user without compromising the quality of the recommendation. It adapts to the specific requirements of each user and provides the corresponding results, so adapting as per user requirements so as the user changes the recommendation list also changes.

The most important job of a good recommending system is to maintain the up-to-date preferences and interests of the customer and provide them updated recommended lists. Maintaining and exploiting the customer.

Advantage of the recommendation is that we can vary it from customer to customer. If the user requirements and interests changes then the system also changes the recommendation list. It provides more options to the user. It makes the experience of the use more personalized, because of this the recommendation system have become the basic feature in the E-Commerce system

## **1.2 Personalization in Context**

The main problems which need to know in the personalization approach are following:

- Representation
- Acquisition
- Exploitation of the user profiles.

The basis of recommending system is the user profile, which can have both the explicit demographic information of the customer (e.g. location, age, sex, etc.) plus the behaviour of customer.

This user profile can be acquired automatically by observing the interaction of the user with system. Representation and exploitation of customer behaviour is not an easy task as preferences of the user are generally vague (“I like the travelling”), complex (“I like travelling to Arab countries but only when their government are stable”) or may be complex (“I like the animals, but not anything that looks like a mouse”).

It performs three basic functions; first it converts browser into buyers, second it increase the cross sale and third and the last but very important is that it builds the loyalty. Higher the no of visitor’s turns into the real customer who purchases the item higher would be the profit.

Similarly the cross sale performs. E.g. a user X comes to purchase an item Y but find that the item in the recommending list also matches his preferences then there is high probability that the customer will also spend his money on that item too. It would increase the company sales.

Building the loyalty in the customer is very important in today's competitive world. User can easily move toward by a single click of his mouse. If the company able to maintain the loyalty in their customer they would not move towards any other website

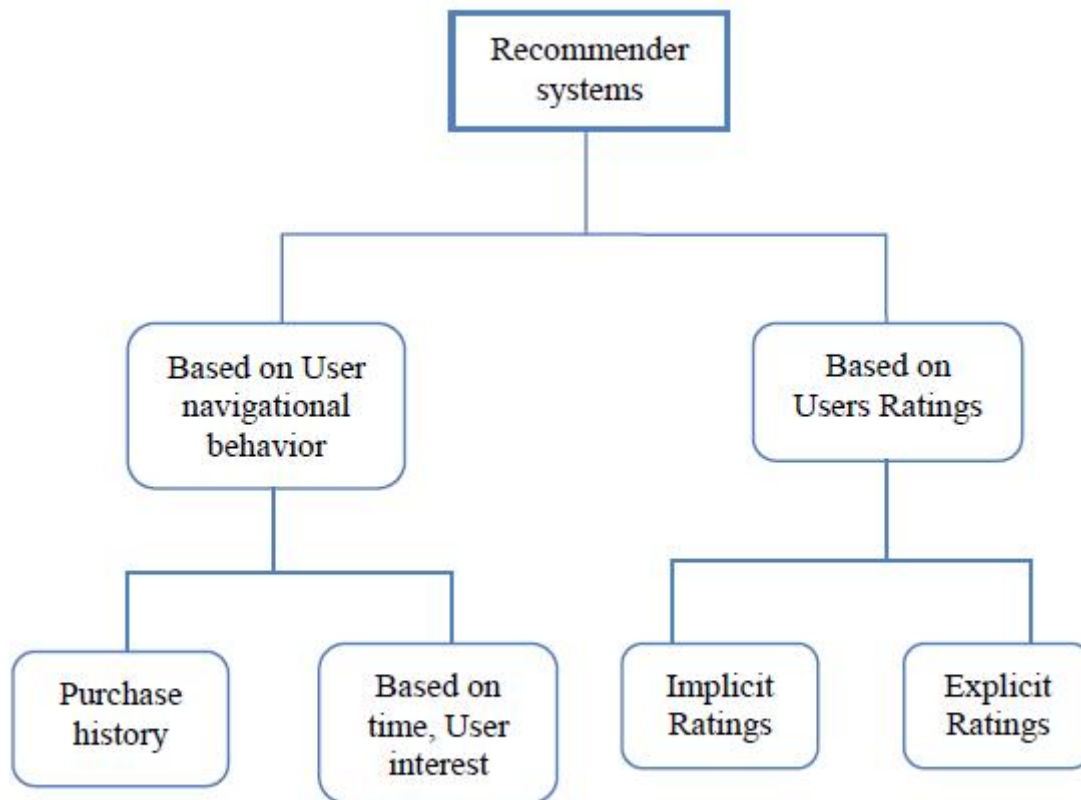


Fig-1.1 Earlier Recommendation System

A good recommender system can use these information to improve the result (improve the recommendation list relevant item). For example person on specific country when search for a novel then it is highly likely that the user may preference novel of his country author as compared to the other country author

Sometimes we can have a full user profile and sometimes we have to use the partial information only. For example, when the user is not signed in to our website, we can have only limited behavioural information by monitoring which links are user clicking and on which page user is staying long and in these cases no demographic details can be exploited. Better the user profile, the recommendation system will give us more accurate suggestions.

User profile can have different types of information. It includes the user preferences so the interest of the user in different items. To increase the efficiency and the removal of loads of data the system can select the select the top preferences only. User profile also contain the user history, so the user interaction with the items in the past. So, the purchase history and the ratings by the customer to the different things.

Property of good recommending system is that it must be able to adapt, as recommendation is a dynamic approach. It must be changed according to time, user preferences and interests. New value must be added and the old value must be deleted as the time progresses.

It must also be efficient. It must represent the good list. If the system is not able to do so then it is of no use.

To improve the performance we would do it using hybrid approach so all of the following would be considered:-

- **Collaborative Based Filtering:** - The idea behind this system is that if two or more users share the same interest in the past then there is high probability that they will also share the same interest in the future. By this method we can know the probability of the purchase of an item by keeping in mind the purchase of the same item by the other similar user (the user with which purchase history is very similar).
- **Content Based Recommendation:** - If the user purchases an item, then the recommendation system will present those items in the recommendation list which shares the same information and the features so those item will be displayed which will have high commonality in the features.
- **Knowledge Based Recommendation Systems:** - In some domain the user does not buy item regularly, generally they buy an electronic item once or twice in a year so in those cases user history is not available. To suggest any item we would need knowledge base in this case. Thus the system needs additional knowledge for the recommendation purpose. We can use constraint based approach. We can limit the constraints. We can also use explicit constraint approach, in which we present item in which some features are relevant. We can also add user profile in the constraint based approach to improve its efficiency and stopping showing of same items to every user just based upon the features.

We can also get the user feedback directly by interacting with the user directly as they can give their requirements explicitly.

The content-based techniques can be divided into the following categories:

- **Search-Based Recommendation:** - This technique selects some of the basic results based on general aspects for example, number of sales, number of clicks, and the number of views. This solution is very easy to understand and requires some of basic statistics on the objects.
- **Classifiers:** - In this we can use the classifier system for building a good recommendation system. We can make the classes of items and based on the classes we can show the recommended result to the users. They can be combined with other recommendation methods.
- **Category-Based Recommendation:** - In this we consider every item in the search base as it belongs to one or more categories. When we choose an item, the system selects categories of interest which would be based upon the user previous history over the items or the categories and then the system would select the most visited items from the selected categories, which would be presented. This technique requires the maintenance of user history.
- **Semantics-Based Recommendation:**- This is an advancement in the previous approach, as it replaces the categories of items by some arbitrarily complex domain models to describe the semantics of objects (for example insectology, grammar, and any other kind of conceptual model) and matches the description of these objects with the semantic models of the users.
- **Information Filtering:**- This technique work on the principal of the fetching the syntactical information of the objects and/or semantic knowledge about their types or categories. The collected information can be unstructured, natural language text or structured descriptions, organized in typed attributes. When a user finds any interest in an object, then the system will recommends those objects that are more similar to it. The similarity will depend upon the type of information. Benefit of this kind of approach is that the system does not need to main the user database and check it.

Fundamentally content based recommendation depends upon the presence of descriptions (tags) of item and profile which assigns the importance to these characteristics.

To illustrate this a high level of content based approach is shown in the following diagram.

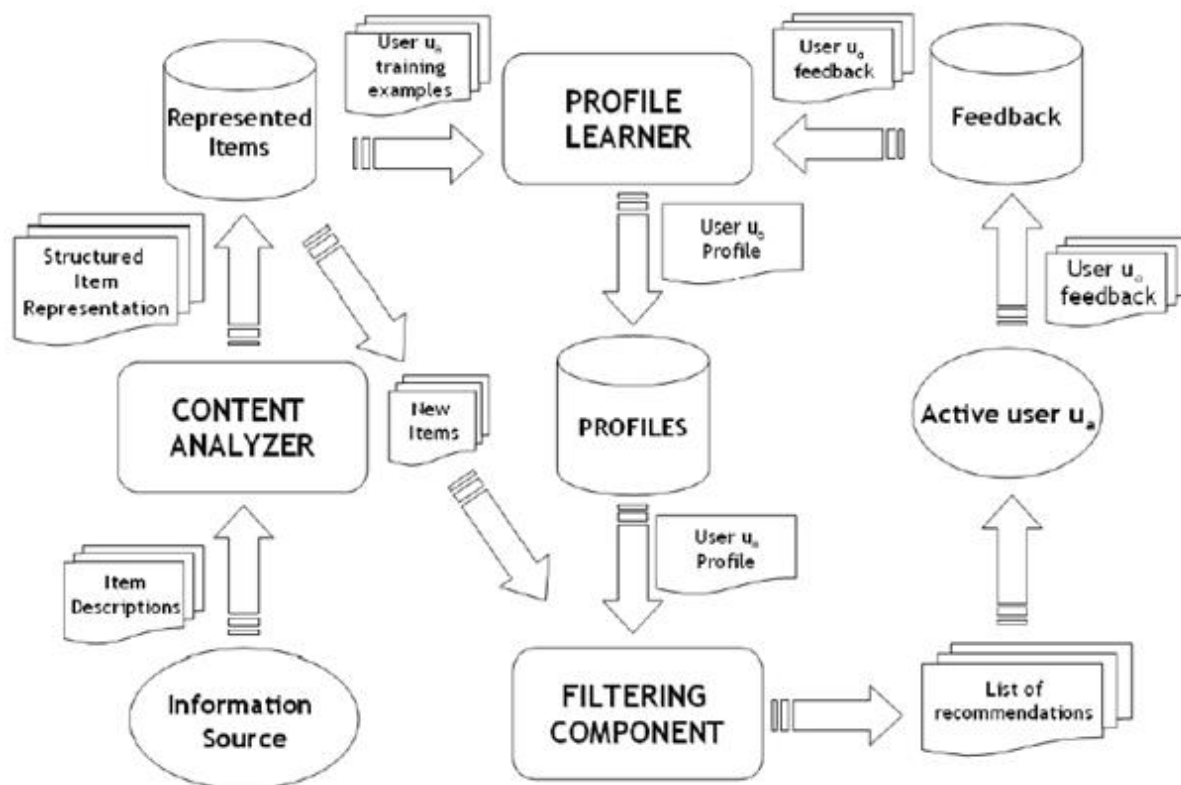


Fig-1.2 High level architecture of a content-based recommender

## Collaborative Filtering Techniques

This is the most famous technique. This technique requires maintaining a database of as many user ratings of items as possible. The technique works on the principal that if some users (at least two) ratings strongly correlate with the each other than recommend those items which are highly rated by other users.

This technique requires Boolean values for the rating purpose, this represent either user purchased the item or not for example 1 would suggest item purchased and 0 would indicate the item not purchased. This represents that the users explicitly declared their choices for an item that the user actually bought that item, or simply the users went through that item

By using this history of the similar behaviour users (users having overlapping history), the recommendation system generate the clusters and recommend the item based on these clusters.

<i>Customers vs. Items:</i>	<b>Item1</b>	<b>Item2</b>	<b>Item3</b>	<b>Item4</b>	<b>Item5</b>	<b>Item6</b>
<i>Customer 1</i>		X	X			
<i>Customer 2</i>	X			X	X	
<i>Customer 3</i>	X				X	X
<i>Customer 4</i>		X	X	X		
<i>Customer 5</i>	X				X	X
<i>Customer 6</i>		X	X			X
<i>Current customer</i>	X				X	

Table-1.1 Customer and items they purchased in the past

A case is shown in the above table, where the X represents that the fact that user purchased that item for example customer 1 purchased the item 2 and item 3. This system must understand that which items must be suggested to customer. For this, the system would compare other customer's behaviour with the current user and decides which purchases are most relevant and good for the current user. In the current example, the current customer behaves similarly to the customer 2, 3, and 5.

As the current customer behaves like customer 2,3 and 5, the system would recommend the items they bought so item 4 and 6 would be recommended strongly

Higher the similarity between the current customer and the other customers, higher would be its weightage in the recommendation; and higher the number of similar customers that bought a particular item, the higher would be the item's rank in the recommendation list.

We can also understand the collaborative filtering by the following diagram. There are some consumers of different types or preferences. We make the group of same type of users (users whose preferences are similar) and according to product selected and their preferences we recommend products to them.



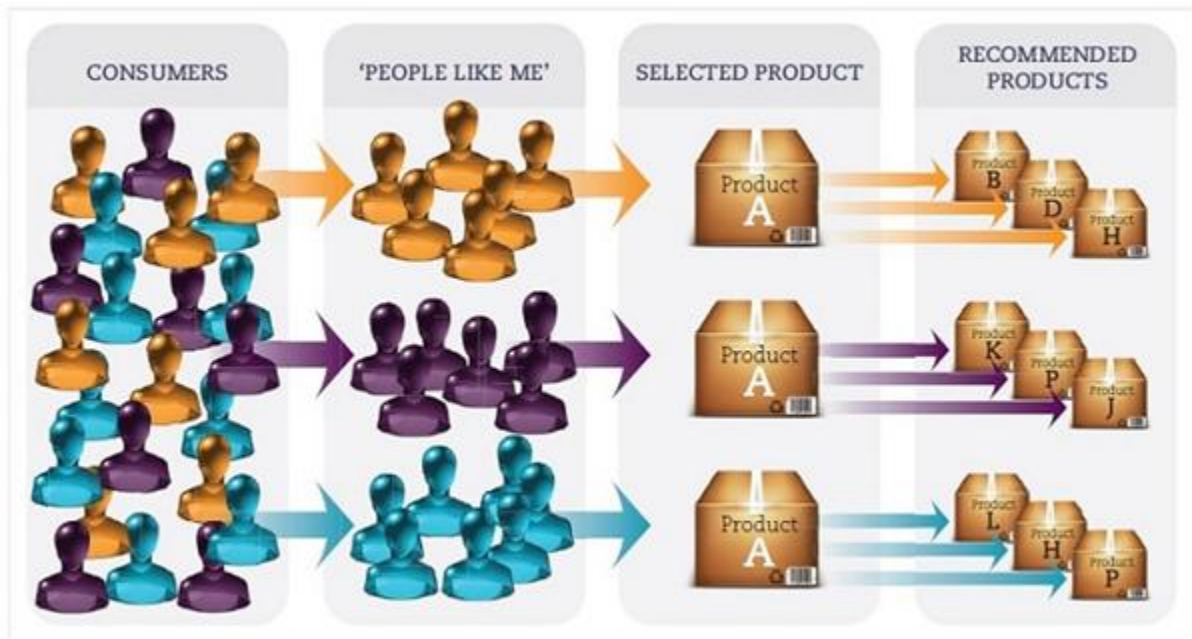


Fig-1.3 A schematic overview of collaborative filtering

Two sub approaches exist in the collaborative filtering technique; User- dependent and the item–dependent. In user- based filtering we require the users whose rating of items and their bought list matches. The collaborative filtering system finds the list of items purchased by the first user and checks the item purchased by the second, remove those items and shows remaining items to the second customer.

The other type of collaborative filtering is item-based collaboration filtering. It first takes the users purchased or rated items and then present the most similar item. E.g. there is an item X, which user purchased and there is an item Y which is very similar to the item X then the system will present the item Y in the recommendation list. It is used in various big E-Commerce websites like Amazon.

This approach is very powerful and efficient, as it able us to deliver very logical recommendations. Obviously, the larger the database and the record of the previous behaviour set, better the recommendations. But because of the volume of the data, it would be difficult to implement it and it ends up like a resource and time consuming approach.



Moreover, this technique would suffer from some well-known problems:

- Cold Start:- In the beginning of system lifecycle, when the machine turns on, there would be no record of any customer past interaction or rating. It makes it impossible to predict any recommendation.
- First Time Rater:- New items that are never purchased would not be recommended by the above technique.
- New User:- The user who has never bought anything can't get any recommendation as they can't be compared to the other users.
- Sparsity:- This is very high because the purchase table is generally very sparsely populated. The typical problem is that, in the database of the millions of products, each customer might have purchased just a few items at most, perhaps visited few items more, and may be rated or commented just one or two items. This problem creates the difficulty in the effective implementation.

To solve these types of issues, we can apply clustering techniques over the users, so aggregating the users that behave almost similarly and hence representing and storing the common past behaviours by the representative selection of the actions. This approach reduces the size of the customer set to be considered for the analysis. Preferences would be given to each cluster, depends upon the likeliness of the users who fits to that cluster. Therefore each user who resides in every cluster would get the recommendations deduced at level of these cluster, and it is very much efficient. As the users can also belong to any number of cluster, that's why clusters can intersect and because of this recommendations are averaged across the clusters based on the weightiness of the participation.

This makes the recommendation at the cluster level slightly less relevant than it is at the individual level, but on the other side it also improves the performance and reduces size of dataset.

### 1.3 Evaluation

Recommender system need to be evaluated on the basis of following parameters:

- 1) Accuracy
- 2) Novelty
- 3) Coverage
- 4) Diversity
- 5) Adaptivity
- 6) Scalability

**Accuracy:** We can divide the accuracy into three types: accuracy of usage predictions, ranking accuracy of the items, and accuracy in the predictions of the given ratings. For different accuracy types, we have to use the different methods of expressing, for instance we generally use Root Mean Squared Error to predict the score, while ordering the objects according to different customer's favorites, we attempt to know the correct order for the collection of the objects for each customer and calculate how near our recommendations comes to that order.

**Novelty:** These recommendations are for those items about which the customer does not know anything. System should also generate the novel items, as these items can also be useful for the customer.

**Coverage:** We can divide the coverage into two different categories. In the first case, part of the objects for which recommendation system can produce the prediction, while in the second case we measure percentage of available item which were ever effectively recommended anyone.

**Diversity:** It is considered as opposite of the similarity. Sometimes when we suggest the some number of similar objects that might not to be valuable to customer, as it can take much time to investigate all the samples from the recommendation list. For example, a same type of 5 product from a single company may not be a good choice as compare to the 5 items from different companies.

**Adaptivity:** There may be cases when the item collection (database) changes rapidly. For example recommendation of the news items. Recommender system must be able to handle these changes efficiently.

**Scalability:** We design the recommender systems help the users in navigating in the large pool of the objects. The effectiveness of system must no degrade as the size of the dataset increases.

## 1.4 Literature Review

Lots of work has been done in this field using various approaches like content based, collaborative based and knowledge based approach.

- 1) **Content based approach:** Lots of work using this approach has been done by various researcher like Ming Chen [5], in this type tags related to any search key item were exploited and these tags were used for recommending the other objects like is the user searches a specific phone then the phone features like its display size, its camera capacity its resolution and other things were used for suggesting the items in the recommendation list. This can be done by rating by different customer to different objects as done by the Vignesh B and Subodh Kant [21]  
System can also question explicitly to the user about his choices, so that system can present the relevant items as explained by the Stefano Ceri Et al. [11]
- 2) **Collaboration based approach:** In this approach, the system uses the similar users and cluster them to use them in the cluster. In this if any user in the cluster search something then the item in the recommendation list shown are the items which the other users in the same group are either purchased or rated. This is at the user level.  
Similarly clusters also forms at item level. In this the system match the user's rated item to the similar item as proposed by the Jeremy York Et al. [20]. This a faster approach as the system does not need to scan the large part of the database to search the similar users as in the previous approach.
- 3) **Knowledge based approach:** We understand by this an example. Typically a user buy a television only after some years. It is not an item which a user purchases frequently

in a short period of a time. So the system cannot make the recommendation for this item. In this the we can only present the top selling television as proposed by Stefano Ceri Et al. [11])

- 4) **K-Mean clustering based approach:** Earlier recommendation system used the k-mean clustering to apply the previous approaches explained in this section as proposed by the Taek-Hun Kim and Sung-Bong Yang [24]. In this clusters are generated both at user level and also at the item level. It is a crisp method of clustering, In this the membership of any data element which is present in data set was only to one cluster means a data point can belong to only one cluster in the cluster set, so the membership of any data point means there is no crisp value of assignment of data points to the clusters

### 2.1 Introduction

Clustering is a grouping of items into clusters in such a manner that item in same group is similar to the other items in the group as compared to the items in the other group.

It is used for data mining operation and used for the data analysis. We can use it in various areas like pattern recognition, information retrieval, image analysis etc.

Cluster analysis is itself not any specific algorithm. There are various algorithm in the clustering like hierarchical clustering, k-means algorithm, DBSCAN, OPTICS, FCM etc. These algorithm differ in the basic method of how the cluster forms. There must be small distance between the data points in the cluster. A cluster algorithm includes various value like distance function, density threshold and the expected number of cluster.

A clustering method can be basically divided into two categories; hard clustering and soft clustering. In the case of hard clustering method each item can belong to only one cluster, no item can be member of more than one cluster (eg. K-Mean clustering). While in the soft clustering technique an item may belong to as many number of one cluster as want (eg. Fuzzy C-Mean).

There are various cluster models. For eg:

- Connectivity model: In hierarchical clustering, models are built upon the distance connectivity.
- Centroid model: In the k-mean, the algorithm represent the clusters by the single mean vector.
- Distribution model: In this the clusters would be made by using the statistical distributions.
- Density model: DBSCAN and OPTICS represents the clusters as the connected dense regions inside the data space.
- Subspace model: In the Bi-clustering, clusters are made using both cluster members and the relevant attributes.

- Group model: Few algorithms don't provide the refined model for their results instead of this they just provide us the grouping information.

In clustering some better distinctions also present, eg of types:

- Strict partitioning: in this every data point can belongs to only one cluster.
- Strict partitioning with outliers: In this type some data points can belong to zero no. of cluster, and would be reflected as outliers.
- Overlapping type: although it is hard clustering, data points can belong to any number of cluster.
- Hierarchical type: data points which belongs to the child cluster can also belong to parent cluster.
- Subspace clustering: Even being a type of overlapping type of clustering, clusters are not expected to overlap within a uniquely defined subspace.

Clustering is a set of these clusters. It can also represent the relationship between these clusters, for eg. a hierarchy of cluster can be embedded in each other

## 2.2 Clustering Types

There are different clustering algorithms. Some of them are illustrated below:

### **Hierarchical clustering**

In this clustering method a hierarchy of clusters forms. Two categories of the hierarchical clustering are following:-

- Agglomerative: It is a bottom up technique. In this technique each cluster merges as one move up in a hierarchy. The complexity of this approach is  $O(n^3)$  and because of this, it is very slow for the large data set.
- Divisive: Its approach is totally different to the previous agglomerative approach as it is a top down approach. In this technique all operations starts in one cluster only. Split performs recursively in downward fashion. The complexity of this approach is  $O(2^n)$  and because of this it is even worse than the agglomerative hierarchical clustering.

To decide which cluster to merge and when a cluster must be break dissimilarity measure is required between the set of observations. Generally we use distance metric (to calculate the distance between the pair of observations) for the measure of dissimilarity. It does not require the value of number of cluster  $k$  in advance, but it requires a termination condition.

## Algorithm

- Start with all instances in their own cluster until there is only one cluster:
- Among the current clusters, determine the two clusters,  $c_i$  and  $c_j$ , which are most similar.
- Replace  $c_i$  and  $c_j$  with a single cluster  $c_i \cup c_j$
- As clusters agglomerate, data points likely to fall into a hierarchy of clusters.

## Weaknesses:

- Time complexity  $O(n^3)$  in agglomerative hierarchical clustering while  $O(2^n)$  in divisive clustering, where  $n$  is no of the total items, which is very high.
- They can never change the work what was done previously.

The following example shows the agglomerative clustering technique using colors.

- 1) In the first step, the orange and red are mixed to form reddish-orange color, while the blue and green performs similarly, mixed to form the aqua color.
- 2) In the second step the reddish orange color and the yellow color are the closest colors, thus on mixing forms a light orange color.
- 3) In the third step we would have three colors remaining; light orange, purple and the aqua.
- 4) In the fourth step we would mix the light orange color and the purple color as these are more similar.
- 5) Now we have only orange-purple combination and aqua color. In the final step we mix the two remaining color to form the final cluster. Following resulting tree is showing the hierarchical clustering.

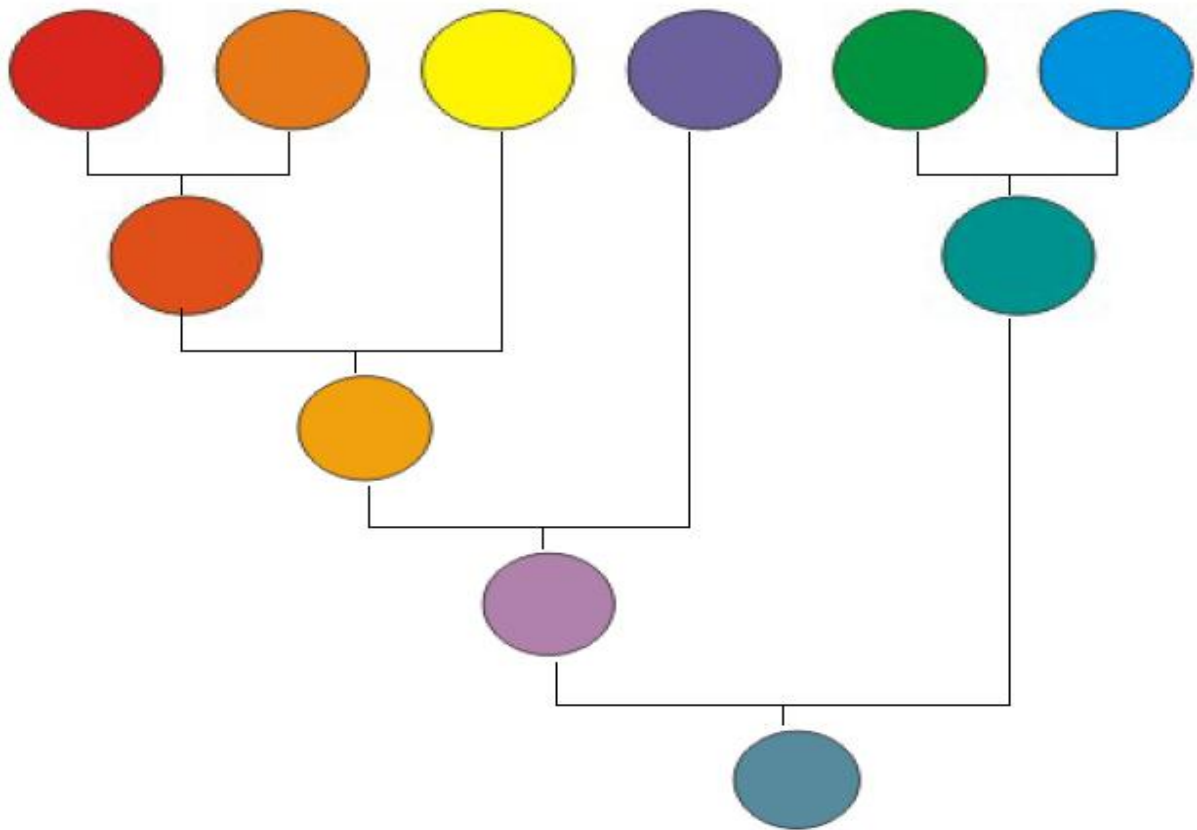


Fig-2.1 Example of Hierarchical clustering

### **K-Mean Clustering**

It is a very simple and unsupervised learning clustering algorithm. It uses the very simple way for the clustering of the given data. It partitions the  $n$  data points into the  $k$  (fixed priori) number of cluster in such type that every item belongs to the cluster with the nearest mean. The basic idea of this approach is to define the  $k$  number of centroids; these centroids will be used one for every clusters. Centroid must exists as much far as possible, because on placing near these can create problems. In the next phase we choose each data point from the data and place them to cluster having closest centroid. Later this step when no data point is pending in the data set, we need to recalculate the centroid for each cluster. After this, we need to recalculate the binding of the data points with the newly deduced nearest centroid. We performs these steps again and again until no movement of centroid occurs means the centroid get fixed



Aim of the K-Mean is to minimize the objective function  $J$ , which is following

$$J = \sum_{j=1}^k \sum_{i=1}^n \left\| \mathbf{x}_i^{(j)} - \mathbf{c}_j \right\|^2$$

Where  $\left\| \mathbf{x}_i^{(j)} - \mathbf{c}_j \right\|^2$  represents the measure of distance between the data points and centroid.

The algorithm is detailed below:-

- 1) Place the  $K$  points from the data point from the data set that need to be clustered. These  $K$  points would denote the initial centroids.
- 2) Now pick each data point from the data set and assign these data points to the group which is having the nearest centroid.
- 3) Now at the point when no data point would be left to assign to the nearest centroid, now we need to again calculate the placements of all these centroids.
- 4) Now perform the steps 2 and 3 again until no centroids of any cluster move longer. Algorithm places the data points into the clusters from which we can calculate the metric which need to be minimized.

Although it is true that the process would always terminate, in spite of this, the algorithm may not always get the most ideal result, comparable to global objective minimum function. This also depends upon initially randomly chosen cluster centroid, because of this we run the algorithm multiple times to reduce the consequence.

It is a very simple algorithm which is adjusted to many problem domains.

### Example

Consider  $n$  number of sample  $x_1, x_2, \dots, x_n$  all from same class, and they would fall into the  $k$  number of clusters and  $k < n$ . Assume  $c_i$  is center of  $i$ th cluster. If clusters are separated in good way then we can use the minimum-distance classifier to isolate those clusters. This means that, if  $\left\| \mathbf{x} - \mathbf{m}_i \right\|$  comes as the minimum of all  $k$  distances, then only we could say that the sample  $\mathbf{x}$  is in the cluster  $i$ . This gives us following method to find  $k$  number of means:

- First make the initial guesses for the means  $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_k$
- Repeat the following sub-step til there is no change in any of the mean
  - Use these calculated means to classify samples into the clusters
  - For  $i$  from 1 to  $k$ 
    - Replace the  $\mathbf{m}_i$  with the mean of all samples for the cluster  $i$

- end\_for
- end\_until

K-Mean is explained in the following diagrams:

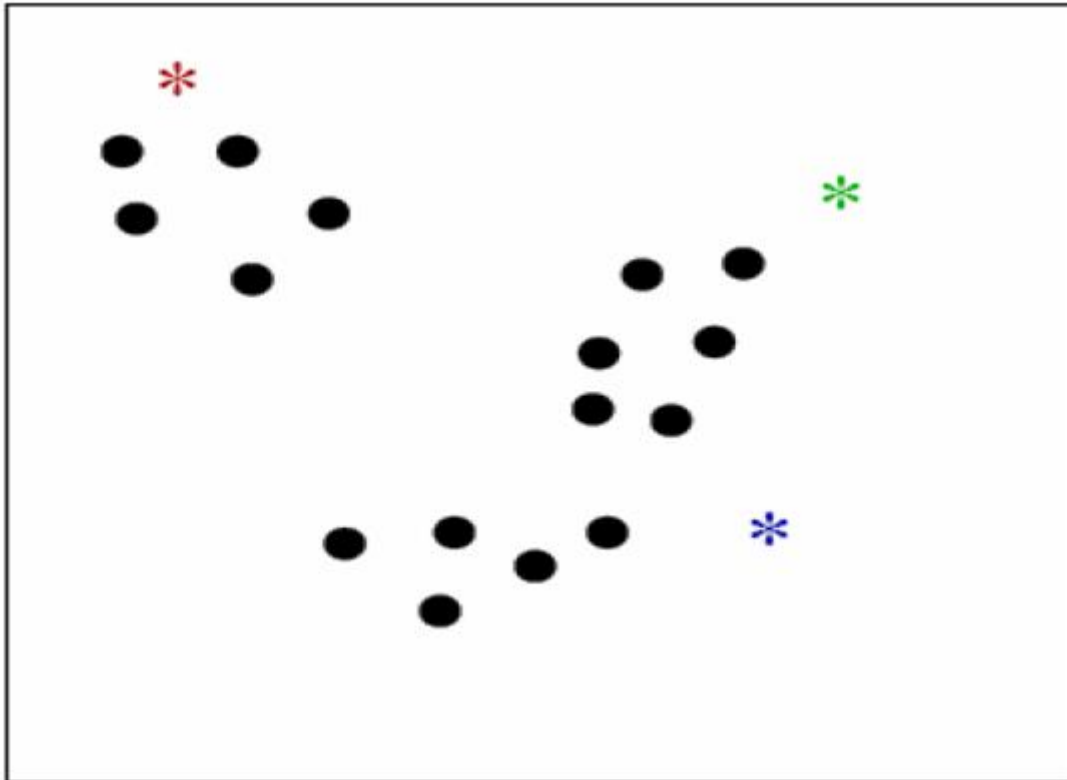


Fig-2.2 Initialize representatives (“means”)

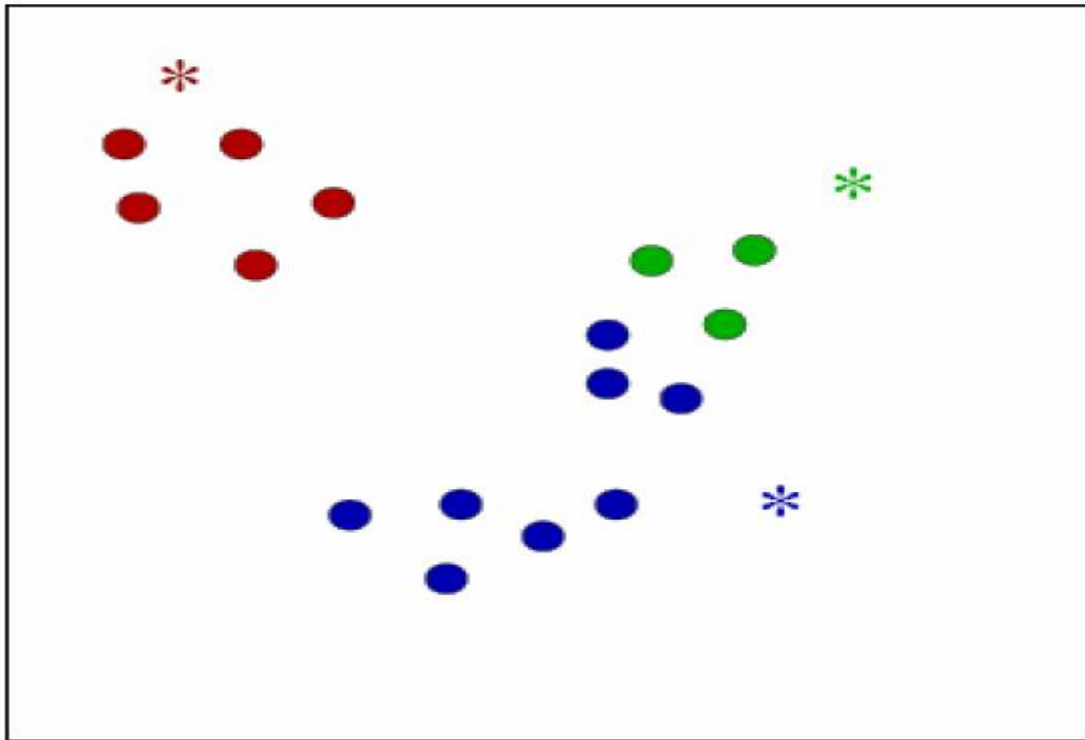


Fig-2.3 Assign to nearest representative

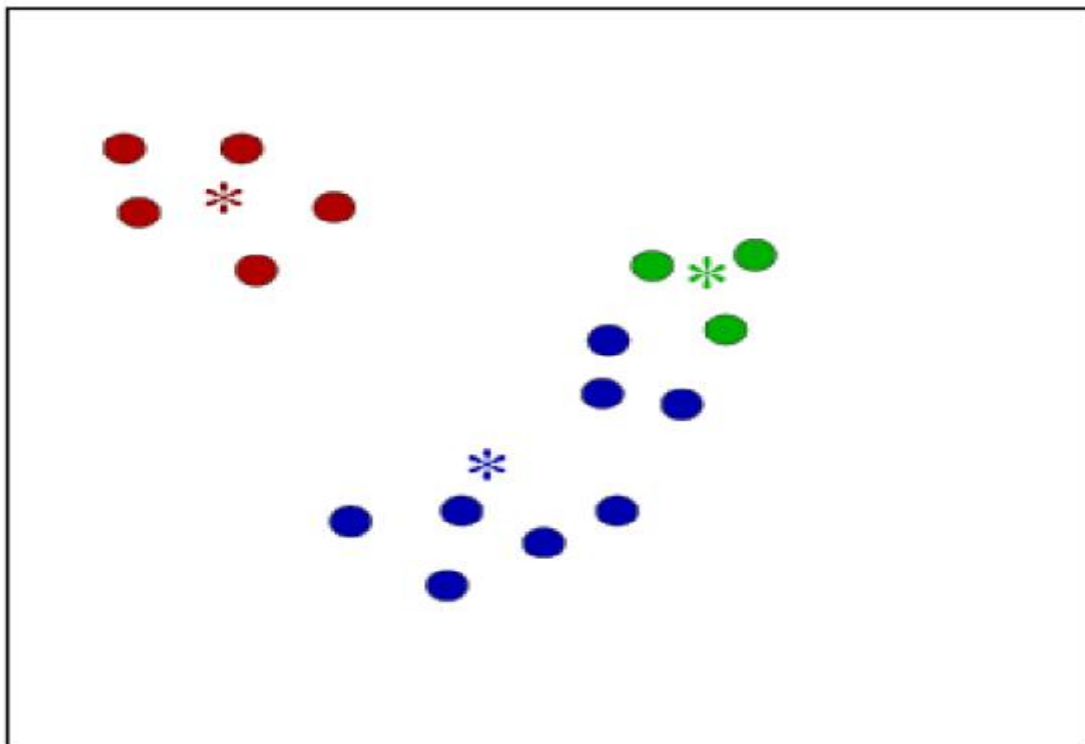


Fig-2.4 Re-estimate means

**Weaknesses:**

- How to calculate the initial mean values is not fixed. Most famous approach on how to begin is randomly pick the  $k$  number of samples.
- Result obtained depends upon initially taken values of the means, and suboptimal partitions are produced frequently. In the standard solution, we try different numbers of starting points.
- It is possible that the data points set nearest to  $\mathbf{c}_i$  is empty, so we cannot update the  $\mathbf{c}_i$ .
- The results depend upon the outcome  $\|\mathbf{x} - \mathbf{c}_i\|$ . We can normalize the variables by their standard deviations, though this is not always necessary.
- Problem with the outliers.
- The results would depend upon the  $k$  (the predefined no of chosen cluster).

The main problem in all the listed above is how to know how many clusters would exist in the starting of the clustering. The problem arises when we need to decide the value of  $k$  as we don't know what value of  $k$  would give us the best result, for example 3-mean clustering can be better or worse than 2-mean. Unluckily, we don't have any fixed way to know that what would be ideal number of clusters for any data set. To solve this problem we can run the problem with different values of  $k$  and compare the results and choose the best value, but we have to be careful as because increase in the value of  $k$  can give us the reduced values of error function, but increase the threat of the over fitting problem.

**Distribution-based clustering**

This type of clustering model is based upon the distribution models. In this clusters are made up of items which belong most probably to same distribution. A good feature of this method is that this method closely looks the same way as artificial data sets generates: by sampling the random items from the distribution.

Theoretically these approach is very good but, they face a big problem called as over-fitting, unless we put the constraints on the complexity of model. We can explain the data better by a more complex model, but it would make the choosing of appropriate model complexity very difficult.

One popular approach is gaussian mixture models. In this the data set is generally made by the fixed (to remove the over fitting problem) no. of the gaussian distributions which are initialized

randomly and whose attributes are iteratively optimized to fit better according to the data set. This would meet to the local optimum, so the multiple runs produces different outputs. To get the hard clustering, items in the data set are generally allotted to gaussian distribution they belong most probably to; in case of the soft clusterings, it is not necessary.

But these algorithms has some shortcomings also as these algorithms creates the extra pressure on customer as they need to pick the suitable data models to improve this and also there is no mathematical model available to optimize many real data sets.

### 3.1 Introduction

Cluster analysis is the most famous technique used in the pattern recognition system. It is used to divide the data points into various clusters such that the similarity between the elements in the same cluster is highest and the dissimilarity between the different clusters are also as high as possible. It is a type of data compression technique, where the large no of data points are grouped into some very less no of clusters. It is unsupervised learning rather than the supervised learning used in various techniques. It is used in various fields like image processing, medical diagnosis, business etc. Previously there was only one type of clustering Hard Clustering (no fuzzy); in which the membership of any element was only to one cluster means an element could belong exactly to one cluster in the cluster set only, so the membership of any data point means there is no crisp value of allocation of these elements to clusters. In the crisp method of clustering there is clear cut boundary between the clusters whereas in the real cases there may not be clear cut boundary between the clustering. Then the soft clustering technique came into existence. In the soft clustering technique every element in the data set can belong to more than one cluster by some degree of belonging range between 0 and 1, where 0 denotes the lowest or no membership and the 1 denotes the highest or full membership. Fuzzy set theory was first used in clustering technique by Bellman, Kalaba, Zadeh and Ruspini . They introduced the fuzzy clustering technique first time.

There are various different types of clustering techniques like k-mean and fuzzy z-mean. In the K-Mean the degree of belonging of the element is hard means either the membership of element to a specific cluster is 0 or 1, no value between these values but it is not the case with the fuzzy c-mean clustering. The belonging of any element to any clusters can be 0, 0.1, 0.9 or 1, so any value between 0 and 1.

This is used for the examination of the clusters in which the distribution of the elements to the clusters are not hard (0 or 1) instead it is fuzzy (attached to some degree of membership) same as the fuzzy logic.

It is the procedure of distributions of the data samples into the clusters such that the elements in same cluster are alike to each other as much as possible, while objects which are not in same clusters are dissimilar to each other as much as possible. It depends upon type of the data and the aim for which we are using clustering. Based on this we can use different methods of similarity to put the elements into different clusters and there similarity method would decide how these clusters would be made. Few instances of these methods can be distance, connectivity, and intensity. On the other hand, in the hard clustering each data point can belong to one cluster only. While in the fuzzy clustering method data element may belong to more than one cluster, and each object associate to each cluster to a set of belonging level. These denotes level of association of each data point and an every related cluster. In fuzzy clustering we use this membership level to assign the data elements to more than one clusters. Fuzzy clustering is very useful when the clusters overlaps.

C-Mean method is the new clustering method based on the K-Means. The distinction between the C-Means and K-Means is that after clustering by C-Means, every sample has some degree of membership to each cluster. So each item belongs to each cluster with some degree of probability. Thereby, recommender system can reduce the cold start problem by suggesting the items which are not rated. If any item is not bought by any customer and also not reviewed (rated) by any user then also that item can be present in the recommendation list, as because with the help of tags exploitation by the content based approach. By this our system can recommend those items to customer to overcome the problem of cold start.

For each point  $x$  there would be set of coefficients which would give us the degree of membership for  $k$ th cluster  $w_k(x)$ . Cluster centroid would be mean of all the points, weighted by their membership degree to that cluster:

$$C_k = \frac{\sum_x W_k(x) x^m}{\sum_x W_k(x)^m}$$

The membership degree  $W_k(x)$  is inversely related to the distance of cluster center to the  $x$ . It depends upon the value of  $m$  which controls how much weight would be given to the nearest center. When we add any new item then we also all some associated feature (tag) to that item, although each customer have some other features manually set in his profile.

Its aim is to minimize the objective function, which is following:

$$J_m = \sum_{i=1}^n \sum_{j=1}^c u_{ij}^m \|x_i - c_j\|^2, \quad 1 < m <$$

In this  $m$  is a real number which is more than 1, while  $u_{ij}$  is degree of the belonging of  $x_i$  in cluster  $j$ ,  $x_i$  is the  $i$ th sample of the calculated data,  $c_j$  is  $p$ -dimension centre of cluster, and  $\|*\|$  is expressing the similarity between the calculated data and centre.

As we have already said, every element is associated to every cluster by the belonging Function, which exhibits fuzzy behavior of the algorithm. To understand this, we need to make a matrix  $U$  whose values (elements) lies between 0 and 1, and express the degree of belonging between the data point and the clusters center.

For the greater knowledge of the system, consider a case. Assume we represent data-set distributed on the axis  $X$ . The following figure illustrate this:



Fig 3.1 Data point distributed on an axis

By examining this figure, we can notice two clusters in nearness of the two data groups. We would call them as 'A' and 'B'. In k-mean approach, we associate every data point to the particular centroid; so, the membership function looks like below diagram:

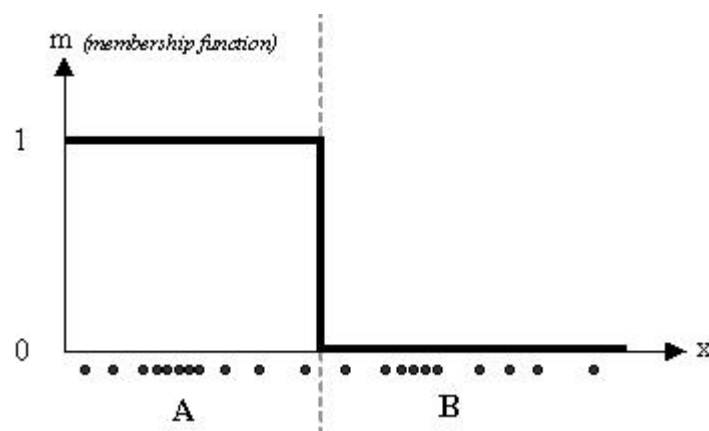


Fig 3.2 Membership function diagram in K-Means



In FCM algorithm, same data point is not an exclusive member of any specific cluster, so we can place it in the middle way. In these cases, membership function follows the smoother line to show that each data point can be a member of several clusters by the different values of belonging function.

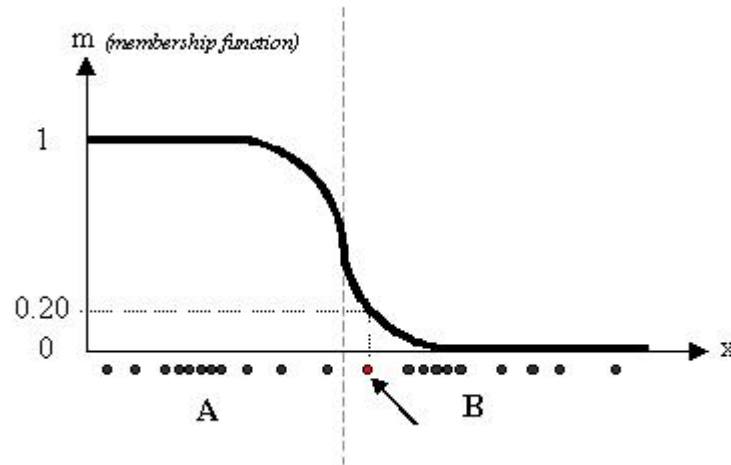


Fig 3.3 Membership function diagram in FCM

In above diagram, red color data point belongs more close to the cluster B as compared to cluster A. 0.2 value of 'm' denotes the degree of belonging to cluster A for that data point.

We would use the Fuzzy C-Means as it is the most famous present technique for this. A fuzzy subset  $\tilde{A}$  of  $X$  is universe of discourse and is defined by its belonging function  $\mu_{\tilde{A}}: X \rightarrow [0, 1]$ . For any  $x \in X$ , value  $\mu_{\tilde{A}}(x)$  specifies degree to what  $x$  belongs to  $\tilde{A}$ . As uncertainty prevails in the field of finding individual differences in our work, therefore a fuzzy technique is required for the classification of the same. The basic task of a classification technique is, divide the  $n$  numbers of items, into  $c$  clusters, where  $2 \leq c < n$ . We divide set of  $n$  elements  $X = \{x_1, \dots, x_n\}$ , where  $X_k = \{x_{k1}, x_{k2}, \dots, x_{kp}\}$ , where  $k = 1, 2, 3, \dots, n$  into the pool of  $c$  set of fuzzy clusters based on already given criterion. Each  $X_k$  is known as feature vector and  $x_{kj}$  where  $j = 1, 2, \dots, p$  is the  $j$ th feature of  $k$ th feature vector. Based on the given data, FCM returns us set of  $c$  cluster centers set  $C = \{c_1, \dots, c_c\}$  and a partition matrix, in which every entry  $Y_{ab}$  denotes the degree by which data point  $X_a$  is attached to the cluster  $C_b$ . Like other algorithms, aim of FCM is also to minimize the objective function.

Partition of dataset  $X$  into some number of the clusters is defined by belonging functions of data points of clusters. Let  $S_1, S_2, \dots, S_c$  denote the clusters with corresponding membership functions  $\mu_{s_1}, \mu_{s_2}, \dots, \mu_{s_c}$ .

A  $c \times n$  matrix containing the membership values of the objects in the clusters  $= [\mu_{si}(x_k)]$  for  $i=1,2,\dots,c$  and  $k=1,2,\dots,n$  is a fuzzy  $c$ - partition if it satisfies the following conditions

$$\sum_{i=1}^c \mu_{s_i}(x_k) = 1 \quad \text{for each } k= 1,2,\dots,n \quad (1)$$

$$0 \leq \sum_{k=1}^n \mu_{s_i}(x_k) \leq n \quad \text{for each } i= 1,2,\dots,c \quad (2)$$

Condition (1) specifies that every feature vector  $x_k$  has the total membership value equal to 1 which is distributed among all the clusters, whereas condition (2) specifies that sum of the membership degrees of the feature vectors in given cluster is not greater than total no. of feature vector.

In the fuzzy clustering method, every point belongs to every cluster by some degree of membership, instead of belonging to just one cluster completely. Because of this a point on border in the cluster will belong to the cluster to lesser degree of membership as compared to point in centre of the clusters.

FCM is very much similar to the k-means algorithm:

- 1) First choose the number of clusters.
- 2) Now assign coefficients for each point randomly for being in the clusters.
- 3) Now we repeat until the algorithm converges (means the coefficients' change between two iterations is not greater than given sensitivity threshold) :
  - a) Calculate centroid for every cluster, by the given formula.
  - b) Now for every point, compute their coefficients of being in the clusters.

This algo also minimizes the intra-cluster variance, but suffer from same problems by which the k-mean algorithm suffers; in this minimum obtained is only local minimum, and result would depend upon the initial choice of the weights.

Another algorithm which is very close to FCM is Soft K-means. FCM is very famous in clustering of objects in an image. The main base behind the classification algorithms is the Clustering of numerical data. Clustering produce the natural combination of the data from the large set of data to generate the concise representation of the system performance.

Consider the item  $I_a$  belong to the cluster  $C_i$  by  $B(C_{ia})$  degree of membership and belong to cluster  $C_j$  by  $B(C_{jb})$  degree of membership. User  $U_a$  belongs to the cluster  $C_u$  by  $B(C_{ua})$  degree of membership and  $U_b$  belongs to the cluster  $C_v$  by  $B(C_{vb})$  degree of membership.

Consider, the user  $U_a$  buy item  $I_a$  and the user  $U_b$  do not buy any item and the item  $I_b$  never buy by any of the users. Now, the cluster  $C_i$  on the items has a relation to the cluster  $C_u$  on the users by  $M(C_{iu})$  while:

So, after the generalization the formula described above become:

$$M(C_{iu}) = \frac{\sum_x^{items} \sum_y^{users} B(C_{ux}) * B(C_{iy})}{m * n}$$

Where, m denotes the numbers of the users and n denotes the numbers of the items. Then, a rule generated for recommending items in cluster  $C_i$  on items to users in cluster  $C_u$  on users. So, user  $U_b$  have a relation with item  $I_b$ .

$$Rec(U_b, I_b) = M(C_{iu}) * B(C_{ib}) * B(C_{ub})$$

Where,  $Rec(U_b, I_b)$  lies between [0, 1] and make a weighted map. If  $Rec(U_b, I_b)$  comes bigger than the threshold, system would recommend item  $I_b$  to the user  $U_b$ . This threshold should be figure out in user interaction interface part (knowledge-based part).

In the starting point of the system, recommendation threshold is zero. Thus, every  $\text{Rec}(U_b, I_b)$  make the recommendation with a like or dislike button only. If user  $U_b$  likes the item  $I_b$ , it means that the recommendation was good but if user  $U_b$  dislikes the item  $I_b$ , then it means that the recommendation was not good and the threshold should be bigger than  $\text{Rec}(U_b, I_b)$ , so that the threshold updates to have better recommendations later.

As there can be differences in the user's preference this threshold should be measures from all users' opinions. A simples method is to calculate the average on every disliked  $\text{Rec}(U_b, I_b)$  to have a better threshold value. As the knowledge base updates rules by the last recommendation result make by the user interaction-based part.

In the below figure, items of different clusters are divided into items of different colours.

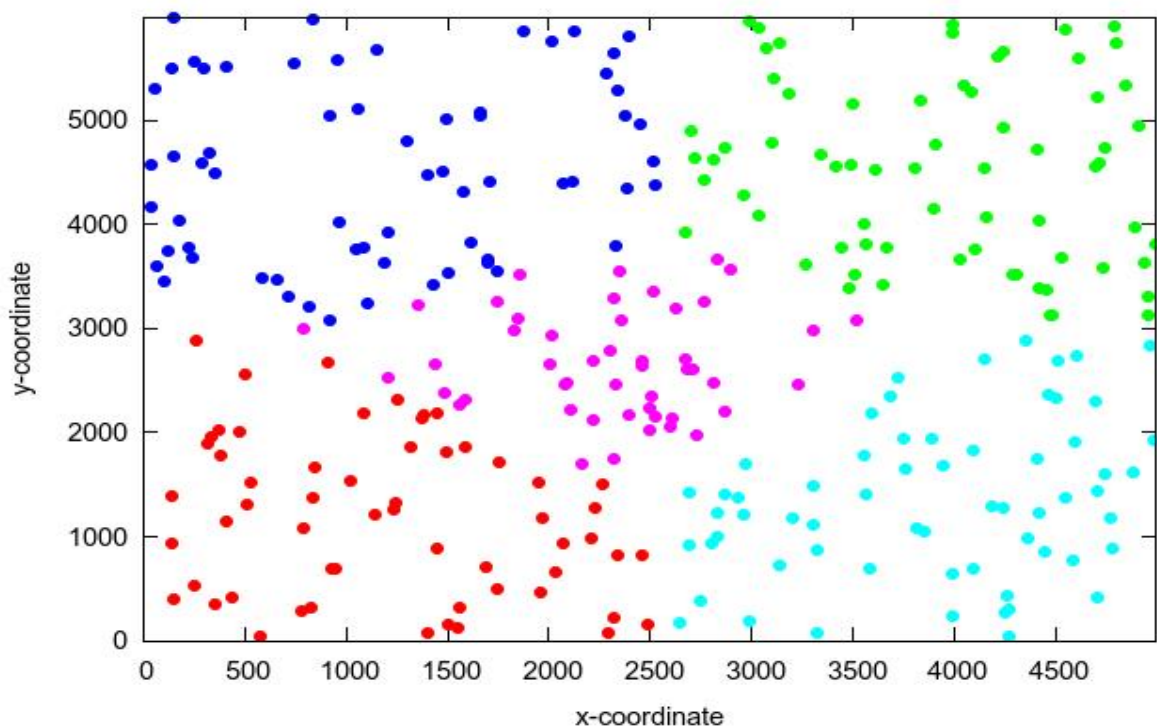


Fig-3.4 FCM clustering example

### 3.2 Comparison

As previously stated there are two types of C-Mean clustering: Hard C-Mean and Soft Fuzzy C-Mean. Difference lies in the way in which they assign the data to clusters, i.e., in hard C-Mean the membership is either full or zero while in fuzzy c – means clustering case we can assign the membership of data- points, membership can be gradual in clusters and it is measured in degrees in [0,1]. This gives the advantage that the data points can be member of more than one cluster.

<b>Hard c-means clustering</b>	<b>Fuzzy c – means clustering</b>
$U_{MC} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ \dots & \dots \\ 0 & 1 \end{bmatrix}$	$U_{MC} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \\ 0.6 & 0.4 \\ \dots & \dots \\ 0.9 & 0.1 \end{bmatrix}$

Fig-3.5 Membership function matrix of hard and fuzzy c-mean clustering

The above matrix U whose elements are the membership functions shows the data for the hard c-mean clustering and fuzzy c-mean clustering. In the present case we take two clusters and we can check the following: in first case (hard clustering) each data point  $p_i$  in the data-set  $X=\{x_1, x_2, \dots, x_n\}$  assigned to one cluster exactly, while in second case a data point can belongs to any number cluster (fuzzy clustering) with degree of belonging between data and centers of clusters. Below figures explain this idea. We can generate the hard clustering from the fuzzy partition by threshold value of membership function.

Membership degrees can also show how definitely or ambiguously a data point should belong to a cluster, as it is shown in the following diagram. Thus, we iteratively optimization the objective function for this partitioning to carry out, with update of the membership ( $U_{ij}$ ) and cluster centers ( $C_j$ ).

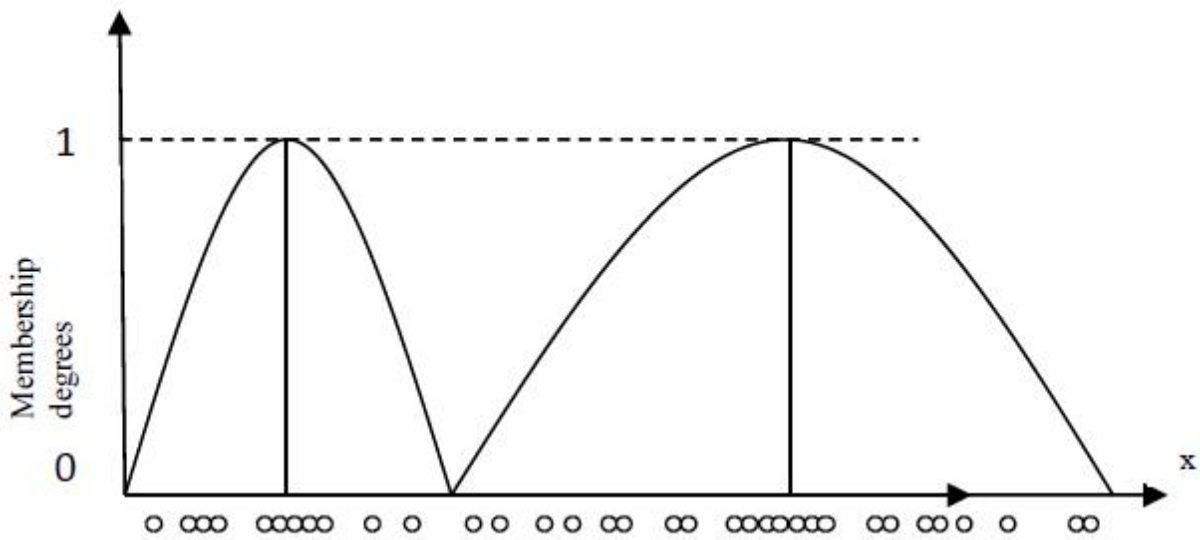


Fig-3.6 Hard Clustering

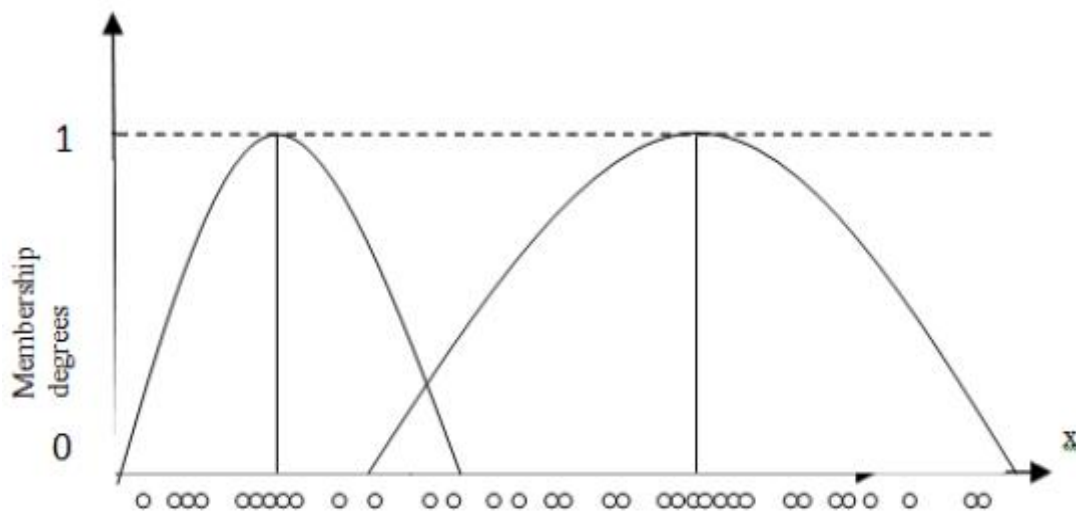


Fig-3.7 Fuzzy Clustering

The objective function is show below:

$$J_m = \sum_{i=1}^N \sum_{j=1}^c u_{ij}^m \|x_i - c_j\|^2, \quad 1 \leq m \leq \infty$$

High belonging value shows high confidence in allocation of the data point to clusters. Fuzzy partitioning is done the continues improvement of objective function, by the update of the membership function  $U_{ij}$  and cluster centers  $C_j$ , by

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m * x_i}{\sum_{i=1}^N u_{ij}^m}$$

Iteration stops when  $\max_{ij} \{|u_{ij}^{(k+1)} - u_{ij}^{(k)}|\} < \epsilon$  where  $\epsilon$  is a termination criterion between 0 and 1, whereas k is no. of iteration steps. This process meets to the local minimum of  $J_m$

The algorithm's detailed steps are below:

1) Initialization:- Selected the following parameters:

- a) The required number of clusters  $N$ ,  $2 < N < k$ ;
- b) Measure distances as Euclidean distance;
- c) A fixed parameter  $q$ ;
- d) Initial (at zero iteration) matrix  $U^{(0)} = (c_i)^{(0)}$  object ownership  $x_i$  with the given initial cluster centers  $C$ .

2). Calculate the centers vectors  $C^{(k)}=[c_j]$  with  $U^{(k)}$  In the t-th iteration step in the known matrix is computed in accordance with the above solution of differential equations

3). Modify the membership measure  $U_{ij}$

$$c_j = \frac{\sum_{i=1}^N u_{ij}^m * x_i}{\sum_{i=1}^N u_{ij}^m}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

4. If  $\| \mathbf{U}^{(k+1)} - \mathbf{U}^{(k)} \| < \epsilon$  then STOP; otherwise return to step 2. Where  $\epsilon$  is a predetermined level of accuracy



## **Chapter 4**

### **Proposed Methodology**

---

Earlier people used generally two or three and on the bases of K-Mean clustering only. Our proposed methodology employs the Fuzzy C-Mean in the recommendation instead of using the conventional K-Means clustering based approach, to improve the relevance of the items in the recommendation list. It would increase the participation of the data point to the different clusters. It gives us the chance to increase the quality of the result in the system.

We would use the hybrid mean of recommendation in our proposed system to enhance the result. There would be three approach in the system, which are following:-

- 1) Content based approach
- 2) Collaborative based approach
- 3) Knowledge based approach

#### **4.1 Content based approach**

In this technique we are using the tags (features) associated with the item. In our system we are using tags associated with books (book name, book publisher, book category) from the database of books. For example if a customer search for any book by its name then the system would generate the cluster taken that book name as the centroid of the generated cluster. If that book is found in the database. For example if the user search for computer network book, then our system not only show the computer network book, it will also show the similar book which the user can buy, based upon the computer network book tags e.g. book publisher etc. Another example is we assign each books some attributes like book its type comic, technology, medical etc. Further we divide these into different categories like author, publications etc.

In the previous recommendation system, there might be the case when we don't find result on entering the search key value in the search bar, it can happen because the search key can be unique. For example in book recommendation system like our system if we enter the book id then result does not show any recommendation the old recommendation system, but in our proposed system we are proposing hierarchical system in the Fuzzy C-Mean. If this type of case happens in our system the clusters would be formed in hierarchical manner. For example if we enter the book

id in the search bar then the system would make the 1<sup>st</sup> cluster based on that id, as the book id is unique no item would enter in the cluster, then the system would go for the other features associated with that book from the book id, based on those features the system would generate the cluster each for each feature. Now based on the degree of belonging of items in cluster, recommendation list would be generated, higher the degree of membership higher will be the order in the recommendation list

So the basic approach behind the content based approach is based upon similarity of item's feature.

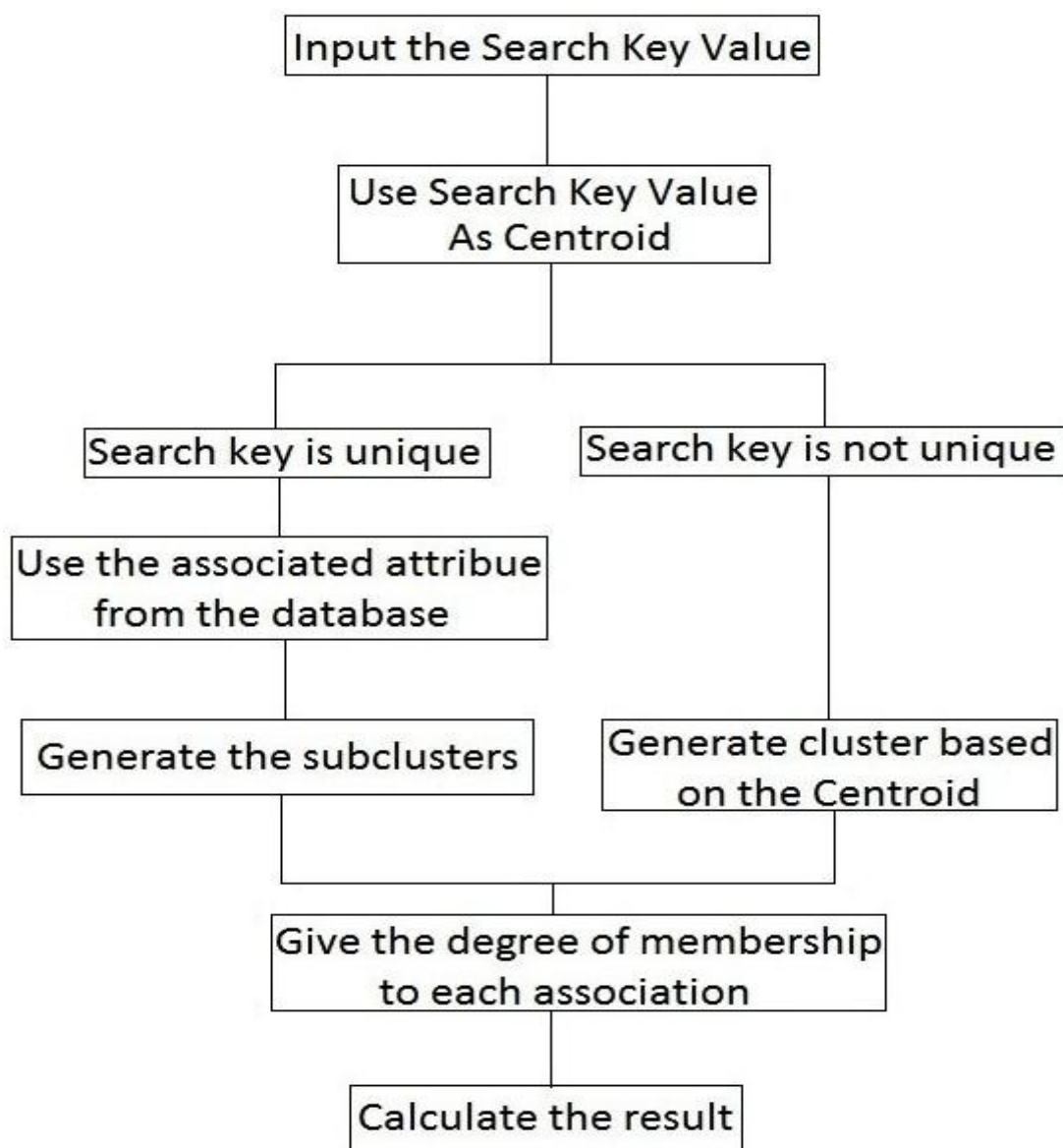


Fig-4.1 Content based approach

## 4.2 Collaborative based approach

We also use the collaborative based approach, it is used by building the database about each customer like the rating given by him or her, the past purchases by him. In this approach we are using the past history of the similar customers, to suggest the items to the current user. For example user A purchased the book Computer Network, DBMS, OS, and C++, if the past history matches the history of the current customer means for e.g. current user purchased the item Computer Network, DBMS and OS then it is very likely that the customer may purchase the book C++ also.

In other words in this approach, user would like those things more that the similar-minded users like. Therefore, this system makes the prediction for the customer, based upon similarity within interest profile of current customer and the other customers.

<i>Customers vs. Items:</i>	<b>Item1</b>	<b>Item2</b>	<b>Item3</b>	<b>Item4</b>	<b>Item5</b>	<b>Item6</b>
<i>Customer 1</i>		X	X			
<i>Customer 2</i>	X			X	X	
<i>Customer 3</i>	X				X	X
<i>Customer 4</i>		X	X	X		
<i>Customer 5</i>	X				X	X
<i>Customer 6</i>		X	X			X
<i>Current customer</i>	X				X	

Table-4.1 Customer item purchased table

A case is shown in the above table, where the X represents that the fact that user purchased that item for example customer 1 purchased the item 2 and item 3. This system must understand that which items must be suggested to the customer. To do this, system would compare other customer's behaviour with the current user and decides which purchases are most relevant and good for the current user. In the current example, the current customer behaves similarly to the customer 2, 3, and 5.

As the current customer behaves like customer 2,3 and 5, the system would recommend the items they bought so item 4 and 6 would be recommended strongly. Higher the similarity between the current customer and the other customers, higher would be its weightage in the recommendation; and higher the number of similar customers that bought a particular item, the higher would be the item's rank in the recommendation list.

Another way of illustration is described in the next page, in which on the basis of rating given by different users to different products, it is shown that how it will be used in making the recommendation. These recommendations are based on the prediction as we cannot guarantee that the user will definitely like those items.

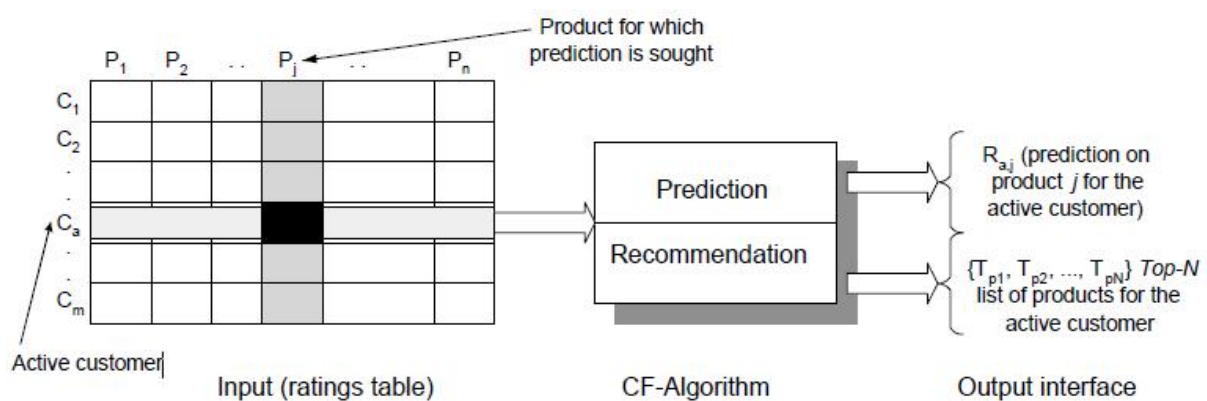


Fig-4.3 Collaborative Filtering Process

Where predictions are the numerical values,  $R_{a,j}$ , express the predicted judgment score of the product  $p_j$  of current user  $c_a$ . The predicted values lies with in the same range as the judgment value provided by the user  $c_a$ .

Recommendation is the set of  $N$  products,  $TPr = \{T_{p1}, T_{p2}, \dots, T_{pn}\}$ , which the current customer would prefer highest. There must not be any item in the list which the current user has already purchased.

Above figure presents the schematic flow of famous collaborative filtering approach. Collaborative filtering algorithm represents the  $m \times n$  ratings matrix,  $A$ , where  $m$  customer gave the ratings for  $n$  products. Each entry of  $a_{i,j}$  in the  $A$  represents rating by  $i$ th customer to  $j$ th product.

The ratings for any product is an integer value and it can also be 0, indicates the item is not rated. These algorithms are successful in many areas, but has shown some limitations also, eg.

- **Sparsity.** The problem with the nearest neighbour algorithms is that they rely heavily upon the exact matches which causes the algorithms to lose the system coverage and the accuracy. As the correlation coefficient depends upon the customer whose rated products are common (at least two), many customer pair would have no correlation at all. Because of this the collaboration algorithms may recommend the product to a specific customer. This effect is called the reduced coverage and it is because of sparse ratings of the neighbours.
- **Scalability.** This algorithm requires the computations which increases in size as no of customers and no. of products both increases. As there are generally millions of customer this algorithm suffer from the problem of scalability.

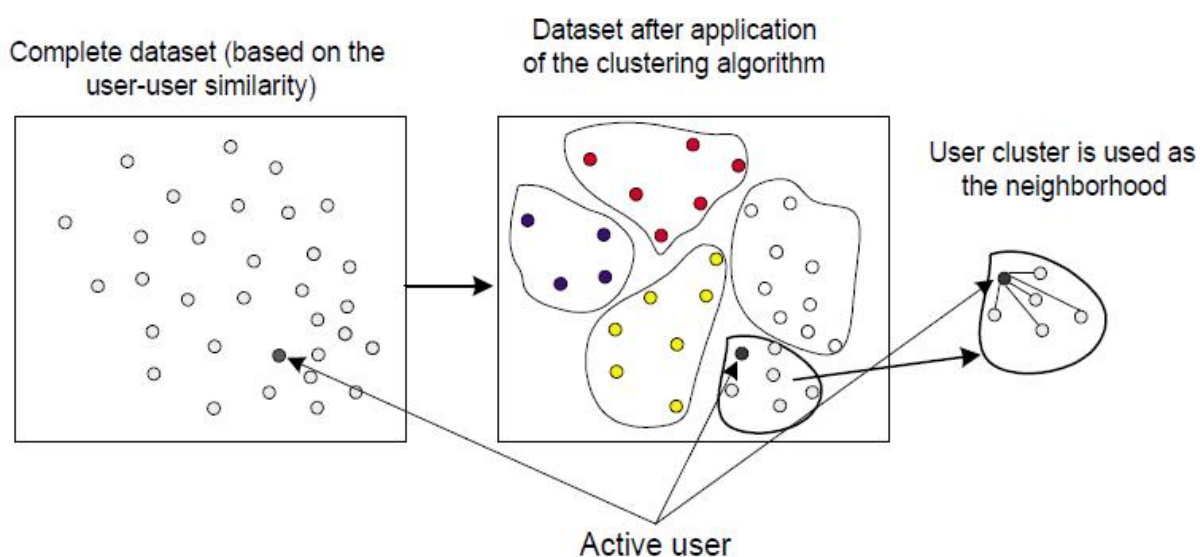


Fig-4.4 Neighborhood formation from clustered partitions

The above diagram shows the cluster formation for the neighbors in the collaboration filtering.

Shortcomings in the closest neighbor approach for the big databases steered us to search the other algorithms for the recommender system. Clustering of the customer can effectively divide the database of ratings and thus enhancement in the scalability power of the recommender system. Earlier researches indicates the benefits of applying the clustering in the recommender systems.

### **4.3 Knowledge based approach**

In this approach we are further are using two approaches in this; in the first approach we are explicitly taking feedback form from the customer and based on this we are filtering the preferences to show the relevant result. It is a very useful form for showing the most relevant results, but there are some problem with this approach also as the user might not be interested in giving its preferences explicitly (filling the form), wasting his time in his own mind. Like in the IEEE website we can filter the results. We explicitly ask the user about his preferences and put the result based on those preferences

Another approach in this method is using the past history of all the item. For example if a user search for a camera and we don't know the user previous history and he just entered the camera and no explicit name then in that case then we would chose the top selling and top rated camera in some past time.

We can also put the query in the search bar and can show the result based on that query. For example on entering the query "which book is best for Data structure" then the system would generate the relevant answer. Because a new user may not know any specific book name in that case it would be very good. Similarly on entering the "best author for digital Circuit book" in the search bar would give the appropriate result.

#### 4.4 Proposed Hybrid Recommendation System

The complete proposed system is illustrated in the following diagram:-

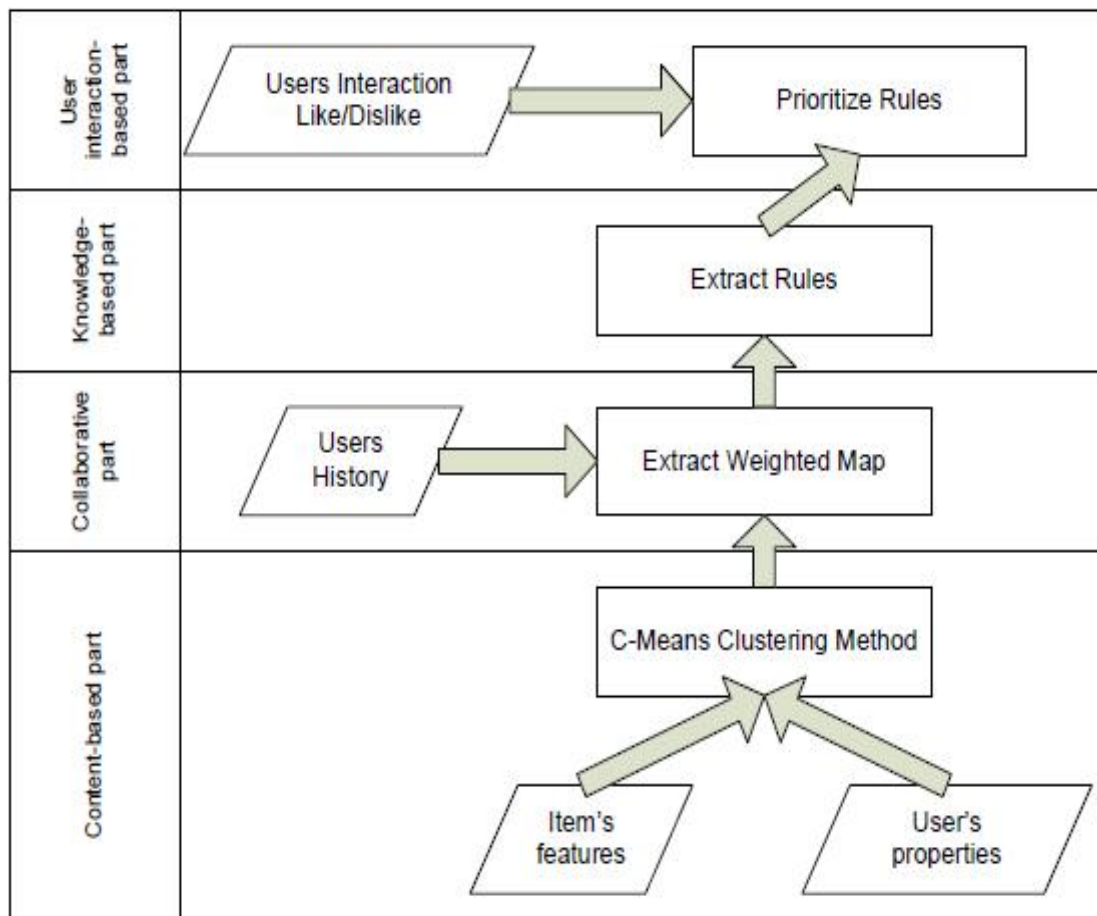


Fig-4.5 Proposed Hybrid Recommendation System

Now we are using Fuzzy C-Mean in this recommendation system. In fuzzy clustering, as we know each object has some degree of membership to each clusters, instead of belonging fully to just one cluster or any other cluster as in K-Mean clustering. Thus, objects on border of the cluster will belong to cluster to less degree of membership than the objects in the cluster center. C-Mean method is new clustering method based upon K-Means. The distinction between C-Means and K-Means is that after clustering by C-Means, every sample has some degree of membership to each cluster. So each item belongs to each cluster with some degree of probability. Thereby, recommender system can reduce the cold start problem to suggest item which is not rated. If any item is not bought by any customer and also not reviewed (rated)

by any user then also that item can be present in the recommendation list, as because with the help of tags exploitation by the content based approach. By this our system can recommend those items to customer to overcome the problem of cold start.

For each point  $x$  there would be set of coefficients which would give us the degree of membership for  $k$ th cluster  $w_k(x)$ . Cluster centroid would be mean of all the items, weighted by their degree of membership to that cluster:

$$C_k = \frac{\sum_x W_k(x)x^m}{\sum_x W_k(x)^m}$$

Membership degree  $W_k(x)$  is inversely related to the distance of cluster center to the  $x$ . It depends upon the value of  $m$  which controls how much weight would be given to the nearest center. When we add any new item then we also add some associated feature (tag) to that item, although each customer has some other features manually set in his profile.

Now the new method uses these features (tags) to cluster the items and customers use C-Means method (content-based part).

Now, we have some clusters on items level and some clusters on user level. To check out the relations between these clusters, it uses the users buy history.

Consider the item  $I_a$  belong to the cluster  $C_i$  by  $B(C_{ia})$  degree of membership and belong to cluster  $C_j$  by  $B(C_{jb})$  degree of membership. User  $U_a$  belongs to the cluster  $C_u$  by  $B(C_{ua})$  degree of membership and  $U_b$  belongs to the cluster  $C_u$  by  $B(C_{ub})$  degree of membership.

Consider, the user  $U_a$  buy item  $I_a$  and the user  $U_b$  do not buy any item and the item  $I_b$  never buy by any of the users. Now, the cluster  $C_i$  on the items has a relation to the cluster  $C_u$  on the users by  $M(C_{iu})$  while:

$$M(C_{iu}) = B(C_{ua}) * B(C_{ia})$$



So, after the generalization the formula described above become:

$$M(C_{iu}) = \frac{\sum_x^{items} \sum_y^{users} B(C_{ux}) * B(C_{iy})}{m * n}$$

Where, m denotes numbers of the customers and n denotes numbers of the items. Then, a rule generated for recommending items in cluster  $C_i$  on items to users in cluster  $C_u$  on users. So, user  $U_b$  have a relation with item  $I_b$ .

$$Rec(U_b, I_b) = M(C_{iu}) * B(C_{ib}) * B(C_{ub})$$

While,  $Rec(U_b, I_b)$  it between  $[0, 1]$  and create a weighted map. If  $Rec(U_b, I_b)$  is bigger that a threshold the system recommend item  $I_b$  to user  $U_b$ . This threshold should figure out in user interaction interface (knowledge-based part).

At the start point of this system the recommendation threshold is zero. So, every  $Rec(U_b, I_b)$  make a recommendation with a like/dislike button. If user  $U_b$  liked item  $I_b$ , it is inferred that this recommendation was good but if user  $U_b$  disliked item  $I_b$ , it is inferred that the recommendation is not good and the threshold should be bigger than  $Rec(U_b, I_b)$ , so the threshold updates to have better recommendations later (user interaction-based part).

Because of differences in user's preference this threshold should be measures from all users' opinions. The simplest method is to measure an average on every disliked  $Rec(U_b, I_b)$  to have a better threshold. After all, the knowledge base updates rules and prioritizes rules by the last recommendation result make by user interaction-based part.

### 5.1 Different Approaches Output

We made the recommendation system using hybrid approach means using all the three approach, content based, collaborative based and knowledge base.

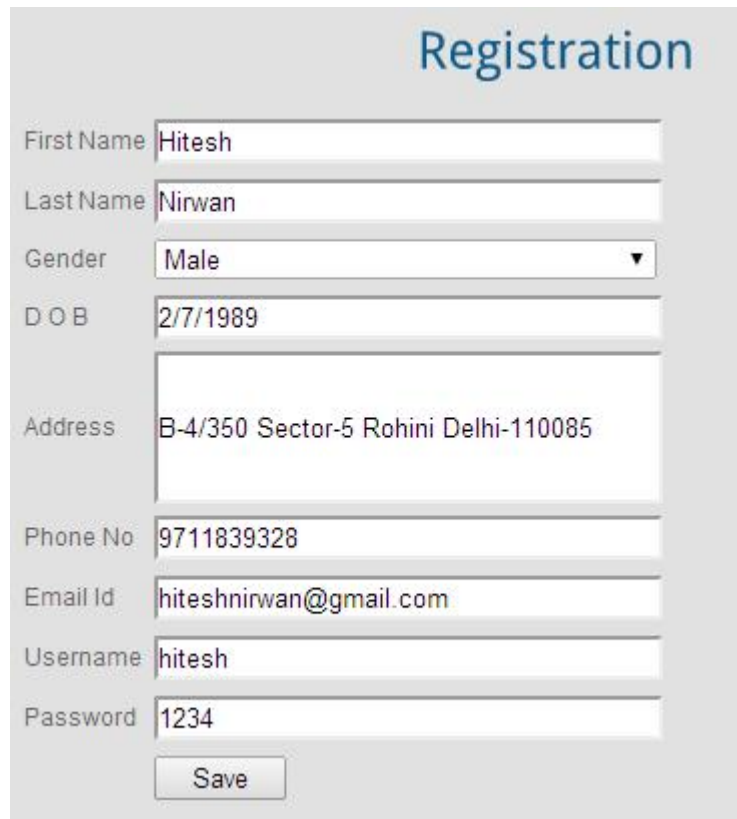
First of all, a simple login and registration system page appears, on clicking a login system following page appears.



The image shows a web form titled "User Login". The form has a dark header with the text "Log In". Below the header, there are two input fields. The first is labeled "User Name:" and contains the text "hitesh". The second is labeled "Password:" and contains masked characters "\*\*\*\*". Below the password field, there is a checkbox labeled "Remember me next time." and a "Log In" button.

Fig-4.1 Login System

Similarly the registration page appears, in which on entering the some relevant information, the system create the database entry for that user .The page which appears during the registration appears is following.



**Registration**

First Name

Last Name

Gender

D O B

Address

Phone No

Email Id

Username

Password

Fig-5.2 Registration page

We can get the book by its name and the books in the list would be generated using content based approach. For example we enter “C++” as this books come under computer language category so other books which are from the same category “computer language” and also exist in the database would also come in the result like Let Us C and Java Programming.




C++

Hybrid Recomend Books Searching

answer	id	author_name	publisher	edition
C++	10	Balagurusamy	TMH	3rd
Let Us C	4	Yashavant kanetkar	PQR	6th
Java Programming	6	The Wikibook	bpb	Ninth Edition

Fig-5.3 Search Example 1

Similarly the system works on entering the “Let Us C” and shows the other related book like the other programming books present in the database.

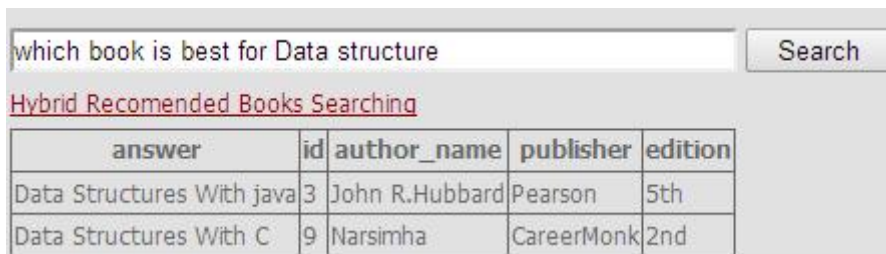


The screenshot shows a search bar with the text "Let Us C" and a "Search" button. Below the search bar, there is a heading "Hybrid Recomendated Books Searching" (note the typo "Recomendated"). Underneath is a table with the following data:

answer	id	author_name	publisher	edition
C++	10	Balagurusamy	TMH	Third
Let Us C	4	Yashavant kanetkar	PQR	Sixth Edition
Java Programming	6	The Wikibook	bpb	Ninth Edition

Fig-5.4 Search Example 2


Similarly the system perform on entering the query. For example on entering the query “which book is best for Data structure” then the system would generate the relevant answer. Because a new user may not know any specific book name in that case it would be very good. Similarly on entering the “best author for digital Circuit book” in the search bar would give the appropriate result.



The screenshot shows a search bar with the text "which book is best for Data structure" and a "Search" button. Below the search bar, there is a heading "Hybrid Recomendated Books Searching" (note the typo "Recomendated"). Underneath is a table with the following data:

answer	id	author_name	publisher	edition
Data Structures With java	3	John R.Hubbard	Pearson	5th
Data Structures With C	9	Narsimha	CareerMonk	2nd

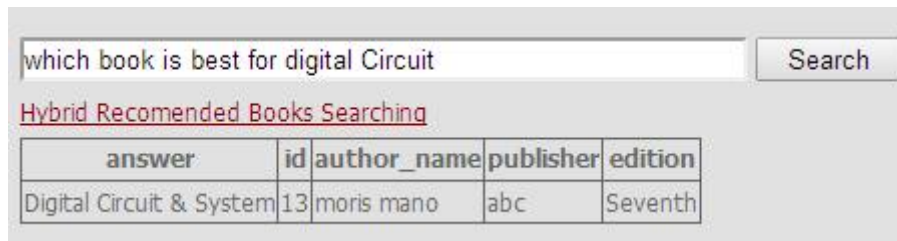
Fig-5.5 Search Example 3



The screenshot shows a search bar with the text "best book for electrical" and a "Search" button. Below the search bar, there is a heading "Hybrid Recomendated Books Searching" (note the typo "Recomendated"). Underneath is a table with the following data:

answer	id	author_name	publisher	edition
Electrical System	12	jb gupta	Pearson	Sixth

Fig-5.6 Search Example 4



which book is best for digital Circuit

Hybrid Recommended Books Searching

answer	id	author_name	publisher	edition
Digital Circuit & System	13	moris mano	abc	Seventh

Fig-5.7 Search Example 5

We can also browse the book by its category (Knowledge Based Recommendation System), like the following



Welcome, hitesh

Select Course	M.Tech
Branch	CS
	<input type="button" value="Go"/>

Fig-5.8 Search Example 6

And the output would be generated as following

hello,hitesh

## Book List



- **Book Id:1**
- Book Code:BBC01
- Book Name:Software Engineering
- Author Name:Pankaj Jalotes

[ViewDeatils](#)



- **Book Id:2**
- Book Code:BBC02
- Book Name:Theory Of Computation
- Author Name:Vivek Kulkarni

[ViewDeatils](#)

Fig-5.9 Previous search output

We can also give the rating to the product (Collaborative Based Recommendation System), so the other user in the cluster can be benefitted by this.

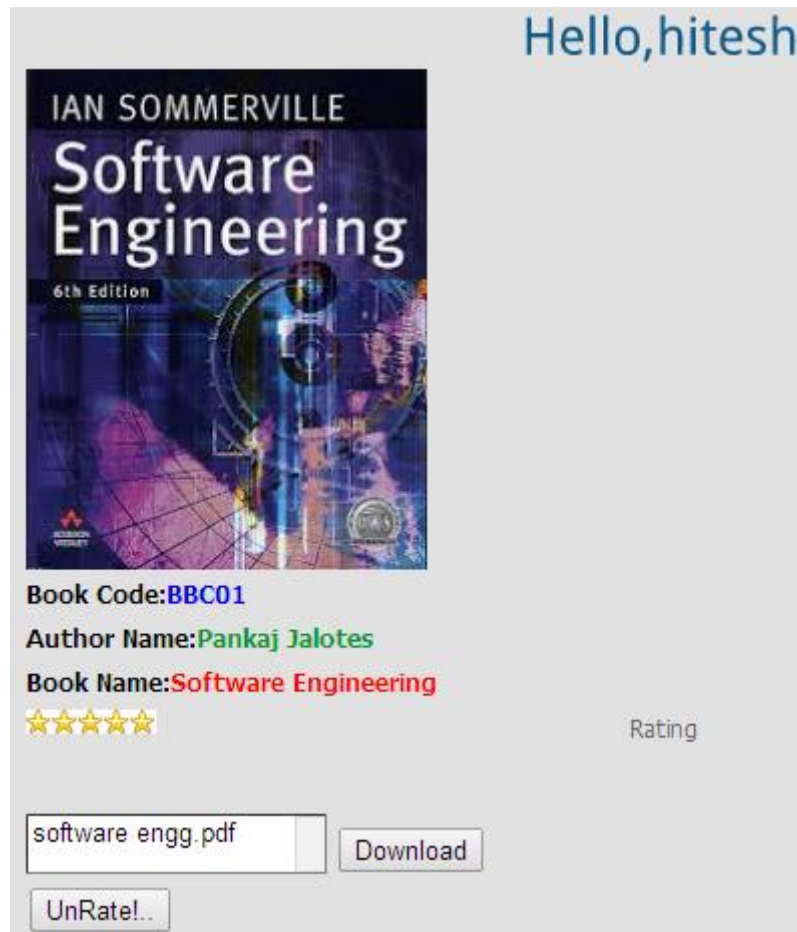


Fig-5.10 Item detail and rating system

We can evaluate the recommendation system by using the value of precision and recall. First understand basic definition of precision and recall

Precision: It is the percentage of relevant item in all of the relevant items in the entire collection set. Let

$X = \{D^1, D^2, \dots, D^k\}$ , where  $X$  is the set of ranked items.

$Y = \{Y_1, Y_2, \dots, Y_k\}$ , where  $Y$  is set of relevant items for current request.

Then,

$$\text{Precision} = \frac{|X \cap Y|}{|X|}$$

$$\text{Recall} = \frac{|X \cap Y|}{|Y|}$$

For testing we search for a book, for example computer network book



computer network

Hybrid Recommended Books Searching

answer	id	author_name	publisher	edition
Computer Network	2	MC-Hill	wsm	Fifth Edition
Computer graphic	14	MC-Hill	ASM	Third Edition
computer	15	Monarch	MSI	Sixth Edition
Computer system	11	Monarch	IJS	Second Edition
Electrical System	12	jb gupta	Pearson	Sixth Edition

Fig-5.11 Search Example 7

We see that in the fifth row there is an irrelevant item “Electrical System”. In this

$$X = \{\text{Computer Network, Computer Graphic, Computer, Computer System, Electrical System}\}$$

$$Y = \{\text{Computer Network, Computer Graphic, Computer, Computer System}\}$$

In this case

$$\text{Precision} = 4/5 = .75$$

$$\text{Recall} = 4/4 = 1$$

Now the value of precision is .75 and recall is 1 in this, which are very good values in the recommendation system.

Similarly



C++

Hybrid Recommended Books Searching

answer	id	author_name	publisher	edition
C++	10	Balagurusamy	TMH	3rd
Let Us C	4	Yashavant kanetkar	PQR	6th
Java Programming	6	The Wikibook	bpb	Ninth Edition

Fig-5.11 Search Example 8

Here  $X = \{\text{C++}, \text{Let Us C}, \text{Java Programming}\}$

$$Y = \{\text{C++}, \text{Let Us C}, \text{Java Programming}, \text{.Net 4.5}\}$$

So in this case

$$\text{Precision} = 3/3 = 1$$



$$\text{Recall} = 3/4 = .75$$

On averaging we get the values,

$$\text{Precision} = (.75+1)/2 = 0.875$$

$$\text{Recall} = (1+.75) = 0.875$$

## 5.2 Conclusion & future work

Recommendation system in E-Commerce is an emerging field. It increase the probability of purchase by the customer by suggesting the various items to the customer, because of this, they are extremely common these days. These are the alternative of the search algorithms. Generally in the earlier system, people mostly choose the content and collaborative approach and that's also with the k mean clustering. We can enhance the results in recommendation system by using various types of clustering. As an example, if we use the mixture of both fuzzy c-mean clustering and hierarchical clustering, we can significantly enhance the results. Using hierarchical clustering with the fuzzy c-mean, we can get the recommendation for the unique search key value. For example in our proposed recommendation system, we can also get the output (recommended items) by searching for the unique book code.

Future work related to the E- Commerce recommendation system to improve the result can be done by blending different algorithms into one, like we can do the mixture of fuzzy c-mean and hierarchical. We can also include some parameters also demographic parameters, which include gender, age, sex, location etc. For example people of a specific country like to choose novel to read written in their local language more.

## REFERENCES

---

1. [www.wikipedia.org](http://www.wikipedia.org)
2. J.S.R. Jang, C.T. Sun, E. Mizutani, "Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence" Prentice Hall, 1997
3. Jiawei han, Michelin Kamber, "Data Mining Concepts and techniques" Morgan Kaufmann, 2001
4. Giovanni Semeraro, Marco de Gemmis and Pasquale Lops, "Content-based Recommender Systems: State of the Art and Trends, "Springer Science and Business Media", 2011
5. Ming Chen, "Research on Recommender Technology in E-commerce Recommendation System", ICETC, 2010
6. Resnick, Paul, Varian Hal, "Recommender system", Communications of the ACM, 1997
7. J. Ben Schafer, Joseph Konstan and John Riedl, "Recommender Systems in E-Commerce", IEEE,
8. Przemyslaw Kazienko and Pawel Kolodziejski, "Personalized Integration of Recommendation Methods for E-commerce", IJSCA, 2006
9. YONG SOO KIM, "Recommender System Based on Product Taxonomy in E-Commerce Sites" JISE, 2013
10. T. Belluf, L. Xavier, R. Giglio, "Case study on the business value impact of personalized recommendations on a large online retailer", Proceedings of the Sixth ACM Conference on Recommender Systems. RecSys'12 (ACM, New York, 2012)
11. Stefano Ceri, Alessandro Bozzon, Marco Brambilla, Emanuele Della Valle, Piero Fraternali and Silvia Quarteroni, "Web Information Retrieval, Springer", 2013
- 12 Giovanni Semeraro, Pasquale Lops, Marco Degemmis, "A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation", User Model. User-Adapt. Interact, 2007

- 13 Schafer, J.B., Frankowski, D., Herlocker, J., and Sen, S., "Collaborative Filtering Recommender Systems", Adaptive Web, Springer,2007
- 14 John Riedl, Joseph Konstan, J. Ben Schafer, "Recommender Systems in E-Commerce", 1st ACM conference on Electronic commerce, New York, 1999
- 15 Jing Yang, Jing Yang, Wen Wu, "Evaluating Recommender Systems", IEEE,2012
- 16 Badrul M. Sarwar, George Karypis, Joseph Konstan, and John Riedl,"Recommender Systems for Large-scale E-Commerce: Scalable Neighborhood Formation Using Clustering, IEEE,
- 17 Billsus, D., and Pazzani, M. J.. Learning Collaborative Information Filters. In Proceedings of ICML"1998
- 18 Resnick, P., and Varian, H. R., "Recommender Systems", Special issue of Communications of the ACM, 1997
- 19 Goldberg D., Nichols D., Oki B. M., and Terry D., "Using Collaborative Filtering to Weave an Information Tapestry". Communications of the ACM, 1992
- 20 Jeremy York, Brent Smith and Greg Linden,"Amazon.com recommendations:Item-to-item collaborative filtering" IEEE,2003
- 21 Vignesh B, Subodh Kant, "Analytical Appraisal of Recommender Systems inE-Commerce", IRACST - International Journal of Computer Science and Information Technology & Security,2012
- 22 Yoon Ho Choa, Jae Kyeong Kimb, Soung Hie Kima, "A personalized recommender system based on web usage mining and decision tree induction", Expert Systems with Applications, 2002.
- 23 Rosario Girardi, Leandro Balby Marinho, "A domain model of Web recommender systems based on usage mining and collaborative filtering", Requirements Engineering Journal, 2006.
- 24 Taek-Hun Kim and Sung-Bong Yang," An Effective Recommendation Algorithm for Clustering-Based Recommender Systems",Springer,2005
- 25 B.M. Sarwar, J.T. Riedl, J.A. Konstan, G. Karypis, "Item-based Collaborative Filtering Recommendation Algorithms", 10th International World Wide Web Conference, 2001
- 26 Marco Degemmis, Pasquale Lops, Giovanni Semeraro, "A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation", User Model. User-Adapt. Interact, 2007
- 27 S.Sahar, Jaideep Srivastava, Kalyan Beemanapalli, Amit Bose, "Incorporating Concept Hierarchies into usage mining based Recommendations", ACM, 2006

- 28 Jianying Mai., Yanguang Shen, Yongjian Fan, “Study of the Model of E-commerce Personalized Recommendation System Based on Data Mining,” International Symposium on Electronic Commerce and Security(ISECS), 2008
- 29 J.T.Evaluating ,Herlocker, J.L., Konstan, J.A., Terveen, L.G., and Riedl, “Collaborative Filtering Recommender Systems”, ACM, 2004