Tag based Recommender System using Pareto Principle

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

Master of Engineering
in
Software Engineering

Submitted By
Akshama Rani
801231001

Under the supervision of:
Dr. Seema Bawa
Professor, Dean (Student Affairs)
CSED

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2014

Certificate

I hereby certify that the work which is being presented in the thesis entitled, **"*Tag based Recommender System using Pareto Principle*"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software 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. Seema Bawa*** and refers other researcher's work which are 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.

(Akshama Rani)

801231001

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

02/July/2014

(Dr. Seema Bawa)
Professor, Dean (Student Affairs)
Computer Science and Engineering Department
Thapar University
Patiala

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgement

Firstly, I am very grateful to almighty for his blessings and for directing me in the right direction. With his continuous support, I am able to achieve my objective.

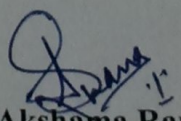
.

It is a great privilege to express my gratitude and admiration towards my respected supervisor **Dr. Seema Bawa** Professor Computer Science & Engineering Department. She has been an esteemed guide and great support behind achieving the task. This work would not have been possible without the encouragement and guidance of my supervisor. Her enthusiasm and optimism made this experience both rewarding and enjoyable. I am truly grateful to her for extending her total co-operation and understanding whenever I needed help and guidance from her.

I am also heartily thankful to **Dr. Deepak Garg**, Associate Professor and Head, Computer Science & Engineering Department and **Dr. Damandeep Kaur**, PG coordinator, for motivation and providing uncanny guidance and support throughout the preparation of the thesis report.

I would also like to thankful to entire faculty and staff members of computer science & Engineering Department for their direct and indirect help, co-operation and affection which made my stay at Thapar University memorable.

Most Importantly, I would like to thanks my parents, my sister and my friends for showing me right direction and help me stay calm in the oddest of the times and keep moving even at time when there was no hope.



(Akshama Rani)
801231001

Abstract

Recommender systems help users to cope up with large amount of data. Today large amount of data comes in very high speed and in different formats. This huge amount of data baffles internet users when they are searching for something, so this data is called Big data. Big data has 4 V model where 4 V stands for Volume, Velocity, Variety and Veracity. Recommender system deals with the Veracity aspect of Big data. Veracity denotes data has noises, abnormality and irrelevant patterns. Recommender system finds relevant data of user's interest from irrelevant patterns. Recommender systems provide personalized and non personalized recommendations to interested users. Recommender systems can be categorized into three ways according to the evolution of web. In Web 1.0 applications, traditional or rating based recommender systems came into existence. In web 2.0 application, social tagging information has been incorporated into recommender systems to improve the performance of traditional recommender systems, called tag based recommender systems. In web 3.0 applications, internet of things has been used for generating recommendations.

In this thesis, Pareto principle has been applied on social tagging dataset for providing good quality item and tag recommendations to users. A pre filtering step based on Pareto principle has been added to the traditional collaborative tagging system for removing irrelevant tags from k-Nearest Neighbour (kNN) selection process. Experiments have been performed on Movie lens datasets and experimental results show good quality item and tag recommendations.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

In the era of Internet, web is a giant source of information. The constantly growing rate of information in the web makes people confused to decide which product is relevant to them. To find relevant product in today's era is very time consuming and tedious task. Everyday a lot of information is uploaded and retrieved from the web. The web is overloaded with information and it is very essential to cop up with this overloaded and overlooked information. Recommender systems are the solution for solving the prevalent information overload problem faced by users of websites [1].

1.1 Big Data

Big data does not mean big size data it means huge amount of data comes in very high speed and in different formats. A large number of organizations are suffering from information overload problem. Due to this, Database sizes are also increasing exponentially. Data is coming from many sources such as sensors, RFID tags, online transactions, e-commerce sites and social networking sites etc. This huge amount of data baffles internet users when they are searching for something. Big data can be defined by 4 V's model and these 4 V's stands for Volume, Variety, Velocity and Veracity [20]. Volume denotes a huge amount of data; variety denotes diversity in data (e.g. data is coming from different sources in structured and unstructured form.); velocity denotes data is coming in very high speed or in real time; Veracity denotes data has noises, abnormality and irrelevant patterns. Veracity is the biggest challenge in big data analytics.

1.2 Big Data Analytics

Big data analytics is used to analyze large amount of data and to extract meaningful information from this bulk of data. Every business application requires big data analytics because in the era of internet, every data intensive application is facing the challenge of generating the relevant response in real time. Big data analytics helps in finding hidden patterns or insights in huge amount of data. It can be denoted as tools (database mining

and searching) and techniques (methods used in the process of analysis of data) [21]. It is used in various fields such as: natural language processing, text mining, spatial analysis, time series analysis, semantic analysis, sentimental analysis, social network analysis, graph analysis and recommender systems. Big data analytics has changed the way of doing business. With the help of it, business can retain their customers by satisfying them. Analytics helps company in knowing their customers liking and disliking and on the basis of that, enhance the user's experience and helps the user to find relevant products. In big data analytics, veracity is the major challenge. Veracity denotes low signal to noise ratio. It specifies huge amount of data contains relevant and irrelevant data. To find out relevant and most interesting patterns is the major issue for all web based companies. The main aim of data analytics is to transform data into knowledge. Google, Yahoo, Amazon and Facebook have the lion's share of ideas, tools and techniques to strengthen big data. For the massive amount of data, complex analytics approaches are required such as Machine learning algorithms, Data clustering, Predictive modelling and Data categorization.

1.3 Machine Learning

It is the branch of study that enables computers to learn without being programmed explicitly [39]. Machine learning algorithms can be divided into followings:

a. **Unsupervised Learning:** It specifies machine learn itself. Clustering algorithms are an example of unsupervised learning. In these algorithms, similar observations or items are clustered together.
b. **Supervised learning:** It specifies to teach machine how to learn. Classification algorithms are supervised machine learning technique.
c. **Recommender System:** It is other kind of learning. It uses both clustering and classification.

1.4 Recommender Systems

Recommender systems are the way to deal with the overload of information reflected in the increasing volume of information artifacts in the web. Recommender systems analyze

existing information on the user activities in order to predict future preferences. Recommendations are not something which user type in search engine and get the results; it is the search result which comes after matching the user's query. Recommendations are provided according to user's interest. Every recommender system implements a different paradigm for generating recommendations in heterogeneous domain [1]. Recommender systems have been recognized as an important tool on web science and e-commerce applications [35]. The process of making recommendations has been widely used since many years in every aspect of life. Before the dawn of Internet, recommender systems were still there but in personalized form. For example, if a person wants his daughter to get married, he takes recommendations from match makers and family members according to suitability of his daughter. If anyone wants to visit any tourist place, he normally takes recommendations from his friends who were known to that place or who have visited there earlier. As internet grows, many users started to take recommendations from the web. In fact, Recommender systems present among people who are not internet savvy and we can say that types of recommenders are wisdom of mouth. In this scenario, a person take recommendations for items from his friends and family members e.g. if a person wants to purchase some items such as clothes, he will take recommendations from his friend whether this clothe suits to him or not (binary rating) and may be ask him to rate his clothe on the scale of 5 or 10 then his friend give rating and on the basis of that rating, person make choice. Sometimes his friend also add keywords (tags) in their reply such as good, bad, good fabric, bad colour contrast etc. Collaborative filtering [1] [8] [16] recommender systems and social tagging recommender systems fulfil this task for internet users. In Movie recommender system, movies are recommended to users on the basis of rating given by users.

1.5 Types of Recommendations

The aim of recommender system is to retain customers by suggesting relevant products from the bulk of data. Recommender systems offer recommendations to users with the help of identifying their preferences. They have been used in various domains such as news, ecommerce sites, movies and social networking sites. They provide personalized and non personalized recommendations to users for items that are of their interest [1] [6].

i. **Personalized**

In personalized, recommendations are given according to user's preferences such as: if a user wants to go to restaurant, he has to give his preferences about location, food and many more things beforehand for getting good recommendations of restaurants according to his taste. Amazon, an E-commerce company provides personalized recommendations to their users.

ii. **Nonpersonalized**

In non personalized recommendations, recommender systems provide recommendations according to the content: For example, news recommender system such as Google news, recommends users news similar to the news that user has been watching.

1.6 Recommender System Taxonomy

Recommender Systems have evolved from traditional or web 1.0 recommender systems to social or web 2.0 recommender systems and then web 3.0 recommender systems [6].

1.6.1 Traditional Recommender Systems

Traditional Recommender systems are based on different approaches for providing effective recommendations to users. These approaches are content based filtering, demographic filtering, collaborative filtering and hybrid. In content based filtering, recommendations are provided to users by considering their past preferences for items (e.g. in e-commerce recommender system, if the user purchased a data mining book in the past, the recommender system will recommend another data mining book that the user has not yet purchased). Demographic filtering make recommendations to active user by considering the ratings of other users who share the same age, sex, location, gender, etc. as the active user [1] [6] [32]. In Collaborative filtering, ratings of a large number of users on a large number of items are considered and on the basis of provided ratings, similarity among users or items is calculated and item is recommended to the active user by considering the similar user's preferences. Ratings can be implicit and explicit. Implicit ratings denote number of times a page is visited and how much time is spent on a particular item. Explicit ratings denote ratings provided by users on the scale of 1 to 5.

Collaborative filtering is the most successful approach among all. In hybrid approach, a combination of two or more than two approaches is taken. In Figure 1.1, Movie is recommended to user by considering the rating behaviour of all users.

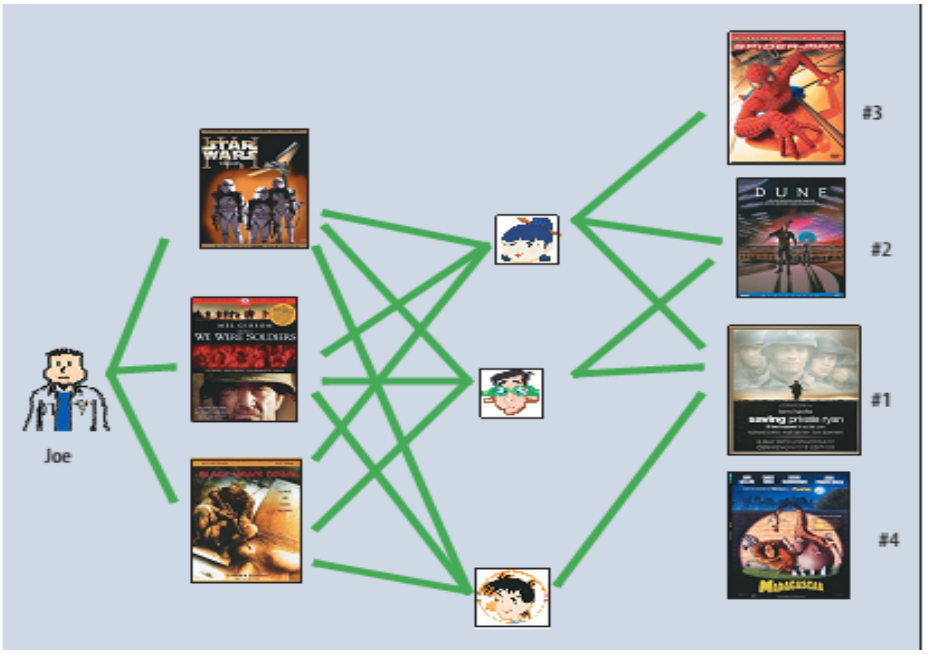


Figure 1.1: Movie recommendation [24]

In left hand side, Jae love to see three movies. For making recommendation of movies for Jae, the recommender system searches same minded users who also love to see those movies, and then tells which other movies they also want to see. In the given scenario, all other three persons liked movie Private Ryan and this is the first recommendation to Jae. Second one is Dune because it is liked by two of them and so on.

1.6.2 Social Recommender Systems

Since the introduction of the Tapestry systems, recommender systems have been in existence [16]. Social media has become famous now days. Eminent examples are social resource sharing sites: Delicious (Bookmarks), Flickr (Images), CiteULike (Bibliographic), Youtube (videos), Slashdot (information) and Social networking sites: MySpace (Music), Twitter (micro blogger), Epinion (Product Review), Flixter (movie review), Facebook, and Linkedin. A social network models relationship between different users and information is exchanged between them according to their relationship [3]. Data in Social network is usually shown with graphs and matrices. Social network is a set

of nodes (actor or user) and edges (relationship between actors) [27]. As shown in Figure 1.2, Relationships between the users are of two types: directional (marriage, cusions) and non directional (friendship, seller–buyer, and employer-employee). In the social recommendations: tag recommendations, people recommendations, and content recommendations take place. In the social network, items can be social entities such as persons or group of persons.

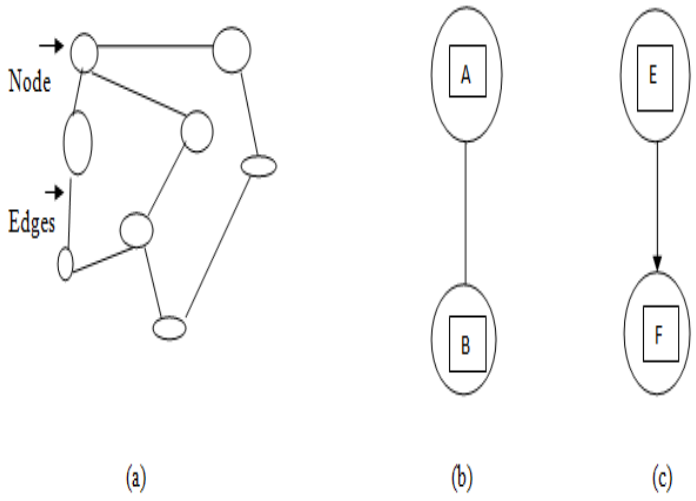


Figure 1.2: Social Relationship a) Social Network (Node-Actor, Edges-Social Relationship), b) Nondirectional Relationship (marriage), c) Directional Relationship (Seller- Buyer)

1.6.3 Tag based Recommender Systems

Social Tagging is a different kind of social network where nobody is actually connected with others but they are connected with only content or information. Social Tagging denotes collaboratively annotate web resources such as web pages, blogs, images, videos, bibliographic references with textual labels or keywords known as tags [4]. Users tag those items that they find interesting and relevant. In Social tagging sites such as Del.icio.us, Flickr and CiteULike, users share web resources like web pages, images, bibliographic references. Social Tags are used to label an item for content indexing, fast searching and retrieval. In social tagging one user makes relation with other user only when one user finds other's content relevant [25]. This kind of relation is less restricted

because there is no mutual agreement between users. Social tagging also creates folksonomy that means relationship between different tags. Folksonomy denotes the classification system that arises from collaborative tagging. Folkosonomy can overcome the problems of web authors of maintaining the meta data by aggregating the free labels tags of many users. Tags introduce semantic relationship among items and it improves the precision and recall [42]. Folkosonomy lacking in semantics such as synonym and polysemy but with the help of tag ontology, these problems can be solved. WordNet dictionary can be used to find out tag's meanings, synonyms etc [40].

1.7 Pareto Principle (80-20 Rule)

Pareto Principle introduced by Vilfredo Pareto, an Italian Economist. This principle states that 80 percent of effects come from 20 percent of causes that's why it is also called 80-20 rule. Figure 1.3 demonstrates this rule as 80 percent of sales come from 20 percent of clients in business management. In mathematical term, Power distribution Law follows Pareto Principle and known as Pareto distribution.

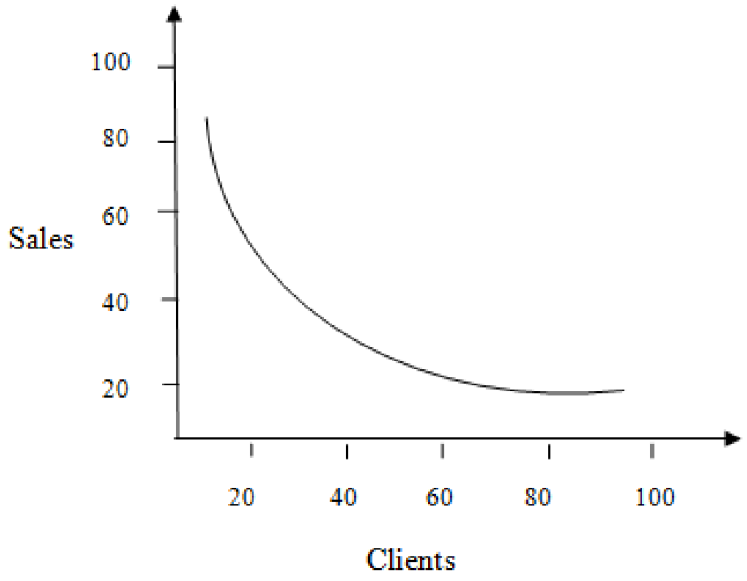


Figure 1.3: 80-20 Rule

Chapter 2 Literature Survey

In this chapter, a brief survey on recommender systems and machine learning approaches that are used in the recommendation processes are discussed and then related work to the taxonomy of recommender system are explained. Recommender systems are classified in three generations: web 1.0 recommender systems, web 2.0 recommender systems and web 3.0 recommender systems.

2.1 Approaches used in Recommender systems

Many famous websites have been using recommender systems for improving their sales such as Netflix, a web based movie rental service and Amazon, an e-commerce company [35]. Netflix Prize contest was a well known contest in the history of recommender systems, it offered 10 million US dollar for improving the collaborative filtering algorithm by 10.6% accuracy. Many approaches are used in recommender systems and each approach has its advantages and limitations. Some of these approaches are discussed as follows:

2.1.1 Collaborative filtering

In the mid 1990's collaborative filtering algorithms were introduced. This approach is very successful approach among all. Recommender systems based on it recommends items according to the rating behavior of users. Ratings of the recommended items are given by the users similar to active user to whom recommendations are offered or similar to the items previously rated by the active user. But there is a contradiction that two users may have similar preferences in one category but different preferences in another. This approach does not work efficiently in sparse datasets which denotes less number of ratings on the items. It suffers from Cold start problem that means there is no rating for newly added items and newly added users do not rated any item then how recommendations is possible for these kinds of users and items. This can be further divided as Memory based approaches and Model based approaches [1-6] [16-19]. Memory based approach further has two types user based and item based.

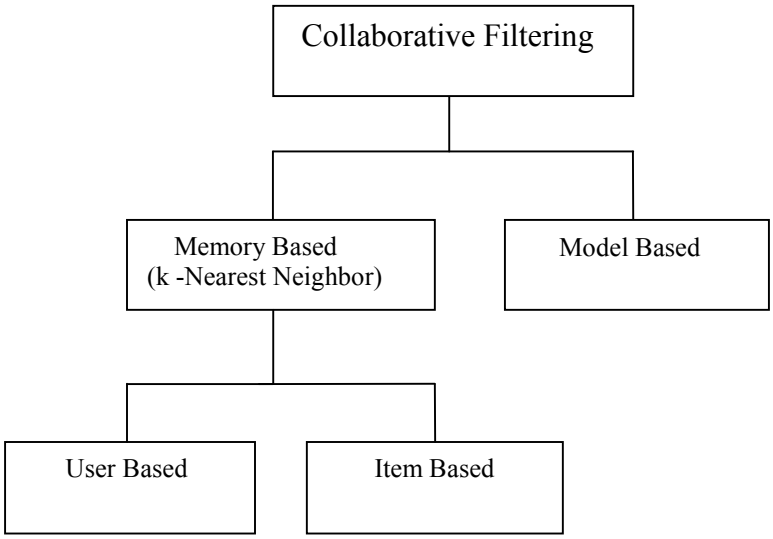


Figure 2.1: Collaborative Filtering Classification

2.1.1.1 Memory based Collaborative Filtering

This approach considers the whole database of user's ratings. On the basis of user's rating behavior, similarity between users or items is calculated with similarity measures. The most widespread memory based algorithm is "neighborhood algorithm" or we can say k-Nearest Neighbor algorithm [1] [7]. In this type of approach, firstly, similarity or weight or distance is calculated with similarity measures such as pearson correlation, cosine similarity, euclidean distance etc.

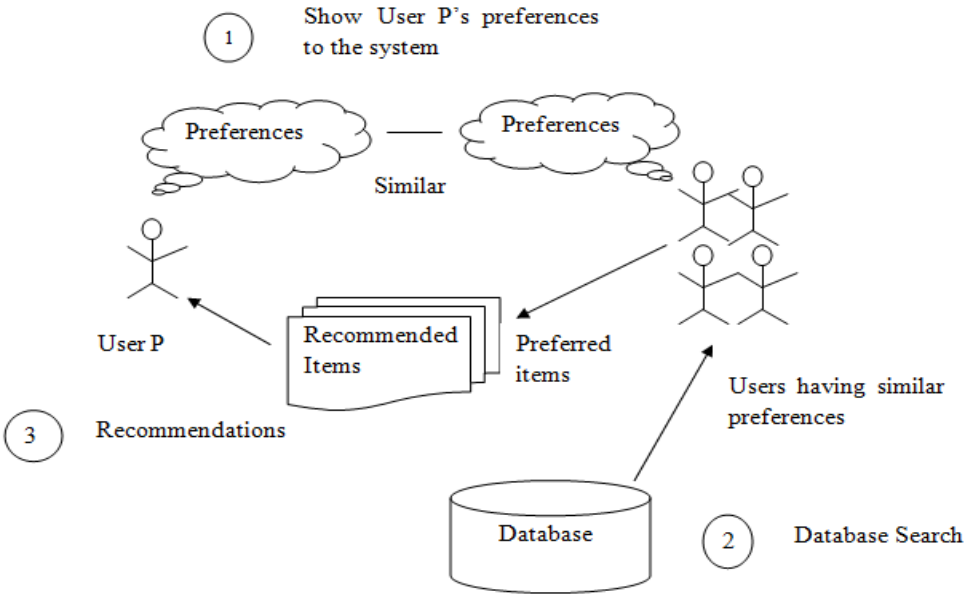


Figure 2.2: Memory based Collaborative Filtering

From similar users, k most similar users are taken and then weighted average of ratings of all similar users or items are taken for generating recommendations. Memory based approach has certain advantages and disadvantages. It is easy to implement. It has main disadvantage that it is completely dependent on human ratings. It does not work well in sparse data. It has further two variants: user based and item based.

2.1.1.1.1 k-Nearest Neighbour (kNN) Classification approach

This approach is also called lazy learning or instance based learning. In this supervised approach, k nearest neighbour of a feature vector is found in the feature space. This approach is used for classification. In Figure 2.3, suppose we have N training vectors and 'b' and 'c' letters as training vectors be dimensions of feature space. The kNN algorithm [7] finds the k nearest neighbour of a feature vector 'a' and then estimate the class of 'a' on the basis of maximum neighbours on a particular class. For finding the class of a, we are taking k = 4 that means 4 nearest neighbour of 'a'. In the 4 nearest neighbour, one vote is for 'b' and 3 vote is for 'c' that shows 'a' is more similar to 'c' and we can say that 'a' belongs to 'c' class as shown in Figure 2.4. These figures show how kNN algorithm works. kNN algorithm is simple to implement.

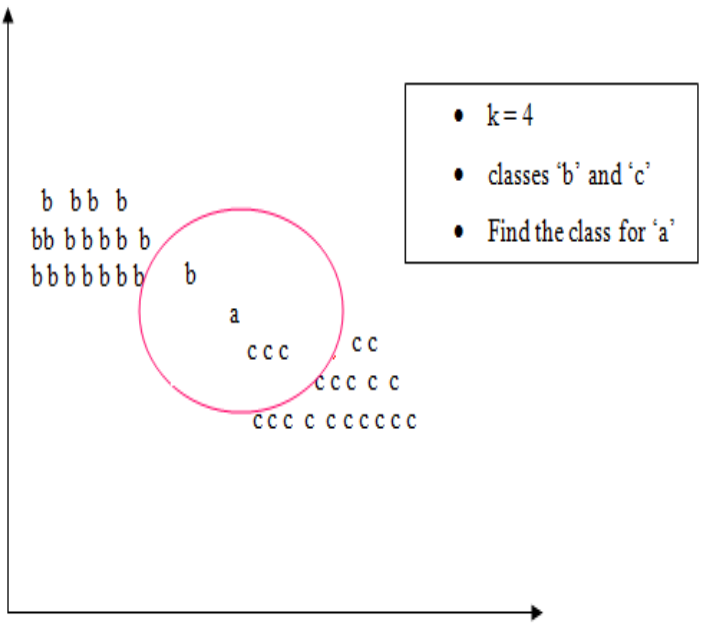


Figure 2.3: Find k Nearest Neighbour of 'a'

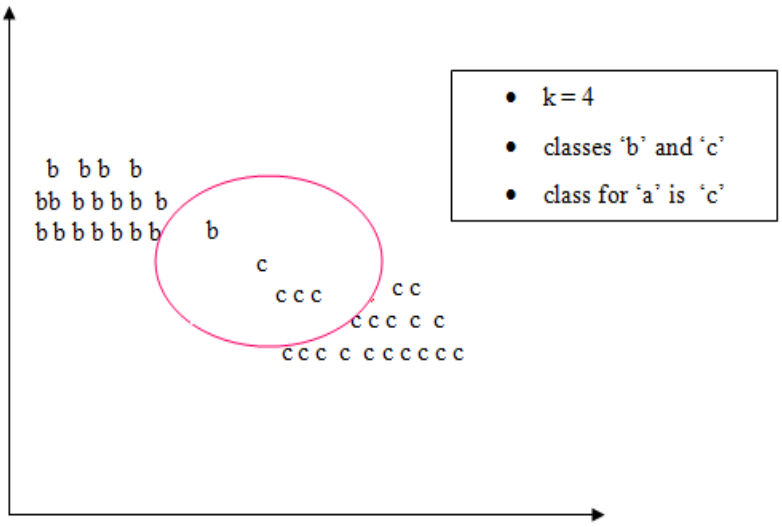


Figure 2.4: Nearest Neighbour of 'a' is 'c'

a. User based approach: User based approaches consider the similarity among users and make the group of similar users. After that algorithms recommend each and every user, the items suggested by other users in the same group. However, there are some potential challenges for user-based collaborative filtering methods [1] [6]. For example, sparsity in the user-item matrix which causes the mining of user similarity difficult and inaccurate. Second is the scalability. In Figure 2.5, item A is liked by both users 1 and 2, item B is liked by only user 3 and item C is liked by all three users 1, 2 and 3. We want to get recommendation for user 3. Item A is recommended to user 3 on the basis of user- user similarity.

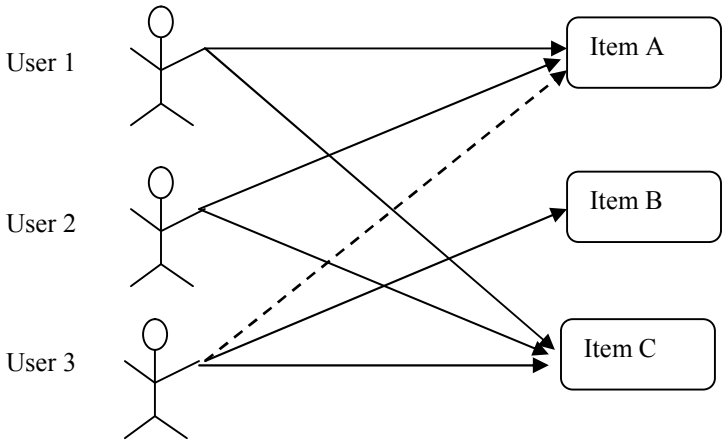


Figure 2.5: User based Approach

b. Item based approach: this method explores the similarity among items. For each user, they suggest the items similar to the preferable ones of the user.

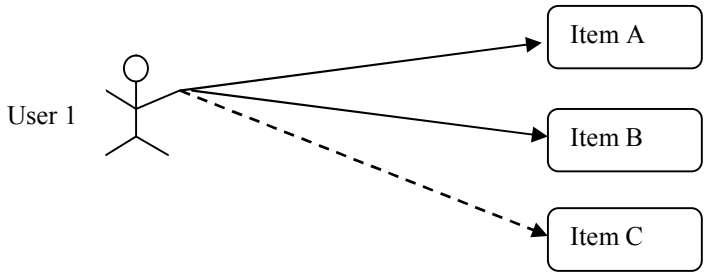


Figure 2.6: Item based Approach

According to the long tail theory, there can be a significant statistics with respect to each item in case of large number of users. There is large amount of metadata for users as compared to for items [1] [6]. Hence, the item-based methods have better scalability. In Figure 2.6, item A and B are liked by user 1 and item C is similar to both A and B. So, Item C is recommended to user 1.

2.1.1.2 Model based Collaborative Filtering

In this approach, a model is prepared using Bayesian belief nets, latent semantic analysis, sparse factor analysis and dimensionality reduction techniques such as SVD and PCA. The whole system is learnt to find complex and hidden patterns with this model using training data. This approach overcomes the shortcomings of memory based collaborative filtering. Model based approach deals with sparsity and scalability issues. But it is an expensive implementation technique. The loss of information takes place when dimensionality reduction techniques are used [1] [6].

2.1.2 Content based Filtering

This approach recommends items to active user according to the content information of items. In the domain of movies, recommender system recommends movies to the users according to the characteristics of movies. This type of recommender systems recommend those items which are similar to the item purchased or seen by user in past. These systems do not bother about ratings given by user to the items. Mainly, Text based

items are recommended by this approach [1] [2] [6] [32] [33].

2.1.3 Demographic Filtering

This approach considers the ratings of users who have same age, sex and location as active user has for the recommendations [1] [6] [32] [33].

2.1.4 Knowledge based Filtering

In this filtering, a matching of user's need and set of options available is occurred and then on the basis of matching, recommendation takes place. These recommender systems have knowledge about how the need of a particular user is met through a particular item. For example knowledge mined from user's profile will be helpful in recommendations [1] [6].

2.1.5 Hybrid Filtering

Hybrid denotes combination of two or more than two approaches. Hybrid approach is used to mitigate the limitations of individual approaches. Collaborative filtering approach can be combined with content based, knowledge based and demographic based approaches. The collaborative filtering recommender systems are the most successful approach among all at this time. Collaborative filtering (CF) approach used by recommender systems is a way to predict the usage of items for users based on the similarity among user's preferences and the preferences of other users. Amazon, Epinion adopted social recommender systems for improving the accuracy of their prediction [1], [6] [9] [33].

2.2 Similarity Measures

Similarity measures are used to calculate the similarity between users. In Memory based collaborative filtering, user to user similarity and item to item similarity is find out by using these similarity measure. In these measures, user's rating is considered and on the basis of rating behavior, similarity is calculated. Memory based CF such as kNN algorithm is dependent on these similarity measures. Pearson correlation coefficient,

Cosine similarity, Adjusted cosine similarity, Mean squared difference, Euclidean Distance and Jaccard index [1] [6] [32]. In Table 2.1, User-item rating matrix is shown.

Table 2.1: User- Item Rating Matrix

Users	**Item**			
	I	***J***	***K***	***L***
A	1	3	ε	4
B	3	ε	2	5
C	E	4	3	ε

I. Pearson Correlation coefficient

It is used to measures the relationship between two variables [1] [6] [32].

$$Corr(a,b) = \frac{\sum_{i \in S_{a,b}} (r_{a,i} - \bar{r}_a).(r_{b,i} - \bar{r}_b)}{\sqrt{\sum_{i \in S_{a,b}} (r_{a,i} - \bar{r}_a)^2 . \sum_{i \in S_{a,b}} (r_{b,i} - \bar{r}_b)^2}}$$

Where $S_{a,b}$ be the set of items rated by both user a and b, $r_{a,i}$ be the rating given by user a to item i, $r_{b,i}$ be the rating given by user b to item i and $\bar{r}_a$, $\bar{r}_b$ be the average rating of user *a* and user *b*. *Corr*(*a*, *b*) = [-1, 0, 1] ; Here, -1 shows negative correlation, 0 shows independent and 1 shows positive correlation

II. Cosine Similarity

It is used to measure the similarity between two vectors by finding cosine angle between them.

$$Cos(a,b) = \frac{\sum_{i \in S_{a,b}} r_{a,i} \cdot r_{b,i}}{\sqrt{\sum_{i \in S_{a,b}} r_{a,i}^2 \cdot \sum_{i \in S_{a,b}} r_{b,i}^2}}$$

Where $S_{a,b}$ be the set of items rated by both user a and b, $r_{a,i}$ be the rating given by user a to item i, $r_{b,i}$ be the rating given by user b to item I [1] [42].

$Cos(\theta) = \frac{a.b}{\|a\|\|b\|}$, where $a.b$ is a dot product of two vectors $\vec{a}$, $\vec{b} \Longrightarrow \|a\| \|b\| \cos\theta$.

III. Adjusted Cosine Similarity

It is used to measure the similarity between two items rated by all users to find out the item to item similarity. It overcomes the drawback of cosine similarity by considering the rating scale of different user.

$$Acos\ (i,j) = \frac{\sum_{x\in X}(r_{x,i} - \bar{r}_x).(r_{x,j} - \bar{r}_x)}{\sqrt{\sum_{x\in X}(r_{x,i} - \bar{r}_x)^2} \cdot \sqrt{\sum_{x\in X}(r_{x,j} - \bar{r}_x)^2}}$$

Where X is a set of users whose rating is considered for finding out similarity between two items. $\bar{r}_x$ is the average rating of user [1] [43].

IV. Mean Squared Difference

This similarity measure firstly calculates the difference between the ratings of users and then calculate mean between them. It is also used to find similarity between users.

$$Msd(a,b) = 1 - \frac{1}{\#S_{a,b}} \sum_{i\in S_{a,b}} \left(\frac{r_{a,i} - r_{b,i}}{max - min}\right)^2$$

Where $S_{a,b}$ be the set of items rated by both user a and b, $\#S_{a,b}$ is the cardinality of set $S_{a,b}$. $r_{a,i}$ be the rating given by user a to item i, $r_{b,i}$ be the rating given by user b to item i. max and min is the maximum and minimumrating value of the system [1] [6] [32] [43].

V. Euclidean Distance

This measure calculates distance between two points or objects in a plane and on the basis of this distance, similarity between two items or points can be found [6].

$$d(a,b) = \sqrt{(b_1 - a_1)^2 (b_2 - a_2)^2 .. (b_n - a_n)^2}$$

$$d(a,b) = d(b,a) = \sqrt{\sum_{i=1}^{n}(b_i - a_i)^2}$$

Where a $(a_1, a_2 \ldots . a_n)$ and b $(b_1, b_2 \ldots . . b_n)$ are two points in Euclidean n- space. The distance from a to b and b to a is given above. The position of point in Euclidean n-space is Euclidean vector.

VI. Jaccard index

This coefficient measure similarity between two sets U and V [32].

$$J(U,V) = \frac{U \cap V}{U \cup V}$$

The size of intersection between two sets is divided by the size of union of two sets U and V. $0 \leq J(U,V) \leq 1$

2.3 Evaluation Metrics

Evaluation is an important part of any recommender system because without evaluation we can not infer whether results of the system are right or not. Comparison between two

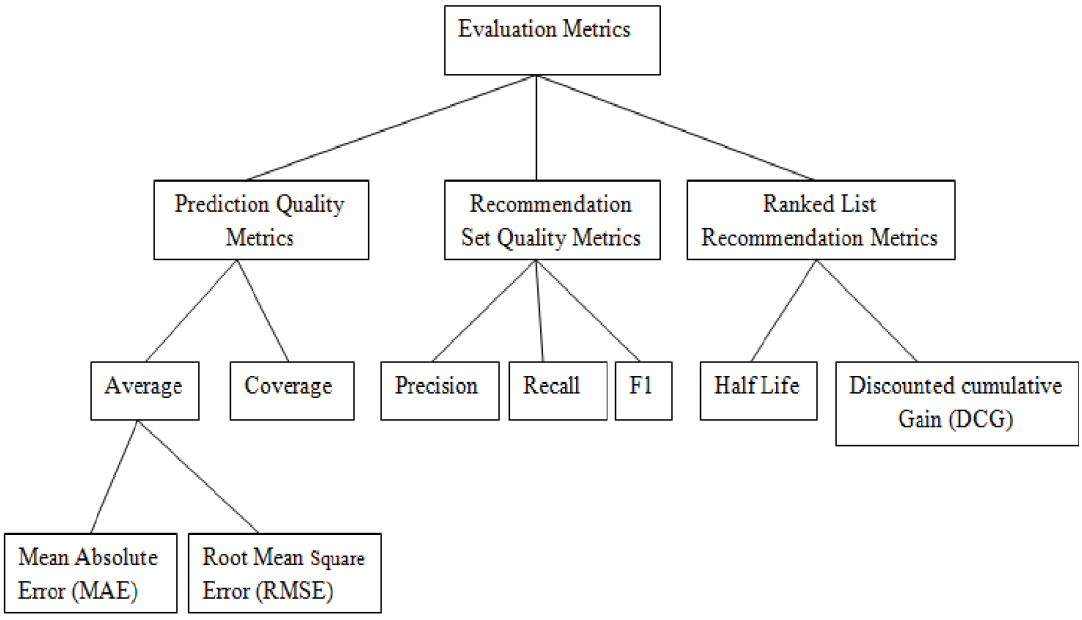


Figure 2.7: Classification of Evaluation Metrics

systems can be possible only after performing evaluation. With the help of evaluation, improvement in the system can be done. Quality of techniques, algorithm and procedure is evaluated for good prediction and recommendation. Evaluation metrics can be classified into different metrics as shown in Figure 2.7 [1] [6] [33] [40] [43].

2.3.1 Prediction Quality Metrics

These metrics are used to evaluate the prediction of the system. Prediction is to know what will user like in future. Accuracy and coverage are important metric for evaluating the prediction. Accuracy is further divided into *MAE* and *RMSE*.

Suppose $D_a = \{i \in I : p_{a,i} \neq \varepsilon \wedge r_{u,i} \neq \varepsilon\}$, set of items rated by user a having prediction and rating value is not equal to null. Error in the prediction is calculated by taking the difference between actual ratings and predicting rating, $|p_{a,i} - r_{a,i}|$ informs the error in the system. Here *MAE* and *RMSE* are discussed.

i. **Mean Absolute Error (*MAE*)**

It is used to calculate deviation between the actual results and predicted results [1].

$$MAE = \frac{1}{\#A}\sum_{a \in A}\left(\frac{1}{\#D_a}\sum_{i \in D_a}|p_{a,i} - r_{a,i}|\right)$$

ii. **Root Mean Square Error (*RMSE*)**

It is also used to calculate error between actual results and predicted results [6].

$$RMSE = \frac{1}{\#A}\sum_{a \in A}\sqrt{\frac{1}{\#D_a}\sum_{i \in D_a}(p_{a,i} - r_{a,i})^2}$$

iii. **Coverage (*C*)**

It is used to measure the capacity of user's neighbors to predict new items. User's coverage is shown in equation [6].

$$C = \frac{1}{\#A}\sum_{a \in A} 100 \times \frac{\#\{i \in I \,|r_{a,i} = \varepsilon \wedge p_{a,i} \neq \varepsilon\}}{\#\{i \in I | r_{u,i} = \varepsilon\}}$$

2.3.2 Recommendation Set Quality Metrics

These measures evaluate the quality of recommendation. User confidence is based on the accuracy and quality of the recommendation offered by the system. Recommendation quality measures inform whether system is recommended relevant content or not. Suppose X_a is the set of recommendations and Z_a is the set of N recommendations Offered to user a. Here, θ is the threshold value of relevancy and it is assumed that all users accept N recommendations.

a) Precision (*P*)

It is the ratio between the relevant recommended items to the total number of item recommended [1] [6] [8-10] [32-38] [42].

$$P = \frac{1}{\#A}\sum_{a\in A}\frac{\#\{i \in Z_a | r_{a,i} \geq \theta\}}{N}$$

b) Recall (*R*)

It is the ratio between the relevant recommended items to the total number of relevant items [1] [6] [8-10] [32-38] [42].

$$R = \frac{1}{\#A}\sum_{a\in A}\frac{\#\{i \in Z_a | r_{a,i} \geq \theta\}}{\#\{i \in Z_a | r_{a,i} \geq \theta\} + \#\{i \in Z_a^c | r_{a,i} \geq \theta\}}$$

c) F1 Metric

The Combination of precision and recall is called F1 metric [1] [6] [8-10] [32-38] [42].

$$F1 = \frac{2 \times P \times R}{P + R}$$

2.3.3 Ranked list Recommendation Metric

When the amount of recommended items is enormously large then user considers only first few items in the list of recommendation. The error occurs in these recommendation are more serious than last recommendation in the list. The ranking Metric considers this scenario. Half Life and Discounted cumulative gain (DCG) are the two most important ranked metrics.

I. Half Life (HL)

It specifies the interest of user decreases exponentially when user move down in the list of recommendation [6].

$$HL = \frac{1}{\#A}\sum_{a\in A}\sum_{j=1}^{n}\frac{max\left(r_{a,p_j} - d_r, 0\right)}{2^{(j-1)/(\alpha-1)}}$$

II. Discounted Cumulative Gain (DCG)

It specifies user's interest decreases logarithmically [6].

$$DCG = \frac{1}{\#A}\sum_{a\in A}\left(r_{a,p_j} + \sum_{j=2}^{K}\frac{r_{a,p_j}}{\log_2 j}\right)$$

Here, recommendation list is represented by $p_1, p_2 \ldots\ldots p_n$, r_{a,p_j} denotes user a's rating for item p_j ,α denotes number of items in the list and there is half chance of user will review it.

2.4 Related Work

In this chapter, some of the research studies related to the web 1.0 and web 2.0 recommender systems are briefly presented here. Web 1.0 recommender systems are those systems which is used in the e-commerce; web 2.0 recommender systems are tag based recommender systems which incorporates tagging information in recommendation process and are used in social networking sites and web 3.0 recommender systems use location information in recommendation process.

2.4.1 Web 1.0 Recommender Systems

Since the web 1.0 recommender systems developed, item recommendations have become the main area of concern for researchers who investigated many approaches of recommendations. In Table 2.2, comparison between approaches of first generation recommender systems is shown.

Goldberg et al. [16] presented a tapestry email system that efficiently uses collaborative filtering approach for filtering the queries of user in mailing list. Users read only those documents which are reviewed by other users before.

Herlocker et al. [18] proposed neighborhood approach of collaborative filtering for finding similarities between users and items. They used automated collaborative filtering approach for increasing the accuracy of recommendations. They also discussed many evaluation metrics such as coverage, accuracy and ROC.

Sarwar et al. [34] explained that collaborative filtering uses item similarity is better than collaborative filtering uses user similarity. They addressed two issues of recommender system: scalability of algorithms and quality of recommendations by introducing new item based approach of collaborative filtering. As relationship between users is dynamic and relationship between items is static, item based approach takes less online computation time. They basically dealt with scalability issue face by user based approach.

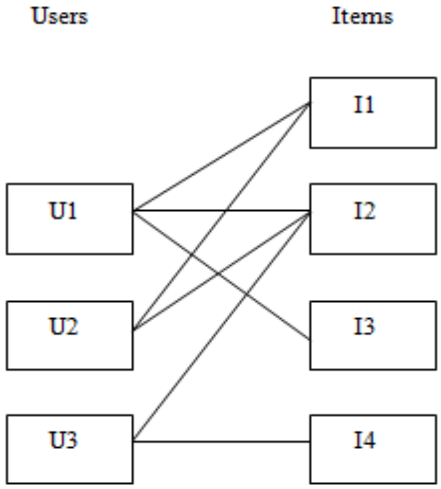


Figure 2.8: Users-Items relation (Binary)

Adomavicius *et al*. [1] presented first generation recommender systems. They also classified recommendation methods into three main approaches: content based approach collaborative approach and Hybrid approach. Content based recommendations consider the past transaction behavior of users and recommend items that are similar to the previously purchased items by analyzing their content. Collaborative filtering recommend items to users on the basis of rating behavior of like-minded users and hybrid approach combine content based and collaborative approaches to mitigate the limitations of each

approach for providing more accurate results. The Web 1.0 recommender systems have two major building blocks: users and items and there is a binary relationship among them. Users rate items on the basis of their preferences and rating can be binary (like or unlike an item) or on the scale of 1 to 5. As shown in Figure 2.8, users (U1, U2 U3) and items (I1, I2, I3, and I4) are related with each other. Items I1, I2 and I3 are liked by user U1; Items I1 and I2 are liked by user U2; Items I2, I3 and I4 are liked by user U3. On the basis of similarity in user's preferences, Items I3 and I4 liked by U1 and U2 correspondingly are recommended to user U2 according to their rank.

Pazaani [33] used a framework of collaborative filtering (CF), content based filtering (CB) and demographic filtering (DF) for recommending restaurants.

Ortega *et al.* [32] proposed an improvement in traditional collaborative filtering recommender system. In traditional collaborative filtering, a posteriori phase is imprecise means a large number of active user's neighbours are selected. In the neighbours of active user some are less representative and some are more representative. The new approach proposed uses Pareto dominance [80-20 rule] to perform pre filtering step in k-nearest neighbor selection process for eliminating the less representative users from the neighborhood of active user.

Koren *et al*. [24] described the matrix factorization techniques and performed comparison between matrix factorization and classic nearest neighbor technique on Netflix database. In the result matrix factorization is superior to nearest-neighbor techniques for generating item recommendations.

Schafer *et al.* [35] explained how a recommender system changes the business of Ecommerce and help Ecommerce sites for increasing its sales. They analyzed several sites that use recommender systems and more than one recommender system. They created taxonomy of recommender systems and discussed technologies that are used in recommender system and input files that are needed from customers.

Bobadilla *et al*. [6] presented the recommender systems evolution. As time passed, recommender systems evolve from first generation, second generation to third generation. In first generation recommender systems, e-commerce recommender system comes into scene and various approaches are used to make them efficient and in second generation, social information is also incorporated in recommendation process for improving the

prediction results and overcoming the limitations of first generation recommender systems and in third or current generation location aware recommender systems are introduced. In Figure 2.9, Taxonomy of Recommender systems are shown.

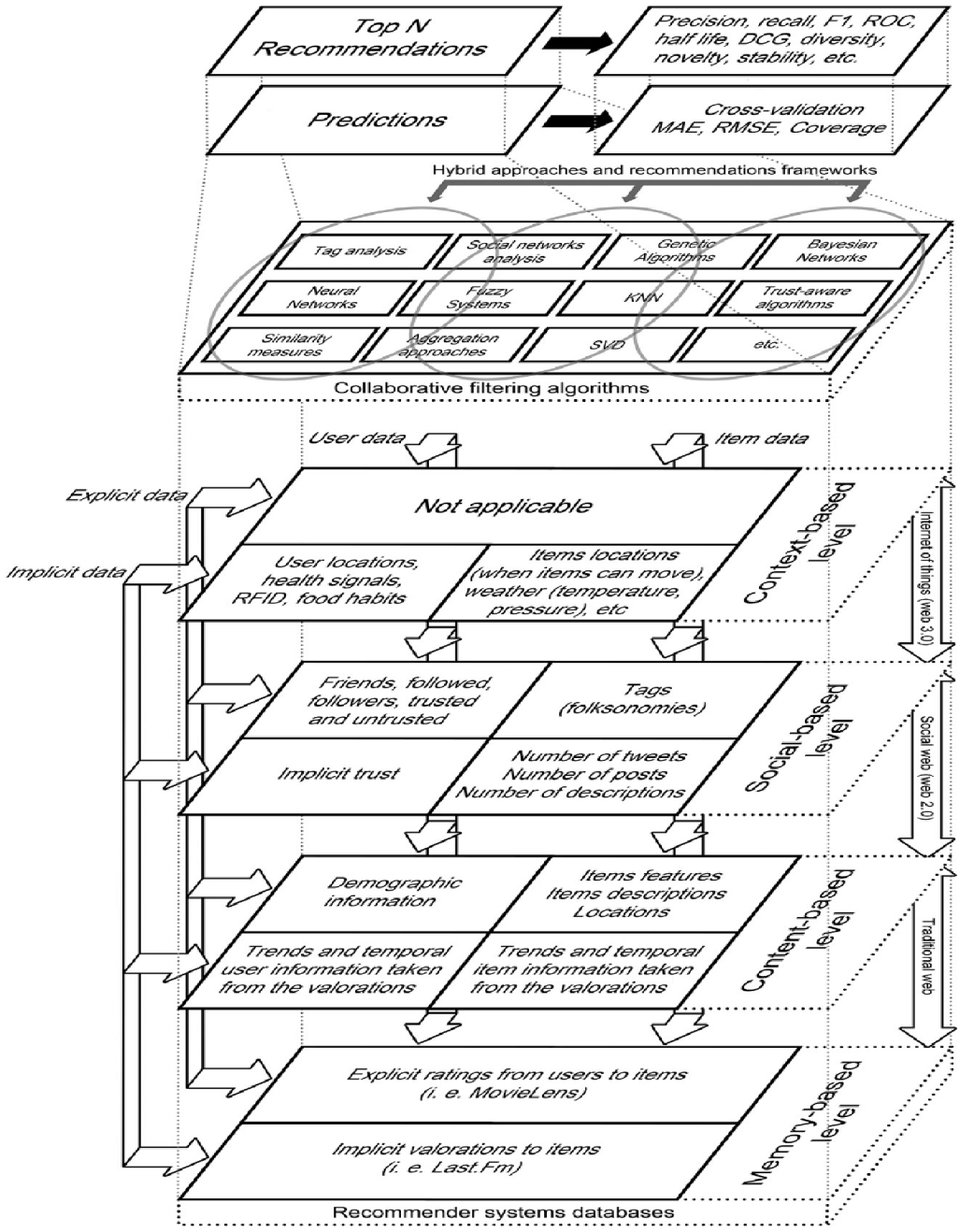


Figure 2.9: Recommender Systems Taxonomy [6]

Bobadilla *et al*. [8] proposed the improvement in traditional similarity measures. In traditional collaborative filtering method, the most similar users are discovered for whom

we want to make recommendation. A new approach that takes care of contextual information and Singularity is considered. If there is greater singularity, there is more similarity. The results are tested on Netflix, Movielens databases and shows excellent behavior.

Bobadilla *et al.* [7] described the improvement in K-nearest neighbor (KNN) algorithm, the core of collaborative filtering. The KNN algorithm is non-scalable in nature and has high execution time. This algorithm is based on repeated execution of similarity metric. They introduced new similarity metric HWSimilarity. This metric has high quality recommendation and employing low- cost hardware circuits.

Walunj *et al.* [39] proposed a successful implementation of a mahout framework provided flexibility in using pre-existing algorithms. It described challenges of collaborative filtering like Scalability, Synonymous, Grey sheep, Shilling attacks, Diversity. It solved the problem of scalability by using the hadoop platform because it is built on the hadoop framework. Apache mahout offers testimony that a recommendation system provides customizable recommendations enable online companies to perform business more effectively.

Table 2.2: Web 1.0 Recommender Systems

Approach	Advantages	Limitations	Domain
Collaborative Filtering [16]	Efficient algorithm for filter queries in e-mail system	Security	Electronic Document
Collaborative Filtering (Neighborhood) [18]	Good Prediction Accuracy	Scalability	Movie (Movielens)
Collaborative Filtering (Item based) [34]	Scalability and High online Performance	Performance decreases as neighborhood size decreases	Movie (Movielens)

Content Based Filtering(TF-IDF) [1]	Fast information retrieval in Text based items	Limited content analysis, Overspecialization and New User Problem	News article, Books and Movies
Collaborative Filtering (Memory based and Model based) [1]	No overspecialization and limited content analysis problem	New user Problem, New item problem (cold start) and sparsity	News article, Books and Movies
Hybrid Filtering (Collaborative and Content based) [1]	Overcome limitations of content based and collaborative based approach	Complex to implement	News article, Books and Movies
Collaborative Filtering, Content based and Demographic Filtering [33]	High Precision	Complex to implement	Restaurants
Improvement in Collaborative Filtering (kNN) Using pareto dominance [32]	Improvement in Quality Measures	Improvement is applied only on memory based Collaborative Filtering	Movie (Movielens, Netflix)
Matrix Factorization Technique(CF) [24]	Accuracy in recommendation as compared to	Loss of information due to factorization	Movie (Netflix)

	classical Nearest Neighbor approach		
Collaborative Filtering using Singularity concept [8]	Improvement in Prediction quality and Recommendation quality	Large Formulation Required	Movie (Movielens, Netflix, Filmaffinity)
Collaborative Filtering using Apache Mahout [39]	Provide flexibility using pre-existing algorithms and solves the scalability problem by using hadoop.	Collaborative filtering with Mahout approach is not suitable for time sensitive applications	E-Commerce

2.4.2 Issues in Recommender Systems

Ghazanfar *et al.* [15] introduced an approach for reducing the error rate in collaborative filtering algorithm caused by gray sheep user problem. They identified gray sheep user by using various clustering approach like K plus means clustering algorithm, various distance measures and improved centroid selection approaches. For generating accurate recommendation, user's profile is used. This is the first attempt in the direction of solving gray sheep user problem and with this attempt, accuracy and coverage of recommendation is also improved.

Blanco-Fernández *et al.* [5] presented a strategy for solving the problem of overspecialization by using the reasoning techniques taken from semantic web. These techniques are implemented in recommender system for digital Television and these techniques provide accurate recommendation.

Lika *et al.* [28] proposed a technique that deals with a problem named cold start. Cold start problem arises in CF system. CF approach recommends items to the user on the basis of the rating of the similar user who share the same interest with active user. But if a user new to the system and did not rate any item yet, recommendation can not be

possible in this scenario because user's neighbor can not be found. If a new item is added to the system and nobody has rated it then this item can also not be recommended. This problem is called cold start problem. Cold Start problem further divided into three parts: a) new user, b) new item, c) new user, and new item. They presented a method for removing the new user problem by incorporating demographic data for finding the similar user and then classified the new user in a particular group and employ prediction mechanism.

Chung *et al.* [12] proposed an approach for shilling attack detection. This approach filtered out the malicious rating from the recommender system. Collaborative filtering (CF) approach is widely used in the recommendation and due to its open nature, it suffers from many vulnerabilities. Many attack detection algorithms such as PCA based algorithm, classification based and detector based on SPC (statistical process control) algorithms have been introduced for handling this issue but all of these are restricted by various constraints. Beta-protection (βP) algorithm is introduced for removing this problem.

Zhan *et al.* [41] presented a privacy preserving approach in recommender system. As recommender system is successfully used in e-Commerce, users want relevant and precise recommendation and this can be possible only when two or more than two companies merge their database for overcoming the problem of limited database. Due to privacy disclosure, there can be numerous hazards that can affect the quality of recommendation. To avoid these hazards, cryptology approaches and scalar product protocol is used.

Huang *et al.* [19] proposed a framework for alleviating the problem of sparsity inherent in collaborative filtering, Collaborative filtering approach is one of the most successful approaches which consider user neighbor's data and feedback for recommendation but this data is sparse. The association retrieval and spreading activation algorithms are used to find out association between users from their past behaviors and feedback.

2.4.3 Web 2.0 Recommender Systems: Tag based Recommender System

Tag recommendation has become the favored topic of interest since the growth of social tagging site. Comparison between approaches of second generation recommender system

is shown in Table 2.1. Xu *et al.* [40] proposed a tag recommendation algorithm that recommends only high quality tags. High quality tags denote appropriate tags that do not include spam and noise because user annotates tags to resources in free form and some of these tags are noise and spam and these tags are not play major role in the tag recommendation. With the help of tag co-occurrence frequency, relationship between two tags is derived. Tag co-occurrence frequency denotes how many times two tags are attached to the same item. If two tags are attached to the items frequently, tag co-occurrence frequency will be high and tags are closely related otherwise co-occurrence frequency will be low.

Lee [25] proved item based similarity shows better result than user based similarity in unilateral relation. Unilaterally relationship denotes one sided relationship (e. g. in micro blogging sites, users are connected with each other without mutual agreement or if user find other's content relevant or worth, he starts following him).

Singurbjőrnsson *et al.* [36] analyzed how and what kind of tags are annotated to items by users and global co–occurrence frequency metric, a metric is used to measure the relationship among tags.

Golder *et al.* [17] analyzed the structural and dynamic aspect of collaborative tagging system and discuss about the tag frequency, popularity and stability in the proportion of tags in the resources.

Barragáns-Martínez *et al.* [4] proposed a tag based recommender system that improves coverage and diversity of recommendations. They used tags to build user and item tag clouds and compared these clouds for recommendations. User tag cloud is made up of tags that user yet not annotated to item and item tag cloud is made up of tags that are annotated to an item.

Zheng *et al.* [43] explored tag and time is two major factors in social tagging systems (STS). They integrate these factors into collaborative filtering to show better performance in recommendation results. They propose three strategies: weight using tag, weight using time, weight using fusion of tag and time to generate ratings in user-item matrix.

Tso-sutter *et al.* [38] proposed tag aware recommender system that incorporates tagging information into traditional recommender systems. Social tagging Systems are based on three building blocks: users, items, and tags for generating recommendations. These

building blocks have three dimensional correlations with each other. To incorporate tagging information into Collaborative filtering, three dimensional correlations <users, items, tags> is reduced to three two dimensional correlations such as < users, items >, < users, tags >, < items, tags>. In three dimensional correlation users (U1, U2, U3) annotates tags (T1, T2, T3, T4, T5, T6, T7, and T8) to items (I1, I2, I3). Users, items and tags have ternary relationship and this relationship splits into three binary relations as shown in Figure 2.10.

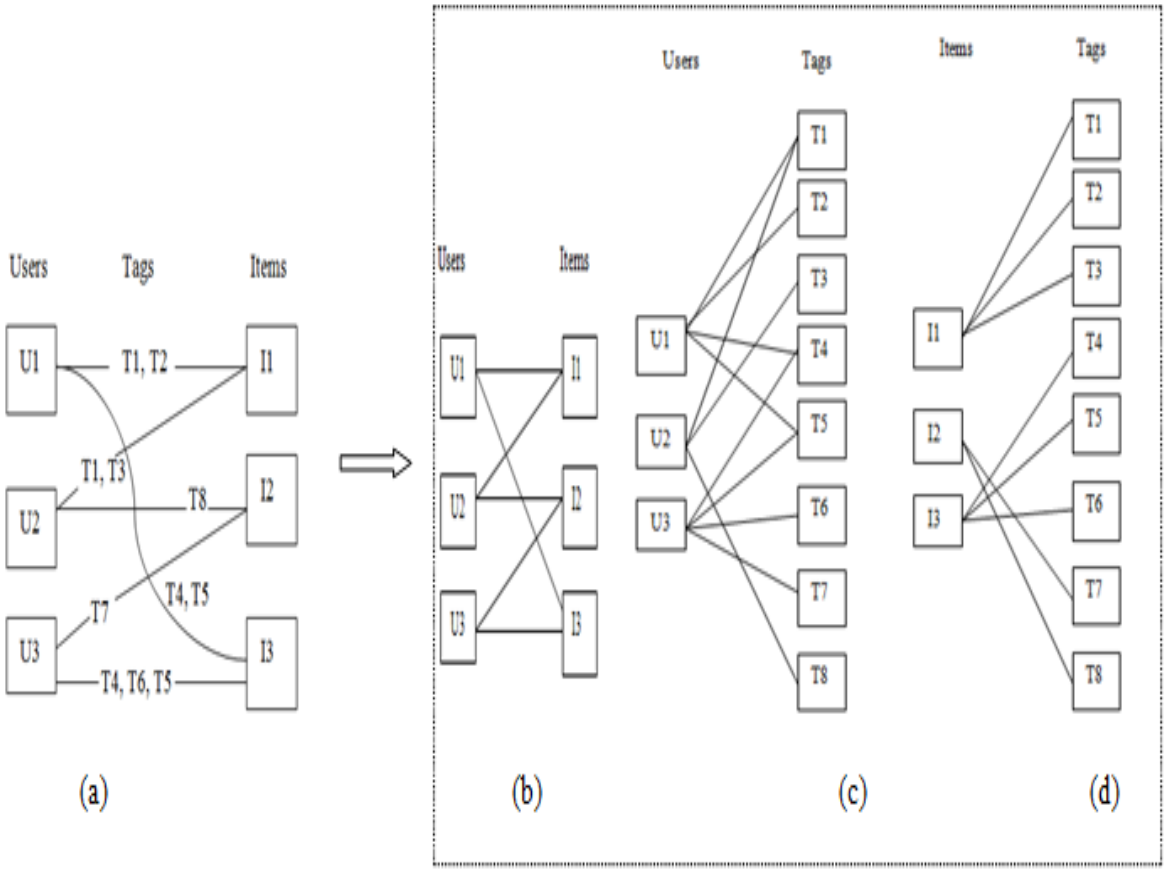


Figure 2.10: (a) Users-Tags-Items relation (ternary), (b) Users-Items relation, (c) Users-Tags relation, (d) Items-Tags relation

Zhao *et al.* [42] presented a novel tag based collaborative filtering approach which uses semantic distance between user generated tags as a measure for finding similarity between users. This approach selects neighbors of user effectively. WordNet is used to find the distance between tags.

Kim *et al.* [23] proposed a hybrid approach for tag aware recommender system that mitigates the limitations of existing approaches in social tagging system. They explored association rule, bigram approach and trust relationship for tag and item recommendations in their proposed framework.

Arazy *et al*. [3] introduced a framework that improves the accuracy of recommender systems and they explain four constructs such as homophily, trust, tie strength and social capital and these construct will impact the advice taking capability of recipients.

Table 2.3: Web 2.0 Recommender Systems

Approach	Advantages	Limitations	Domain
Collaborative Tagging [40]	High quality tags	Loss of information	My web 2.0
Traditional Collaborative Filtering with social network [25]	Overcome the shilling attacks	Semantically rich information is important	Social bookmarking sites (Delicious)
Collective knowledge in tagging [36]	Gracefully handle the expansion of vocabulary	Not implemented online	Online photo sharing site (Flickr)
Collaborative Filterinng, Content Based and Tagging [4]	Tag remains present whether item is present or not	Does not consider time in the weight of tag	TV program (quevo.tv)
Collaborative Tagging with Temporal Information [43]	Information drift with time is considered	Approach considers only one dataset	Social bookmarking site (CiteULike)

Incorporation of Tag with standard CF algorithms (Fusion approach) [38]	Compare performance with or without incorporation of tags in CF algorithms and proves better quality recommendation results with tagging information	Only item recommendation is considered (tag recommendation is not considered)	Music community website (Last.Fm)
Collaborative Filtering with Tagging information [42]	Semantic information of tag is added in recommendation process and generate accurate recommendations	Community wisdom is not considered	Bookmarking system (Dogear with IBM Lotus connections)
Social Tagging System with Hybrid Framework [23]	Implicit Trust information is incorporated	Scalability Problem	Movie (Movielens)
Recommender System Incorporating social Network information [3]	Improvement in Recommendation	Privacy	Theoretical Framework (Social Network Information)

2.4.4 Web 3.0 Recommender Systems

Web 3.0 recommender systems come into existence after the increasing use of mobile devices. Recommender system using internet of things and location based recommender systems are becoming more widespread. In Table 2.4, comparison between approaches of

third generation recommender system is shown. Muñoz-Organero *et al.* [31] they discussed collaborative filtering algorithms using Internet of Things shown in Figure 2.11. This recommender is relying on user- object interaction and space-time interaction patterns. They used time and location for finding the similar user in this IoT based recommender. For finding similarities, they used NFC (near field communities) based social networking information from users who belongs to the same region. In IoT recommender, Sensed user-object pattern and user-location pattern is far better than user rating approach for finding similarities. NFC is one of the major components of RFID tags. It is used for face to face communication. It is used to identify the content embedded in physical object. NFC technology in mobile devices is used to communicate with other NFC device. It allows peer to peer communication between two mobile device and read or write RFID tags. By touching the RFID enabled device, user can read description of that object for example Bar code.

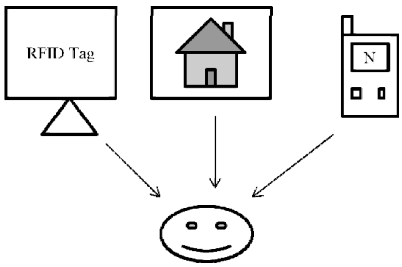


Figure 2.11: RFID Tagged Objects

Levandoski *et al.* [26] proposed a recommender system named Location aware recommender system (LARS). This Recommender system considers location based rating and spatial properties of user. This system enhances scalability and recommendation quality of system. LARS knows item location by using a technique that is called travel penality.

Table 2.4: Web 3.0 Recommender Systems

Approach	Advantages	Limitations	Domain
Collaborative recommender with space and time	Location and time based item similarity is better than Rating	Expensive Implementation	Customized NFC Tagged Item

similarity [31]	based similarity in IoT environment		
Location aware recommender system(LARS) [26]	Better Quality Recommendation	Complex Implementation	Spatial Rating Dataset (FourSquare)

Chapter 3 Problem Formulation

Recommender systems can be used in various real time applications such as movie recommendations, book recommendations, hotel recommendations, clothes recommendations, news recommendations and friend recommendations in social networking sites. Now, Gap analysis is performed to know why tag based recommender system comes into existence.

3.1 Gap Analysis: The Need of Tag based Recommender System

If a person wants to purchase clothes, he will take recommendations from his friend whether this clothe suits to him or not (binary rating) and may be ask him to rate his clothes on the scale of 5 or 10 then his friend give rating and on the basis of that rating, persons make their choice. Sometimes his friend also add keywords (tags) in their reply such as good, bad, good fabric, bad colour contrast etc.). Collaborative filtering recommender systems and social tagging recommender systems fulfil this task for internet users. The main aim of collaborative filtering is to provide series of recommendations to users about items of their interest. But this approach is suffered from many problems: sparsity, shilling attacks and recommendations in diverse environment. To overcome these limitations, social tagging information is also incorporated into system. Tagging information with collaborative filtering recommender systems is used for tag and item recommendations. Many researchers have performed qualitative work in this direction. Precision and recall are two main factors of knowing the performance of any recommender system and these factors are dependent on the error in the outcome. So, these factors can be improved by reducing the error between actual and predicted results.

3.2 Problem Statement

In this thesis, Pareto principle (80-20 rule) in collaborative tagging has been used for good quality recommendations. Only those users are considered who annotate only one tag per item and then 80-20 rule is applied on these users for good quality tag and item

recommendations. This approach has been used for finding similar users who have the same opinions about items or who have tagged common items. Then items will be recommended to active user, which are tagged by users in the same group of active user who is not aware of those items and tags will be recommended also. This approach is combination of collaborative filtering with tagging information named collaborative tagging. The proposed approach has been divided into three parts:

a. Firstly k-Nearest Neighbour approach is used for finding the k most similar users of active user. This is the main step which affects the quality of recommendations. In this thesis, a novel approach is introduced for getting the most promising users in the neighborhood of active user.

b. In the neighbourhood of active user, some users are dominated or some are dominating and in simple terms, some users are more similar to active user than others. The candidate set of dominating users is selected and out of this set, k nearest neighbours are obtained.

c. Recommend the items and tags to active user who did not tag those items yet, by observing the tagging behaviour of k neighbours of active users.

Chapter 4 Proposed Framework for Tag based Recommender System

As user annotates tag to the items in social tagging sites or in movie recommendation sites, recommender system recommends items on the basis of that tagging information. Tags annotated by users are sometimes relevant or sometimes irrelevant. For finding relevant tags annotated by users, "Pareto (80-20) principle that means only 20 percent users annotate relevant tags to the items" is used. This principle is applied on users [32]. One user is dominated by other users on the basis of relevant tags annotated by them.

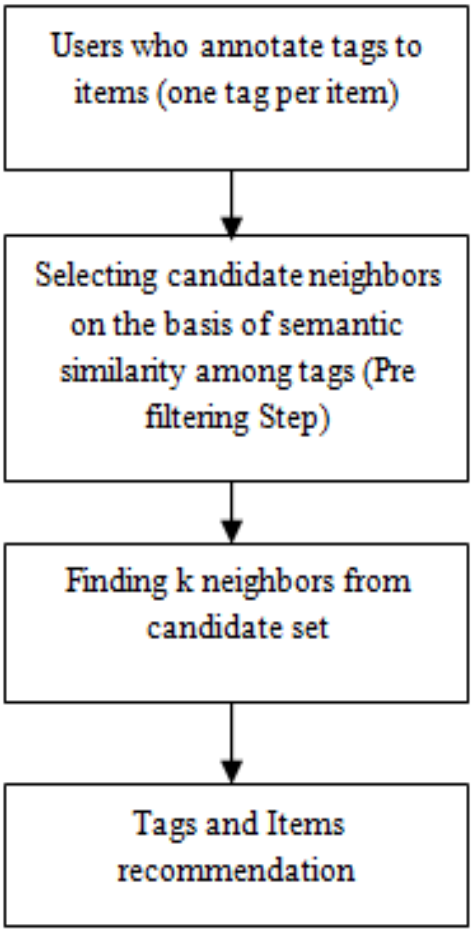


Figure 4.1: Basic concept of proposed method

If two users or more than two users annotate tags on the same movie that means they are interested in that movie and these users make a cluster and in the cluster every user is

neighbour of each other. Before finding the k-Nearest neighbour of active user, a pre filtering step is performed that filters out those users who annotated irrelevant tags. In Figure 4.1, steps of proposed approach are shown. In the Table 2.5, the user's tags of item are displayed.

Table 2.5: User-Tag-Item Matrix

Users	**Items**				
	I1	***I2***	***I3***	***I4***	***I5***
U1	T1	ε	T2	T3	T4
U2	T5	*T6*	T2	T7	ε
U3	ε	T8	ε	T3	T9

The semantic similarity distance between tags has been computed using Word Net [40]. WordNet is a large public dictionary of English words and grouped into nouns, verbs, adverbs and adjectives. The structural collection of these words and their synonyms is called synsets which is basic unit of large repository. A word can have different meanings and every meaning belongs to different synsets. A word with the same meaning belongs to same synsets (e.g. the word sad and sorrow constitutes a synsets with the gloss: an object of unhappiness. Word Net can be expressed in graphical structure in which nodes are synsets and edges are semantic relations. But English words are suffered from many problems such as polysemy (a word can have multiple meanings e.g. crane – a bird or an equipment used for construction), ambiguous words (apple-fruit or iPod). Morphological variations can exist in two tags such as apple vs apples. For mitigating these problems, count the number of common words in the meaning of two words and the more common words they have, the more closely they are related (e.g. mouse and keyboard).

4.1 Preliminaries

Recommender system based on collaborative tagging have *m* users who annotate one and only one tag per movie in the collection of *n* movies where as lacking of tags is denoted by $\mathcal{E}$.

U be the set of users who annotate tags to items.

$$U = \{u \in \mathbb{N} \mid 1 \leq u \leq m\} \quad (1)$$

I be the set of items that are annotated with tags by users.

$$I = \{i \in \mathbb{N} | 1 \leq i \leq n\} \quad (2)$$

T be the set of tags that are used by all users to annotate items. *F* is a folksonomy set, a collection of all the tags that are used in the system and $T \subseteq F$, T is a subset of the collection of tags *F*. *#F* is a cardinality of set *F*.

$$F = \{t \in \mathbb{N} | 1 \leq t \leq l\} \quad (3)$$

$$T = \{t \in \mathbb{N} | 1 \leq t \leq \#F\} \quad (4)$$

P_u be the set of tags that are used by a user *u* to annotate items and that is called personomy. The tagging of user u on item i can be defined as as $t_{u,i} = t$.

$$P_u = \{t \in T \mid t_{u,i} \neq \varepsilon\} \quad (5)$$

In Figure 4.2, relation between folksonomy and personomy is shown. The sets related to users, items and tags are defined in equation (1) to (5):

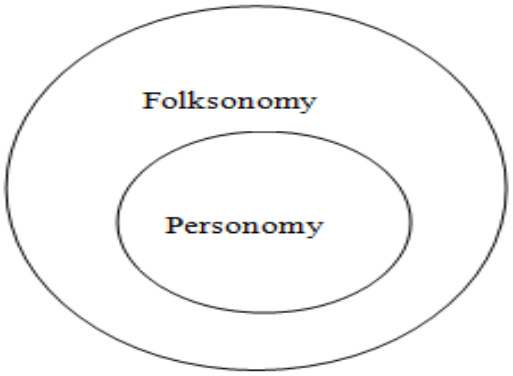


Figure 4.2: Relation between Folksonomy and Personomy

4.2 Selecting Candidate Neighbours (Pre filtering Step)

Candidate neighbours are selected on the basis of semantic similarity and Pareto Principle. In this thesis, dominating users who uses relevant tags have been found. In equation (8), with respect to user *u*, *a* dominate *b* (*a* shows greater similarity in opinion to user *u* than *b*). Dominated users have been discarded because they do not show any improvement in recommendations.

$I_u = \{i \in I \mid t_{u,i} \neq \varepsilon\}$ be the set of items tagged by user *u*. (6)

$d(t_{x,i}, t_{y,i})$ is the semantic distance between two tags used by user *x* and user *y* to the item *i* as described in equation (7).

$$d(t_{x,i}, t_{y,i}) = \begin{cases} |t_{x,i} - t_{y,i}| & t_{y,i} \neq \varepsilon \\ \infty & t_{y,i} = \varepsilon \end{cases} \quad (7)$$

In equation (8), *a* dominate *b* with respect to user *u*.

$$a\ dom_u\ b \Leftrightarrow \forall i \in I_u : d(t_{u,i} - t_{a,i}) \leq d(t_{u,i} - t_{b,i}) \wedge \exists j \in I_u \mid d(t_{u,j} - t_{a,j}) < d(t_{u,j} - t_{b,j}) \quad (8)$$

Suppose dominated users are denoted as u_D, a set of users who are dominated by at least one user with respect to u and u_c are the candidate neighbours of user u. Candidate neighbours have been selected by applying equation (9). From candidate neighbours, k nearest neighbour of active user will be selected.

$$u_c \subset U, u \notin u_c, u_c = U - \{u_D \cup u\}, \forall b \in u_D, \exists a \in u_c \mid a\ dom_u\ b \quad (9)$$

4.3 Finding *k*-Nearest Neighbours

For finding *k* neighbours of active user, similarity between active user and each of the users who belong to candidate set is calculated. Then, *k* more similar users to active user have been taken. Social tagging recommender system provides high quality recommendations in heterogeneous environment. Memory based CF such as *k*NN algorithm is dependent on these similarity measures. Similarity measures are used to

calculate similarity between users. In Memory based collaborative filtering, user to user similarity and item to item similarity find out by using these similarity measures.

The set of items tagged by both user *a* and *b* are shown in equation (10).

$$S_{a,b} = \{i \in I \big| t_{a,i} \neq \varepsilon \wedge t_{b,i} \neq \varepsilon \} \quad (10)$$

In this thesis, mean square difference (Msd) is used to calculate similarity among users on the basis of semantic difference between tags used by two users as shown in equation (11).

$$Msd(a,b) = 1 - \frac{1}{\#S_{a,b}} \sum_{i \in S_{a,b}} (t_{a,i} - t_{b,i})^2 \Leftrightarrow b \in a_c \quad (11)$$

a_c is the set of candidate neighbours of *a*. In equation (11), $t_{a,i} - t_{b,i}$ is semantic difference between two tags calculated with the help of world net dictionary. u_k is the set of *k* neighbours of active user. A parameter *k* is an optimal value that is chosen for quality recommendations.

$$u_k \subseteq \mathrm{u_c} \wedge u_k \leq k \wedge \mathrm{u} \notin u_k \quad (12)$$

$$\forall a \in u_k \, , \forall b \in (u_c - u_k) : msd(u,a) \geq msd(u,b) \quad (13)$$

k parameter depends on many factors: user's opinion , user's tagging behavior, size of the data set. A brute force method is used. Several values of k have been tested for finding an optimum solution.

4.4 Items and Tags Recommendation

The last step of the system is recommendation, a process by which items and tags are recommended to the users on the basis of previous steps. Recommendations to active user are provided by considering *k* neighbor's items and tags used.

R_i be the set of recommended items.

$$R_i = \{i \in I \big| t_{u,i} = \varepsilon \wedge t_{c,i} \neq \varepsilon\} \Leftrightarrow c \in u_k \wedge R_i \subset I \quad (14)$$

R_t be the set of recommended tags where P_c denoted the personomy of user c who belongs to k nearest neighbour set.

$$R_t = \{t \in P_c \mid t_{u,i} = \varepsilon \wedge t_{c,i} \neq \varepsilon\} \Leftrightarrow \forall c \in u_k, \forall i \in R_i \wedge R_t \subset T \quad (15)$$

4.5 Proposed Framework

An overall framework of Tag based Recommender System is shown in Figure 4.3. In step 1, users annotate tags to items and here those items are considered which are tagged with one tag by one user or one user can not annotate more than one tag to the same movie. In step 2, tagged dataset is stored in database.

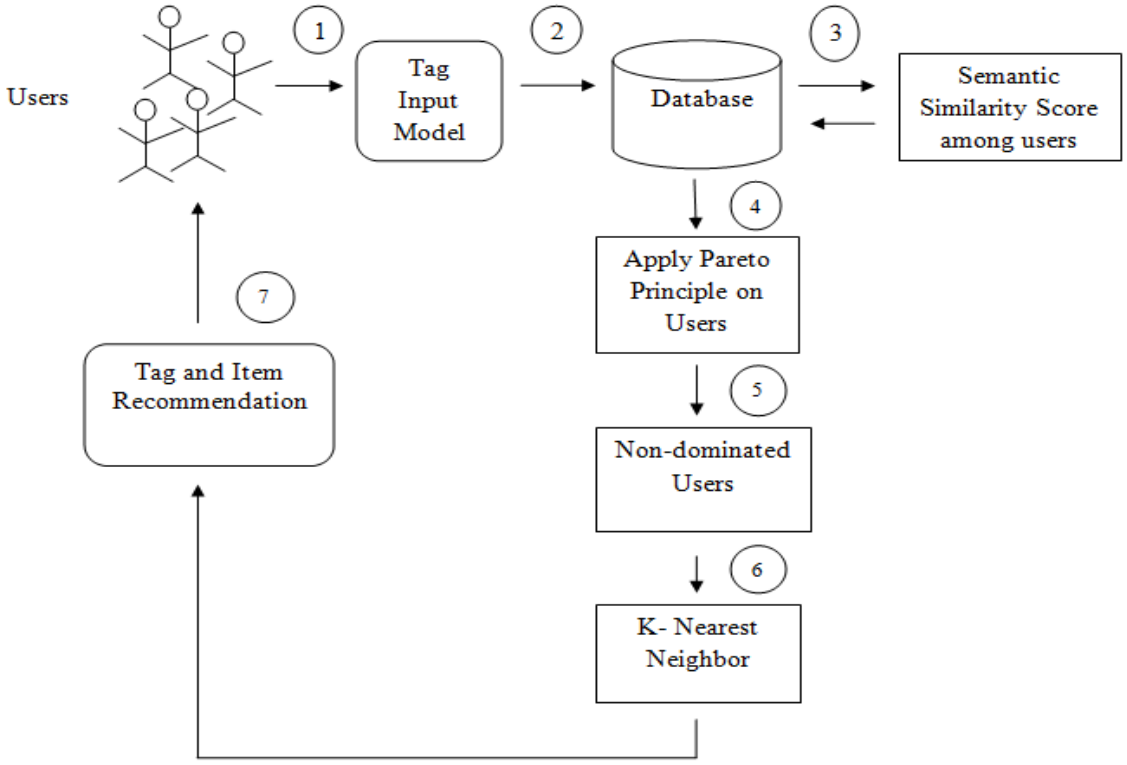


Figure 4.3: Proposed Framework

In step 3, Semantic similarity between tags using word net dictionary has been found and then similarity score among users has been calculated. In step 4, Pareto principle is applied on users. In step 5, with the help of this principle, only those users are considered who annotate only relevant tags and these users are called non-dominated users. In step 6, among all these users, k nearest neighbor has been selected and then in final step, tags and items are recommended to interested users.

Chapter 5 Implementation Details and Experimental Results

In this chapter, firstly experimental design of proposed framework is described and then implementation and results are discussed. Experimental data analysis has been performed on real movie lens dataset. Evaluation metrics are discussed for measuring the quality of recommendations and finally the results are shown.

5.1 Experimental Data

Experiments have been performed on movie lens datasets. Datasets contain total 71567 users, 10681 movies and 95580 tags. In this dataset, 1501 users are unique users who annotate only one tag per movie and 2508 users are those users who tag more than one movie or one movie is tagged by more than one tag and total 4009 users have tagged movies. Total 7601 movies that are tagged and from these movies, 1429 movies are uniquely tagged by users, 6172 movies are tagged by more than one tag or more than one user and 3080 movies are not tagged by anyone. 16529 distinct tags are used. Dataset is divided into several tables: users, items, tags, users-items, users-tags and items-tags. Each user's items are divided into 80 and 20 ratio on the basis of time recency feature of tagging. Recent 20 % items are used for test the system and 80 % items are used for train the system as shown in Figure 5.1. Recommender system performance has been evaluated by hit ratio of number of recommended items into the test set of user.

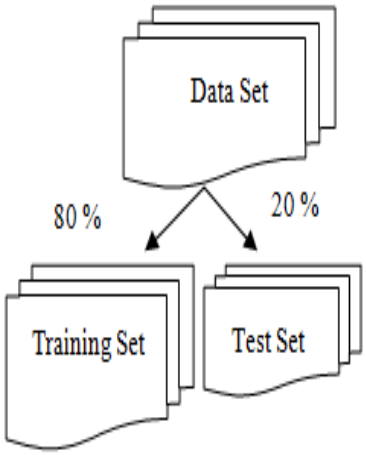


Figure 5.1: Training Set and Test Set

5.2 Test Plan

The ultimate goal of recommender system is to provide accurate and precise recommendations to users. For knowing the accuracy of the system, mean absolute error (*MAE*) metric are used. The impact of neighborhood size on the accuracy of recommender system is checked. An optimal value of k at which error decreases and accuracy increases is chosen. For good quality recommendations, precision and recall metrics are used.

5.2.1 Evaluation Metrics

Evaluation is an important part of any recommender system because without evaluation it cannot be inferred whether results of the system are right or not. Comparison between two systems can be possible only after performing the evaluation. With the help of evaluation, improvement in the system can be done. Recommendation Set Quality Measures evaluate the quality of recommendations. User confidence is based on the accuracy and quality of the recommendations offered by the system. Recommendation quality measure informs whether system is recommended relevant content or not. Precision, Recall, and Mean absolute Error is used to measure the quality of proposed system.

Precision (*P*) is defined as the ratio between the total number of relevant recommended resources to the total number of resources recommended.

$$P = \frac{\text{Total number of relevant recommended resources}}{\text{Total number of resources recommended}}$$

Recall (*R*) is defined as the ratio between the total number of relevant recommended resources to the total number of relevant resources.

$$R = \frac{\text{Total number of relevant recommended resources}}{\text{Total number of relevant resources}}$$

Mean Absolute Error (*MAE*) is used to measure the accuracy of system. Suppose D_a = { i ∈ I : $p_{a,i} \neq \mathcal{E} \wedge r_{u,i} \neq \mathcal{E}$}, a set of items rated by users having prediction and rating value is not equal to null. *A* is the set of users. Error in the prediction is calculated by

taking the difference between actual ratings and predicted ratings, $|p_{a,i} - r_{a,i}|$ informs error in the system.

$$MAE = \frac{1}{\#A}\sum_{a\in A}\left(\frac{1}{\#D_a}\sum_{i\in D_a}|p_{a,i} - r_{a,i}|\right)$$

5.3 Implementation

The Proposed Approach has been implemented on movielens dataset. This data set has so many csv files. One csv file that contains user id, movie id, tags and timestamp values is picked. Then this file containing 95580 rows is stored into Mysql database as shown in Figure 5.2.

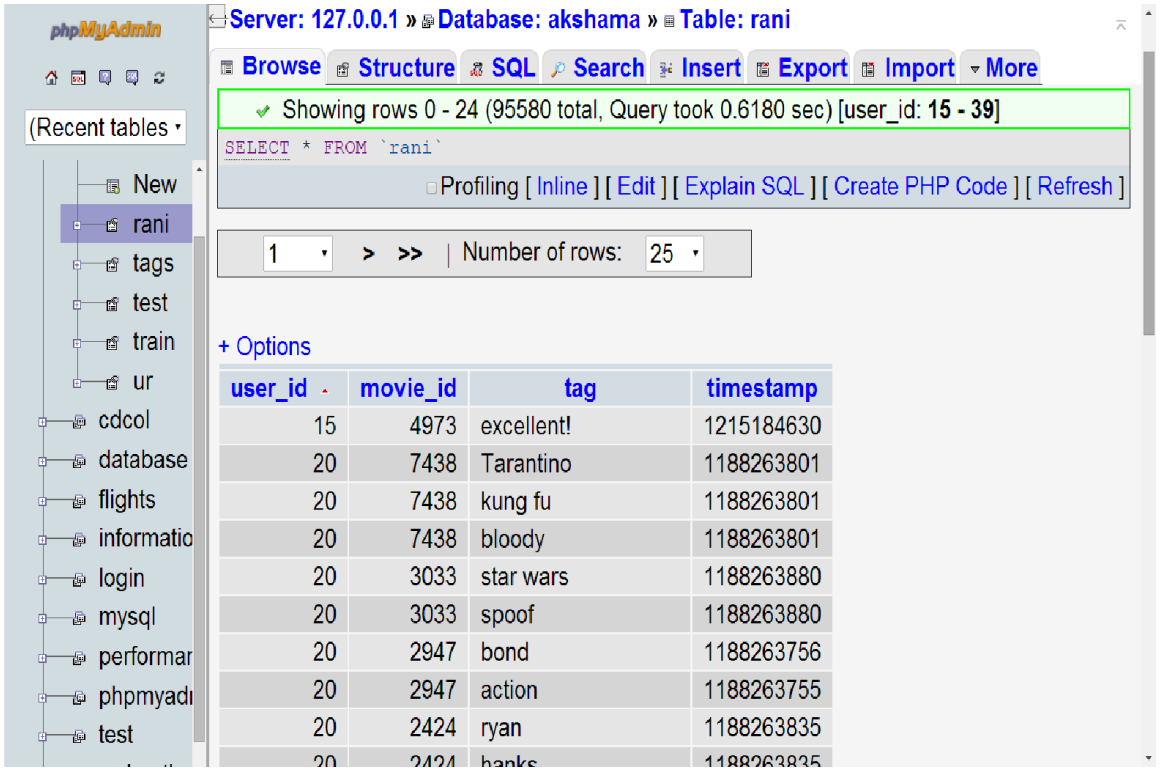


Figure 5.2: User-tag-movie database file

From this dataset, those users who annotated movie with only one tag are selected. As dataset is very large, it is very difficult to work with this. So out of this, 1000 rows are selected for checking whether our approach is giving right recommendations or not as shown in Figure 5.3.

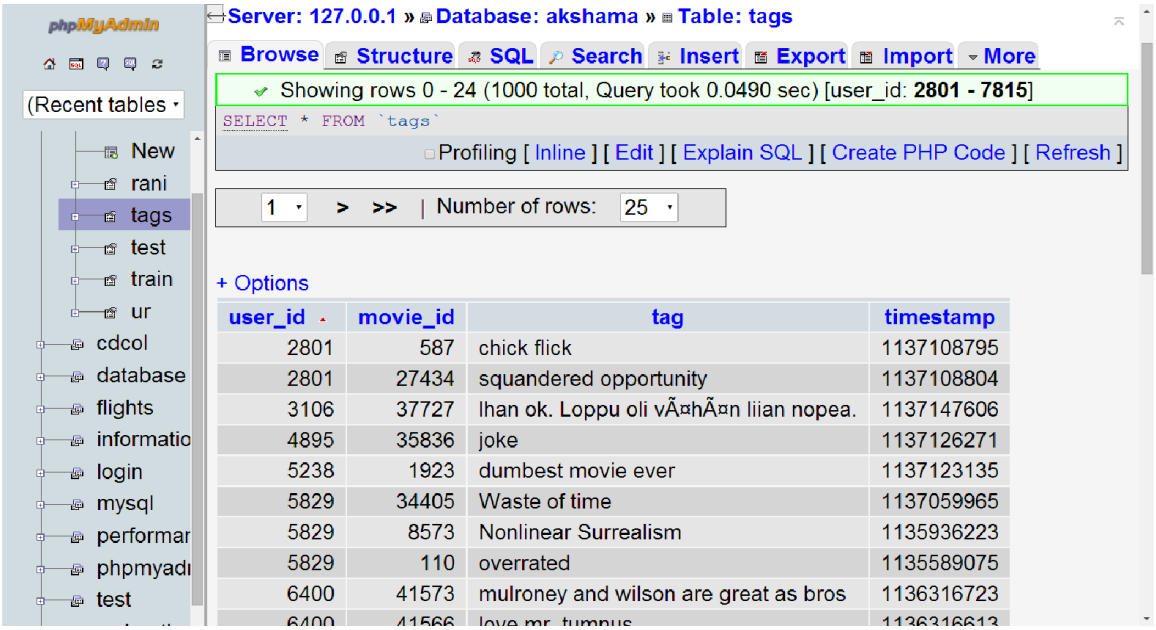


Figure 5.3: Dataset of 1000 users

Dataset has been divided into training set and test set in the ratio of 80 % and 20 %. 800 rows and 200 rows are picked for training set and test set respectively on the basis of time of annotation of tags to movies as shown in Figure 5.4 and 5.5.

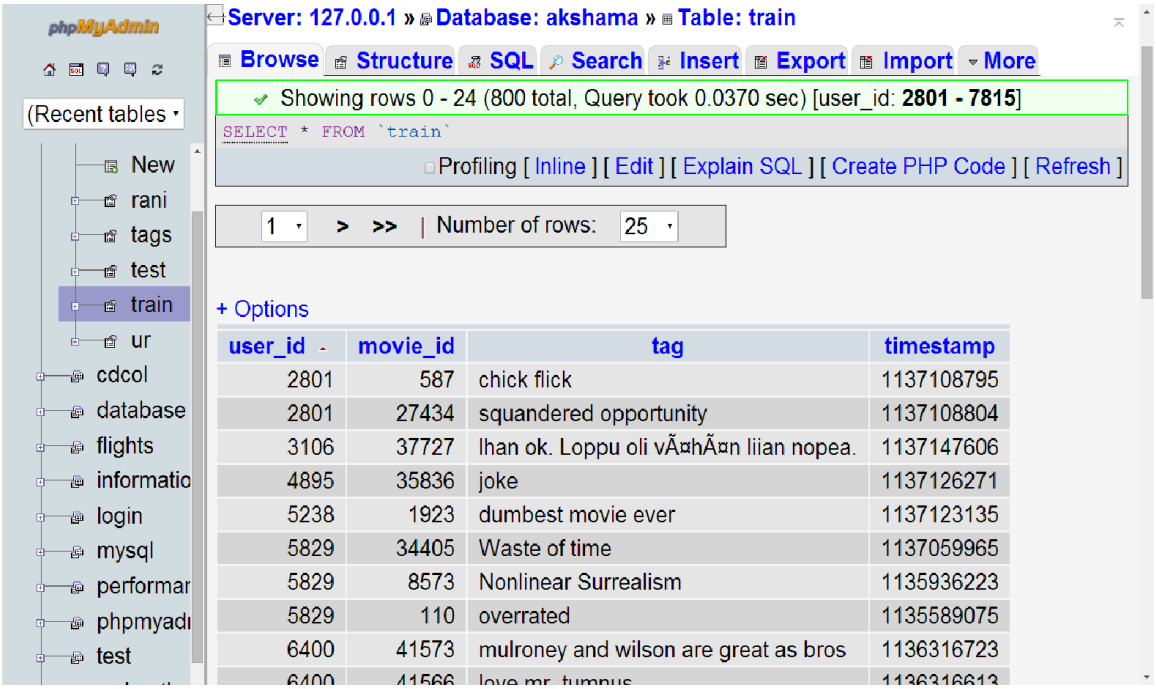


Figure 5.4: Training set used for movie recommender system

Only those users are taken in test set who annotate tag recently to item on the basis of timestamp and earlier users are taken in training set, who recommend movies to the user of test set. Then, whether recommended movies and rated movies by the users of the test set is matching or not checked.

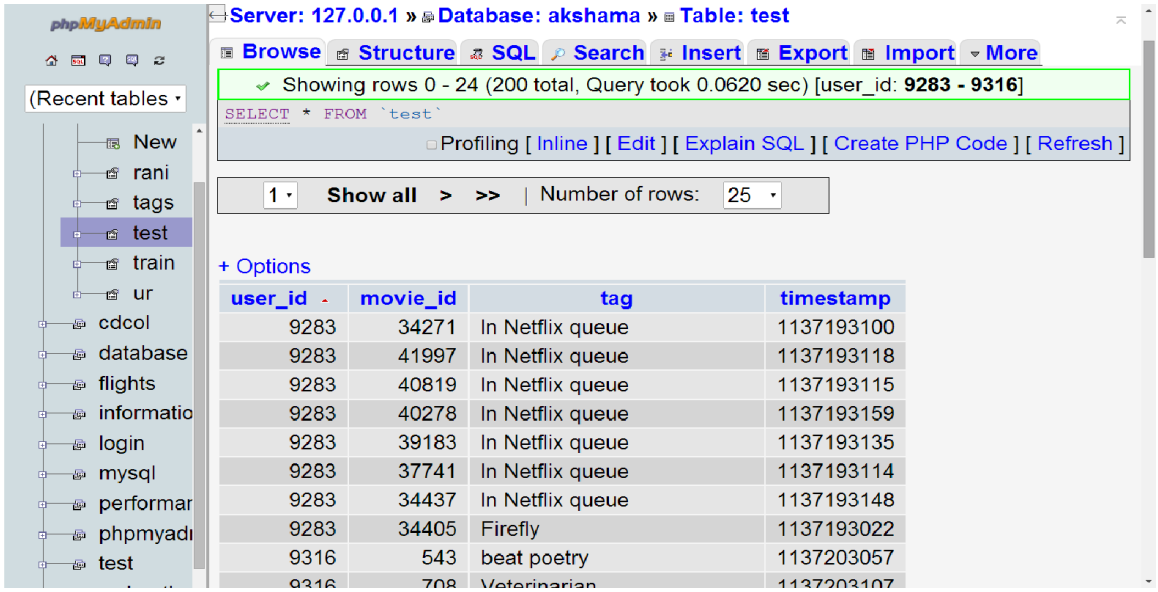


Figure 5.5: Test set used for movie recommender system

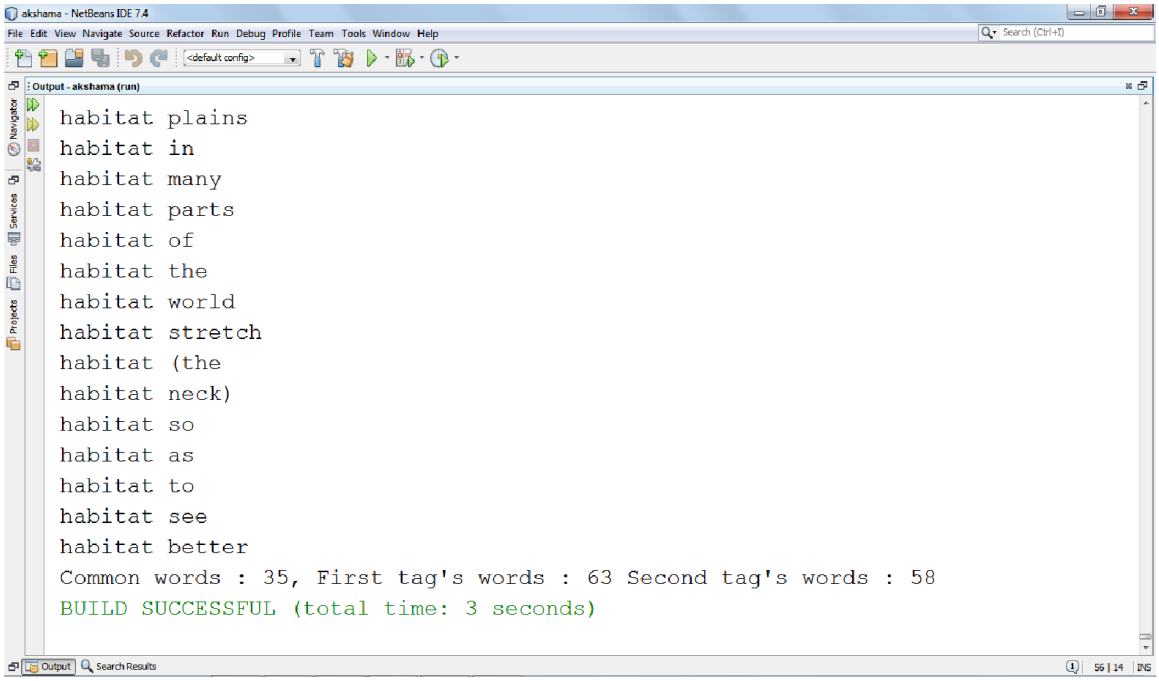


Figure 5.6: Semantic Similarity score between two words

Semantic distance or score (inversely proportional to distance) is calculated among all users. Then, Pareto principle is applied on them. In Figure 5.6, similarity score between two words is calculated using WorldNet dictionary. Here, two words are taken such as: a bird and crane. These words having so many synonyms and all synonyms have been checked for finding similarity score.

In Figure 5.7, score between different users has been found using word net dictionary. Score and distance has an inverse relationship. If score is more between two users that means distance between two users is less or we can say that two users are more similar. We are calculating score on the basis of common items between two users. If two users annotate an item with the same tag or same meaning tag then score of their similarity will be high. Finally, users are taken on the basis of dominance principle as explained earlier.

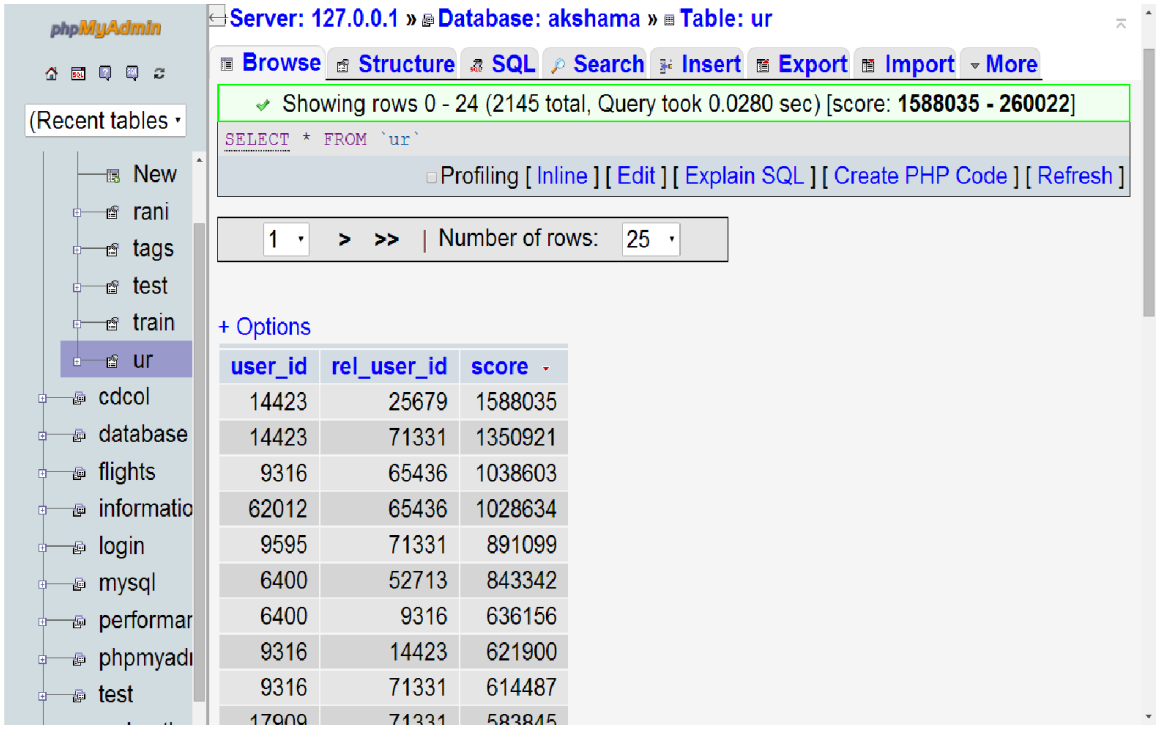


Figure 5.7: Similarity Score between users

In Figure 5.8, active user having id 9316 has so many common users. Common users are those users who have good score with active user. User having id 9316 has been picked from test set and recommendation has been checked with the tagging behavior of active user.

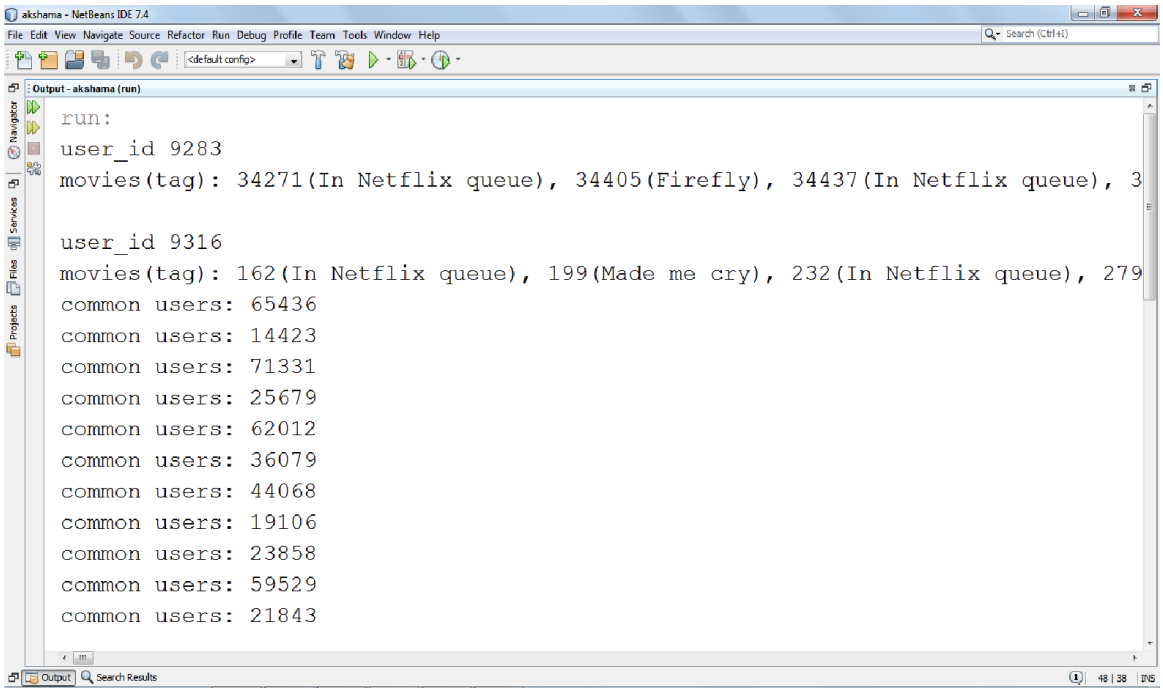


Figure 5.8: User 9316's similar users

In Figure 5.9, firstly tagged movies of active user are shown. After that movies and tags have been recommended by common users or similar users, to active user are also shown.

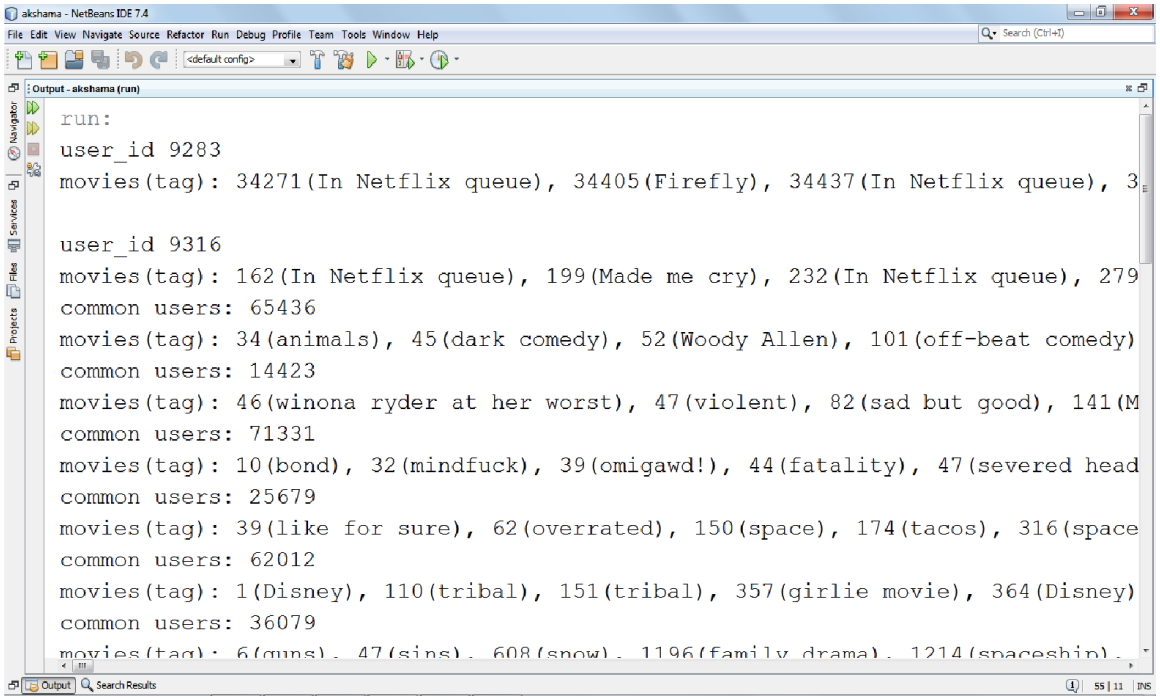


Figure 5.9: Recommended to and tagged movies by user 9316

5.4 Test Results

Results of experiment performed on movie lens dataset are shown in Figure 5.10 to 5.13. Movie distribution over users is shown in Figure 5.10, large number of users preferred only some movies and small number of users preferred large number of movies.

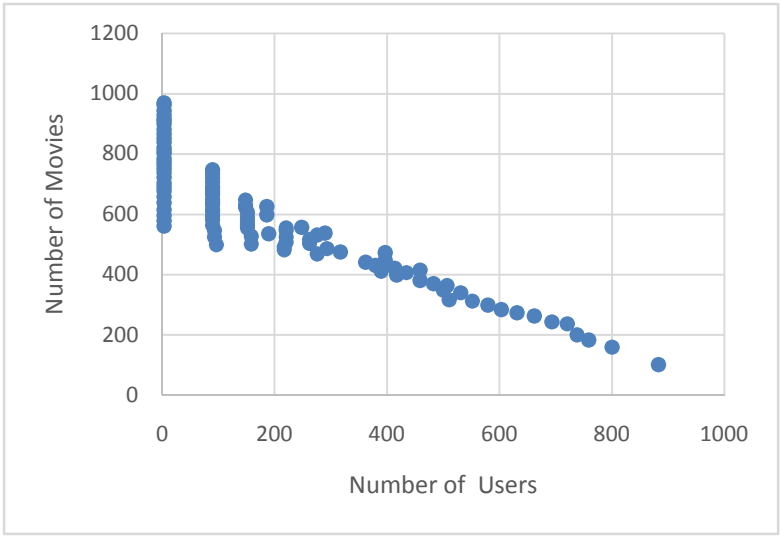


Figure 5.10: Movie Distribution over users

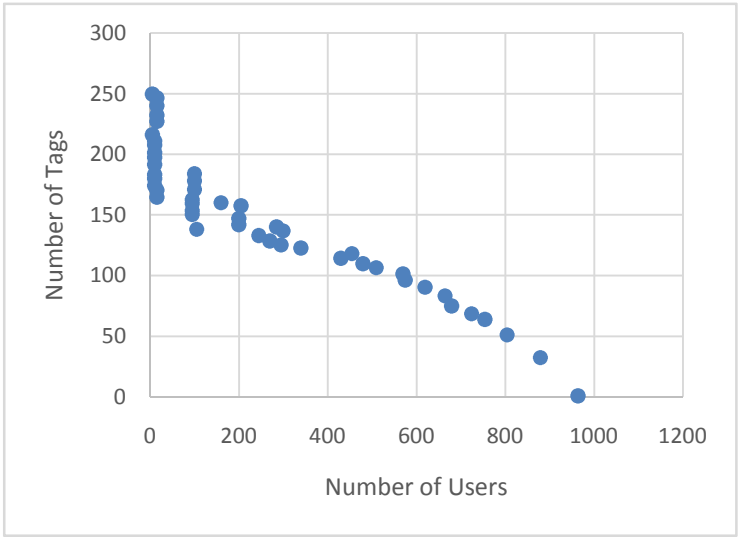


Figure 5.11: Tag Distribution over users

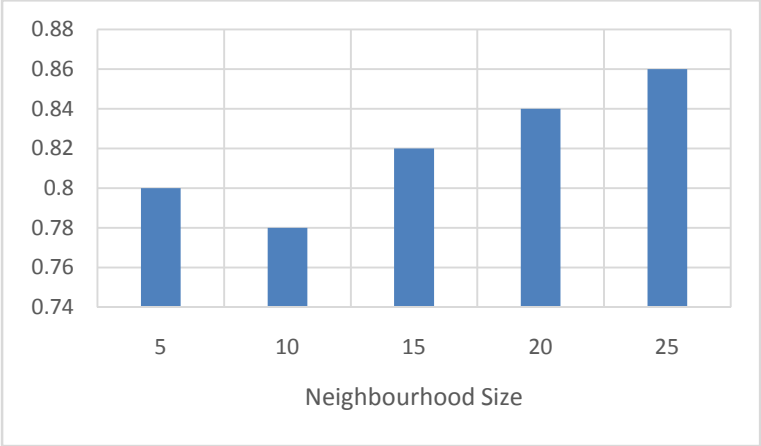


Figure 5.12: *MAE* comparison for different neighbourhood size

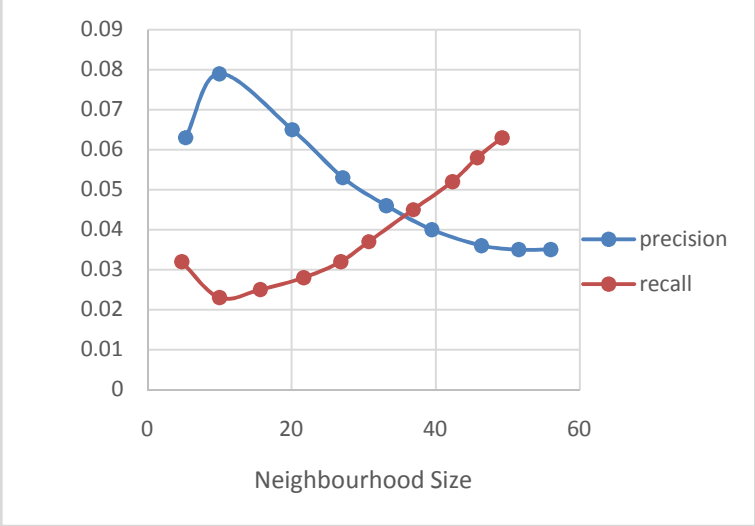


Figure 5.13: Precision and Recall

In Figure 5.11, Tag distribution over user is shown. Tagging dataset is very less as compared to movie's rating data set because users find annotating tags is an annoying task. One fourth users annotate 100 or more than 100 tags and remaining users annotate 10 or less than 10 tags only. We are comparing the influence of different k values on recommendation results. In Figure 5.12, *MAE* value is used to measure the accuracy of system. We choose different neighbourhood size for checking the accuracy of recommendations. Number of users in neighbourhood is represented with k. At k =5,

MAE value is 0.8 and at k =10, *MAE* value decreases and after that it increases sharply. If *MAE* value decreases, system's accuracy increases. We choose k value 10. In Figure 5.13, precision and recall quality metrics are represented. Our system shows good precision. At k=5 system shows 0.06 precision, at k=10 system shows 0.08 precision and after that precision decreases as size of neighbourhood increases and after some time it becomes constant. Recall and precision has an inverse relation. Recall increases as neighbourhood size increases and after some time it also shows no change in the values.

In this chapter, implementation and Test results of proposed framework is shown.

Chapter 6 Conclusion and Future Scope

In this chapter, a brief overview of thesis work and how this work can be extended further is described.

6.1 Conclusion

A framework for tag based recommender system using Pareto Principle is proposed in this thesis. Pareto (80-20 rule) principle enables generation of good quality tag and item recommendations. A pre filtering step using Pareto principle is added to traditional collaborative tagging system for eliminating less representative users from k-Nearest Neighbour selection process. Semantic difference between tags using Word Net dictionary is calculated and this difference is used further to find semantic similarity between tags. To conclude Pareto principle retains most promising users. Experiments have been performed on movielens dataset and experimental result show good quality tag and item recommendations.

6.2 Future Scope

In current work, those users are taken who tagged a movie only once. In future, those users can be taken who have annotated more than one tag to items and the same framework can be applied on them for analyzing improvement in recommendation results.

a. The proposed Framework can be applied on other social tagging sites such as Delicious, CiteULike, etc.
b. Tag based recommender system can be deployed on hadoop for Scalable recommendations.

References

[1] Adomavicius, G. and Tuzhilin, A. (2005), "Towards 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, pp. 734-749.

[2] Antonopoulus, N. and Salter, J. (2006), "Cinema screen recommender agent: combining collaborative and content-based filtering", *IEEE Intelligent Systems*, vol. 21, pp.35–41.

[3] Arazy, O., Kumar, N. and Shapira, B. (2009), "Improving Social Recommender Systems", *IT Professional*, vol. 11, pp. 38-44.

[4] Barragáns-Martínez, A., Rey-López, M., Costa-Montenegro, E. and Mikic-Fonte, F. et al (2010), "Exploiting Social Tagging in a Web 2.0 Recommender System", *IEEE Internet Computing*, vol. 14, pp. 23-30.

[5] Blanco-Fernández, Y., Pazos-Arias, J.J., Gil-Solla, A., Ramos-Cabrer, M. and López-Nores, M. (2008), "Providing Entertainment by Content-based Filtering and Semantic Reasoning in Intelligent Recommender Systems", *IEEE Transactions on Consumer Electronics*, vol. 54, pp. 727-735.

[6] Bobadilla, J., Ortega, F., Hernando, A. and Gutiérrez, A. (2013), "Recommender systems survey", *Knowledge-Based Systems*, vol. 46, pp. 109-132.

[7] Bobadilla, J., Ortega, F. and Hernando, A. and. Glez-de-Rivera G. (2013), "A similarity metric designed to speed up, using hardware, the recommender systems k-nearest neighbors algorithm," vol.51, pp.27-34.

[8] Bobadilla, J., Ortega, F. and Hernando, A. (2012), "A collaborative filtering similarity measure based on singularities", *Information Processing and Management*, vol.48, pp.204-217.

[9] Burke, R. (2002), "Hybrid recommender systems: survey and experiments", *User Modeling and User-Adapted Interaction*, vol. 12, pp. 331-370.

[10] Carrer-Neto, W., Hernandez-Alcaraz, M., L., Valencia-Garcia, R. and Garcia-Sanchez, F. (2012), "Social knowledge based recommender systems. Application

to the movie domain", *Expert Systems with Applications*, vol. 39, pp. 10990-11000.

[11] Chen, C., Zeng, J., Zheng, X., Chen, D. (2013), "Recommender systems Based on Social Trust Relationships", *IEEE 10th International Conference on e-Business Engineering*, pp. 32-37.

[12] Chung, C., Hsu, P. and Huang, S. (2013), "βP: A novel approach to filter out malicious rating profiles from recommender systems", *Decision Support Systems*, vol. 55, pp. 314-325.

[13] Fong, A., Zhou, B., Hui, S., Hong, G. and Do, A. (2011), "Web Content Recommender System based on Consumer Behavior Modeling", *IEEE Transactions on Consumer Electronics*, vol. 57, pp. 962-969.

[14] Gedikli, F., Jannach, D. and Ge, M. (2014), "How should I explain? A comparison of different explanation types for recommender systems",*International Journal Human-Computer Studies*, Vol. 72, pp. 367-382.

[15] Ghazanfara, M. and Prügel-Bennett, A. (2014), "Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems",*Expert Systems with Applications*, vol. 41, pp. 3261-3275.

[16] Goldberg, D., Nichols, D., Oki, B.M. and Terry, D. (1992), "Using Collaborative Filtering to weave an information Tapestry", *Communications of the ACM*, pp.61-70.

[17] Golder, S. and Huberman, B. (2006), "Usage patterns of collaborative tagging systems", *Journal of Information Science*, vol. 32 pp. 198–208.

[18] Herlocker, J., Konstan, J., Brochers, A., and Riedl, J. (1999), "An algorithmic framework for performing collaborative filtering", *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 230-237.

[19] Hung, Z., Chen, H. and Zeng, D. (2004), "Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering", *ACM Transactions on Information Systems (TOIS)*, vol. 22, pp. 116-142.

[20] Hunter, P. (2013), "Journey to the centre of big data", *IET Journals & Magazines*, vol. 8, pp. 56-59.

[21] Ji, C., Li, Y., Qiu, W., Awada, U. and Li, K. (2012), "Big Data Processing in Cloud Computing Environments", *12th International Symposium on Pervasive Systems, Algorithms and Networks (ISPAN),* pp.17-23.

[22] Kardan, A. and Hooman, M. (2013), "Targeted Advertisement in Social Networks using Recommender Systems", *7th International Conference on e-Commerce in Developing Countries: With Focus on e-security (ECDC),* pp. 1-13.

[23] Kim, H. and Kim, H. (2014), "A framework for tag-aware recommender systems", *Expert Systems with Applications*, vol. 41, pp. 4000-4009.

[24] Koren, Y., Bell, R. and Volinsky, C. (2009), "Matrix Factorization Techniques For Recommender Systems", *IEEE Computer Society*, vol. 42, pp. 30-37.

[25] Lee, D. (2010), "How to measure the information similarity in unilateral relations: the case study of *Delicious", In 10 Proceedings of the International Workshop on Modeling Social Media, ACM*.

[26] Levandoski, J., Sarwat, M., Eldawy, A. and Mokbel, M. (2012), "LARS: A Location-Aware Recommender System", 2012 IEEE 28th International Conference on Data Engineering (ICDE), pp. 450-461.

[27] Li, Y., Wu, C. and Lai, C. (2013), "A social recommender mechanism for e-commerce: combining similarity, trust, and relationship", *Decision Support Systems*, Vol. 55, pp. 740-752.

[28] Lika, B., Kolomvatsos K. and Hadjiefthymiades, S. (2014), "Facing the cold start problem in recommender system", *Expert Systems with Applications*, Vol. 41, pp. 2065-2073.

[29] Ma, H., Zhou, T.C., Lyu, M.R. and King, I. (2011), "Improving recommender systems by incorporating social contextual information", ACM Transactions on Information Systems, vol. 29, pp. 1-23.

[30] Melamed, D., Shapira, B. and Elovici, Y. (2007), "Marcol: A Market-Based Recommender System", *IEEE Intelligent Systems*, Vol. 22, pp. 74-78.

[31] Muñoz-Organero, M., Ramírez-González, G.A., Muñoz-Merino, P.J. and Kloos, C.D. (2010), "A Collaborative Recommender System Based on Space-Time Similarities", IEEE Pervasive Computing, Vol. 9, pp. 81-87.

[32] Ortega, F., Sánchez, J., Bobadilla, J. and Gutiérrez, A. (2013), "Improving collaborative filtering-based recommender systems results using Pareto dominance", *Information Sciences*, vol.239, pp.50-61.

[33] Pazaani, M. (1999), "A Framework for collaborative, content based and Demographic Filtering", *Artificial Intelligent Review*, vol. 13, pp. 393-408.

[34] Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. (2001), "Item-based collaborative filtering recommendation algorithms", *Proceedings of the 10th international conference on World Wide Web*, pp. 285-295.

[35] Schafer, J., Konstan, J. and Riedl, J. (1999), "Recommender Systems in E-Commerce", *In proceeding of ACM conference on Electronic commerce in Computing Systems,* pp. 158-166.

[36] Sigurbjörnsson, B. and Zwol, R.V. (2008), "Flickr tag recommendation based on collective knowledge", *In Proceedings of the 17th international conference on worldwide web, ACM,* pp. 327–336.

[37] Tan, S., Bu, J., Chen, C. and He, X. (2011), "Using rich social media information for music recommendations via hypergraph model", *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 7, pp. 1-22.

[38] Tso-sutter, K., Marinho, L. And Schmidt-Thieme, L. (2008), "Tag-aware recommender systems by fusion of collaborative filtering algorithms", *In Proceedings of the 2008 ACM symposium on Applied computing*, pp. 1995-1999.

[39] Walunj, S. and Sadafale, K. (2013), "An Online Recommendation System for E-commerce Based on Apache Mahout Framework,'' *ACM*.

[40] Xu, Z., Fu, Y., Mao, J. and Su, D. (2006), "Towards the semantic web: collaborative tag suggestions", *In Collaborative web tagging workshop Scotland*.

[41] Zhan, J., Hsieh, C., Wang, I., Hsu, T., Liau, C. and Wang, D. (2010), "Privacy-Preserving Collaborative Recommender Systems", *IEEE Transactions on Systems, Man, and Cybernetcis, Part C: Applications and Reviews*, vol. 40, pp. 472-476.

[42] Zhao, S., Du, N., Nauerz, A., Zhang, X., Yuan, Q. and Fu, R. (2008), "Improved recommendation based on collaborative tagging behaviors", *In Proceedings of the 13th international conference on Intelligent user interfaces*, pp. 413-416.

[43] Zheng, N. and Li, Q. (2011), "A recommender system based on tag and time information for social tagging systems", *Expert Systems with applications*, vol. 38, pp. 4575-4587.

[44] Movie Data Set [online] http://grouplens.org/datasets/movielens/

List of Publications

Accepted

[1] Rani, A. and Bawa, S. "Improvement in Collaborative Tagging Systems using Pareto principle", *In Proceedings of International Conference on Emerging Research in Computing, Information, Communication and Applications (ERCICA-14)*, August 1-2, 2014, ***Elsevier***.

Communicated

[1] Rani, A. and Bawa, S. "A Social Recommender System for Event Planning", *In Journal of Social Network Analysis and Mining*, 2014, ***Springer***.

[2] Rani, A. and Bawa, S. "A Survey on the Generation of Recommender Systems", *Seventh International Conference on Contemporary Computing (IC3-2014),* Noida, India, August 7-9, 2014, ***IEEE***.