Research Journal of Applied Sciences, Engineering and Technology 5(16): 4168-4182, 2013
ISSN: 2040-7459; e-ISSN: 2040-7467
© Maxwell Scientific Organization, 2013
Submitted: August 16, 2012 Accepted: December 01, 2012 Published: April 30, 2013

Collaborative Filtering Recommender Systems

[1]Mehrbakhsh Nilashi, [1, 2]Karamollah Bagherifard, [1]Othman Ibrahim, [3]Hamid Alizadeh, [4]Lasisi Ayodele Nojeem and [1]Nazanin Roozegar
[1]Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, Malaysia
[2]Department of Computer Engineering, Islamic Azad University, Yasooj Branch, Yasooj, Iran
[3]Department of Executive Management, E-compus Branch, Islamic Azad University, Tehran, Iran
[4]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn, Malaysia

Abstract: Recommender Systems are software tools and techniques for suggesting items to users by considering their preferences in an automated fashion. The suggestions provided are aimed at support users in various decision-making processes. Technically, recommender system has their origins in different fields such as Information Retrieval (IR), text classification, machine learning and Decision Support Systems (DSS). Recommender systems are used to address the Information Overload (IO) problem by recommending potentially interesting or useful items to users. They have proven to be worthy tools for online users to deal with the IO and have become one of the most popular and powerful tools in E-commerce. Many existing recommender systems rely on the Collaborative Filtering (CF) and have been extensively used in E-commerce .They have proven to be very effective with powerful techniques in many famous E-commerce companies. This study presents an overview of the field of recommender systems with current generation of recommendation methods and examines comprehensively CF systems with its algorithms.

Keywords: Collaborative filtering, item-based, prediction, rating, recommender system, user-based, recommendation

INTRODUCTION

The history of recommender systems dates back to the year 1979 with relation to cognitive science (Rich, 1979). Recommender systems gained prominence among other application areas such as approximation theory (Powell, 1981), information retrieval (Salton, 1989), forecasting theories (Armstrong, 2001), management science (Murthi and Sarkar, 2003) and consumer choice modeling in marketing (Lilien *et al*., 2003). In the mid-1990s, recommender systems became active in the research domain when the focus was shifted to recommendation problems by researchers that explicitly rely on user rating structure and also emerged as an independent research area (Anand and Mobasher, 2005; McSherry and Mironov, 2009; Goldberg *et al*., 1992)

In coping with information overload problems, recommender systems have proved in recent years to be a force to reckon with as a valuable means in tackling such problems. In addressing this phenomenon, recommender system guides user towards new, unknown experienced items that may be relevant to the user's current task. In the aftermath of user making a request, articulated depending on the recommendation approach by the user's context and need, there exist a generation of recommendations aided by the use of various types of knowledge and data about the users, the available items and previous transactions stored in customized databases.

Recommender system research, aside from its theoretical contribution, is conducted with a strong emphasis on practice and commercial applications generally aimed at practically improving commercial recommender systems. The recommender system research therefore, involves practical aspects that apply to the implementation of these systems which are relevant in different stages of the life cycle of the system namely, the design of the system, its implementation, maintenance and enhancement during operation.

In the design stage, there are factors which build the aspects that might affect the choice of the algorithm. The application domain, which is the first factor in consideration, has a major effect on the algorithmic approach. In the study of Lopez and colleagues, they provided taxonomy of recommender systems and classification of existing recommender system applications into specific application domains (López and de la Rosa, 2003).

Corresponding Author: Mehrbakhsh Nilashi, Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, Malaysia

This study presents an overview of the field of recommender systems with current generation of recommendation methods and examines comprehensively CF with its algorithms in prediction and recommendation process. It is hoped that this research will accentuate the importance of recommender systems and provide researchers with insight and future direction on recommender systems.

DATA AND KNOWLEDGE SOURCES

Recommendation techniques can either be knowledge poor or knowledge dependent. While knowledge poor is the use of simple and basic data such as user ratings/evaluations for items, knowledge dependent is using ontological descriptions of the users or the items, or constraints, or social relations and activities of the users. Thus, as a general classification, three kinds of elements namely items, users and transactions construct the data used by recommender systems.

Items T = {t_1, t_2,…,t_n}: Items are the products in the recommender system for suggesting to the user .All domain relevant items are stored in set *T*. The possibly unique item identifiers can either be proprietary product codes from an ecommerce site such as Amazon.com's ASINs or globally accepted codes such as ISBNs, ISSNs, etc. Items may be characterized by their complexity and their value or utility. The value of an item is considered positive if the item is of any usefulness for the user and the value is negative if the item is inappropriate due to the user making a wrong decision in selection process. According to their core technology, recommender systems can use a range of properties and features of the items. As an example, in a movie recommender system, the genre (such as comedy, thriller, etc.), as well as the director and actors can be used to describe a movie and to learn how the utility of an item depends on its features. Items can be represented using various information and representation approaches, e.g., in a minimalist way as a single id code, or in a richer form, as a set of attributes, but even as a concept in an ontological representation of the domain. The complexities and values of items can either be low or large. Examples of low are news, Web pages, books, CDs, movies and those of larger complexity and values are digital cameras, mobile phones, PCs, etc. Insurance policies, financial investments, travels and jobs are considered the most complex items (Montaner *et al*., 2003).

Users U = {u_1,u_2,…,u_n}: Elements of U comprises of all the users that have browsed items or contributed to the item ratings in the sites. As a way to personalize the recommendations and human computer interaction, recommender systems exploit a range of information, which can be structured in various ways, about the users who may have diverse goals and characteristics and the selection of what information to model depends on the recommendation technique. In CF for instance, users are modeled as a simple list containing the ratings provided by the user for some items. Socio-demographic attributes such as age, gender, profession and education within the demographic recommender system are used. Also, the behavioral pattern data, for example, site browsing patterns in a Web-based recommender system (Taghipour *et al*., 2007) or travel search patterns in a travel recommender system (Mahmood and Ricci, 2009) can be used to describe the users. The relations between users such as the trust level of these relations may include user data and all the above examples of information are used as a model by recommender system to recommend items to user that were preferred by similar or trusted users.

Transactions: Transaction, a recorded interaction between a user and the recommender system, are log-like data consisting of stored information generated during human-computer interaction used for the recommendation generation algorithm by the system. A reference to an item selected by the user and a description of the context (e.g., the user goal/query) for that particular recommendation is an instance of a transaction log and an explicit feedback, such as the rating for the selected item, during a transaction is provided by the user. The rating is in fact the most popular form of transaction data collected by a recommender system which may be explicitly or implicitly. In the explicit rating, the user is asked to provide about an item on a rating scale.

There are varieties of forms that ratings could adopt (Schafer *et al*., 2007a) as follows:

- Numerical ratings represented by a number from either a discrete or a continuous rating scale and in most cases with a limited range. Discrete rating scale are ratings on a scale from zero to five stars or Likert response scaled commonly used in questionnaires while continuous rating scale could be a slider set by a user and translated to a real value.
- Binary rating scale allowing users to assign items to two different classes (like/dislike) and a good example is YouTube that allows users to rate movies with either thump up or down.
- Ordinal ratings, such as "strongly agree, agree, neutral, disagree, strongly disagree" where users

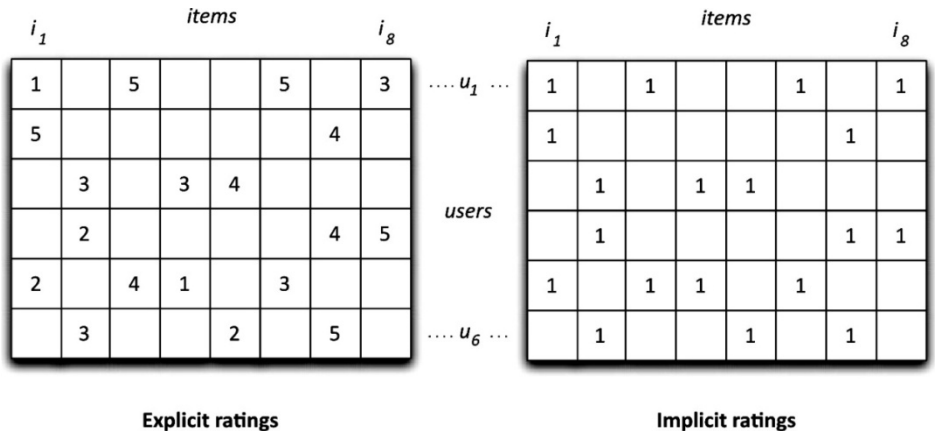

Fig. 1: Matrix ratings in the case of explicit user opinions (left), or unary in the case of implicit user activity (right)

are asked to select the term that best indicates their opinion regarding an item (usually via questionnaire).

- Unary rating, by contrast, allows users to assign items only to a single class, which is positive in most cases and a prominent example is the Facebook's "Like"-button. Purchased products in a web shop or clicked links on a new page can be implicit unary ratings and also in addition, unary rating can signal the purchase or observation of an item by users, or rating them positively. With the above cases, the absence of a rating indicates that there is no information relating the user to the item (perhaps a purchase was made somewhere else).

There are two techniques to getting the information needed to establish a user profile. The first, an explicit strategy, depends on the users to provide the information, the second, an implicit strategy, tries to gather this information without relating to the user directly.

Explicit Profiling: The most apparent way to collect a customer's preferences is simply to ask the user to provide the information. This explicit profiling is often achieved by demanding the user to complete a preliminary set of questions detail any appropriate preferences or background, for example, MyYahoo. It can also be obtained (or refined) by asking the user future information content on an ongoing basis, according to how appropriate or interesting the individual found each item/product (Fig. 1).

Implicit Profiling: Implicit profiling techniques build individual information by inferring users rating from so-called interest indicators depending on customer's interactions with the system (Claypool *et al*., 2001; Goecks and Shavlik, 1999). We can draw out this information unquestioningly by tracking the customer's behavior as they get navigate or use a service and using these findings to infer what the customer's preferences are (Fig. 1).. On the other hand, the implicit rating constructs the user-item matrix by tracking users' behaviors such as whether or not an activity (e.g., buy, access, save, print) is conducted to the product, how long they invest some time on studying, for example, the product and how many times they have browsed the product and so on Lee *et al*. (2005) and Nichols (1998).

User-item matrix: User-item matrix is a matrix of customers against products that have components as the explicit ratings of customers to products (user to item). Some of the user-matrix cells are not loaded, as there are products that are not rated by any user.

For M items and K users, the user profiles are represented in a K×M user-item matrix X. Each element $x_{k,m} = r$ indicates that user k rated item m by r, where $r \in \{1, ..., |r|\}$. If the item has been rated and $x_{k,m} = 0$; means that the rating is unknown.

The user-item matrix can be decomposed into row vectors:

$$X = [u_1, \ldots, u_K]^T, u_k = [x_{k,1}, \ldots, x_{k,M}]^T, k = 1, \ldots, K \quad (1)$$

where, T denotes transpose. Each row vector u^T_k corresponds to a user profile and represents a particular user's item ratings. As discussed below, this decomposition leads to user based CF. Alternatively, the matrix can also be represented by its column vectors:

$$X = [i_1, ..., i_M], i_m = [x_{1,m}, ..., i_{K,m}]^T, m = 1, ..., M \quad (2)$$

where, each column vector i_m corresponds to a specific item's ratings by all K users.

This representation results in item based recommendation algorithms.

Recommendation problem: The recommendation problem can be formulated as follows (Adomavicius and Tuzhilin, 2005). Consider U (u_1, u_2, …, u_m) be the group of all available users in a recommender system and consider I(i_1, i_2, …, i_n) be the group of all items users have access to in the system. Let f: U×I→R, where R indicates an entirely ordered set be a utility function such that $f(u_m, i_n)$ computes the usefulness of item i_n to user u_m. Then, for every user $u_m \in U$, the system chooses an item $i^{max,um} \in I$, unknown to the active user, which maximizes the utility function f.

Recommendation techniques: To be able to apply recommender system core function, determining the useful products, a recommender system must examine products which are valuable for suggesting to the target user. The system must be able to predict the application of some of them, or at least evaluate the utilization of some products and then choose which products are suitable to suggest depending on this evaluation.

Recommendation methods have a variety of possible categories (Resnick and Varian, 1997; Schafer *et al*., 1999). For arranging a first review of the different kinds of recommender systems, we want to quotation a taxonomy offered by Burke (2007a) that has become a traditional way of identifying between recommender techniques. Burke (2007b) differentiates between six different classes of recommendation approaches as:

Content-based: Content recommender systems try to suggest products that are similar to the ones that the user liked in the past. The likeness of items is determined depending on the traits associated with the compared items. For example, if an individual user has favorably rated a movie that connected to the comedy category, then the program can understand to suggest other movies from this category. Furthermore, Content-based recommenders treat suggestions as a user-specific category problem and learn a classifier for the customer's preferences depending on product traits.

According to Ziegler (2004), techniques applying a content-based recommendation strategy evaluate a set of documents and/or details of products previously ranked by a user and develop a model or user profile of user passions depending on the features of the things rated by that user.

Content-based recommender system can be used in a variety of domains ranges i.e., recommending web pages, news articles, jobs, television programs and products for sale.

According to Pazzani and Billsus (2007), Generally CF-based recommendation systems:

- Construct a user profile from rating information of each user on items;
- Identify like-minded users who rate items similarly to a target user using a similarity function such as cosine similarity, Pearson correlation coefficient, or distance-based similarity
- Recommend top n items that the like-minded users preferred after their ratings are predicted as an average, weighted sum or adjusted weighted sum of ratings given on items by the identified like-minded users.

Collaborative filtering: Based on the genuine and ordinary of this strategy (Schafer *et al*., 2007b) the items that other users with similar tastes liked in the past are recommended to the target user. The likeness in taste of two users is computed with regards to the likeness in the past ratings of the users.

All CF methods share a capability to utilize the past ratings of users in order to predict or recommend new content that an individual user will like. The real assumption is highly based in the idea of likeness between users or between products, with the similarity being expressed as a function of agreement between past ratings or preferences. Two basic variants of CF approach can be classified as user-based and item-based.

According to Breese *et al*. (1999), methods for collaborative recommendations can be classified into two groups: memory-based and model-based. Memory-based methods (Breese *et al*., 1999; Resnick *et al*., 1994a) are heuristics that make ratings predictions depending on the whole collection of items formerly rated by users. These techniques require all ratings, items and users to be maintained in memory. Model-based methods (Goldberg *et al*., 2001) use the group selection of ratings to learn a model, which is then used to make rating predictions. These techniques regularly make a concise of ratings patterns off-line.

Figure 2 demonstrates the interaction of an online user with a collaborative recommender system through a web interface.

Demographic: This type of system suggests items depending on the user demographic profile. The supposition is that different recommendations should be produced for different demographic records. Many

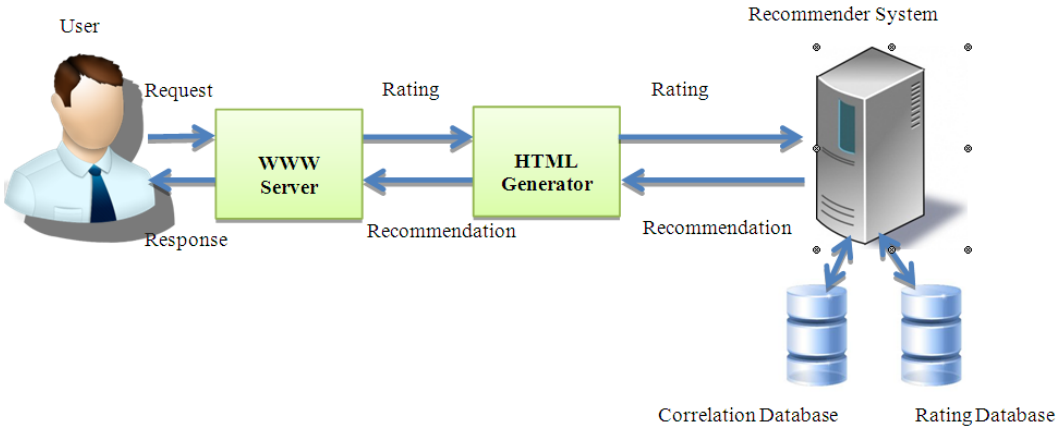

Fig. 2: CF System architecture

websites embrace simple and effective customization solution depending on demographic. For example, customers are dispatched to specific websites depending on their language or nation. Or recommendations may be personalized according to the age of the user. While these methods have been quite popular in the marketing aspect, there has been relatively little appropriate recommender system research into demographic systems (Ricci and Nguyen, 2006a). Pazzani (1999) uses machine learning methods to acquire a classifier depending on demographic information. The advantage of a demographic approach is that usually is not needed a record of user rates for a sort that is needed in the type of collaborative and content-based methods that. Because of difficulty to capture information in this type of recommendation, there are not many recommender techniques using demographic date.

Knowledge-based: In this type of recommender system, Knowledge-Based (KB) systems recommend items based on particular domain knowledge about how certain item features fulfill users' needs and preferences and, eventually, how the item is useful for the user. Remarkable knowledge-based recommender systems typically are Case-Based (CB) (Bridge *et al*., 2006; Ricci *et al*., 2006b). For any KB systems a similarity function considers problem description (needs) and solution of the problem (match the recommendation) and estimates how much the user needs them. The similarity score also can be directly expounded as the utility of the recommendation for the user.

Based on functional knowledge KB approaches are recognized that in them a knowledge causes to how a particular item meets a particular user need and also can reason about the association between a need and a possible recommendation (Burke, 2002). In KB system, the user profile can be a source for knowledge structure to support the mentioned inference .We can consider Google as a simplest case that uses query of a user for its recommendations. Towle and Quinn (2000) mentioned a more detailed representation of the user need.

Case-Based Reasoning (CBR) is a specific kind in this case which is implemented by KB systems.
Typically KB systems find a solution for solving a new problem seeking a similar solved in the past.

According to Lorenzi and Ricci (2003), retrieve, reuse, adaptation and retain are four main steps of a CBR recommender .The recommendations of KB systems do not depend on abase of user ratings therefore it do not have a ramp-up problem ("early rater" problem and the "sparse ratings" problem). Therefore KB approach can be a complement to other recommender approaches (Burke, 2000)

Community-based: This kind of system works on the preferences of the users friends to recommend items. Evidence demonstrates that customers tend to rely more on recommendations from their friends than on recommendations from similar but anonymous users (Zhang *et al*., 2002).

Community-based is also connected with the raising popularity of open social networks with a growing in community-based or social recommender systems (Gabrilovich and Markovitch, 2006). This kind of recommender systems models obtains information from the social relationships of the users groups and the preferences of the user's friends in that group.

For identifying the community of the social relationships in community-based recommender system, several statistical and graph-based approaches have been applied. In this case a few instances are Bayesian generative models (Delong and Erickson,

2008), graph clustering approaches, hierarchical clustering and modularity-based methods (Fortunato, 2010).

Hybrid recommender systems: Hybrid recommender system can be obtained from a combination of mentioned techniques by combining two or more techniques that tries to alleviate disadvantages of them. A hybrid approaches more have been used by combing collaborative and content-based methods, which tries to improve shortcomings of both (Burke, 2002; Adomavicius and Tuzhilin, 2005; Burke, 2007a; Li and Kim, 2003; Liu *et al*., 2009, 2010; Salter and Antonopoulos, 2006; Wei *et al*., 2008). Moreover, a combination for developing hybrid recommender system is depending on the domain and data characteristics. Seven categories of hybrid recommendation systems, weighted, switching, mixed, feature combination, feature augmentation, cascade and meta-level have been introduced by Burke (2002).

COLLABORATIVE FILTERING METHODS

CF method uses the opinion of other community for recommendation to target user. Generally, prediction for target user is made based on many other similar users by gathering taste information (Schafer *et al*., 2007a).

Thereby CF supposed that those users agreed in the past tend to agree also in the future. Usually processing the mass of information, including enlarged datasets such as in electronic commerce and web applications are needed to be performed by CF systems. Within the last decade CF has been enhanced constantly and lastly became one of the most popular customization methods in the area of recommendation techniques.

Nowadays computers and the internet let us to consider the views of huge connected areas with a huge number of members (Schafer *et al*., 2007b). Individuals can benefit from a groups of community, in that they obtain access to the knowledge of other customers and their encounter about different products. Furthermore, these details can assists users to create their own customized perspective or to determine regarding the products that were rated. To be more particular, users utilize CF techniques for discovering new products they might like, getting recommendation on particular products and linking to other customers with same references.

Collaborative filtering processes: The objective of a CF algorithm is to recommend new products or to estimate the utility of a certain product for a specific user depending on the customer's past likings and the views of other like-minded users. There are two tasks that a CF can perform, leading to two unique kinds of result.

Rating prediction: The first task is the rating prediction process-that of predicting the rating that a given unseen product will have for the target user.

Prediction, $P_{a,j}$, can be defined as a numerical value that indicates the predicted likeliness of item $i_j \notin I_{ua}$ for the target user u_a. Also same scale that is considered for the opinion values provided by u_a also is considered for predicted value within the scale (e.g., from 1 to 5).

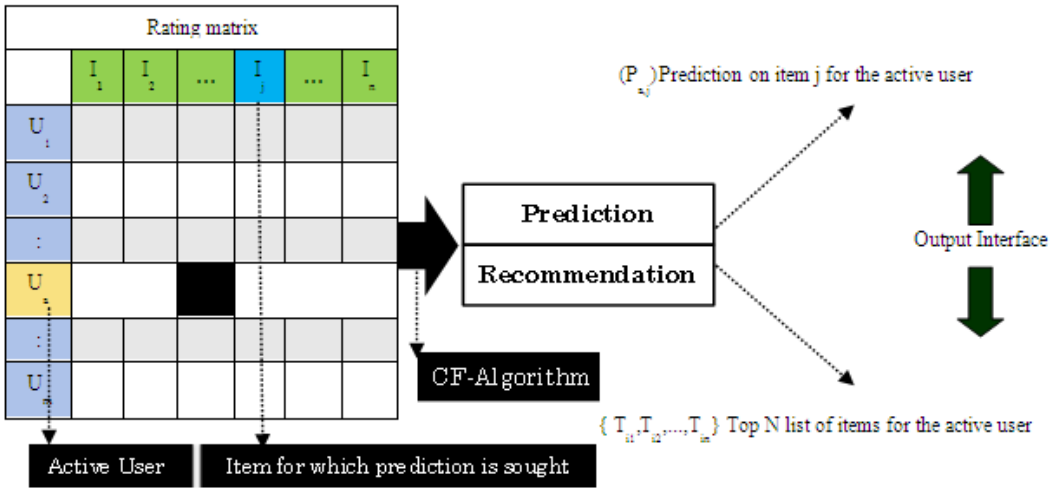

Fig. 3: The CF processes

Recommendation task: Typically, recommendation task as a second task in CF provides top-N recommendation list of unseen relevant items for the target user.

Recommendation can be defined as a list of *N* items, $I_r \subset I$, that the target user will like the most that can generated after rating prediction. Recall that the suggested list must be on items not already purchased by the target user, i.e., $I_r \cap I_{ua} = \Phi$.

In a general CF scenario, a list of *m* users as $U = \{u_1, u1, \ldots, u_m\}$ and a list of *n* items as $I = \{i_1, i1, \ldots, i_n\}$ are needed for both tasks of prediction and recommendation .

The general schematic diagram of the CF process is shown in Fig. 3.

In the prior researches, authors have classified CF systems in accord with whether they adopt a memory-based or a model-based approach (Adomavicius and Tuzhilin, 2005; Deshpande and Karypis, 2004).

MODEL-BASED AND MEMORY-BASED COLLABORATIVE FILTERING

Model-based CF: Model-based CF can often suggest significant usefulness over memory-based algorithms regarding the efficiency but until more recently have not presented the same level of accuracy. Model-based CF adopts an eager learning strategy that get getting a probabilistic approach for two tasks, predicting or recommending content, that a model of the information, i.e., the users data, items data with their ratings for those items in the recommender system, is pre-computed. For Deriving the model of model-based filtering in the past (Sarwar *et al*., 2001), machine learning algorithms such as Bayesian networks (Breese *et al*., 1998), clustering (Basu *et al*., 1998; Breese *et al*., 1998; Ungar and Foster, 1998) and rule-based approaches (Sarwar *et al*., 2000a) have been commonly used.

Memory-based algorithms: Memory-based approaches are more typical in the literary works than model-based but in this method an intensive memory is needed for implementing. Memory-based has become a well-known design of CF. It has been applied impressively in many online systems, especially Amazon. This tension of CF approaches is slothful in it learning sense and simply leaves all computations until a position prediction or a recommendation is required. Moreover, in memory-based algorithms pre-computation is not necessary to be done and no off-line design is developed. For making a prediction or recommendation in memory-based CF algorithms all information including the most recent transaction information is immediately available therefore it can be a main advantage of these algorithms.

For finding K-nearest neighbours to the active user or target item, memory-based CF algorithms typically use statistical techniques based on a history of shared

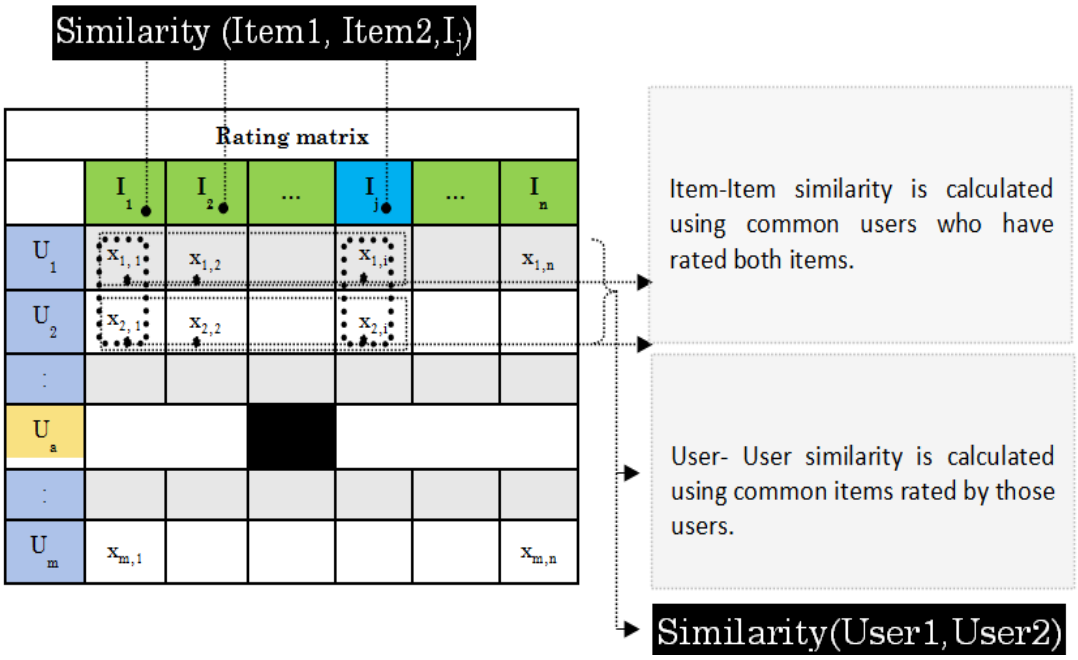

Fig. 4: Item and user based similarity memory-based CF

ratings. In this case, based on neighbour distance or correlation from active user, each neighbour receive a weight and then the algorithm in some manner combines the preferences of the nearest neighbours to generate a prediction or recommendation for the target user.

Furthermore, memory-based CF has been more usually associated to as neighbour-based CF, reflecting its heavy confidence on the k-nearest neighbour algorithm. Commonly, user-based nearest neighborhood and item-based nearest neighborhood are two basic NNH techniques that a memory-based CF algorithm employs them in its tasks (Schafer *et al*., 2007a). Figure 4 demonstrates the item and user based similarity memory-based CF.

User-based neighborhood: User-based neighborhood methods first seek who shared the same rating pattern with the target user and then use the ratings of the similar users to estimate the predictions and then recommendation. This method for calculating the rating for a yet unrated item of the active user, average the ratings of the nearest neighbors about this particular item. In order to generate more accurate predictions, rating values of neighbor are assigned with weights according their similarity to the target user. This method for generating the more precise prediction, weights allocate to the values of neighbor based their similarity to the active user.

Item-based neighborhood: The transpose of user-based algorithms is item-based nearest neighbor algorithms that produce predictions based on similarities between items. An item-based method exploits the similarity among the items. This method looks into the set of items that a user has rated and computes the similarity among the target item (to decide whether is worth to recommend it to the user or not).

Similarity metrics in collaborative filtering: One crucial step in the CF algorithm is to calculate the similarity between items and users and finally to choose a group of nearest neighbours as recommendation partners for an active user. After establishing a set of profiles by the recommender system, it is possible to reason about the similarities between users or items and finally chooses a group of nearest neighbours as recommendation partners for an active user. Because of importance of similarity matrices, some of the most common similarity metrics that used in CF will be examined in detail.

Cosine similarity: Usually cosine similarity metric is used for estimate the similarity between two objects a and b in information retrieval that the objects are in the shape of two vectors x_a and x_b and calculating the Cosine Vector (CV) (or Vector Space) similarity between these vectors indicate the distance of them to each other Billsus and Pazzani (1998, 2000) and Lang (1995):

$$\cos(X_a, X_b) = \frac{X_a . X_b}{\| X_a \|^2 * \| X_b \|^2} \tag{3}$$

In the context of item recommendation, for computing user similarities, this measure can be employed in which a user u indicates vector $x_u \in R^{|I|}$ where $x_{ui} = r_{ui}$ if user u has rated item i and for unrated item considers 0. The similarity between two users' u and v would then be calculated as:

$$CV(u,v) = \cos(X_u, X_v) = \frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2} \sqrt{\sum_{j \in I_v} r_{vi}^2}} \tag{4}$$

where, I_{uv} once more indicates the items rated by both u and v. A shortcoming of this measure is that it does not examine the differences in the mean and variance of the ratings made by user's u and v.

Cosine similarity is calculated on a scale between -1 and +1, where -1 implies the objects are completely dissimilar, +1 implies they are completely similar and 0 implies that the objects do not have any relationship to each other.

In prior researches, vector similarity has been proven to work well in information retrieval (Salton and Buckley, 1998) but it has not been found to carry out as well as Pearson's for user-based CF (Breese *et al*., 1998).

Pearson correlation: Pearson Correlation (PC) is a well-known metric that compares ratings where the effects of mean and variance have been eliminated is the Pearson Correlation (PC) similarity:

$$PC(u,v) = \frac{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)(r_{vi} - r_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)^2 \sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}} \tag{5}$$

Also, for acquiring the similarity between two items i and j the ratings given by users that have rated both of these items is compared:

$$PC(i,j) = \frac{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_i)(r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_i)^2 \sum_{u \in U_{ij}} (r_{uj} - \bar{r}_j)^2}} \tag{6}$$

Spearman's correlation coefficient: Spearman's correlation coefficient is a rank coefficient that independent of the actual item rating values, estimates the difference in the ranking of the items in the profiles. First user's list of ratings is turned into a list of ranks, where the user's highest rating takes the rank of 1 and tied ratings take the average of the ranks for their spot (Herlocker *et al*., 2002). Herlocker *et al*. (1999) showed that Spearman's performs similarly to Pearson's for user-based CF:

$$SRC(i,j) = \frac{\sum_{i \in I}(r_{a,i} - \bar{r}_a)(r_{b,i} - \bar{r}_b)}{\sqrt{\sum_{i \in I}(r_{a,i} - \bar{r}_a)^2 \sum_{i \in I}(r_{b,i} - r_b)^2}} \quad (7)$$

The Spearman Correlation Coefficient for user-user similarity between two users a and b have been represented in Eq. (7). It is declared regarding the set of all co-rated items (I) that $r_{a,i}$ and $r_{b,i}$ indicate rank each user gave to each item i and $\bar{r}_a$ and $\bar{r}_b$ finally indicate each user's average rank. Once again, the correlation is measured on a scale between -1 to +1 where , -1 implies the objects are completely dissimilar, +1 implies they are completely similar and 0 implies that the objects do not have any relationship to each other.

Adjusted cosine similarity: To consider a shortcoming of standard cosine similarities metric for item-based CF that does not take individual users' rating scales into account, this method was presented by Sarwar *et al*. (2001). After calculating the similarity between two items i and j, by subtracting the user's average rating from each co-rated pair, the adjusted metric compensates result. The formula seems similar to the Pearson coefficient for item similarities but it considers user average rather than the item average that is subtracted from each co-rated pair. Equation 8 represent the similarity between items i and j:

$$Sim(i,j) = \frac{\sum_{u \in U}(r_{u,j} - \bar{r}_u)(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U}(r_{u,i} - \bar{r}_u)^2 \sum_{u \in U}(r_{u,j} - \bar{r}_u)^2}} \quad (8)$$

Mean Squared Difference (MSD): For estimating the similarity between two users u and v MSD mature is applied as the reverse of the average squared difference between the ratings made by u and v on the same items (Shardanand and Maes, 1995):

$$MSD(u,v) = \frac{|I_{uv}|}{\sum_{i \in I_{uv}}(r_{ui} - r_{vi})^2} \quad (9)$$

While it could be modified to compute the differences on normalized ratings, the MSD similarity compared to PC similarity has one shortcoming that it does not capture negative correlations between user preferences or the appreciation of different items but having such negative correlations can improve the rating prediction accuracy (Gori and Pucci, 2007).

The Jaccard coefficient: The Jaccard coefficient is a measure for calculating the similarity between two users with binary profiles, i.e. ratings are not taken into account. Equation 10 shows it as the similarity between two users u and v, determined by the profile intersection as a fraction of the profile union that values range between 0 and 1 are result of this measure, where 0 indicates there is no similarity and 1 indicates there is perfect similarity.

$$Sim(u,v) = \frac{|U \cap V|}{|U \cup V|} \quad (10)$$

Conditional probability-based similarity: Karypis (2001) proposed the conditional probability-based metric as a similarity metric for item-based CF Top-N recommendations. The similarity between two items i and j is simply the probability of purchasing (rating) one given that the other has already been purchased. Thus the probability of purchasing a given that b has been purchased is determined as the number of users that purchased both items divided by the total number of users that purchased b. Note that this metric gives asymmetric similarities since (P(i|j) ∉ P(j|i)). The similarity of i to j is given in Eq. (11) as:

$$Sim(i,j) = P(i \mid j) = \frac{freq(i,j)}{freq(j)} \quad (11)$$

According to the Deshpande and Karypis (2004), one of the shortcomings of an asymmetric metric is that each item tends to have high conditional probabilities with regard to the most favored items. To allay this shortcoming, the following form of the conditional probability is presented in Deshpande and Karypis (2004):

$$Sim(a,b) = P(a \mid b) = \frac{\sum_{\forall u: R_{u,b} > 0} R_{u,b}}{freq(a), (freq(b)^{\alpha})} \quad (12)$$

where, α ∈ [0, 1] and Freq (a) indicates the number of users that have a transaction on item i in the training data and R (u, b) is the (u, b) element in the normalized user-item matrix.

Predictions and recommendations in collaborative filtering: The ultimate step in CF is to generate some significant outcome for the individual user that should help him in his choice of future products. There are two tasks that a CF is can carry out, leading to two unique types of results. The first is the rating prediction task that makes a prediction rating to give the unseen item to target user. The second is the recommendation task - that of producing a top-N recommendation list of unseen relevant items for the target user.

Producing rating predictions: In user-based CF the standard way of producing a prediction is presented in Resnick *et al*. (1994b) and shown in Eq. (13). Essentially the prediction is a weighted sum of the ratings of the target user's k nearest neighbours which are selected using Pearson's correlation coefficient:

$$r_{t,i} = \bar{r}_t + \frac{\sum_{u \in N} (r_{u,i} - \bar{r}_u) \times S_{t,u}}{\sum_{u \in N} |S_{t,u}|} \quad (13)$$

In item-based CF the typical prediction algorithm is similar, however the similarity this time is estimated using the adjusted cosine similarity metric rather than person's correlation coefficient, since the study by Sarwar *et al*. (2001) demonstrated it worked best for item-based CF. The formula is given in Eq. (14).

$$r_{t,i} = \bar{r}_i + \frac{\sum_{j \in N} (r_{t,i} - \bar{r}_j) \times S_{i,j}}{\sum_{j \in N} S_{i,j}} \quad (14)$$

Producing recommendations in collaborative filtering: Often the ultimate objective of a CF system is not so much to estimate particular ratings that an active user will provide to particular items, but rather to compile a top-N list of recommendations for target user. We describe in the following some of the general methods for producing recommendation.

Frequency-based: In concept of recommender systems a candidate set of items C for recommendations is shaped by taking the set of all items that arise in N that do not already occur in the target users profile.

In frequency-based the items that occur in C are located in decreasing order of their frequency in N. The n most frequently occurring items are recommended to the target user. Frequency-based approach has been shown in Equation 15 for an item i and a neighborhood N. This approach was used by Sarwar *et al*. (2000b) and Karypis (2001):

$$Weight(i) = Freq(i; N) \quad (15)$$

Prediction-based: Based on candidate set of C, the Eq. (16) shows as the weight for an item i in C using the prediction formula described in section producing rating predictions which predicts the rating that the target user t will give to i. Items are then located in decreasing order of their predicted ratings. This approach is used in Lam and Riedl (2004) and Ziegler *et al*. (2005):

$$Weight(i) = \bar{r}_t + \frac{\sum_{u \in N} (r_{u,i} - \bar{r}_u) \times S_{t,u}}{\sum_{u \in N} |S_{t,u}|} \quad (16)$$

Ratings-based: Based on candidate set of C, the items that occur in C are located in decreasing order of the sum of the ratings they obtained across the profiles in N. Equation 17 shows the formula for computing the item weight for an item i and a neighbourhood N:

$$Weight(i) = \sum_{u \in N: i \in n} sim(n, t) \quad (17)$$

Similarity-based: Based on candidate set of C, the items in C are weighted in regard of the sum of the similarities of the profiles in N in which they occur. Equation 18 shows the formula for computing the item weight for an item i and a neighbourhood N and an active user t.

$$Weight(i) = \sum_{u \in N: i \in n} r_i \quad (18)$$

EVALUATING COLLABORATIVE FILTERING ACCURACY

Recommender systems are implemented to help users in recognizing desirable information. Accuracy is one of the widely used performances metric for recommendation, which quantifies the degree of errors between actual and predicted ratings. It is also the most common criterion used to evaluate the success of a recommender system both related to the rating predictions and to the recommendations.

Accuracy can be approximately categorized into predictive accuracy, decision-support accuracy and rank accuracy (Herlocker *et al*., 2004).

Predictive accuracy metrics: Predictive accuracy metrics evaluate the conforming to true user ratings from predictions of a recommender system.

MAE is the metric generally employed for this purpose and is much more widely used than the other metrics. The MAE is determined as the average absolute deviation between predicted ratings and true ratings that showed in Eq. (19).

In Table 1 we display an example computation of the MAE for a user based the predicted and actual ratings. In this case the 5-point rating scale is selected

Table 1: Example MAE calculation from real and predicted ratings

Item ID	21	39	05	77
Actual rating	5	3	2	1
Predictions	4	3	4	2

for ratings and the MAE score of 1 indicates that the distance between predicted ratings and actual ratings on average is 1 point.

$$MAE(pred, act) = \sum_{i=1}^{N} \left| \frac{pred_{u,i} - act_{u,i}}{N} \right| \quad (19)$$

$$MAE(pred, act) = \frac{|5-4|+|3-3|+|2-4|+|1-2|}{4} = \frac{4}{4} = 1$$

Root Mean Squared Error (RMSE) is another metric in predictive accuracy that is the statistical accuracy calculated preferred in the Netflix Prize Competition. In concept it is similar to MAE, however the squaring of the error results in more emphasis on large errors than would be given using MAE.

$$RMSE(pred, act) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(pred_{u,i} - act_{u,i})^2} \quad (20)$$

Decision-support accuracy metrics: Decision-support precision metrics evaluate the acceptance rate by the user. Reversal rate, ROC (receiver operating characteristics) curve, precision and recall are general metrics of this type. Reversal rate is an evaluating of the frequency and identify poor recommendation that might corrupt a user's confidence in the system .A high reversal rate indicates that the system frequently makes poor recommendations respecting whether a user will strongly like or dislike an item. For identifying that which the system presents relevant information, precision and recall are measures of the degree in this issue. We can define the precision as the ratio of relevant items selected to the number of items selected. Also recall is defined as the ratio of relevant items selected to the total number of relevant items available. F-Measure is also often utilized for combing precision and recall (Herlocker *et al*., 2004).

$$Precision(Reclist) = \frac{|\{relevant\ document\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} \quad (21)$$

$$Precision(Reclist) = \frac{|\{relevant\ document\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} \quad (22)$$

$$f(Reclist) = 2.\frac{precision.recall}{precision + recall} \quad (23)$$

Rank accuracy metrics: Rank precision metrics evaluate the ability of a recommendation algorithm to generate a suggested purchasing of items that suits how the individual would have requested the same items. Therefore, such metrics are proper for assessing methods that will be used to present a ranked recommendation record to users. Exceptional metrics comprise correlation coefficient, half-life utility and the normalized Distance-Based Performance Measures (NDPM). The correlation coefficient indicates the power and the path of a liner line association between two random variables; three of the well-known coefficients are the Pearson's coefficient, Spearman's and Kendall's. The half-life utility measurement analyses a ranked list from the user with regards to the difference between the user rating and default rating for products. For weakly requested rankings, NDPM is also employed. It is worth noting that, according to Herlocker *et al*. (2004), these precision metrics are carefully correlated.

- **Half-life utility metric:** Breese *et al*. (1998) provided a new assessment measurement for recommender techniques that is developed for projects where the individual is provided with a ranked list of outcomes and is unlikely to browse through very profoundly into the rated list. Information of this measurement can be found in Heckerman *et al*. (2000). The process for which the measurement is developed is an online web-page recommender. They declare that most Web customers will not browse very profoundly into results given by search engine .Half-life utility measurement tries to assess the utility of a ranked list to the user. The utility is determined as the distinction between the user's rating for item and the "default rating" for items. The default ranking is usually a fairly neutral or a little bit negative ranking. The chances that an individual will observe each subsequent product is described with an exponential decay function, where the durability of the exponential is described by a half-life parameter.

The expected utility (R_α) is shown in Eq. (24). In this Equation, $r_{a,j}$ signifies the rating of user a on product j of the ranked list, d indicates the default rating and α implies the half-life. The half-life is the rank of the item on the list such that there is a 50% chance that the user will view that item. Breese *et al*. (1998) used a half-life of 5 for his tests, but mentioned that using a half-life of 10 triggered little extra understanding of outcomes:

$$R_a = \sum_j \frac{\max(r_{a,j} - d, 0)}{2^{(j-1)(\alpha-1)}} \quad (24)$$

The overall ranking for a dataset across all users (R) is presented in Eq. (25). Ramax is the highest possible utility if the system ranked the products in the actual purchase that the individual rated them:

$$R = 100 \frac{\sum_\alpha R_a}{\sum_\alpha R_a^{\max}} \quad (25)$$

The NDPM measure: NDPM was initially suggested by Yao (1995) and used to assess the precision of the FAB recommender system (Balabanovíc and Shoham, 1997). Yao designed NDPM theoretically, using an approach from decision and measurement theory.concept. NDPM indicates "normalized distance-based performance measure".

NDPM Eq. (26) can be used to contrast two different weakly-ordered rankings. Moreover, NDPM metric is comparable among different datasets (it is normalized), because the numerator stand for the distance and the denominator represents worst possible distance. NDPM is identical in type to the Spearman and Kendall's Tau rank correlations, but produce a more precise interpretation of the impact of linked user ranks:

$$NDPM = \frac{2C^- + C}{2C^i} \quad (26)$$

- **Correlation between ratings and predictions:** A third metric that is used to assess the precision of predicted ratings is to evaluate the correlation between vector of predicted ratings and vector of actual ratings (Hill *et al*., 1995; Sarwar *et al*., 1998; Konstan *et al*., 1997). Person's Correlation Coefficient uses Eq. (5) to determine the correlation. Recall that the correlation is calculated on a range of-1 to+1 where-1 signifies a perfect negative correlation and+1 signifies a perfect positive correlation.

CONCLUSION

Recommender systems play an important role in the providing of user-specific services by filtering the large variety of available data to draw out information on user preferences.

In this study we highlighted the importance of recommender systems and analyzed CF technique with its techniques extensively. Furthermore, the most common recommender systems were introduced and also two main tasks in the CF, recommendation and prediction were investigated with specifying their related techniques. Finally, we presented the most common evaluation metrics of collaborative recommender systems and explained the evaluation of these systems based on the accuracy into three classes, predictive accuracy, decision-support accuracy and rank accuracy.

REFERENCES

Adomavicius, G. and A. Tuzhilin, 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE T. Knowl. Data En., 17(6): 734-749.

Anand, S.S. and B. Mobasher, 2005. Intelligent Techniques for Web Personalization. In: Carbonell, J.G. and J. Siekmann (Eds.), ITWP 2003. LNCS (LNAI), Springer, Heidelberg, 3169: 1-36.

Armstrong, J.S., 2001. Combining Forecasts. Principles of Forecasting: A Handbook for Researchers and Practitioners. Kluwer Academic, Norwell, MA, pp: 417-439.

Balabanovíc, M. and Y. Shoham, 1997. Fab: Content-based: Collaborative recommendation. Commun. ACM, 40: 66-72.

Basu, C., H. Hirsh and W. Cohen, 1998. Recommendation as classification: Using social and content-based information in recommendation. Proceedings of the 15th National Conference on Artificial Intelligence (AAAI). Madison, Wisconsin, USA, pp: 714-720.

Billsus, D. and M.J. Pazzani, 1998. Learning collaborative information filters. Proceeding of the 15th International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp: 46-54.

Billsus, D. and M.J. Pazzani, 2000. User modeling for adaptive news access. User Mod. User-adapted Interac., 10(2-3): 147-180.

Breese, J., D. Heckerman and C. Kadie, 1998. Empirical analysis of predictive algorithms for collaborative filtering. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI). Morgan Kaufmann Publishers, Madison, Wisconsin, USA, pp: 43-52.

Breese, J.S., D. Heckerman and C. Kadie, 1999. Empirical analysis of predictive algorithms for collaborative filtering. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp: 43-52.

Bridge, D., M. G¨oker, L. McGinty and B. Smyth, 2006. Case-based recommender systems. Knowl. Eng. Rev., 20(3): 315-320.

Burke, R., 2000. Knowledge-based recommender systems. Encycl. Lib. Inform. Syst., 69(32).

Burke, R., 2002. Hybrid recommender systems: Survey and experiments. User Mod. User-adapted Interac., 12(4): 331-370.

Burke, R., 2007a. Hybrid Web Recommender Systems. The Adaptive Web, Springer Berlin, Heidelberg, pp: 377-408.

Burke, R.D., 2007b. Hybrid web recommender systems. Lect. Notes Comput. Sc., 4321: 377-408.

Claypool, M., P. Le, M. Wased and D. Brown, 2001. Implicit interest indicators. Proceeding of International Conference on Intelligent User Interfaces, pp: 33-40.

Delong, C. and K. Erickson, 2008. Social topic models for community extraction categories and subject descriptors. October, 2008.

Deshpande, M. and G. Karypis, 2004. Item-based top-N recommendation algorithms. ACM T. Inform. Syst., 22(1): 143-177.

Fortunato, S., 2010. Community detection in graphs. Phys. Reports, 486(3-5): 75-174.

Gabrilovich, E. and S. Markovitch, 2006. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. Proceedings of the 21th National Conference on Artificial Intelligence, AAAI Press, pp: 1301-1306.

Goecks, J. and J. Shavlik, 1999. Automatically labelling web pages based on normal user interactions. Proceedings of the UCAI Workshop on Machine Learning for Information Filtering. Stockholm, Sweden.

Goldberg, D., D. Nichols, B.M. Oki and D. Terry, 1992. Using collaborative filtering to weave an information tapestry. Commun. ACM, 35(12): 61-70.

Goldberg, P.K., T. Roeder, D. Gupta and C. Perkins, 2001. Eigentaste: A constant time collaborative filtering algorithm. Inform. Retriev. J., 4(2): 133-151.

Gori, M. and A. Pucci, 2007. Itemrank: A random-walk based scoring algorithm for recommender engines. Proceeding of the IJCAI Conference, pp: 2766-2771.

Heckerman, D., D.M. Chickering, C. Meek, R. Rounthwaite and C. Kadie, 2000. Dependency networks for inference: Collaborative filtering and data visualization. J. Mach. Learn. Res., 1: 49-75.

Herlocker, J.L., J.A. Konstan, A. Borchers and J. Riedl, 1999. An algorithmic framework for performing collaborative filtering. Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR). Berkeley, California, USA, ACM, pp: 230-237.

Herlocker, J.L., J.A. Konstan and J. Riedl, 2002. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Inform. Retriev., 5(4): 287-310.

Herlocker, J.L., J.A. Konstan, L.G. Terveen and J. Riedl, 2004. Evaluating collaborative filtering recommender systems. ACM T. Inform. Syst., 22(1): 5-53.

Hill, W.C., L. Stead, M. Rosenstein and G. Furnas, 1995. Recommending and evaluating choices in a virtual community of use. Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI). ACM Press/Addison-Wesley Publishing Co., Denver, Colorado, USA, pp: 194-201.

Karypis, G., 2001. Experimental evaluation of item-based top-N recommendation algorithms. Proceedings of 10th International Conference on Information and Knowledge Management (CIKM). ACM, Atlanta, Georgia, USA, pp: 247-254.

Konstan, J.A., B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon and J. Riedl, 1997. Group lens: Applying collaborative filtering to usenet news. Commun. ACM, 40: 77-87.

Lam, S.K. and J. Riedl, 2004. Shilling recommender systems for fun and profit. Proceeding of International World Wide Web Conference. ACM, New York, USA, pp: 392-402.

Lang, K., 1995. News weeder: Learning to filter netnews. Proceeding of the 12th International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., San Mateo, CA, USA, pp: 331-339.

Lee, J.S., C.H. Jun, J. Lee and S. Kim, 2005. Classification-based collaborative filtering using market basket data. Exp. Syst. Appl., 29(3): 700-704.

Li, Q. and B. Kim, 2003. Clustering approach for hybrid recommender system. Proceedings of the International Conference on Web Intelligence, pp: 33-38.

Lilien, G., P. Kotler and K. Moorthy, 2003. Marketing Models. Prentice-Hall of India, New Delhi, pp: 803, ISBN: 8120314751.

Liu, D.R., C.H. Lai and W.J. Lee, 2009. A hybrid of sequential rules and collaborative filtering for product recommendation. Inform. Sci., 179(20): 3505-3519.

Liu, Z., W. Qu, H. Li and C. Xie, 2010. A hybrid collaborative filtering recommendation mechanism for P2P networks. Future Generat. Comp. Syst., 26(8): 1409-1417.

López, M.B. and J.L. de la Rosa, 2003. A taxonomy of recommender agents on the internet. Artific. Intell. Rev., 19(4): 285-330.

Lorenzi, F. and F. Ricci, 2003. Case-based Recommender Systems: A Unifying View. In: Mobasher, B. and S.S. Anand (Eds.), ITWP. Springer, Heidelberg, pp: 89-113.

Mahmood, T. and F. Ricci, 2009. Improving recommender systems with adaptive conversational strategies. Proceedings of the 20th ACM Conference on Hypertext and Hypermedia. ACM, New York, pp: 73-82.

McSherry, F. and I. Mironov, 2009. Differentially private recommender systems: Building privacy into the net. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York, USA, pp: 627-636.

Montaner, M., B. L´opez and J.L. de la Rosa, 2003. Taxonomy of recommender agents on the internet. Artif. Intell. Rev., 19(4): 285-330.

Murthi, B. and S. Sarkar, 2003. The role of the management sciences in research on personalization. Manag. Sci., 49(10): 1344-1362.

Nichols, D., 1998. Implicit rating and filtering. Proceeding of the 5th DELOS Workshop on Filtering and Collaborative Filtering, pp: 31-36.

Pazzani, M.J., 1999. A framework for collaborative: Content-based and demographic filtering. Artif. Intell. Rev., 13(5-6): 393-408.

Pazzani, M.J. and D. Billsus, 2007. Content-based Recommendation Systems. In: Brusilovsky, P., A. Kobsa and W. Nejdl (Eds.), the Adaptive Web. LNCS 4321, Springer-Verlag, Berlin Heidelberg, pp: 325-341.

Powell, D., 1981. Approximation Theory and Methods. Cambridge University Press, Cambridge, pp: 352, ISBN: 0521295149.

Resnick, N., M. Iacovou, P. Suchack, J.T. Bergstrom and G.L. Riedl, 1994a. An open architecture for collaborative filtering of netnews. Proceedings of the ACM Conference on Computer Supported Cooperative Work, pp: 175-186.

Resnick, P. and H.R. Varian, 1997. Recommender systems. Commun. ACM, 40(3): 56-58.

Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl, 1994b. GroupLens: An open architecture for collaborative filtering of netnews. Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW). ACM, Chapel Hill, North Carolina, USA, pp: 175-186.

Ricci, F. and Q.N. Nguyen, 2006a. Mobyrek: A Conversational Recommender System for on-the-move Travelers. In: Fesenmaier, D.R., H. Werthner and K.W. Wober (Eds.), Destination Recommendation Systems: Behavioural Foundations and Applications. CABI Publishing, London, pp: 281-294.

Ricci, F., D. Cavada, N. Mirzadeh and A. Venturini, 2006b. Case-based Travel Recommendations. In: Fesenmaier, D.R., K. Woeber and H. Werthner (Eds.), Destination Recommendation Systems: Behavioural Foundations and Applications. CABI, London, pp: 67-93.

Rich, E., 1979. User modeling via stereotypes. Cognitive Sci., 3: 329-354.

Salter, J. and N. Antonopoulos, 2006. Cinema screen recommender agent: Combining collaborative and content-based filtering. IEEE Intell. Syst., 21(1): 35-41.

Salton, G. and C. Buckley, 1998. Term-weighting approaches in automatic text retrieval. Inform. Proc. Manag. Int. J., 24(5): 513-523.

Salton, G.G., 1989. Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer. Addison-Wesley, Reading, Mass, pp: 530, ISBN: 0201122278.

Sarwar, B.M., G. Karypis, J.A. Konstan and J. Riedl, 2000a. Application of dimensionality reduction in recommender system: A case study. Proceedings of the ACM WebKDD Workshop at the ACM-SIGKDD Conference on Knowledge Discovery in Databases (KDD). Boston, Massachusetts, USA.

Sarwar, B.M., G. Karypis, J.A. Konstan and J. Riedl, 2000b. Analysis of recommendation algorithms for e-commerce. Proceedings of the 2nd ACM Conference on Electronic Commerce. ACM, Minneapolis, Minnesota, USA, pp: 158-167.

Sarwar, B.M., G. Karypis, J.A. Konstan and J. Riedl, 2001. Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th International World Wide Web Conference (WWW). ACM, Hong Kong, pp: 285-295.

Sarwar, B.M., J.A. Konstan, A. Borchers, J.L. Herlocker, B.N. Miller and J. Riedl, 1998. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW). Seattle. ACM, Washington, USA, pp: 345-354.

Schafer, J.B., D. Frankowski, J. Herlocker and S. Sen, 2007a. Collaborative Filtering Recommender Systems. In: Brusilovsky, P., A. Kobsa and W. Nejdl (Eds.), the Adaptive Web. Springer Berlin, Heidelberg, pp: 291-324.

Schafer, J.B., F. Dan, H. Jon and S. Shilad, 2007b. Collaborative Filtering Recommender Systems. In: Brusilovsky, P., K. Alfred and N. Wolfgang (Eds.), The Adaptive Web of Lecture Notes in Computer Science. Springer-Verlag, Berlin, Germany, 4321: 291-324.

Schafer, J.B., J. Konstan and J. Riedl, 1999. Recommender systems in E-commerce. Proceedings of the 1st ACM Conference on Electronic Commerce. Denver, CO, pp: 158-166.

Shardanand, U. and P. Maes, 1995. Social information filtering: Algorithms for automating word of mouth. Proceeding of the SIGCHI Conference on Human Factors in Computing Systems. ACM Press/Addison-Wesley Publishing Co., New York, USA, pp: 210-217.

Taghipour, N., A. Kardan and S.S. Ghidary, 2007. Usage-based web recommendations: A reinforcement learning approach. Proceedings of the ACM Conference on Recommender Systems. Minneapolis, MN, USA, pp: 113-120.

Towle, B. and C. Quinn, 2000. Knowledge Based Recommender Systems using Explicit User Models. In Papers from the AAAI Workshop, AAAI Technical Report WS-00-04, AAAI Press, Menlo Park, CA, pp: 74-77.

Ungar, L.H. and D.P. Foster, 1998. Clustering methods for collaborative filtering. Proceedings of the Workshop on Recommender Systems at the 15th National Conference on Artificial Intelligence (AAAI). Madison, Wisconsin, USA, pp: 112-125.

Wei, C.P., C.S. Yang and H.W. Hsiao, 2008. A collaborative filtering-based approach to personalized document clustering. Decision Supp. Syst., 45(3): 413-428.

Yao, Y.Y., 1995. Measuring retrieval effectiveness based on user preference of documents. J. Am. Soc. Inform. Sci., 46: 133-145.

Zhang, Y., J. Callan and T. Minka, 2002. Novelty and redundancy detection in adaptive filtering. Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, pp: 81-88.

Ziegler, C., 2004. Semantic web recommender systems. Proceedings of the EDBT Workshop, pp: 78-89.

Ziegler, C.N., S.M. Mcnee, J.A. Konstan and L. Georg, 2005. Improving recommendation lists through topic diversification. Proceeding of the 14th International World Wide Web Conference (WWW). ACM, Chiba, Japan, pp: 22-32.