A COMPREHENSIVE STUDY OF RECOMMENDER SYSTEMS: PROSPECTS AND CHALLENGES

Adetoba B. Tiwalola., Yekini N. Asafe

1Department of Computer Science, Yaba College of Technology, Lagos Nigeria
2Department of Computer Engineering, Yaba College of Technology, Lagos Nigeria

Abstract - Recommender systems (RSs) automate some of these strategies with the goal of providing affordable, personal, and high-quality recommendations. Recommender Systems are software tools and techniques aimed at providing suggestion to support users in various decision-making processes. Development of recommender systems is a multi-disciplinary effort which involves experts from various fields such as Artificial intelligence (AI), Human Computer Interaction (HCI), Information Technology (IT), Data Mining, Statistics, Adaptive User Interfaces, Decision Support Systems (DSS), Marketing, or Consumer Behaviour. Recommender systems have proven to be valuable means for online users to cope with information overload and various techniques for recommendation algorithms have been proposed and successfully deployed in commercial environments. In this paper, a comprehensive study of recommendation systems and various approaches are provided with their major strengths and limitations thereby providing future research possibilities in recommendation systems.

Keywords: Artificial intelligence (AI), Recommender systems (RSs), Computer Interaction (HCI), decision-making processes, Data Mining,

I. INTRODUCTION

Recommender Systems or Recommendation Systems (RSs) are software tools and techniques aimed at providing suggestion to support users in various decision-making processes such as what items to buy, what music to listen, or what news to read (Ricci, 2012).

In the simplest form, personalized recommendations are offered as ranked lists of items. In performing the ranking, RSs predict what the most suitable products or services are, based on the interest and constraints of the users. In order to complete these computational task, RSs collect from users their preferences (interest), which are either explicitly stated, e.g. ratings for products, or are inferred by interpreting user actions. For instance, a RS may consider the navigation to a particular product page as an implicit sign of preference for the items shown on that page. Personalized recommender systems are used by E-commerce sites to suggest products to their customers. The products can be recommended based on the top sellers of a site, demographics of the customer, or analysis of the past buying behaviour of the customer as a prediction for future buying behaviour, for example eBay (Ricci et. al., 2011). These techniques help the sites spread over the World Wide Web to adapt itself to each customer requirements thus enabling individual personalisation for each customer (Prasad & Kumari, 2012).

Non-personalized recommender systems recommend products to customers based on what other customers have said about the products on average. The recommendations are independent of the customer, so each customer gets the same recommendations. Non-personalized recommender systems are automatic, because they require little customer effort to generate the recommendations and are momentary. These recommendations are completely independent of the particular customer targeted by the recommender system. For example, Amazon.com and Moviefinder.com websites are treated as non-personalized recommender systems (Prasad & Kumari, 2012).

RSs development initiated from a rather simple observation: individuals often rely on recommendations provided by others in making routine or daily decisions (Mahmood & Ricci, 2009). The recommendations were for items that similar users (those with similar tastes) had liked. This approach is termed collaborative-filtering and its rationale is that if the active user agreed in the past with some users, then the other recommendations coming from these similar users should be relevant as well and of interest to the active user (Ricci et al., 2010).

Recommender systems have proven to be valuable means for online users to cope with information overload; and various techniques for recommendation algorithms have been proposed and successfully deployed in commercial environments. Existing recommender systems use collaborative filtering or content-based or hybrid methods that combine both techniques. Several data mining techniques (such as Similarity measures, Sampling, Dimensionality Reduction, Classification, Association-Rule-Mining (ARM) and Clustering) are frequently used for recommendation technology to enhance on-line business.

A. Problem Statement

The explosive growth and variety of information available on the Web and the rapid introduction of new e-business services (buying products, product comparison, auction, etc.) frequently overwhelmed or confused users, leading them to make poor decisions. RSs have proved, in recent years, to be a valuable means for coping with this information overload problem. Although many different approaches to recommender systems have been developed over past years but the interest in this area still remains high due to growing demand on practical applications (real life applications), which are able to provide personalized recommendations and to deal with information overload. These growing demands pose some key challenges to recommender systems and to deal with these problems many advanced techniques are proposed, like content-based collaborative filtering, clustering-based filtering, combining item-based and user-based similarity and many more. Despite all these advances, recommender systems still require improvement and thus becoming a rich research area.

B. Objectives

The objective of this paper is to present an overview of recommendation systems, the various techniques used and also propose future research topics. The content of this paper mainly includes the following:

- To explain recommendation system as a tool in solving information overload problem.
- To provide a comprehensive study on various techniques/approaches used in recommender systems especially the traditional methods with their major strengths and limitations.
- To provide future direction in recommendation systems.

C. Motivation

The interest in RSs has dramatically increased based on the following reasons:

- E-commerce is at its infancy in developing countries and RSs play an important role in this area.
- There are opportunities for researcher to obtain useful information from dedicated conferences and workshop related to the field e.g ACM SIGIR Special Interest Group on Information Retrieval (SIGIR), User Modelling, Adaptation and Personalization (UMAP); and ACM's Special Interest Group on Management of Data (SIGMOD).
- It's an opportunity to develop syllabus for graduates and undergraduate students in to support e-commerce in developing countries.

The above reasons motivate the author of this work.

II. CLASSIFICATION OF RECOMMENDER SYSTEMS

There are various types of recommendation techniques, some are knowledge poor, i.e., they use very simple and basic data, such as user ratings/evaluations for items. Others are much more knowledge dependent, e.g., using ontological descriptions of the users or the items or constraints or social relations and activities of the users. Recommendation systems use a number of different technologies. We can classify these approaches into two: Traditional Recommendation Approach and Modern Recommendation Approach (Wanaskar et. al., 2013). More emphasise will be on the traditional approaches – Collaborative filtering, Content-based, Knowledge-based, Hybrid, Community-based and Demographic-filtering approach.

A. Traditional Recommendation Approach

Collaborative filtering: Collaborative filtering became one of the most researched techniques of recommender systems since this approach was mentioned and described by Paul Resnick and Hal Varian in 1997 (Resnik & Varian, 1997). It is also the most widely used and successful methods for implementing RSs (Oliver & Pujol, 2011). The idea of collaborative filtering is, finding the users in a community that share appreciations (Takacs et. al., 2009). If two users have same or almost same rated items in common, then they have similar tastes. Such users build a group or a so called neighbourhood. A user gets recommendations to choose items that he/she has not rated before, but that were already positively rated by users in his/her neighborhood. Collaborative filtering is widely used in e-commerce. Customers can rate books, songs, movies and then get recommendations regarding those issues in future. Moreover collaborative filtering is utilized in browsing of certain documents e.g. documents among scientific works, articles, and magazines (Orlando et al., 2004).

Figures 1 and 2 show the user-based and item-based methods respectively:

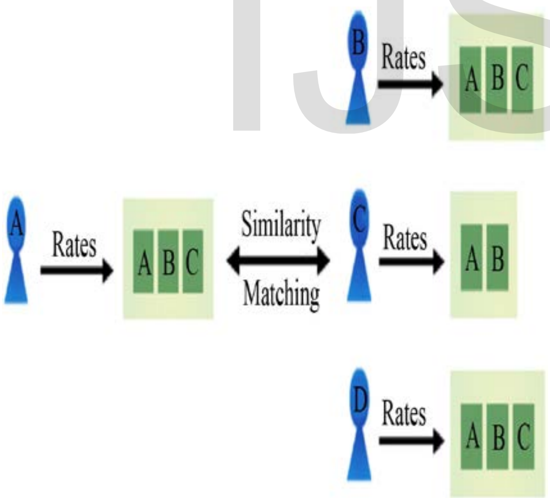

Figure 1: User-based approach

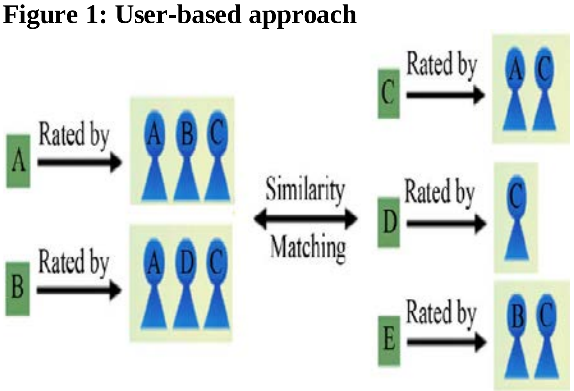

Figure 2: Item-based approach (Wanaskar et al., 2013)

A key advantage of this approach is that it does not rely on machine analyzable content and therefore it is capable of accurately recommending complex items such as movies without requiring any "understanding" of the item itself. Many algorithms have been used in measuring user similarity or item similarity in recommender systems. For instance, the k-nearest neighbour (k-NN) approach (Sarwar et al., 2000) and the Pearson Correlation. When building a model from a user's profile, distinction is often made between implicit and explicit forms of data collection.

Examples of implicit data collection are listed below:

- Observing items a user view in an online store.
- Analyzing item/user viewing times
- Keeping records of items that a user purchases online.

Examples of explicit data collection are listed below:

- Asking users to rate an item on a sliding scale.
- Asking users to rank a collection of items from favourite to least favourite.
- Presenting two items to a user and asking him/her to choose the one that is better.
- Asking user to create a list of items that he/she likes.

One of the most famous examples of collaborative filtering is item-to-item collaborative filtering (people who buy x also buy y), an algorithm popularized by Amazon.com's recommender system. Other examples include:

- Last.fm recommends music based on a comparison of the listening habits of similar users.
- Facebook, MySpace, LinkedIn, and other social networks use collaborative filtering to recommend new friends, groups, and other social connections (by examining the network of connections between a user and their friends) (Ricci et al., 2011).

Collaborative filtering recommender approach can be further divided into two categories (Bobadilla et al., 2012): Memory-based Collaborative Filtering (Neighbourhood based); and Model-based Collaborative Filtering. In memory-based collaborative filtering systems, a set of items or users is generated according to the relevance of user or item. This system works on the ratings whether implicit or explicit. The user-item ratings are stored in the system and helps in generating the list of items or users to be recommended. There are two types of memory-based collaborative filtering approach known as item-based collaborative filtering and user-based collaborative filtering (Adomavicius et. al., 2005; Su & Khoshgoftaar, 2009).

The user-based collaborative filtering approaches evaluate the interest of a user for an item using the ratings given by other users for that item. On the

other hand, item-based collaborative filtering predicts the rating of an item for a user, based on the similar items, liked by the user.
The algorithm can be in this form:

User-based recommendations:

If User A likes Items 1,2,3,4, and 5,
And User B likes Items 1,2,3, and 4
Then User B is quite likely to also like Item 5 (see fig 1)

Item-based recommendations:

If Users who purchase item 1 are also disproportionately likely to purchase item 2
And User A purchased item 1
Then User A will probably be interested in item 2 (see fig 2)

Model-based collaborative filtering systems uses the user item rating stored in the system to learn a predictive model. The basic idea behind this approach is to model the user item interactions with main characteristic and features extraction. The model is then trained by using these data. This model then comes in handy to predict ratings of user for new items. Several different machine learning algorithms are used in this approach like Bayesian Clustering (Breeze, 1998), Probabilistic Latent Semantic Analysis (PLSA) Hofmann, 2003), Latent Dirichlet Allocation (LDA) Blei et al., 2003), Maximum Entropy (Zitnick et al., 2004), Boltzmann Machines (Salakhutginov et al., 2007), Support Vector Machines (SVM) (Grcar et al., 2005), Singular Value Decomposition (SVD) (Paterek, 2007).

Collaborative filtering Recommender systems application: One of most popular applications in the field of collaborative filtering is Amazon's recommendation system. Amazon uses recommendations for marketing campaigns and personal adaptation of its homepage. Customers have the possibility to receive individual suggestion on Amazon products, based on the articles they purchased previously. For the reason that ordinary CF algorithms cannot scale Amazon's massive datasets, they developed their own Item-To-Item Collaborative Filtering method (Sarwar et al., 2000). The main difference to traditional item-based CF techniques is that the Amazon algorithm prunes items which have no common customers or belong to unlike product catalogs. However, in order to provide real time recommendations, the expensive item-similarity matrix is computed offline.

Figure 3 presents the interaction of a user with an online collaborative recommender system through a web interface. In order to suggest products to a user, web server and recommender need to communicate with each other. Usually, the web server application forwards user feedback to the recommender, and receives personalized recommendations in return. User ratings and item correlations are both stored on the recommender platform to ensure real time results.

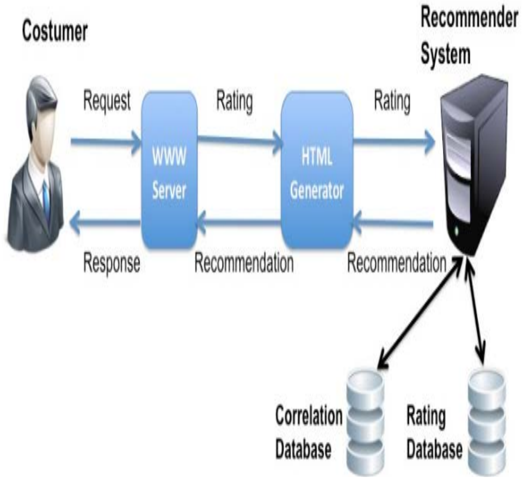

Fig 3: Collaborative Recommender System Architecture (Sarwar et al., 2000)

Advantages of Collaborative filtering Recommender systems: The main advantages of collaborative filtering recommender systems are that they are more effective when it comes to customer satisfaction as they recommend the most appropriate items to users (Koenigstein et al., 2011). The collaborative filtering algorithms are designed such that the accuracy of their prediction increases tremendously over items as more user preferences are added to the database, irrespective of the size of the database (Lopez-Nores et al., 2012).

Disadvantages of Collaborative filtering Recommender systems: Collaborative filtering approaches often suffer from the following problems: cold start, scalability, sparsity and Gray sheep (Lee et al., 2007)

- Cold Start: These systems often require a large amount of existing data on a user in order to make accurate recommendations.
- Scalability: In many of the environments that these systems make recommendations in, there are millions of users and products. Thus, a large amount of computation power is often necessary to calculate recommendations.
- Sparsity: The number of items sold on major e-commerce sites is extremely large. The most active users will only have rated a small subset of the overall database. Thus, even the most popular items have very few ratings.
- Gray sheep problem: The main drawback of this method is that the system would not be very effective when user preferences change unexpectedly, as the system still focuses on past interests of the user. It is also known as gray sheep problem (Lopez-nores et al., 2012. A particular type of collaborative filtering

algorithm uses matrix factorization, a low-rank matrix approximation technique (Sachan & Richariya, 2013).

Content-based filtering: Another common approach when designing recommender systems is content-based filtering (Gunawardana & Shani, 2009). These are based on information about and characteristics of the items that are going to be recommended. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past (or is examining in the present). In particular, various candidate preferences are compared with items previously rated by the user and the best-matching items are recommended. This approach has its roots in information retrieval and information filtering research (Wikipedia, 2013).

Basically, these methods use an item profile (i.e., a set of discrete attributes and features) characterizing the item within the system. The system creates a content-based profile of users based on a weighted vector of item features. The weights denote the importance of each feature to the user and can be computed from individually rated content vectors using a variety of techniques. Simple approaches use the average values of the rated item vector while other sophisticated methods use machine learning techniques such as Bayesian Classifiers, cluster analysis, decision trees, and artificial neural networks in order to estimate the probability that the user is going to like the item. Direct feedback from a user, usually in the form of a like or dislike button, can be used to assign higher or lower weights on the importance of certain attributes (using Rocchio Classification or other similar techniques) (Krishna & Devi, 2012).

A key issue with content-based filtering is whether the system is able to learn user preferences from user's actions regarding one content source and use them across other content types. When the system is limited to recommending content of the same type as the user is already using, the value from the recommendation system is significantly less than when other content types from other services can be recommended. For example, recommending news articles based on browsing of news is useful, but it's much more useful when music, videos, products, discussions etc. from different services can be recommended based on news browsing.

A high level architecture of a content based recommender system is depicted in Figure 4. The recommendation process is performed in three steps, each of which is handled by a separate component:

- Content analyser: When information has no structure (e.g. text), some kind of pre-processing step is needed to extract structured relevant information. The main responsibility of the component is to represent the content of items (e.g. documents, Web pages, news, product descriptions, etc.) coming from information sources in a form suitable for the next processing steps. Data items are analyzed by feature extraction techniques in order to shift item representation from the original information space to the target one (e.g.Web pages represented as keyword vectors). This representation is the input to the profile learner and filtering component;
- Profile Learner: This module collects data representative of the user preferences and tries to generalize this data, in order to construct the user profile. Usually, the generalization strategy is realized through machine learning techniques, which are able to infer a model of user interests starting from items liked or disliked in the past. For instance, the PROFILE LEARNER of a Web page recommender can implement a relevance feedback method (Rocchio, 1971) in which the learning technique combines vectors of positive and negative examples into a prototype vector representing the user profile. Training examples are Web pages on which a positive or negative feedback has been provided by the user;
- Filtering Component: This module exploits the user profile to suggest relevant items by matching the profile representation against that of items to be recommended. The result is a binary or continuous relevance judgment (computed using some similarity metrics (Herlocker et al., 2004), the latter case resulting in a ranked list of potentially interesting items. In the above mentioned example, the matching is realized by computing the cosine similarity between the prototype vector and the item vectors.

The first step of the recommendation process is the one performed by the content analyzer, which usually borrows techniques from Information Retrieval systems (Baeza-Yates & Ribeiro-Neto, 1999). Item descriptions coming from *Information Source* are processed by the content analyzer, that extracts features from unstructured text to produce a structured item representation, stored in the repository *Represented Items*.

In order to construct and update the *profile* of the *active user ua* (user for which recommendations must be provided) her reactions to items are collected in some way and recorded in the repository *Feedback*. These reactions, called *annotations* or *feedback*, together with the related item descriptions, are exploited during the process of learning a model useful to predict the actual relevance of newly presented items. Users can also explicitly define their areas of interest as an initial profile without providing any feedback.

ISSN 2229-5518

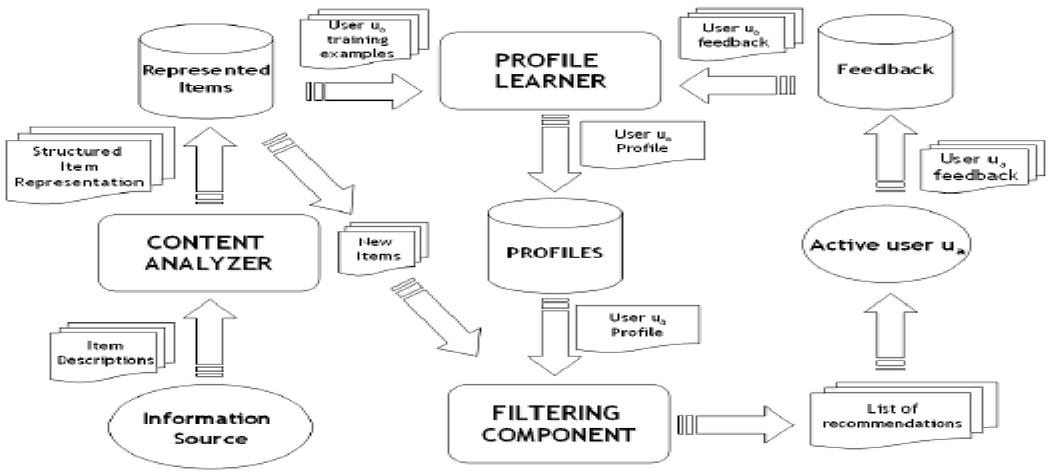

Fig 4: High level architecture of a Content-based Recommender (Ricci et al., 2011)

Content-based Applications: Pandora Radio is a popular example of a content-based recommender system that plays music with similar characteristics to that of a song provided by the user as an initial seed. There are also a large number of content-based recommender systems aimed at providing movie recommendations, a few such examples include Rotten Tomatoes, Internet Movie Database, Jinni, Rovi Corporation.

Advantages of Content-based recommender systems: The main advantage of this method is that it does not depend on the user ratings of items in the database and hence, even if the database does not contain user preferences, the prediction accuracy is not affected. Even if the user preferences change, it has the capacity to adjust its recommendations in a short span of time.

Disadvantages of Content based recommender systems: The main drawback of this approach is the need to know all the details of an item really well, even where the features of the item is stored in the database in a way where it cannot be retrieved directly.

B. Hybrid Recommender Systems

Hybrid approach combines both collaborative filtering and content-based filtering, this can be implemented in many ways: by making collaborative-based and content-based predictions separately and then combining them; by adding collaborative-based capabilities to a content-based approach, and vice versa; or by unifying the approaches into one model. Several studies empirically compare the performance of the hybrid with the pure collaborative and content-based methods and demonstrate that the hybrid methods can provide more accurate recommendations than pure approaches. These methods can also be used to overcome some of the common problems in recommender systems such as cold start and the sparsity problem. A good example of hybrid systems is Netflix which makes recommendations by comparing the watching and searching habits of similar users (i.e. collaborative filtering) as well as by offering movies that share characteristics with films that a user has rated highly (content-based filtering). A hybrid recommender system is one that combines multiple techniques together to achieve some synergy between them. Hybrid recommender system is used to describe any recommender system that combines multiple recommendation techniques together to produce its output. There is no reason why several different techniques of the same type could not be hybridized, for example, two different content-based recommenders could work together, and a number of projects have investigated this type of hybrid: NewsDude, which uses both naive Bayes and kNN classifiers in its news recommendations is just one example (Burke, 2007).

Hybrid recommender systems combine both of the earlier methods to produce better results by involving all the advantages of the two techniques and by removing their drawbacks at the same time (Bobadilla et al., 2012). Burke (2002) introduced taxonomy for the hybrid recommendation systems. He classified them into seven categories, weighted, switching, mixed, feature combination, feature augmentation, cascade, and meta-level.

- Weighted hybrid – This hybrid combines scores from each component using linear formula. Therefore, components must be able to produce its recommendation score which can be linearly combinable. Also, the components have to be consistent relative accuracy across the product space and to perform uniformly.
- Switching hybrid – The issue of this hybrid is selecting one recommender among candidates. This selection is made according to the situation it is experiencing. The criterion for the selection like confidence value or external criteria should exist and the components might

ISSN 2229-5518

have different performance with different situations.

- Mixed hybrid – This is a hybrid which is based on the merging and presentation of multiple ranked lists into one. Each component of this hybrid should be able to produce recommendation lists with ranks and the core algorithm of mixed hybrid merges them into a single ranked list. The issue here is how the new rank scores should be produced.
- Feature combination hybrid – There exist two very different recommendation components for this hybrid, contributing and actual recommender. The actual recommender works with data modified by the contributing one. The contributing one injects features of one source to the source of the other component.
- Feature augmentation hybrid – This is similar to the feature combination hybrids but different in that the contributor generates new features. It is more flexible and adds smaller dimension than feature combination method.
- Cascade hybrid – This is a tie breaker. The secondary recommender is just a tie breaker and does refinements.
- Meta-level hybrid – For this, contributing and actual recommenders exist but the former one completely replaces the data for the latter one, not just part of it.

C. An Hybrid Recommender System Framework

This architecture is based on both collaborative filtering and content-based filtering methods by using multi-based clustering, illustrated in Fig. 5. It includes two major components: an off-line component and an online component. The off-line component is a batch processing unit that runs periodically. It consists of two modules as follows.

- The training module is the main process that generates the predicted data rating by using multi-based clustering method. A dual based group, a similar users based group and a similar items based group is formed as credible information sources. Then, it analyzes each group's influence on the target users from the target items. This approach takes advantage of user correlations and item correlations embedded in the user-item matrix. Hence the new items can be included in the recommendations. The predicted data rating are then generated and stored in database. The predicted data rating is computed by combining the item-based predicted ratings and the user-based predicted ratings to make a high accuracy predicted ratings on those items for users.
- In the clustering module, we cluster the predicted rating matrix to find groups of like-minded users. These groups are small in size compared to the original set, thus making the technique scalable.

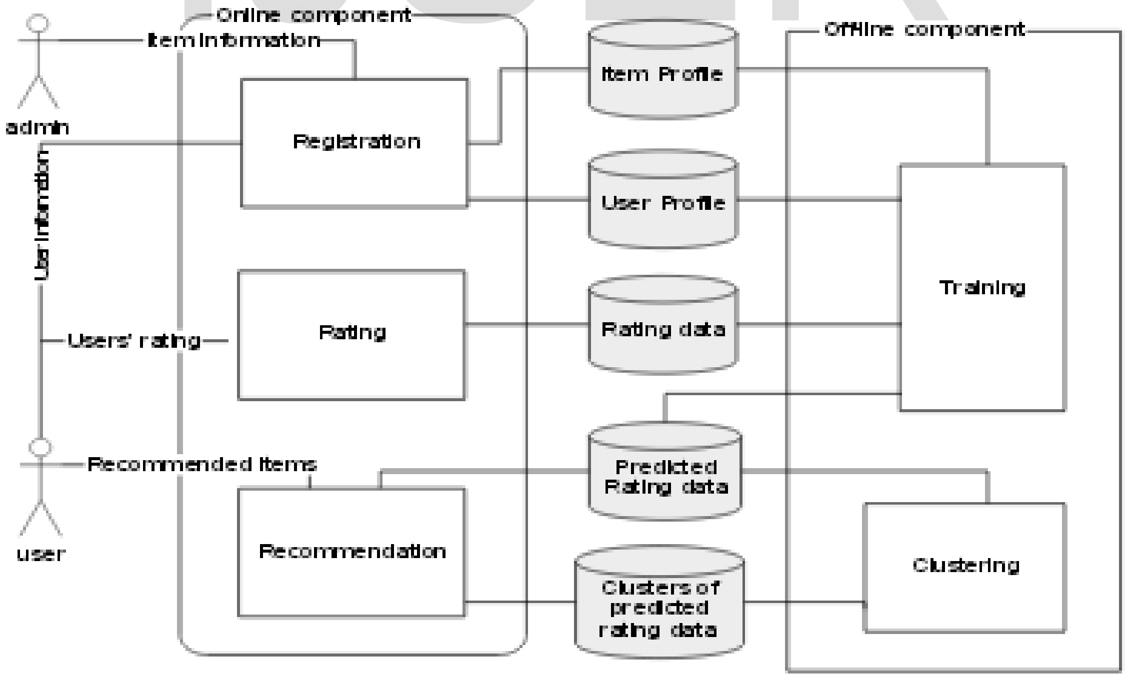

Figure 5: Hybrid Recommender System Architecture (Puntheeranurak & Tsuji, 2009)

D. Demographic-based recommender systems

This system provides recommendations based on a demographic profile of the user. Recommended products can be produced for different demographic niches, by combining the

ratings of users in those niches. Many websites adopt simple and effective personalization solutions based on demographics (Wang & Reinders, 2003; Montainer et. al., 2003; Gemmis et. al, 2009). For example, users are dispatched to particular Websites based on their language or country. Or suggestions may be customized according to the age of the user. Other advantage is that the quality of recommendation improves over the span of time as it builds the profile for user preferences (Gemmis et al., 2009).
The main disadvantage of this approach is that it suffers from cold-start problem because it needs user preferences to recommend better and also it needs huge demographic information. It also suffers from gray-sheep problem (Montainer et al., 2003; Gemmis et. al, 2009).

E. Knowledge-based system

Knowledge-based recommender system integrates the knowledge of users and products to do recommendation (Burke, 2000). It is a recommender system that suggests products based on inferences about a user's needs and preferences. This knowledge will sometimes contain explicit functional knowledge about how certain product features meet user needs (Burke, 2013). This system does not recommend based on generalizing the long-term description of user but it does recommend on the evaluation of users' need and optional set and calculate the preference of the product to user. This approach doesn't suffers from cold-start problem or overspecialization because it is independent of other user, rating and statistical evaluation (Burke, 2000). This is the main advantage of using this approach. However, this approach heavily depends upon user profile. This approach needs three types of knowledge: knowledge about the users, knowledge about the items and knowledge about the matching between the item and user's need and this is one its disadvantage (Shishehchi et. al., 2011). Most knowledge based recommender systems are case based (Mahmood & Ricci, 2007).

F. Community-based recommender system

This is a recommender system where system recommends items, based on preferences of user's friends. It has been observed that people tend to rely more on recommendations from their friends rather than anonymous individuals (Shimon et al., 2007). This observation generates a rising interest in community-based systems; they are usually referred to as social recommender systems (Sinha & Swearingen, 2001).

Table 1: Comparison of Recommender System Techniques Mentioned With Advantages and Disadvantages

Techniques	Advantages/Strengths	Disadvantages/Weaknesses	Author & Year of Publishing
Collaborative Filtering	▪ No domain knowledge required. ▪ Quality of recommendation increases over time.	▪ New-user problem (Cold-start problem). ▪ New-item problem (First rater problem). ▪ Sparsity problem. ▪ Gray sheep problem. ▪ Scalability problem.	▪ Burke, 1999 ▪ Sharma & Gera, 2013. ▪ Wanaskar et. al., 2013. ▪ Khoshgoftaar, 2009. ▪ Su and Khoshgoftaar, 2009
Content based Filtering	▪ -No domain knowledge required. ▪ -Quality of recommendation increases over time. ▪ -New Item recommendation	▪ New-user problem (Cold-start problem). ▪ Limited content analysis problem. ▪ Overspecialization	Hu & Pu, 2011
Hybrid based Filtering	▪ No domain knowledge required. ▪ Quality of recommendation increases over time. ▪ New Item recommendation ▪ No Sparsity problem		Ali & Ghani, 2012
Demographic	▪ No domain knowledge required. ▪ Quality of recommendation increases over time. ▪ No new-item problem.	▪ New-user problem (Cold-start problem). Gray sheep. ▪ Need demographic information	▪ Montaner et. al., 2003. ▪ Gemmis et. al., 2009
Knowledge based Filtering	▪ No cold-start problem. ▪ No overspecialization problem. ▪ No sparsity problem. ▪ Prone to preference changes.	▪ Needs domain knowledge. ▪ Does not learn over time	Burke, 1999. Burke, 2000.
Community based	▪ No domain knowledge required. ▪ Quality of recommendation increases over time. ▪ No new-item problem	▪ New-user problem (Cold-start problem). ▪ Gray sheep. ▪ Need community ▪ Information	Herlocker et. al., 2004 Lopez-Nores et. al., 2009.

G. Modern Recommendation Approaches

Context-based recommender system: The majority of existing approaches to recommender systems focus on recommending the most relevant items to individual users and do not take into consideration any contextual information, such as time, place and the company of other people (e.g., for watching movies or dining out). It is also important to incorporate the contextual information into the recommendation process in order to recommend items to users under certain circumstances. For example, a travel recommender system would provide a vacation recommendation in the winter that can be very different from the one in the summer. Therefore, accurate prediction of consumer preferences depends upon the degree to which the recommender system has incorporated the relevant contextual information into a recommendation method (Ricci et. al., 2010).

Semantic Based Approaches: Most of the descriptions of items, users in recommender systems and the rest of the web are presented in the web in a textual form. Using tags and keywords without any semantic meanings doesn't improve the accuracy of recommendations in all cases, as some keywords may be homonyms. That is why understanding and structuring of text is a very significant part recommendation. Traditional text mining approaches that base on lexical and syntactical analysis show descriptions that can be understood by a user but not a computer or a recommender system. That was a reason of creating new text mining techniques that were based on semantic analysis. Recommender systems with such techniques are called semantic based recommender systems. The performance of semantic recommender systems are based on knowledge based usually defined as a concept diagram (like taxonomy) or ontology (Wang and Kong, 2007).

Cross-Domain Based Approaches: Finding similar users and building an accurate neighbourhood is an important part of recommending process of collaborative recommender systems. Similarities of two users are discovered based on their appreciation of items. But similar appreciations in one domain do not surely mean that in another domain valuations are similar as well (Winoto & Tang, 2008).

Peer-to-Peer Approaches: The recommender systems with P2P approaches are decentralized. Each peer can relate itself to a group of other peers with same interests and get recommendations from the users of that group. Recommendations can also be given based on the history of a peer. Decentralization of recommender system can solve the scalability problem (Shavitt et al., 2010).

Cross-lingual Approaches: The recommender system based on cross-lingual approach lets the users receive recommendations to the items that have descriptions in languages they don't speak and understand. Yang, Chen and Wu purposed an approach for a cross lingual news group recommendation. The main idea is to map both text and keywords in different languages into a single feature space, that is to say a probability distribution over latent topics. From the descriptions of items the system parses keywords than translates them in one defined language using dictionaries. After that, using collaborative or other filtering, the system gives recommendations to users (Yang et al., 2008)

III. EXISTING WORK

There have been a lot of researches in the area of recommender systems. Several recommender systems have been developed so far according to different recommendation techniques discussed above. These recommender systems are related to various fields of applications, such as news, music, e-commerce, movies, etc. Each domain presents different problems that require different solutions. Table 2 illustrates 10 different domains where Recommendation applications exist.

TABLE 2: Recommender Systems with Different Domains (Sharma & Suman, 2011)

Domain	Risk	Churn	Heterogeneous	Preferences	Interaction Style	Scrutiny	Examples	Recommendation Technology
E-commerce	Low	High	High	Stable	Implicit	Not required	Amazon.com, eBay	Collaborative Filtering
Financial services and Life insurance	High	Low	Low	Stable	Explicit	Required	Koba4MS (Felfernig, 2005)) FSAdviser (Felfernig & Kiener, 2005)	Knowledge Based
Job Search Recruiting	High	Low	Low	Stable	Explicit	Required	CASPER(Lee, 2009) and (Keim et. al., 2006)	Content based
Movie	Low	Low	Low	Stable	Implicit	Not required	Netflix (Paterek, 2007) INTIMATE (Bollacker et. al., 98) Movies2Go (Mak et. al.,2003)	Collaborative and Content based
Music	Low	Low	Low	Stable	Implicit	Not required	Pandora and (Hayes, 2000)	Content based Hybrid
News	Low	High	Low	Stable	Implicit	Not required	Yahoo News (Billus &Pazzani, 2006) ACR news (Mobasher et al, 2000) and Google news (DAS et. al, 2007) INFOrmer NewsDude	Content based Collaborative – Filtering
Real Estate	High		Low	Stable	Explicit	Required	RentMe (Burke, 2000) FlatFinder (Viappiani, 2007	Knowledge based
Scientific Research papers	Low	Low	Low	Stable	Explicit Implicit	Not required	QuickStep system (Middleton, 2002) Citeseer (Boone, 1998)	Content based
Software Engineering	Low	Low	Low	Stable	Explicit Implicit	Required	Castro-Herrera et. al., 2008	Hybrid and Content based
Tourism	High	Low	Low	Unstable	Explicit	Required	Travel Recommender (Ricci, 2002)	Content based Knowledge based
TV Program	Low	Low	Low	Unstable	Implicit Explicit	Not required	AVTAR (Blanco-Fernandez et. al., 2008)	Content based
Web Page Recommender	Low	High	High	Unstable	Implicit	Not required	Zaiane et. al., 2004 Letna (Letizia, 1995)	Collaborative Filtering Hybrid

Recommendation domains can be distinguished by the degree of risk that a user incurs in accepting a recommendation. Risk determines the user's tolerance for false positives among the recommendations. In some domains, false negatives may also be important if there is a cost or risk associated with not considering some options. High-risk domains are generally considered under knowledge-based recommendation.
A high churn domain is one in which items come and go rapidly. In such a domain, a recommender system faces a continual stream of new items to be integrated into its knowledge sources. This greatly increases the sparsity of any kind of opinion data, as new items will necessarily have been seen by very few users.

User preferences can also have varying degrees of duration. For example, when one's favourite basketball team is in a big tournament, stories about it become highly preferred, but if they are knocked out or when the tournament is over, the user's preferences will change.
Scrutiny is also a good predictor of knowledge-based recommendation. Heterogeneous domains are handled largely with collaborative recommendation. Webpage recommendation looks a bit contradictory when we consider high churn and preference instability, which would seem to militate against collaborative methods. However, database size can compensate for preference instability and these recommenders collect large amounts of implicit preference data in each session. It has been observed that the recommender systems with social knowledge, requires high heterogeneity.

In the area of e-commerce famous literature includes both content-based and collaborative approaches. In content-based approach, the idea is to detect items that are most "similar" to the user's existing profile. A user profile is composed of his/her previous transaction history, such as what he/she viewed or purchased before. After the user profile is set up, how to determine similarity between an item and the profile is the key challenge. Various approaches have been developed (Zhang & Zhang, 2010), such as cosine similarity, Bayesian classifiers, clustering, etc.

IV. APPLICATIONS OF RECOMMENDER SYSTEMS

Recommender systems applications are mostly common in the following areas

- Entertainment: recommendations for movies, music, and IPTV.
  Ringo is an online music recommendation system which uses social information filtering to make personalized recommendation. Using correlation algorithm (Pearson Correlation technique), Ringo compares the user profiles to determine whether the users have similar taste in music and it computes the weighted average of the ratings for that particular artist, song or album using the information in them (Shardananda & Maes, 1995).
- Content: personalized newspapers, recommendation for documents, recommendations of Web pages, e-learning applications, and e-mail filters.
- E-commerce: recommendations for consumers of products to buy, such as books, cameras, PCs etc. Learning Intelligent Book Recommending Agent (LIBRA) used by Amazon, makes use of the database which holds the information of all the books extracted from the web pages at Amazon (Burke, 2002). The actual texts of the items are not used in this system. Amazon.com uses topic diversification algorithms to improve recommendations (Ziegler et al., 2005).
- Services: recommendations of travel services, recommendation of experts for consultation, recommendation of houses to rent, or matchmaking services.

V. Recommender Systems Algorithms

In this paper we will consider the three popular algorithms used in recommendation approaches and these are: Pearson Correlation, Cosine Similarity and Item-to-Item Similarity.
Pearson Correlation is used to compare the linear dependence between two variables (Almazro et al., 2010). In context recommender systems, it is used to compare the ratings of the items which are rated by a reader or a user (u) and the number of neighbours (n).

$$w(u,n) = \frac{\sum_{i \in I}(r_{u,i} - r_u)(r_{n,i} - r_n)}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{i \in I}(r_{n,i} - \bar{r}_n)^2}}$$

(1)

Where, I is the set of items rated by both users; $r_{n,i}$ is the rating given to item i by user n.
The predicted rating for u over item j, where j has been rated by both u and n can be calculated as follows:

$$p_{u,j} = \bar{r}_u + \frac{\sum_{u \in K} w(u,i)(r_{i,j} - \bar{r}_i)}{\sum_{u \in K} |w(u,i)|} \quad (2)$$

Where $P_{u,j}$ is the prediction for active user u for item j, $w(u,i)$ is the similarity between users u and i, and K is the neighbourhood or set of similar users.

Cosine Similarity: The cosine similarity or vector similarity (Gunawardana, 2009) is used to measure similarity using the cosine angle formed by the frequency vectors (for example NewsWeeder). In memory based collaborative filtering algorithms, the similarity between two documents are

ISSN 2229-5518

measured using this technique. The document is considered as a vector of frequencies of words and the cosine angle between the two vectors of frequencies are calculated to determine similarity. In the calculation of cosine similarity, only positive ratings are considered and negative ratings are rejected. The formula is shown below:

$$w(u,i) = \sum_{x \in I_{u,i}} \frac{v_{u,x}}{\sqrt{\sum_{y \in I_u} v_{u,k}^2}} \frac{v_{i,x}}{\sqrt{\sum_{y \in I_i} v_{i,k}^2}} \quad \textbf{(3)}$$

Where $I_{u.i}$ is the set of items which both the users rated positively. I_i is the item which is rated positively by the user i. Only positive ratings are considered for computing the cosine similarity.
Using this, the predicted score for a user is computed as below:

$$I_{u,i} = K \left| \sum_{x=1}^{n} w(u,x) c_{x,y} \right. \quad \textbf{(4)}$$

Item to Item Similarity: This measure is used to compute the similarity between items (Linden et al., 2003; Jojic et al., 2011). Most of the times, the maximum likelihood estimate (MLE) is used to calculate conditional probability of items. The conditional probability of items in binary usage can be calculated as below:

$$P(i_1 \mid i_2) = \frac{|I_{i_1,i_2}|}{|I_{i_2}|} \quad \textbf{(5)}$$

Where, I_x represents the number of users who had used item x, and the number of users who had rated both i_1 and i_2 is represented by $I_{i1,i2}$

Table 3: Summary of recommender systems and their algorithms (Ali & Ghani, 2012)

Recommender System	Type	Algorithm	Rating Scale
GroupLens	Hybrid	Correlation	1 to 5
Ringo	Collaborative	Correlation	1 to 7
LIBRA	Hybrid	Bayesian Learning Algorithm	1 to 10
NewsWeeder	Content based	Cosine Similarity	1 to 5
Book Crossing	Collaborative	Topic Diversification	1 to 10
Jester 2.0	Collaborative	Cluster	Real-Valued

GroupLens (Adomavicious & Tuzhilin, 2005) is an architecture used for distributing ratings, which can be modified by anybody who wants to alter or improve a news client to use predicted scores or to make a news client to allow entry of evaluations. It was introduced to mine the reactions of people who read news articles and the architecture was successful in scaling up to a very large number of users and ratings. The system employed collaborative filtering methods on vast number of news articles (or content) present in Netnews in order to help users to find useful articles in a much easier way.
Jester 2.0 (Gupta et al., 2011) is a web based recommender system for jokes, which is used to predict the jokes which might interest the user.
NewsWeeder: This is a text recommender which uses the words in the texts as features. It is basically a Netnews filtering system which lets the user rate his interest level (from 1 to 5) for each article and learns their user profile based on the information obtained. This eliminates the need to depend on the user to create their profiles altogether (Lang, 1995).

VI. EVALUATION METRICS OF RECOMMENDER SYSTEMS

Several metrics are used to evaluate recommendation algorithms (Herlocker et al., 2004). The quality of a recommender system can be evaluated by comparing recommendations to a test set of known user ratings. These systems are typically measured using *predictive accuracy metrics*, where the predicted ratings are directly compared to actual user ratings. The most commonly used metric in the literature is *Mean Absolute Error* (MAE) which is defined as the average absolute difference between predicted ratings and actual ratings, given by:

$$MAE = \frac{\sum_{\{u,i\}} |p_{u,i} - r_{u,i}|}{N} \quad (6)$$

Where $P_{u,i}$ is the predicted rating for user u on item i, $r_{u,i}$ is the actual rating, and N is the total number of ratings in the test set.

A related commonly-used metric, *Root Mean Squared Error* (RMSE), puts more emphasis on larger absolute errors, and is given by:

$$RMSE = \sqrt{\frac{\sum_{\{u,i\}} (p_{u,i} - r_{u,i})^2}{N}} \quad \textbf{(7)}$$

Predictive accuracy metrics treat all items equally. However, for most recommender systems we are primarily concerned with accurately predicting the items a user will like. As such, researchers often view recommending as predicting *good*, i.e. items with high ratings versus *bad* or poorly-rated items. In the context of Information Retrieval (IR), identifying the good from the background of bad items can be viewed as discriminating between “relevant” and “irrelevant” items; and as such, standard IR measures, like *Precision*, *Recall* and *Area Under the ROC (Receiver operating characteristics) Curve (AUC)* (Bamber, 1975) *or precision recall* can be utilized. These, and several other measures, such as *F1-measure*, *Pearson's product-moment correlation*, *Kendall's* τ, *mean average precision*, *half-life utility*, and *normalized distance-based performance measure* are discussed in more detail by Herlocker et al. (2004).

VII. CHALLENGES OF EXISTING RSs

Various techniques used in a recommender system experiences some of the hurdles that may be described in terms of basic problems such as:

Cold-Start: Cold start problem refers to the situation when a new user or item just enters the system. There are three kinds of cold start problems: new user problem, new item problem and new system problem (Sharma & Mann, 2013). In such cases, it is really very difficult to provide recommendation as in case of new user, there is very less information about user that is available and also for a new item, no ratings are usually available and thus collaborative filtering cannot make useful recommendations as in the case where we have new item as well as new user. However, content-based methods can provide recommendations if there is a new item as they do not depend on any previous rating information of other users. These problems can be solved using the hybrid approach.

Scalability: Scalability is the property of a system which indicates its ability to handle growing amount of information in a graceful manner. With enormous growth in information over internet, it is obvious that the recommender systems are having an explosion of data and thus it is a great challenge to handle this continuously growing demand. Some of the recommender system algorithms deal with the computations which increase with growing number of users and items. In CF computations grow exponentially and get expensive, sometimes leading to inaccurate results. Methods proposed for handling this scalability problem and speeding up recommendation formulation are based on approximation mechanisms. Even if they improve performance, most of the time they result in accuracy reduction (Papagelis et al., 2005).

Over Specialization Problem: Users are restricted to getting recommendations which resemble those already known or defined in their profiles in some cases, and it is termed as over specialization problem (Chen et al., 2011). It prevents user from discovering new items and other available options. However, diversity of recommendations is a desirable feature of all recommendation systems. After solving the problem using genetic algorithms, user will be provided with a set of different and a wide range of alternatives.

Sparsity: Sparsity problem is one of the major problems encountered by recommender systems; and data sparsity has great influence on the quality of recommendation. Generally, data of system like MovieLens is represented in form of user-item matrix populated by ratings given to movies and as number of users and items increase the matrix dimensions and sparsity evolves. The main reason behind data sparsity is that most users do not rate most of the items and the available ratings are usually sparse. Collaborative filtering suffers from this problem because it is dependent on the rating matrix in most cases. Many researchers have attempted to reduce this problem; still this area demands more research (Sharma & Gera, 2013).

Trust: The voices of people with a short history may not be that relevant as the voices of those who have rich history in their profiles. The issue of trust arises towards evaluations of a certain customer. The problem could be solved by distribution of priorities to the users (Herlocker et al., 2000; Bonhard et al., 2006; Cramer et al., 2008; Pu and Chen, 2006).

Privacy: Privacy has been the most important problem. In order to receive the most accurate and correct recommendation, the system must acquire the most amount of information possible about the user, including demographic data, and data about the location of a particular user. Naturally, the question of reliability, security and confidentiality of the given information arises. Many online shops offer effective protection of privacy of the users by utilizing specialized algorithms and programs (Wanaskar et al., 2013).

Fraud: As Recommender Systems are being increasingly adopted by commercial websites, they have started to play a significant role in affecting the profitability of sellers. This has led to many unscrupulous vendors engaging in different forms of fraud to game recommender systems for their benefit. Typically, they attempt to inflate the perceived desirability of their own products (push attacks) or lower the ratings of their competitors (nuke attacks). These types of attack have been broadly studied as shilling attacks (Shyong et al., 2004), or profile injection attacks (Burke et al., 2005). Such attacks usually involve setting up dummy profiles, and assume different amounts of knowledge about the system. For instance, the average attack (Shyong et. al., 2004) assumes knowledge of the average rating for each item; and the attacker assigns values randomly distributed around this average, along with a high rating for the item being pushed. Studies have shown that such attacks can be quite detrimental to predicted ratings, though item-based Collaborative Filtering tends to be more robust to these attacks (Lam & Reidl, 2004). Obviously, content-based methods, which only rely on a user's past ratings, are unaffected by profile injection attacks. While pure content-based methods avoid some of the pitfalls discussed above, Collaborative Filtering still has some key advantages over them.

Firstly, CF can perform in domains where there is not much content associated with items, or where the content is difficult for a computer to analyze, such as ideas, opinions, etc.

Secondly, a CF system has the ability to provide serendipitous recommendations, i.e. it can recommend items that are relevant to the user, but do not contain content from the user's profile.

VIII. CONCLUSION

Despite all the advancements made recommendation systems, RSs still require further improvements in its recommendation methods or development of new algorithms in the following real-life applications: Recommending vacations in area of tourism (i.e. Tourism recommender systems); Recommending suitable candidate for jobs to benefit both the employee and employers (Job recommender systems); Recommending certain types of financial services to investors, for example stock recommendation; Recommending products to purchase in an online store; Better methods for representing user behaviour and the information about the items to be recommended; More advanced recommendation modelling methods through incorporation of various contextual information into the recommendation process; Utilization of multi criteria ratings; Development of less intrusive and more flexible recommendation methods that also rely on the measures that more effectively determine performance of recommender systems.

Also, new application areas for recommender systems emerge with the popularity of the Social Web. This Social Web therefore provides huge opportunities for recommender technology and in turn recommender technologies can play a part in fuelling the success of the Social Web phenomenon.

Finally, privacy-protection considerations are also a challenge. Recommender algorithms can identify patterns individuals might not even know exist. A recent example is the case of a large company that could calculate a pregnancy-prediction score based on purchasing habits. Through the use of targeted ads, a father was surprised to learn that his teenage daughter was pregnant. The company's predictor was so accurate that it could predict a prospective mother's due date based on products she purchased.

REFERENCES

Adomavicius, G., & Tuzhilin, A. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transaction on Knowledge and Data Eng., Vol. 17*, No. 6, pp. 734-749.

Adomavicius, G., & Tuzhilin, A. (2008). *Context-aware recommender systems*. In Proceedings of the 2008 ACM conference on Recommender systems, Lausanne, Switzerland.

Amatriain, X., Pujol, J. M., & Oliver, N. (2011). Data Mining Methods for Recommender
Systems. In F. Ricci et al. (eds.), *Recommender Systems Handbook*,
DOI 10.1007/978-0-387-85820-3_2, © Springer Science+Business Media, LLC

Balabanovíc, M., & Shoham, Y. (1997). *FAB: content-based, collaborative recommendation.*
Commun. *ACM* *40* (3), 66–72. DOI http://doi.acm.org/10.1145/245108.245124

Bhasker, B. &Srikumar, K. (2010). *Recommender Systems in E-Commerce*. CUP.
ISBN 978-0-07-068067-8.

Beel, J., Langer, S., Genzmehr, M., & Gipp, B. (2013). *Research Paper Recommender System Evaluation: A Quantitative Literature Survey*. Proceedings of the Workshop on Reproducibility and Replication in Recommender Systems Evaluation (Rep Sys) at the ACM Recommender System Conference (RecSys).

Beel, J., Langer, S., Nürnberger, A. & Genzmehr, M. (2013). The impact of demographics(Age and Gender) and other user characteristics on Evaluating Recommender systems. In T. Aalberg, M. Dobreva, C. Papatheodorou, G. Tsakonas, and C. Farrugia. *Proceedings of the 17th International Conference on Theory and Practice of Digital Libraries (TPDL 2013).* Springer. pp. 400–404.

Bennett, J., & Lanning, S. (2007). *The Netflix prize*. Proceedings of KDD Cup and Workshop.

Billsus, D., Brunk, C., Evans, C., Gladish, B., & Pazzani, M. (2002). Adaptive Interfaces
for Ubiquitous Web Access. *Comm. ACM, vol. 45*, No. 5, pp. 34-38.

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

Burke, R. (2002). *Hybrid Recommender Systems: Survey and Experiments, User Modeling and User-Adapted Interaction,* **vol. 12**, pp. 331-370.

Burke, R. (2007). *Hybrid web recommender systems*. In *The Adaptive Web*, pp. 377–408. Springer Berlin / Heidelberg.

Burke, R., Mobasher, B., Bhaumik, R., & Williams, C. (2005). Segment-based injection attacks against collaborative filtering recommender systems. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 577–580, Washington, DC, USA. IEEE Computer Society.

Chen, Y., Wu, C., Xie, M., & Guo, X. (2011). Solving the sparcity problem in recommender systems using association retrieval. *Journal of Computers, Vol. 6*, No. 9.

Ghani, I., Lee, C. Y., Jeong, S. R., & Juhn, S. H. (2010). "A Role- Oriented Content-based Filtering Approach: Personalized Enterprise Architecture Management Perspective," *International Journal of Computer Science and Information Security*, ***vol. 8***.

Gunawardana, A., & Meek, C. (2009). *A unified approach to building hybrid recommender systems*. In RecSys '09: Proceedings of the third ACM conference on Recommender systems, pages 117–124, NY, USA: ACM.

Gunawardana, A. and Shani, G. (2009). A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. *Journal of Machine Learning Research*, ***vol. 10***, pp. 2935-2962

Han, E., & Karypis, G. (2005). *Feature-based recommendation system*. In CIKM '05: Proceedings of the 14th ACM international conference on Information and Knowledge management, pages 446–452, New York, NY, USA: ACM.

Herlocker, J. L., Konstan, J. A., Terveen, L., & Riedl, J. T. (2004). Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems (TOIS)*, ***vol. 22***, pp. 5-53.

Hill, W., Stead, L., Rosenstein, M., & Furnas, G. (1995). *Recommending and Evaluating Choices in a Virtual Community of Use*. Proc. Conf. Human Factors in Computing Systems.

Jafarkarimi, H., Sim, A. T., & Saadatdoost, R. (2012). A Naïve Recommendation Model for Large Databases, *International Journal of Information and Education Technology*, vol 2 No.3, 216 – 219.

Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2013). *Recommender Systems: An Introduction*. London: Cambridge University Press.

- Jennings, A., & Higuchi, H. (1992). A personal news service based on a user model neural network. *IEICE Transactions on Information and Systems* ***75*** (2), 198–209.

Koenigstein, N., Dror, G., & Koren, Y. (2011). *Yahoo! Music Recommendations: Modelling Music Ratings with Temporal Dynamics and Item Taxonomy*, in Proceedings of the fifth ACM Conference on Recommender systems RecSys '11, pp. 165-172

Krishna, D., & Devi, K. (2012). Improving accumulated recommendation system a comparative study of diversity using Pagerank algorithm techniques. *IJESAT vol. 2*.

Lam, K. S., & Riedl, J. (2004). *Shilling recommender systems for fun and profit*. In WWW '04: Proceedings of the 13th international conference on World Wide Web, pages 393–402. New York, NY, USA: ACM.

Lang, K. (1995). *NewsWeeder: Learning to Filter Netnews*, in In Proceedings of the 12th International Machine Learning Conference (ML-95).

Lee, H., Park, S. J., & Moners (2007). *A news recommender for the mobile web*. Expert Systems with Applications 32(1), 143–150 (2007).

Linden, G., Smith, B., & York, J. (2003). *Amazon.com Recommendations: Item-to-Item Collaborative Filtering*. IEEE Internet Computing,

Liu, N., & Yang, Q. (2008). *Eigenrank: a ranking-oriented approach to collaborative filtering*. In SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 83–90. New York, USA: ACM.

Lopez-Nores, M., Blanco-Fernandez, F., Pazos-Arias, J. J., & Gil-Solla, A.(2012). Property-based collaborative filtering for health-aware recommender systems. *Expert Systems with Applications*, ***vol. 39***, pp. 7451-7457.

Mahmood, T., & Ricci, F. (2009). *Improving recommender systems with adaptive conversational strategies*. In C. Cattuto, G. Ruffo, & F. Menczer (Eds.), *Hypertext*, pp. 73–82. ACM.

Melville, P., & Sindhwani, V. (2010). *Recommender systems*. Encyclopedia of Machine Learning, Springer.

Miller, B. N. I., Albert, S. K., Lam, J. A., Konstan, J. & Riedl, J. (2003). *MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System*, Proc. Int'l Conf. Intelligent User Interfaces.

Montaner, M., Lopez, B., & Rosa, J. L. D. (2003). A Taxonomy of Recommender Agents on the Internet. *Artificial Intelligence Review*, ***vol. 19***, pp. 285-330.

Mooney, R., & Roy, L. (2000). *Content-Based Book Recommending Using Learning for Text Categorization*, in Proceedings of the fifth ACM Conference on Digital libraries.

Papagelis, M., Rousidis, I., Plexousakis, D., & Theoharopoulos, E. (2005). *Incremental Collaborative Filtering for Highly Scalable Recommendations Algorithms*. ISMIS, LNAI 3488 pp553-561.

Paterek, A. (2007). *Improving regularized singular value decomposition for collaborative filtering*. Proceedings of the KDD Cup and Workshop.

Prasad, R., & Kumari, V. (2012). A Categorical Review of Recommender Systems. *International Journal of Distributed and Parallel Systems (IJDPS) Vol.3*, No.5,

Puntheeranurak S., & Tsuji, H. (2009). An Improved Hybrid Recommender System Using Multi-Based Clustering Method. *IEEJ Transactions on Electronics, Information and Systems, Volume 129*, Issue 1, pp. 125-132.

Ricci, F., Rokach, L., & Shapira, B. (2010). *Introduction to recommender Systems Handbook*-Page 2.

Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (Eds.) (2011). *Recommender Systems Handbook*. Springer.

Ricci, F. (2012). *Context-Aware Music Recommender Systems*. WWW 2012 – AdMIRe'12 Workshop. Lyon, France

Rich, E. (1979). *User modeling via stereotypes*. Cognitive science (Elsevier) **3** (4), 329–354. ISSN 0364-0213

Roy, B., & Yan, X. (2009). *Manipulation-resistant collaborative filtering systems*. In RecSys '09: Proceedings of the third ACM conference on Recommender systems, pp. 165–172. New York, NY, USA: ACM.

Sachan, A., & Richariya, V. (2013). Reduction of Data sparsity in collaborative Filtering based on fuzzy inference rule. *International Journal of Advanced Computer Research*, *3*(10), 101 - 107.

Sarwar, B., Karypis, G., Konstan, J., & Reidl, J. (2000). *Analysis of recommendation algorithms for e-commerce*. In Proceedings of the 2nd ACM conference on Electronic commerce, pages 158–167, New York, USA: ACM.

Sarwar, B., Karypis, G., Konstan, J., & Reidl, J. (2001). *Item-based collaborative filtering recommendation algorithms*. In WWW 01 international conference on World Wide Web, Hong Kong.

Schafer, J. B., Frankowski, D., Herlocker, D., & Sen, S. (2007). Collaborative Filtering Recommender Systems. *Lecture Notes in Computer Science*, ***vol. 4321***, pp. 291-324.

Shani, G., Heckerman, D., & Brafman, R. (2005). An MDP-based recommender system. *Journal of Mach. Learn. Res.*, *6*, 1265–1295.

Sharma, S. K., & Suman, U. (2011). Design and Implementation of Architectural Framework of Recommender System for e-Commerce IRACST. *International Journal of Computer Science and Information Technology & Security (IJCSITS)*, Volume 1, No. 2, ISSN 2249-9555.

Sharma, M., & Mann, S. (2013). A survey of Recommender Systems: Approaches and Limitations. *International Journals of Innovation in Engineering and Technology*, ISSN:2319-1058.

Sinha, R., & Swearingen, K. (2001). *Comparing recommendations made by online systems and friends*. In DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries.

Wade, W. (2003). *A Grocery Cart that Holds Bread, Butter, and Preferences*. New York: Times.

Wang, J., de Vries, A. P., & Reinders, M. J. T. (2006). *Unifying user-based and item-based Collaborative filtering approaches by similarity fusion*. pp. 501–508. New York, USA: ACM Press, DOI http://doi.acm.org/10.1145/1148170.1148257

Wang, J., & Reinders, T. (2003). *Music Recommender System for Wi-Fil Walkman," ICT* Group Technical Report ICT-2003-01.

Wang P. Y.; and Yang,H. C. (2012). Using Collaborative Filtering to Support College Students' Use of Online Forum for English Learning, *Computers and Education*, ***vol. 59***, pp. 628-637.

Wanaskar, U. H., Vij, S. R., & Mukhopadhyay, D. (2013). *A Hybrid Web Recommendation System based on the Improved Association Rule Mining Algorithm.* Department of Computer Engineering, Pune, India.

Wu, P., Liu, W., & Jin, C. (2010). *A novel recommender system fusing the opinions from experts and ordinary people*. In Proceedings of the Workshop on Context-Aware Movie Recommendation (CAMRa '10). pp. 41–44. New York, NY: ACM.

Zhang L., & Zhang, Y. (2010). *Discriminative factored prior models for personalized Content based recommendation*. In Proceedings of the 19th ACM international conference on Information and knowledge management, pages 1569–1572. ACM.