A Literature Review on Recommender Systems Algorithms, Techniques and Evaluations

Kasra Madadipouya
Asia Pacific University of Technology & Innovation, Kuala Lumpur, Malaysia
Kasra_mp@live.com

Sivananthan Chelliah
Asia Pacific University of Technology & Innovation, Kuala Lumpur, Malaysia
Sivananthan@apiit.edu.my

Abstract

One of the most crucial issues, nowadays, is to provide personalized services to each individual based on their preferences. To achieve this goal, recommender system could be utilized as a tool to help the users in decision-making process offering different items and options. They are utilized to predict and recommend relevant items to end users. In this case an item could be anything such as a document, a location, a movie, an article or even a user (friend suggestion). The main objective of the recommender systems is to suggest items which have great potential to be liked by users. In modern recommender systems, various methods are combined together with the aim of extracting patterns in available datasets. Combination of different algorithms make prediction more convoluted since various parameters should be taken into account in providing recommendations.

Recommendations could be personalized or non-personalized. In non-personalized type, selection of the items for a user is based on the number of the times that an item has been visited in the past by other users. However, in the personalized type, the main objective is to provide the best items to the user based on her taste and preferences. Although, in many domains recommender systems gained significant improvements and provide better services for users, it still requires further research to improve accuracy of recommendations in many aspects. In fact, the current available recommender systems are far from the ideal model of the recommender system.

This paper reviews state of art in recommender systems algorithms and techniques which is necessary to identify the gaps and improvement areas. In addition to that, we provide possible solutions to overcome shortages and known issues of recommender systems as well as discussing about recommender systems evaluation methods and metrics in details.

Keywords: Recommender Systems, Collaborative Filtering, Content-based Filtering, Recommendation, Evaluation Metrics

1. Introduction

In the last few years WWW (World Wide Web) has faced with enormous growth in information and number of online users, in particular with the creation of Web 2.0. In the first version of web (web 1.0) users have accessed to databases of information and knowledge; however, they were limited in terms of contribution and information accessibility. On the other hand, in the second generation of WWW which is known as Web 2.0, the Internet faced significant improvements in user interactions and behaviors. Web 2.0 can be characterized as a paradigm that facilitates communication, interoperability, user-centered design, and information sharing and collaboration on the Web (O'Reilly, 2005; Sharma, 2008; Madadipouya, 2013). Furthermore, in this generation we see an obvious shift from solitary and local to worldwide collaboration and contribution. Moreover, improvements of Web 2.0 have changed the way of creating and accessing information. By contrast of Web 1.0, Web 2.0 information management and accessibility are distributed and done by users' collaborations instead of accessing, creating, and managing information on a specific computer or browser. For instance, Wikipedia is an obvious example of this collaboration which everybody can access to its content and able to edit or even add new materials to it. Other examples are videos and pictures sharing sites, blogs, and more importantly social networks which are gaining daily popularity.

However, the growth of Web 2.0 that facilitates ease of access to the vast amount of data at anytime, anywhere resulted in a new issue called, information overload which makes finding appropriate information tedious. As a result, users are facing difficulty of understanding an issue and effectively making decisions when one has too much information about that issue.

In order to overcome with aforementioned challenge and assist users to select suitable option among many possibilities, online recommender systems have come to existence to provide a technological proxy (Chen, 2011), to determine if a user would like a specific item via making prediction, or recommending top items to the user based on her preferences and analyzing the user behavior.

Recommender Systems (RS) can be widely applied in different areas (Jannach, 2010). For instance, they are used in e-commerce websites, online auctions, online music stores, social networks, and media streaming platforms. Among various range of recommender systems, people mostly use recommender systems in social network, e-commerce, and online streaming sites, where in social networks as an instance, a user receives friend recommendations based on people who has in her friend list (mutual friend). In e-commerce website, also people use recommender systems extensively in order to get product suggestions and information which may are useful and helping them to make better purchase decision. Suggested items could be from top seller list or based on the analysis of the past purchases (Kim et al., 2005).

Recommender systems differ with Information Retrieval (IR) in many areas, although both focus on providing better information to users. It can be said that also recommender systems established based on IR and IF.

Additionally, IR has differences with RS in contextual and practical situations. Each is used for different purposes and tasks. These differences could be classified in three main categories. One associated with objects, other related to users' issues and the last one considers on general environment. Hence, it can be seen that RS fundamentally is the same as IR system according to Baeza-Yates and RibeiroNeto (1999) in terms of goals and purposes, however, the operation methods are dissimilar.

In this work, we review state of art in recommender systems algorithms and techniques which is necessary to identify the gaps and improvement areas. In addition to that, we provide possible solutions to overcome shortages and known issues of recommender systems as well as discussing about recommender systems evaluation methods and metrics in details.

The remainder of this paper is organized as follows: Section 2 provides an in-depth review of recommender system types. Section 3 presents the shortcoming of the current recommender systems. In section 4 we provided some solutions to address the issues highlighted in previous section. Then we reviewed different approaches on how to evaluation recommender systems.

Finally, in the last section we draw some conclusions and provide recommendations for further research on this topic.

2. Types of Recommender Systems

Two types of recommender systems are widely adapted which are collaborative filtering and content-based filtering respectively. In collaborative filtering, that is also referred as social filtering (Shardanand & Maes, 1995), items are selected based on the correlation between the current (active) user and other users of the system (Su & Khoshgoftaar, 2009). However, in content-based filtering items are recommended based on the correlation between items and the user preferences (Adomavicius et al., 2005). The user interests are firstly analyzed and the result of the user profile analysis compared with available items on the system to provide recommendations to the user. It can be said that in collaborative filtering similarity between users are considered in the recommendation process, whereas, in content-based filtering the focus is on similarity between the user and items (Bogers & Van Den Bosch, 2009). In addition to mentioned techniques there are also different hybrid approaches which developed by combination of collaborative filtering and content-based filtering in order to overcome with certain limitations that each technique has. The following subsections discuss each technique in details.

2.1. Content-based Filtering

Due to information overloading in the web which is explained before, different approaches provided to tackle with this issue. One common method that used to automatically categorize, filter, and provide recommendations to users is content-based filtering (Ferman et al., 2002). This method compares the available items with a user's profile and item rated previously to find the best match to generate recommendations. This technique is different from IF and IR methods. For instance, information retrieval allows users to specify their interest explicitly in form of keywords. This means that users retrieve their needed information, however, content-based filtering is based on finding suitable information automatically. In this technique, at first, data about the target user is gathered. This data could be acquired explicitly and implicitly with doing analysis on the user profile. Then based on the analysis the user's tastes identified and as a result a set of items will be recommended to the users. Although, IR has significant differences with recommender systems, however, content-based algorithms are driven from IR (Jannach, 2010).

In content-based filtering available items, rated items and users' preferences are shown as vectors (Yih, 2009). CF uses this technique by letting each user profile be represented by a vector, and then compare user similarities by interpreting the vectors.

In addition, some IR techniques can be utilized in content-based filtering such as *Boolean search indexes* (Arnold & Voss, 2004). In this technique keywords are combined with Boolean operators as a part of the recommendation process (Cummins, 2008). Probabilistic retrieval systems are also another information retrieval technique that is used in content-based filtering. In technique, probabilistic reasoning is employed for getting the probability of a document in order to know whether the document is met the user's needs (Lee & Lee, 2005). Natural language query is another method that looks for queries in natural sentences (Siddiqui & Tiwary, 2008).

Many different recommender system applications leverage on content-based filtering to provide recommendations to users. For instance, StubleUpon is a recommender system that assists users in web browsing (Trivedi et al., 2010). In this system, the user behavior is tracked and based on the history and interaction appropriate web pages are recommended to the user. Content-based filtering is also widely used in music domain. Last.Fm (Petersen & Hansen, 2011) is a successful example of content-based recommender system in the music industry. In Last.Fm, predictions are given to users based on previous items that they rated over time.

Google News is a successful news recommender system which is used both collaborative and content-based filtering approaches to provide recommendations. In this system, the user clicks behavior is modeled in order to identify the likelihood interest of articles by the user (Liu, Dolan & Pedersen, 2010).

All the mentioned examples have one feature in common. They are functioning on textual contents or textual metadata or meta-description. Textual information could be easily parsed and categorized automatically with available techniques. However, for other types of media such as audios, videos and pictures, still there is no unified technique to categorize them automatically without human intervention. Most of the time these types of media are categorized manually which time and resource consuming (Massa & Avesani, 2007). Due to this reason content-based recommender system is not suitable for large dynamic environments which keep vast various amount of multimedia information. However, if information could be categorized without direct human involvement the problem could be mitigated to some extends.

2.2. Collaborative Filtering

Collaborative filtering (CF) functions in a different way from content-based filtering. In collaborative filtering, instead of comparing user interests with available items, the user is compared with the other users and similar users are found (Adomavicius & Tuzhilin, 2005). Then in the next step based on the similar users, peer users, recommendations are provided for the active user.

Additionally, in collaborative approach all users are involved in rating based on their interests. In this way, similar users who have the same tastes are linked together (put in the same group/community). Therefore, it can be said that collaborative filtering does not have content-based filtering limitation since it does not depend on the contents; as a result, filtering information from any sources is possible. Moreover, using collaborative filtering makes complex and deep relationships between users and items, such as the needed quality or taste of a certain user. This feature makes it possible to make difference between poor document and well written document by contrast of pure content-based approach (Ricci, Rokach & Shapira, 2011). Lastly, collaborative filtering returns more accurate results since it utilizes real users ranking instead of pure machine made predictions. For instance, in music recommender area, a user who has listened to several poor-quality jazz songs might conclude that jazz music is not interesting. Meanwhile, another user that also disliked those poor jazz songs such as the first user may find a new interesting jazz song and rates a new jazz song high rate. In this situation, since both users have similar taste, the system recommends the interesting jazz song to the first user. Then the first user might discover that jazz genre is not that much bad as he previously considered. Finally, CF helps to create communities, as explained before which none of mentioned facts would be possible using content-based filtering.

It is vital to know that collaborative filtering does not compete with content-based approach. Both of these technologies are mostly combined together in order to make better and more accurate system. This mixture made a new type of recommender system which is called as hybrid recommender systems. Some successful systems that used hybrid approach are Pandora (Petersen & Hansen, 2011), Facebook, and MovieLens (Devi & Venkatesh, 2009). Hybrid approach is discussed in the next section in details.

2.2.1. Collaborative Filtering Techniques

Collaborative filtering could be implemented in the forms of memory-based (usually user-based) or model-based.

In memory-based approach the entire user database is kept in memory and the entire database is traversed in each operation. In this approach recommendations are more accurate but when system database is very large, this approach is almost impractical due to existed limitation in primary memory for keeping the database. In addition, when database is big traversing the entire database could be time consuming.

Model-based approach on the other hand does not have the aforementioned limitations of memory-based. In this method, instead of keeping the entire database in memory just specific collection of data which are already trained using machine learning methods are kept in memory.

Although, in some extends model-based approach has similar related limitations to memory-based, this approach is more efficient and feasible to implement in real world scenarios. Both approaches are addressed in details with their strengths and weaknesses in next sections.

2.2.1.1. Memory-based Collaborative Filtering

A memory-based CF (nearest-neighbor) approach, mostly called as a form of implementation of the “Word of Mouth” phenomenon (Jin, Chai & Si, 2004) since the entire user database with their preferences are kept in memory. For each prediction computation is performed on the whole database. This method could predict a user interests on a specific item based on the rating information of similar user profiles. It reflects where the prediction of a specific item (belonging to a specific user) is done by sorting the row vectors (user profiles) by its dissimilarity toward the user. In this method, more rating by more similar users leads to more rating prediction.

Various types of memory-based recommender systems have been developed. Decker and Lenz (2007) stated that Goldberg on 1992 developed certain type of memory-based CF system which is called Tapestry. This system relies on each user to identify similar users manually.

Ringo and GroupLens also developed separately developed the systems which in prediction task were automated for the first time. In GroupLens method Pearson correlation coefficient was utilized to provide automatic predictions (Nguyen & Haddawy, 1998), however, in Ringo project the main concern was testing various metrics for finding similarity between users such as correlation and means squared.

In a work by Madadipouya (2015a) a new collaborative filtering proposed to take into account users' location. The proposed method has been implemented using modified Pearson Correlation and applied in a movie dataset. The end result demonstrated some enhancement over the baseline Pearson method (Madadipouya, 2015b).

Jannach (2010) proposed a similarity vector based on the cosine measure to find the similarity between the users. This approach mostly is used in information retrieval systems. Apart from developments which have been done by researchers, some commercial websites also have developed their own version of memory-based collaborative filtering. The most successful and significance examples are Amazoon and CDNow.

Several equations and formulas could be used in the memory-based approach in order to find the similarity between two users which are called mostly as similarity index function. One of the most used and the best of them is Pearson Correlation Coefficient which is used to find similarity between two users based on the rated items of the user (Jannach, 2010). In addition to that this method could be used to find missed values in the database. Equation (1) demonstrates Pearson Correlation Coefficient.

$$sim(a,b) = \frac{\sum_{p\in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p\in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p\in P}(r_{b,p} - \bar{r}_b)^2}} \qquad (1)$$

In the equation, we use $U = \{u_1, ..., u_n\}$ to denote the set of users, $P = \{p_1, \ldots, p_m\}$ for the set of products (items), and R as an $n \times m$ matrix of ratings $r_{i,j}$, with $i \in\{1\ldots\text{m}\}$. The equation (1) returns a real number between to -1. If two users strongly have similar tastes to each other the output of the equation is closer to 1 whereas if their tastes are dissimilar the output be closer to -1.

With respect to the determination of the set of similar users, one common measure used in recommender systems is Pearson's correlation coefficient. The similarity *sim(a,b)* of users *a* and *b*, given the rating matrix *R*, is defined in above equation. The symbol r_x corresponds to the average rating of user *x* which in the above equation is users *a*, *b*.

In addition to Pearson Coefficient Correlation other similarity index is also could be used in order to find the similarity between two users. One of them is Adjusted Cosine Similarity. The metric measures the similarity between two *n–dimensional* vectors based on the angle between them. This measure is also commonly used in the fields of information retrieval and text mining to compare two text documents, in which documents are represented as vectors of terms (Jannach, 2010). The similarity between two items *a* and *b* – viewed as the corresponding rating vectors $\vec{a}$ and $\vec{b}$ – is formally defined as follow, equation (2).

$$sim(\vec{a},\vec{b}) = \frac{\vec{a}\cdot\vec{b}}{|\vec{a}|*|\vec{b}|} \qquad (2)$$

The $\cdot$ symbol is the dot product of vectors, $|\vec{a}|$ is the Euclidian length of the vector that defines as the square root of the dot product of the vector.

Although many others similarity indexes are existed to evaluate the similarity between users in the recommender system, according to Herlocker, et al. (2004), empirical analyses show that for user-based recommender systems – and at least for the best studied recommendation domains – the Pearson coefficient outperforms other measures.

Memory-based recommender systems are very popular and reached to high level of using since it is simple and intuitive on a conceptual level while avoiding the complications of a potentially expensive model-building stage. However, it has its own issues which are mentioned as following (Hofmann, 2004; Sarwar et al., 2000a):

- **Sparsity:** In practice, memory-based approach ought to be used to evaluate large item sets. In such systems, even a very active user barely could reach to visit or rate 1 percent of all available items. Therefore, in some extends memory-based method is unable to give any recommendations to a particular user or the recommendation results would be poor.
- **Scalability:** Most of memory-based recommender systems suffer from scalability problem since growing number of users and items cause tremendous growth in computations which sometimes could impose huge load. As a result, in systems which have millions of users and items scalability of the system could cause serious issues.
- **Learning:** In memory-based approach, systems do not learn about the user profile, hence it is unable to the previous interactions or observations in the following operations to improve the recommendations quality.

The weaknesses of memory-based CF systems, especially the learning and scalability problems led to the exploration of an alternative, model-based CF, approach.

2.2.1.2. Model-based Collaborative Filtering

The motivation of model-based collaborative filtering arose when memory-based approach faced with problems. In model-based approach, a model that reflects user preferences is created. This task is accomplished, firstly, by compiling the entire dataset into a descriptive model of users, ratings and items. The model could be built offline in order to reduce time and avoid computation load. The procedure of compiling a model might take up to several hours or even days depend of the dataset size. After creating the model, recommendations could be provided based on the model. Recommendations are provided usually based on the similarity between items instead of users. Prediction is provided to a user by calculating the average of similar items that the user rated (Sarwar et al., 2000b; Sarwar et al., 2001; Linden, Smith & York, 2003; Deshpande & Karypis, 2004). However, the sorting process is the same as memory-based collaborative filtering, which is based on differences between items. One possible distinctive point could be that in memory-based method, the column of vectors is sorted toward specific user, however, in model-based CF, this sort is done toward the specific item. Sorting grantees that ratings from highly similar items are received stronger and better weight.

The size of created model is considerably smaller than the actual database size. This model then could be kept in memory which boosts the performance of the recommendation system as well. Early research on this approach evaluated two probabilistic models, Bayesian clustering and Bayesian networks (Ricci, Rokach & Shapira, 2011). In the first method, similar users are clustered into one group or class. Given the user's class membership, the ratings are assumed to be independent. The number of classes and model parameters could be different in each dataset and the values are not constant. However, in Bayesian network, every node in the network refers to an item in the database. Each node state reflects the possible rate of the item. Both the structures of the network, which encodes the dependencies between items, and the conditional probabilities, are learned from the dataset.

In some context clustering is known as a natural preprocessing for collaborative filtering Rongfei, Maozhong & Chao (2010). In preprocessing step items and users are clustered and they are classified in groups based on the similarities with other items or users. In every class of users, estimation about each class of items is made. Estimations are made with the use of different probability, techniques and statistical which used to compare estimations with synthetic and real data.

Additionally, some works have done on rule-based approach for doing model-based collaborative filtering (Lin, Alvarez & Ruiz, 2002; Abel et al., 2008; Najafabadi et al., 2017). In this approach, some association rules are applied in order to figure out the relationship between co-purchased items. Then the system provides recommendation, based on items relationships (Sarwar et al., 2000a).

As discussed earlier, memory-based approaches have shortcomings such as scalability and data sparsity. In addition to the mentioned techniques, dimensionality reduction methods can be applied to overcome scalability and data sparsity issues in particular (Wang, Vries & Reinders, 2004). Another method by Huang, Chen & Zeng (2004) tries to solve data sparsity problem with utilizing transitive association among items and users. Additionally, some graph-based methods are explored for solving data sparsity. However, in dimensionality reduction approaches such as SVD (Sarwar et al., 2000b) some useful information could be discarded due to the matrix reduction operation (Xue et al., 2005) and as a result the accuracy of recommendations might be decreased.

In order to have more effective recommender system both model-based and memory-based approaches could also be combined together (Xue et al., 2005). In a work by Wang, Vries and Reinders (2006) a framework suggested to include recommendation results of model-based approach into final prediction which eliminates need of data clustering.

In comparison between model-based and memory-based approaches, it can be concluded that model-based approach has some advantages over the other method which discussed as follow:
First of all, model-based CF might add extra values to predictive abilities. This could be done, with highlighting specific correlations in data. Second of all, the amount of memory which is needed for this approach are considerably less than the memory-based approach. Lastly, the prediction could be made quicker since the model size is smaller and it is also kept in memory, however, on the other hand compiling the model may take long time like several hours or days based on the database size which can be done offline. Additionally, adding the new items or users require the trained model to be created again.

Model-based approach could be suitable for environments which preference or item changes occur gradually such as book or even car recommender systems which new items are not added hourly or even daily. However, for very dynamic environments that items are changing rapidly, on daily or hourly basis, model-based approach does not perform effectively since creating or updating models is a time-consuming process. For instance, news items are constantly changing and have a very short lifespan, usually between few hours to few days, as a result having a model-based RS for such a case requires updating the models very frequently, few times in a day, which imposes heavy computational loads as constructing models usually takes between few hours to few days.

2.3. Hybrid Approach

Many commercial and non-commercial systems are utilized hybrid approach which is made by combining both content-based and collaborative methods. This could be very effective way to cross the limitations that each collaborative and content-based method faces with (Adomavicius and

Tuzhilin, 2005). Four main approaches could be created by combining collaborative filtering and content-based filtering which are described as below.

2.3.1. Combining Results Separately

In this approach, each method functions separately but at the final recommendation stage, results of individual method are combined together to deliver to users. In other word, the best results or recommendations of individual system are mixed together for formation of final recommendations (Li & Kim, 2003).

2.3.2. Adding Content-based Filtering Characteristics to Collaborative Filtering

The main focus of this approach is to combine some features of content-based approach into collaborative. In this method, content-based profiles and items are used to calculate similarity between two users. The goal of using this is to overcome data sparsity issue to some extends and minimizing its side effects which could be resulted in providing poor recommendations to users (Pazzani, 1999). Another advantage of using this method is that, the given recommendations are not only rated highly by similar users but also it is scores highly against the active user profile.

2.3.3. Adding Collaborative Filtering Characteristics to Content-based Filtering

In this approach, some collaborative features are combined into content-based approach by contrast of the previous approach. For instance, in this method, a collaborative view of relation between users could be created which users' profiles are illustrated by vectors term (Li and Kim, 2003). This improves the performance of content-based filtering.

2.3.4. Developing a Single Unified Recommender System

In this method, a new unify model is created which utilizes both methods characteristics. Gunawardana and Meek (2009) proposed a new recommender system based on using content-based and collaborative characteristics. In their work, for instance the age, gender of users or the genre of movies are used in a single rule-based classifier.

3. Recommender Systems Shortcomings

Regardless of techniques to use, recommender systems have some shortcomings. In this section, we discuss the issues that are common in RS.

3.1. Cold Start

Recommender systems might suffer from the cold start problem (Siddiqui & Tiwary, 2008). To generate recommendations reasonable amount of information are needed. Therefore, there are some situations that lack of data causes RS to not make recommendations or the generated recommendations are poor. Cold start problem could be occurring due to adding new item, adding new user or launching a new system.

- **New user:** When a user signs up for an account for the first time, the system is unable to suggest personalized recommendations to the user, since; the system has not had any information about the user. Additionally, the user has not rated any items yet. This means that the system is unable to provide accurate recommendation to user. The new user problem exists in both content-based and collaborative approaches. In order to build the user profile and produce coherent results, there should be enough user feedback, which is generally the ratings that are given to the items (Lika, Kolomvatsos & Hadjiefthymiades, 2014). Hybrid, approach could overcome, the cold start issue to some extends. For instance, when a user signs up to a system for the first time, the system can ask about the user interests explicitly

- **New item:** Adding new item is similar to new user problem which causes RS to not make recommendations based on the newly added items. This is due to lack of enough rating for the items or insufficient user reviews and feedback. This problem is more obvious in collaborative filtering techniques, specifically, those rely on item to item approach for making recommendations. Therefore, when a new item is added there is no data about that item (Lika, Kolomvatsos & Hadjiefthymiades, 2014). For instance, in movie domain when a new movie is added to the system, there is no rating available to demonstrate the movie quality or popularity. As a result, the movie could not be recommended to users until sufficient numbers of ratings are gathered. To mollify the impact partially items metadata can be used. When a new item is added the metadata of the item could be leveraged to give recommendations or comparing the item with other items.
- **New system:** New system problem is the synthesis of the new user and new item problems which occurs clearly when a system has just been constructed.

3.2. Data Sparsity

Data sparsity is a crucial problem in recommender systems. Data sparsity plays an important role in recommendation systems. In a work by Ricci, Rokach & Shapira (2011), data sparsity problem is discussed in collaborative filtering approach. In the work, it is concluded that the data sparsity negatively affects in the recommendations which is provided by collaborative filtering.

For instance, in newspaper domain, some recent news is just rated by few people. In such condition, even if the news is highly important or crucial for people and have high rating only by few people, the chances of being recommended to other users is very slim (Sarwar et al., 2001).

Amazon Company put a lot of effort to remove data sparsity problem. In a work by Linden, Smith & York (2003), a new method is suggested to improve similarity matrices under sparsity data with using gathered data from Movie-lens. Various experiments are also done regarding sparsity problem and they conducted that Random Walk recommender algorithm has better performance in comparison with the other approaches when the dataset is sparse.

Additionally, experiments have proven that using matrix factorization techniques such as SVD could reduce the side effects of data sparsity (Sarwar et al., 2001).

3.3. Lack of Sufficient Context-awareness

Even though some context such as location, time, season, date, and so on are taken into consideration to some extends, there are still various factors such as user emotion, mood, and other parameters should be catered since they influence user decisions in reality. User demand might change with the aforementioned context which cannot be satisfied thoroughly via available state of the art recommender systems.

3.4. Over-specialization

Another issue of recommender system is over-specialization especially in content-based filtering. In such systems, the main objective is to recommend items that highly match with user preferences, however, this could lead to suggest items that the user already visited. For instance, in news recommender system that is established base on content-based filtering, a user may receive recommended news that she has already read and not different ones that the user may like (Massa & Avesani, 2007). This problem as stated before happens in pure content-based filtering system (Melville, Mooney & Nagarajan, 2002).

The common solution for this problem is to add some random items in recommendation list.

In a work by Park, Yoo and Cho (2006), authors proposed a method that uses genetic algorithm to reduce the chance of recommending very similar items. Beside this, Outside-The-Box (OTB) recommendation by Abbassi et al. (2009) reflects that taking some risks are helpful to cope with over-specialization issue.

As over-specialization problem is related with content-based filtering, collaborative filtering techniques can be used in order to eliminate the problem. However, Resnick et al. (1994) deals with over-specialization problem by presenting a personalization strategy without making use of collaborative filtering approaches. In the system, a different reasoning mechanism is used which offers semantically related items, instead of using semantic approaches and finally, the obtained system is used for recommending TV-programs.

4. Solution to Improve Recommender Systems

To provide high quality recommender system several approaches could be utilized. This section introduces some methods for improving recommender systems capabilities.

4.1. Intrusiveness

Based on users' opinions and interests' recommendations are provided to them. Their opinions could be gathered through items that rated or explicitly asking preferences. In other word, gathered information could be divided into two intrusive (explicit) or non-intrusive (implicit).

In an explicit way, a user idea is reflected by expressing her interests directly via rating which is normally saved in a binary form (like or dislike) or in numerical scale. In the binary form the user just can indicate whether she likes or dislikes certain items, whereas, in numerical scale the user has more options and freedom for expressing the degree of interest. One example of intrusive binary rating is a famous video-sharing website Youtube which allows registered users to reflect their feelings, by clicking on thumbs up or thumbs down. Amazon by contrast of Youtube utilizes intrusive numerical rating which scale of one (bad) to five (good) are used to rate items.

Implicit rating is done via analyzing user behavior selections to predict rating or provide recommendations. The user preferences could be gathered based on browsing data in web applications, purchasing history or other forms. However, this approach seems often inaccurate and unable to make accurate predictions. In fact, non-intrusive method could not be used as a replacement of explicit method.

Intrusiveness could overcome cold start as well as data sparsity problems. To tackle the former, a recommender system could enforce certain obligations including requiring users to explicitly specifying their preferences or rate certain items on sign up before be able to actually using the system. However, this may not work for item-based recommender systems. One possible solution is to assign certain users as referees to rate and review items upon adding to the system.

Similar approaches as stated above could be applied with some enhancements to minimize the side effects of data sparsity. First, a crowd sourcing mechanism can be established to the system where redeemable reward points will be given to loyal users to rate or provide reviews of determined items. Second, the average of ratings can be calculated for items that barely received rates and this average be used to reduce user-item matrix sparsity. The latter method is less accurate since the average does not demonstrate the accurate rating for all users.

4.2. Richer Context-awareness

Providing accurate recommendation is the vital task of recommender systems. Poor and incorrect recommendations cause user disappointment, which affects user trust. Several recommendation systems exist that consider about fixed user preferences and they are not able to provide accurate recommendations when user preferences are changed based on the context.

With the advancement of technologies contextual data has become one of the most valuable knowledge sources to improve recommendations and provide more user specific recommendations under similar circumstances that are related with similar user preferences in future (Adomavicius &Tuzhilin, 2015; Baltrunas et al., 2012; Gorgoglione, Panniello & Tuzhilin, 2011).

Context in recommender systems could be any form of information that is utilized to portray an entity situation (Adomavicius et al., 2011). The entity could be defined as location of a user, identity of people near the user, the objects around, and the changes in these elements which considered for the interaction between a user and an application, including the user and applications themselves (Wang, 2011).

For instance, LBS technologies can be leveraged to track user location and activities (Madadipouya, 2014) to significantly boost quality of recommendations. In the most basic form, this can overcome a classic recommender system problem where a user located in a city, let's say Chicago, seeking for restaurants recommendations may receive suggestions that are located in different city, let's in Seattle (Levandoski et al., 2012).

In addition to that, context can assist to infer latent relationships between data in recommender systems. For instance, by considering user location as the simplest form of context, Levandoski et al. (2012) proved that most of users located in the same or near geographical location have similar tastes to each other.

This demonstrates that usefulness of a specific recommendation might depend on its context. However, context is not limited to location only. In fact, it is just a sub-category of context-aware recommender systems (CARS) which most of available CARSs leverage on and other contextual parameters are partially neglected (Meehan et al., 2012).

As a result, taking into consideration other contextual parameters (mood, emotion, stereotypes, etc.) in effective ways could improve the quality of recommendations significantly.

4.3. Manual Modeling of Content Similarity

Metadata could be added to item manually via content providers, or by allowing users to add related information when using recommender systems. For instance, in movie domain metadata such as genre and other features could be provided and added to a movie by its producer or this information could be provided by users to set a movie properties. In a movie recommender system by Garden and Dudek (2006) this facility was provided to allow user to even set new keywords for movies which resulted in providing more accurate recommendations based on features set by users.

The big advantage of this solution is that it adapts changes regarding what the users find important, something which can be changed over time.

As mentioned earlier, manually categorization of content can be expensive, time-consuming, error-prone and highly subjective. Due to this, many systems aim to automate the procedure.

4.4. Computational modeling of Multimedia Content Similarity

Many works have been conducted to automate multimedia content similarity modeling in recommender systems. As stated earlier content labeling for the multimedia items are usually done manually, however, having some mechanisms to automate multimedia items metadata labeling can be highly beneficial. Compare with manual labeling this approach is significantly faster and is less resource intensive.

Some research has been conducted for instance, in music domain with the aim of automating classification procedure based on genre and artist. In 2004, a world-wide cross-validation of music similarity systems was conducted, in the form of a public competition during the International Symposium on Music Information Retrieval (ISMIR). The result of this competition reflected that it is possible to classify 729 music tracks into 6 various genres with 78.8% accuracy. They were also able to identify artists from a collection of 120 music titles out of a list of 40 artists with 24% accuracy. MusicSurfer is another example of content-based recommender system which fetches meta descriptions (song descriptions) such as harmony, instrumentation and rhythm from music signals automatically. Cano, Koppenberger & Wack (2005) also developed a car music recommender system which is able to classify wide range of songs. In that system, some musical features could be extracted automatically from songs and music without any prior information or data. Then the extracted information could be saved in a server with music. In addition to that, a user could listen to song based on moods, and the system is able to provide personalized recommendations based on previously selected songs.

5. Evaluation of Recommender Systems

Evaluation of recommender systems is a vital part to benchmark performance of a proposed approach. Usually coverage and accuracy metrics are utilized to evaluate performance of a recommender system (Wu, He & Yang, 2012).

In coverage metric, percentage of items that a recommender system could make prediction is considered (Ge, Delgado-Battenfeld & Jannach, 2010). However, in accuracy metric, effectiveness of the system could be reflected in either statistical or decision support (Wu, He & Yang, 2012).

The most known techniques to evaluate coverage and accuracy are Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE).

Decision support by these metrics shows how well a recommender system is able to generate recommendations which could be highly rated by a user. For instance, recall and precision measures are included which precision refers to percentage of receiving high rates among other predictions that supposed to get high rate and recall demonstrates percentage of correctly predicted high ratings among all the ratings known to be high (Wu He et al., 2012).

All mentioned measures have certain issues to some extends. One of the problem is that users just only intend to rate those items that they select to rate. Therefore, a set of rated items is more likely to give incorrect view of the users' preferences since, the users just rated items that they like not those which they do not like. Thus, the metrics just reflect the accuracy of the items which user attempted to rate, though, the metrics unable to properly evaluate a random item that has not been rated.

The other problem that most of the evaluation metrics have is “quality” evaluation and recommendations “usefulness”. This means that most of the metrics do not consider recommendations quality or degree of usefulness.

6. Conclusion

This work provided a broad overview of available recommender systems techniques with their pros and cons. In addition to that shortcomings of each technique have been discussed and some suggestions have been made which can be leveraged to improve quality of recommendations.

Finally, methods to evaluate recommender systems have been discussed.

Recommender systems are closely related to information filtering with the idea of having personalized decision guides for users. There are mainly three algorithmic techniques for computing recommendations. Content-based filtering selects items to recommend based on the correlation between contents and a user profile. Collaborative filtering chooses items based on the correlation between users with similar preferences. CF are often classified as memory-based or model-based, which means that rating predictions are based on the entire collection of previously rated items, or on a model that reflects previously rated items respectively. In addition, third type of recommender system is hybrid filtering approach which tries to avoid certain limitations related to content-based and collaborative filtering by combining both approaches to cover each other disadvantages.

Current recommender systems have shortcomings such as cold start, sparsity, and lack of enough context awareness. These issues could be partially addressed by taking applying some role enforcement mechanism, crowd sourcing, and taking further contextual parameters into account.

These are the open areas in recommender systems.

Various available evaluation metrics can also be applied to evaluate performance of recommender systems by measuring their coverage and accuracy, but the current metrics are insufficient for evaluating quality and usefulness of recommender systems.

References

Abbassi, Z., Amer-Yahia, S., Lakshmanan, L. V., Vassilvitskii, S., & Yu, C. (2009, October). Getting recommender systems to think outside the box. In *Proceedings of the third ACM conference on Recommender systems* (pp. 285-288). ACM.

Abel, F., Bittencourt, I. I., Henze, N., Krause, D., & Vassileva, J. (2008, July). A rule-based recommender system for online discussion forums. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems* (pp. 12-21). Springer Berlin Heidelberg.

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, *17*(6), 734-749.

Adomavicius, G., & Tuzhilin, A. (2015). Context-aware recommender systems. In Recommender systems handbook (pp. 191-226). Springer US.

Adomavicius, G., Mobasher, M., Ricci, F., & Tuzhilin, A. (2011). Context-aware recommender systems. In AI Magazine, AI Access Foundation, 32(3), 67-80.

Adomavicius, G., Sankaranarayanan, R., Sen, S., & Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems (TOIS)*, *23*(1), 103-145.

Arnold, J. F., & Voss, L. L. (2004). U.S. Patent No. 6,745,161. Washington, DC: U.S. Patent and Trademark Office.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval* (Vol. 463). New York: ACM press.

Baltrunas, L., Ludwig, B., Peer, S., & Ricci, F. (2012). Context relevance assessment and exploitation in mobile recommender systems. Personal and Ubiquitous Computing, 16(5), 507-526.

Bogers, T., & Van den Bosch, A. (2009). Collaborative and content-based filtering for item recommendation on social bookmarking websites. *Submitted to CIKM*, *9*.

Cano, P., Koppenberger, M., & Wack, N. (2005, November). Content-based music audio recommendation. In *Proceedings of the 13th annual ACM international conference on Multimedia* (pp. 211-212). ACM.

Chen, Y. (2011, October). Interface and interaction design for group and social recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems* (pp. 363-366). ACM.

Cummins, R. (2008). *The evolution and analysis of term-weighting schemes in information retrieval* (Doctoral dissertation, National University of Ireland, Galway).

Decker, R., & Lenz, H. J. (Eds.). (2007). *Advances in Data Analysis: Proceedings of the 30th Annual Conference of The Gesellschaft Für Klassifikation EV, Freie Universität Berlin, March 8-10, 2006*. Springer Science & Business Media.

Deshpande, M., & Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, *22*(1), 143-177.

Devi, M. K., & Venkatesh, P. (2009, December). An improved collaborative recommender system. In *Networks and Communications, 2009. NETCOM'09. First International Conference on* (pp. 386-391). IEEE.

Ferman, A. M., Errico, J. H., Beek, P. V., & Sezan, M. I. (2002, July). Content-based filtering and personalization using structured metadata. In Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries (pp. 393-393). ACM.

Garden, M., & Dudek, G. (2006, July). Mixed collaborative and content-based filtering with user-contributed semantic features. In *AAAI* (Vol. 6, pp. 1307-1312).

Ge, M., Delgado-Battenfeld, C., & Jannach, D. (2010). Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems* (pp. 257-260). ACM.

Gorgoglione, M., Panniello, U., & Tuzhilin, A. (2011, October). The effect of context-aware recommendations on customer purchasing behavior and trust. In Proceedings of the fifth ACM conference on Recommender systems (pp. 85-92). ACM.

Gunawardana, A., & Meek, C. (2009, October). A unified approach to building hybrid recommender systems. In *Proceedings of the third ACM conference on Recommender systems* (pp. 117-124). ACM.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, *22*(1), 5-53.

Hofmann, T. (2004). Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, *22*(1), 89-115.

Huang, Z., Chen, H., & Zeng, D. (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, *22*(1), 116-142.

Jannach, D., Zanker, M., Felfernig, A., & Friedrich, G. (2010). Recommender systems: an introduction. Cambridge University Press.

Jin, R., Chai, J. Y., & Si, L. (2004, July). An automatic weighting scheme for collaborative filtering. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 337-344). ACM.

Kim, Y. S., Yum, B. J., Song, J., & Kim, S. M. (2005). Development of a recommender system based on navigational and behavioral patterns of customers in e-commerce sites. *Expert Systems with Applications*, *28*(2), 381-393.

Lee, C., & Lee, G. G. (2005). Probabilistic information retrieval model for a dependency structured indexing system. *Information processing & management*, *41*(2), 161-175.

Levandoski, J. J., Sarwat, M., Eldawy, A., & Mokbel, M. F. (2012, April). Lars: A location-aware recommender system. In Data Engineering (ICDE), 2012 IEEE 28th International Conference on (pp. 450-461). IEEE.

Li, Q., & Kim, B. M. (2003, July). An approach for combining content-based and collaborative filters. In *Proceedings of the sixth international workshop on Information retrieval with Asian languages-Volume 11* (pp. 17-24). Association for Computational Linguistics.

Lika, B., Kolomvatsos, K., & Hadjiefthymiades, S. (2014). Facing the cold start problem in recommender systems. *Expert Systems with Applications*, *41*(4), 2065-2073.

Lin, W., Alvarez, S. A., & Ruiz, C. (2002). Efficient adaptive-support association rule mining for recommender systems. *Data mining and knowledge discovery*, *6*(1), 83-105.

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, *7*(1), 76-80.

Liu, J., Dolan, P., & Pedersen, E. R. (2010, February). Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces* (pp. 31-40). ACM.

Madadipouya, K. (2013). Survey on how Web 2.0 can facilitate knowledge management. IJCER, 2(6), 693-695.

Madadipouya, K. (2014). An Examination and Report on Potential Methods of Strategic Location-Based Service Applications on Mobile Networks and Devices. arXiv preprint arXiv:1411.4294.

Madadipouya, K. (2015a). A Location-based recommender system framework to improve accuracy in user-based collaborative filtering.

Madadipouya, K. (2015b). A Location-Based Movie Recommender System Using Collaborative Filtering. arXiv preprint arXiv:1508.01696.

Massa, P., & Avesani, P. (2007, October). Trust-aware recommender systems. In Proceedings of the 2007 ACM conference on Recommender systems (pp. 17-24). ACM.

Meehan, K., Lunney, T., Curran, K., & McCaughey, A. (2012). VISIT: Virtual Intelligent System for Informing Tourists. PGNET, Liverpool.

Melville, P., Mooney, R. J., & Nagarajan, R. (2002, July). Content-boosted collaborative filtering for improved recommendations. In *Aaai/iaai* (pp. 187-192).

Najafabadi, M. K., Mahrin, M. N. R., Chuprat, S., & Sarkan, H. M. (2017). Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data. *Computers in Human Behavior*, *67*, 113-128.

Nguyen, H., & Haddawy, P. (1998, July). DIVA: applying decision theory to collaborative filtering. In *Proc. of the AAAI Workshop on Recommender Systems*.

O'reilly, T. (2005). What is web 2.0.

Park, H. S., Yoo, J. O., & Cho, S. B. (2006, September). A context-aware music recommendation system using fuzzy bayesian networks with utility theory. In *International Conference on Fuzzy Systems and Knowledge Discovery* (pp. 970-979). Springer Berlin Heidelberg.

Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artificial intelligence review*, *13*(5-6), 393-408.

Petersen, M. K., & Hansen, L. K. (2011, March). Emotional nodes among lines of lyrics. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on* (pp. 821-826). IEEE.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994, October). GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work* (pp. 175-186). ACM.

Ricci, F., Rokach, L., & Shapira, B. (2011). *Introduction to recommender systems handbook* (pp. 1-35). Springer US.

Rongfei, J., Maozhong, J., & Chao, L. (2010, June). A new clustering method for collaborative filtering. In *Networking and Information Technology (ICNIT), 2010 International Conference on* (pp. 488-492). IEEE.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000a, October). Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM conference on Electronic commerce* (pp. 158-167). ACM.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000b). *Application of dimensionality reduction in recommender system-a case study* (No. TR-00-043). Minnesota Univ Minneapolis Dept of Computer Science.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (pp. 285-295). ACM.

Shardanand, U., & Maes, P. (1995, May). Social information filtering: algorithms for automating "word of mouth". In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 210-217). ACM Press/Addison-Wesley Publishing Co.

Sharma, P. (2008). Core characteristics of web 2.0 services. *Startup Reviews| Tech news| Tech events| Tech tips| Business Technology Tips» Blog Archive*.

Siddiqui, T. J., & Tiwary, U. S. (2008). Utilizing local context for effective information retrieval. *International Journal of Information Technology & Decision Making*, *7*(01), 5-21.

Su, X. & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. Advances in artificial intelligence, 2009, 4.

Takeuchi, Y. & Sugimoto, M. (2006). CityVoyager: an outdoor recommendation system based on user location history. *Ubiquitous intelligence and computing*, 625-636.

Trivedi, A., Rai, P., DuVall, S. L., & Daumé III, H. (2010, October). Exploiting tag and word correlations for improved webpage clustering. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents* (pp. 3-12). ACM.

Wang, J., De Vries, A. P., & Reinders, M. J. (2006, August). Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 501-508). ACM.

Wang, L. (2011, July). Understanding and using contextual information in recommender systems. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval* (pp. 1329-1330). ACM.

Wu, W., He, L., & Yang, J. (2012, November). Evaluating recommender systems. In *Digital Information Management (ICDIM), 2012 Seventh International Conference on* (pp. 56-61). IEEE.

Xue, G. R., Lin, C., Yang, Q., Xi, W., Zeng, H. J., Yu, Y., & Chen, Z. (2005, August). Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 114-121). ACM.

Yih, W. T. (2009, August). Learning term-weighting functions for similarity measures. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2* (pp. 793-802). Association for Computational Linguistics.

Kasra Madadipouya received his M.Sc. degree in software engineering from Asia Pacific University of Technology & Innovation.

His research interests include recommender systems, data mining, sentiment analysis, and machine learning.

Sivananthan Chelliah holds a Masters in Software Engineering. Presently, Sivananthan is a Senior Lecturer at Asia Pacific University of Technology & Innovation for the past 17 years, supervising Masters and Undergraduate research projects and teaching a broad spectrum of Software Engineering Modules. Prior to joining the University, Sivananthan has 22 years of active involvement Software Consultancy and in developing and managing Software projects spanning over a number of industries including mining, housing, manufacturing, hospitality, financial, shipping, software quality management and infrastructure.

Sivananthan