International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 11, November 2012)

A Survey of Recommender Systems Techniques, Challenges and Evaluation Metrics

Tranos Zuva[1], Sunday O. Ojo[2], Seleman M. Ngwira[1] and Keneilwe Zuva[3]

[1]*Department of Computer Systems Engineering, Soshanguve South Campus, South Africa*
[2]*Faculty of Information and Communication Technology, Soshanguve South Campus, South Africa*
[3]*Department of Computer Science, University of Botswana, Gaborone, Botswana*

***Abstract* - Recommender systems are software applications that belong to a class of personalized information filtering technologies that aim to support decision making in large information space. There are various techniques being used to achieve this goal in traditional and mobile recommender systems. The recommender systems techniques are usually classified in four main categories: Collaborative Filtering (CF), Content Based Filtering (CBF), Knowledge Based Filtering (KBF) and Hybrid Filtering (HF). In this paper an overview of these techniques, challenges and evaluation metrics of recommender systems is discussed.**

***Keywords* — Recommender System, Decision Support, Information Filtering, Evaluation Metrics**

I. INTRODUCTION

Recommender systems belong to a class of personalized information filtering technologies that aim to meaningfully suggest which items or products available might be of interest to a particular user [1-2]. These systems make recommendations using three fundamental steps: preferences acquisition (acquiring preferences from the user's input data), recommendation computation (computing recommendations using proper methods) and recommendation presentation (presenting the recommendation to the user) [3]. Based on various techniques used in recommendation computation existing recommendation systems can be classified into four fundamental categories shown in Figure 1, that is, Collaborative Filtering (CF), Content-Based Filtering (CBF), Knowledge-Based filtering (KBF) and Hybrid Filtering (HF). Surveys and reviews give researchers an overview of developments, achievements, challenges, direction and open issues within a given area.

This paper is organized as follows: Collaborative Approach, Content Based Approach, Knowledge- Based Approach, Hybrid Approach, Challenges, Performance Evaluation and Summary.

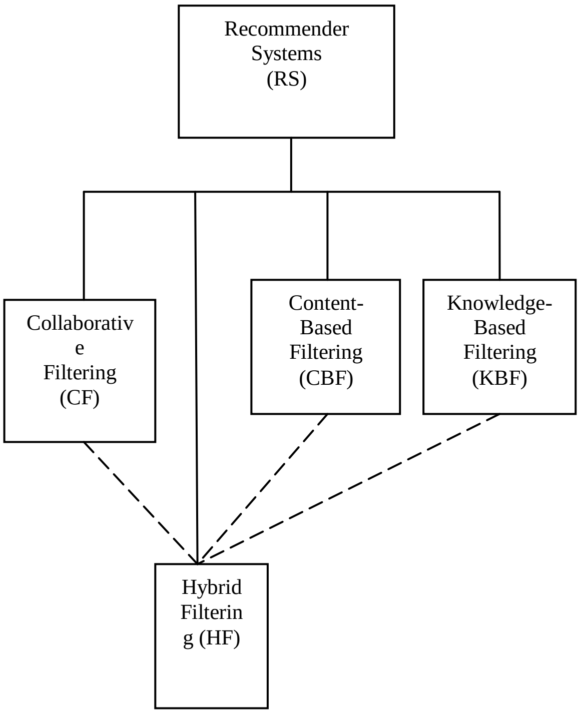
Recommender Systems (RS)
Collaborativ e Filtering (CF)
Content-Based Filtering (CBF)
Knowledge-Based Filtering (KBF)
Hybrid Filterin g (HF)

Figure 1: Classification of Recommender Systems

II. COLLABORATIVE FILTERING (CF)

CF systems obtain user feedback in the form of ratings in a given application domain then exploit similarities and differences among profiles of several users to generate recommendations [4]. Algorithms for CF recommender systems can be grouped into two general classes: memory based (algorithms that require all ratings, items and users be stored in memory) and model based (algorithms that periodically create a summary of ratings patterns offline) [5-6]. Most commonly used are the model based algorithms due to the fact that run-time complexities are reduced.

CF techniques can also be grouped into non-probabilistic and probabilistic algorithms. Probabilistic CF algorithms are those that are based on an underlying probabilistic model. Non-probabilistic CF algorithms are not based on probabilistic model. The non-probabilistic CF algorithms are the most commonly used [5-7]. Nearest neighbour algorithms are well-known CF non-probabilistic algorithms. There are two different classes of nearest neighbour CF algorithms that are User-based nearest neighbour and Item-based nearest neighbour. CF algorithms use a ratings matrix, R, to represent the complete mXn user-item data, m represents the m^{th} user and n^{th} item. Each entry $R_{u,i}$ is the score of item i rated by user u within a certain numerical scale. The matrix is illustrated in table 1 below.

Table 1
User Rating Data Matrix R

	$Item_1$	$Item_2$	$Item_{......}$	$Item_i$	$Item_{.......}$	$Item_n$
$User_1$	$R_{1,1}$	$R_{1,2}$	$R_{1,...}$	$R_{1,i}$	$R_{1,...}$	$R_{1,n}$
$User_2$	$R_{2,1}$	$R_{2,2}$	$R_{2,...}$	$R_{2,i}$	$R_{2,...}$	$R_{2,n}$
$User_{......}$	$R_{...,1}$	$R_{....,2}$	$R_{.....,...}$	$R_{...,i}$	$R_{.....,...}$	$R_{...,n}$
$User_u$	$R_{u,1}$	$R_{u,2}$	$R_{u,...}$	$R_{u,i}$	$R_{u,...}$	$R_{u,n}$
$User_{....}$	$R_{...,1}$	$R_{...,2}$	$R_{.....,...}$	$R_{...,i}$	$R_{.....,...}$	$R_{.....,n}$
$User_m$	$R_{m,1}$	$R_{m,2}$	$R_{m,...}$	$R_{m,i}$	$R_{m,...}$	$R_{m,n}$

This section will discuss the user-based nearest neighbour and item-based nearest neighbour algorithms then the practical challenges of CF algorithms in general.

A User-based Nearest Neighbour

In the user-based neighbour collaborative filtering recommendation systems, the prediction of likeness of an item for an active user u is based on ratings from similar users. These users are called neighbours of u. User-based algorithms generate a prediction for an item i by analyzing ratings for i from users in the u's neighbourhood. Suppose we have a user-item rating matrix R_{m*n}, which means m is the number of all users n is the number of all items and $R_{u,i}$ is the score of item i rated by user u, showing the user's degree of preference for item as in table 1. The most significant step in user-base neighbour CF algorithm is to search the neighbour of the target user u_t. To be able to find the neighbour of the target user u_t, similarity algorithm is used.

There are two most used to compute similarity methods: cosine similarity and Pearson correlation coefficient similarity. The formula for Pearson is given in equations (1).

$$Usersim(u_t,u) = \frac{\sum_{i\varepsilon I_{u,u_t}} \left(R_{u,i} - \bar{R}_u\right)\left(R_{u_t,i} - \bar{R}_{u_t}\right)}{\sqrt{\sum_{i\varepsilon I_{u,u_t}} \left(R_{u,i} - \bar{R}_u\right)^2} \sqrt{\sum_{i\varepsilon I_{u,u_t}} \left(R_{u_t,i} - \bar{R}_{u_t}\right)^2}}$$

(1)

Where $Usersim(u,u_t)$ represent the similarity between user $u\ and\ u_t$, $I_{uu_t} = I(u) \cap I(u_t)$ means the item set rated simultaneously by user $u\ and\ u_t$, $R_{u,i}\ and\ R_{u_t,i}$ are the scores of item i rated by users $u\ and\ u_t$ respectively, $\bar{R}_u\ and\ \bar{R}_{u_t}$ represent the average scores of users $u\ and\ u_t$ respectively.

The last step is when N_{u_t} denotes the target user u_t's neighbour set. We would want to predict u_t rating for item j. The following equation (2) will be used.

$$P_{user-based}(u_t, j) = \bar{A}_{u_t} + \frac{\sum \left(R_{u_n,j} - \bar{R}_{u_n}\right) * sim(u_t, u_n)}{\sum_{u_n \varepsilon N_{u_t}} | sim(u_t u_n) |}$$

(2)

Where $\bar{A}_{u_t}$ represents the average score for user u_t for the rated items, $R_{u_n,j}$ is the score of item j rated by neighbour user u_n, $\bar{R}_{u_n}$ means the average score of neighbour u_n for the rated items, $sim(u_t, u_n)$ means the similarity between user u_t and the neighbour u_n.

This will be used to recommend an item to target user. For cosine based similarity algorithm refer to (Bigdeli, 2008).

B Item-based Nearest Neighbour

Item-based nearest neighbour algorithms are transpose of the user-based nearest neighbour algorithms. Item-based algorithms create predictions based on similarities between items [5].

There are many ways to calculate the similarity between items. Some of the most popular algorithms are cosine based similarity, correlation based similarity and adjusted-cosine similarity. The formula for Adjusted-based cosine which is the most popular and believed to be the most accurate [5, 8] is given in equation (3).

$$Itemsim(i,j) = \frac{\sum_{u \varepsilon U_{i,j}} (R_{u,i} - \bar{R}_u)(R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \varepsilon U_{i,j}} (R_{u,i} - \bar{R}_u)^2}\sqrt{\sum_{u \varepsilon U_{i,j}} R_{u,j} - \bar{R}_u)^2}} \quad (3)$$

Where $R_{u,i}$ *and* $R_{u,j}$ represents the rating of user u on items i *and* j respectively, $\bar{R}_u$ is the mean of the u^{th} user's ratings and $U_{i,j}$ represents all users who have rated items i *and* j.

The prediction calculation for item based nearest neighbour algorithm for user u and item j is carried out using formula (4) below.

$$P_{item-based}(u_t, j) = \frac{\sum_{i \varepsilon R_{u_t}} Itemsim(i,j) * R_{u_t,j}}{\sum_{i \varepsilon R_{u_t}} Itemsim(i,j)} \quad (4)$$

If the predicted rating is high then the system recommends the item to user. The item-based nearest neighbour algorithms are more accurate in predicting ratings than user based nearest neighbour algorithms [5].

III. Content-Based Filtering

CBF approaches recommend items that are similar in content to the items the user liked in the past or march to the attributes of the user [9-10]. In content based filtering recommender systems every item is represented by a feature vector or an attribute profile. The feature hold numeric or nominal values representing certain aspects of the item like colour, price, etc. A variety of (dis) similarity measures between the feature vectors may be used to compute the similarity of two items. The Euclidean or cosine (dis)similarity algorithms can be used and they are given in equations (5) and (6) respectively.

Euclidean dissimilarity

$$dissim(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} = \| x - y \|_2 \quad (5)$$

Cosine similarity

$$sim(x,y) = \frac{\sum_{i=1}^{n} x_i * y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \quad (6)$$

Where x *and* y are an items vectors with n elements in them, $dissim(x,y)$ *and* $sim(x,y)$ measure the distance apart and closeness respectively.

The (dis)similarity values are then used to obtain a ranked list of recommended items. These approaches are based on information retrieval because content associated with the user's preferences is treated as a query and unrated objects are scored with similarity to the query. This approach can give recommendations in any domain. Content based recommender systems work well if the items can be properly represented as a set of features.

IV. Knowledge Based Recommender Systems

Knowledge based systems use knowledge structure to make inference about the user needs and preferences [11]. Knowledge based approaches are well-known in that they have functional knowledge: they have knowledge about how a particular item satisfies a particular user need, and can therefore reason about the relationship between a need and possible recommendation [12]. The user profile can be any knowledge structure that supports this inference.

V. Hybrid Recommender Systems

A hybrid is combination of at least two techniques in order to overcome the deficiencies of a single method used in isolation [10]. One way is to combine content based and collaborative filtering algorithms in such a way that they produce separate ranked lists of recommendations then merge them to make up the final recommendations [9]. Some notable examples of hybrid recommender systems are Weighted and Switching hybrid recommender systems. A weighted hybrid recommender is one in which the score of a recommended item is calculated from the results of all of the available recommendation algorithms in the system. For example the simplest combined hybrid recommender systems would be a linear combination of recommendation scores. Switching Hybrid recommender system (SH) uses some criterion to switch between recommendation techniques. Example of (SH) recommender system is the DailyLearner that uses a content\collaborative hybrid. In this hybrid content based recommendation algorithm is employed first then collaborative if the first results are not satisfactory [13-14].

VI. Challenges Of Recommendation Techniques

Recommender systems techniques have been very successful in past, but their extensive use has exposed some real challenges. Some of the challenges are: Data Sparsity, Cold Start Problem, Fraud, Scalability, Gray sheep, Shilling attack and synonymy [6-7, 9, 15].

Data Sparsity: In practice, many commercial recommender systems are used to evaluate very large item sets (e.g. Amazon.com, CDnow.com). In these systems, even active users may have purchased one percent of the items (1% of two million of books is 20 000 books). The user-item matrix used for CF will be extremely sparse and a recommender system based on nearest neighbour algorithms may be unable to make any item recommendations for a particular user. The system becomes very ineffective. Under data sparsity there is also reduced coverage and neighbour transitivity [5, 7]. Coverage can be defined as the percentage of items that the system could provide recommendations for. The reduced coverage problem arises when the number of users' ratings may be very small compared with the large number of items in the system and the recommender system may fail to generate the recommendations for them. Neighbour transitivity refers to a problem with sparse databases, in which users with similar tastes may not be identified if they have not rated the same items. Content based approaches can also solve the problem since they do not require ratings from other users.

Cold start problem describes a situation in which a recommender system is unable to make meaningful recommendations due to an initial lack of ratings. Cold start occurs when a new user or item has just entered the system, it is very difficult to find similar ones due to inadequate enough information. New items cannot be recommended until some users rate them. The new item problem affects collaborative filtering recommender systems. Since content based filtering recommender systems do not dependent on ratings from other users, they can be used to produce recommendations for all items provided attributes of the items are available. New users are very unlikely to be given good recommendations because of lack of their rating or purchase history. Research to solve the new user problem is focusing on effectively selecting items to be rated by the user to quickly get the user preferences to improve the recommendation performance [9].

Scalability: When the population of existing users and items grow tremendously, the traditional recommender systems algorithms will suffer serious scalability problems, with computational resources going beyond practical or acceptable levels.

Synonymy: When a number of the same or very similar items have a different name and recommender systems fail to discover this latent association then treat these products differently.

Gray Sheep and Black Sheep: When a user whose opinions do not consistently correlate in agreement or disagreement with any group of people and thus not benefit from the system. The gray sheep users problem is also responsible for increased error rate in collaborative filtering recommender systems [16], which often result in failure of recommender systems. Black sheep are those users who have no or very few people who they correlate with. This situation makes it very difficult to make recommendation for them [12].

Fraud: Recommender systems are increasingly being adopted by commercial websites due to their economic benefits to the retailers and service providers. Unprincipled competing vendors have started to engage in different forms of fraud in order to cheat the recommender systems to their advantage. They have endeavoured to inflate the perceived attractiveness of their own commodities (push attacks) or reduce the ratings of their rivals (nuke attacks). These attacks are also known as shilling attacks [7, 9].

With all these challenges encountered in the use of recommendation systems, there is need to evaluate the performance of the developed systems. The evaluation of the systems enables to determine the accuracy of the systems.

VII. Evaluation Metrics For Recommender Systems

The performance of recommender system can be evaluated by comparing recommendations to a test set of known user ratings. These systems are commonly measured using predictive accuracy metrics, where the predicted ratings are directly compared to actual user ratings [9]. The commonly used metrics are Mean Absolute Error (MAE) and Root Mean Error (RME) as formulated in equations (5) and (6) respectively [9].

$$MAE = \frac{\sum | P_{u,i} - R_{u,i} |}{N} \quad \textbf{(6)}$$

$$RME = \sqrt{\frac{\left(P_{u,i} - R_{u,i}\right)^2}{N}} \quad \textbf{(7)}$$

Where $P_{u,i}$ is the predicted ratings for u on item i, $R_{u,i}$ is the actual rating and N is the total number of ratings in the test set. Predictive accuracy metrics treat all items equally.

VIII. Conclusion

In this paper, techniques that are used to construct recommender systems have been highlighted. The challenges of these techniques have been discussed. The performance evaluation techniques of recommender systems have also been looked at. In summary, recommender systems have added-value to business and corporation, at the same time supporting decision making to customers in choosing the product or service from a vast information space.

REFERENCES

[1] T. Bogers and A. v. d. Bosch, "Collaborative and Content-based Filtering for item Recommendation on Social Bookmarking Websites," in ACM RecSys '09 Workshop on Recommender Systems and the Social Web, New York, USA, 2009, pp. 9-16.

[2] A. Gunawardana and C. Meek, "A Unified Approach to Building Hybrid Recommender Systems," in Proceedings of the 2009 ACM Conference on Recommender Systems, New York, 2009, pp. 117-124.

[3] C.-L. Huang and W.-L. Huang, "Handling sequential pattern decay:Developing a two-stage collaborative recommender system," Electronic Commerce Research and Applications, vol. 8, pp. 117-129, 2009.

[4] O. O. Olugbara, et al., "Exploiting Image Content in Location-Based Shopping Recommender Systems for Mobile Users," International Journal of Information Technology & Decision Making, vol. 9, pp. 759-778, 2010.

[5] J. B. Schafer, et al., "Collaborative Filtering Recommender Systems," in The Adaptive web, Springer-Verlag, Ed., ed Berlin, Heidelberg, 2007, pp. 291-324.

[6] Z. Chen, et al., "A Collaborative Filtering Recommendation Algorithm Based on User Interest Change and Trust Evaluation," Internation Journal of Digital Content Technology and its Applications vol. 4, pp. 106-113, 2010.

[7] X. Su and T. M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," Advances in Artificial Intelligence, vol. 2009, pp. 1-19, 2009.

[8] J. Zhang, et al., "An Optimized Item-Based Collaborative Filtering Recommendation Algorithm," in IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC), Beijing, 2009, pp. 414-418.

[9] P. Melville and V. Sindhwani, "Recommender Systems," in Encyclopedia of Machine Learning, S. Verlag, Ed., ed Berlin: Springer, 2010, pp. 1-9.

[10] M. J. Pazzani and D. Billsus, "Content-based Recommendation Systems," in The Adaptive Web, methods and Strategies of Web Personalization, 2007, pp. 325-341.

[11] F. Ricci, "Mobile Recommender Systems," IT & Tourism, vol. 12, pp. 205-231, 2010.

[12] M. d. Gemmis, et al., "Preference Learning in Recommender Systems," in European Conference on Machine Learning and Principles and Practice of knowledge Discovery in Databases (ECML PKDD 2009), Bled, Slovenia, 2009, pp. 41-55.

[13] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," User Modeling and User-Adapted Interaction, vol. 12, pp. 331-370, 2002.

[14] M. A. Ghazanfar and A. Prugel-Bennett, "An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering," in Proceedings of the International MultiConference of Engineers and Computer Science (IMECS), Hong Kong, 2010.

[15] B. M. Sarwar, et al., "Recommender Systems for Large-Scale E-Commerce: Scalable Neighborhood Formation Using Clustering," in In Proceedings of the Fifth International Conference on Computer and Information Technology, Dhaka, Bangladesh, 2002.

[16] M. A. Ghazanfar and A. Prugel-Bennett, "Fulfilling the Needs of Gray-Sheep Users in Recommender Systems, A Clustering Solution," in In 2011 International Conference on Information Systems and Computational Intelligence, Harbin, China, 2011.