International Journal For Technological Research In Engineering
Volume 4, Issue 7, March-2017 *ISSN (Online): 2347 - 4718*

A SURVEY ON RECOMMENDER SYSTEM TECHNIQUES

Vidhya Raval[1], Risha Tiwari[2]
[1]Research Scholar, [2]Assistant Professor
Computer Engineering, Hasmukh Goswami Collage of Engineering, Ahmedabad, Gujarat, India

ABSTRACT: Recommender Systems are software tools and techniques providing suggestions for items to be of use to a user [1]. Recommender systems are combinations of various algorithms that collects various kinds of data implicitly or explicitly from the users. Recommender system uses these data to predict other user's preferences. They provides suggestions to the users helps them in various decision making processes, such as what items to buy, what music to listen to, or what online news to read [1]. Web portals use these kinds of recommendations to assist their users. This survey paper analyzes various techniques used in Recommender Systems.

I. INTRODUCTION

Recommender systems were originally defined as ones in which "people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients" (Resnick& Varian 1997). The term now has a broader connotation, describing any system that produces personalized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options. Such systems have an obvious appeal in an environment where the amount of on-line information vastly outstrips any individual's capability to survey it [2]. Recommender systems are now an integral part of some e-commerce sites such as Amazon.com (Schafer, Konstan&Riedl, 1999). Recommender systems need to have knowledge about users to provide them with suggestions. There are various techniques available to provide recommendations to the user. We are going to consider the techniques based on the data source used to provide recommendations to the users. Based on the data source, recommender system techniques can be categorized as follows:

- Collaborative Filtering based Recommender System
- Content-Based Recommender System
- Demographic Recommender System
- Knowledge-Based Recommender System
- Community Based Recommender System
- Hybrid Recommender System

In the following section I will discuss the approaches in detail.

II. RELATED WORK

2.1 Collaborative Filtering based Recommender Systems

Collaborative Filtering (CF) based Recommender Systems are most basic techniques of recommending items to the users. The simplest and original implementation of this approach [3] recommends to the active user the items that other users with similar tastes liked in the past. The similarity in taste of two users is calculated based on the similarity in the rating history of the users [1]. Collaborative Filtering (CF) systems work by collecting user feedback in the form of ratings for items in a given domain and exploiting similarities in rating behavior amongst several users in determining how to recommend an item. Collaborative Filtering (CF) methods can be further subdivided into neighborhood-based and model-based approaches. Collaborative recommendation is probably the most familiar, most widely implemented and most mature of the technologies. Collaborative recommender systems aggregate ratings or recommendations of objects, recognize commonalities between users on the basis of their ratings, and generate new recommendations based on inter-user comparisons. A typical user profile in a collaborative system consists of a vector of items and their ratings, continuously augmented as the user interacts with the system over time[1]. The greatest strength of collaborative techniques is that they are completely independent of any machine-readable representation of the objects being recommended, and work well for complex objects such as music and movies where variations in taste are responsible for much of the variation in preferences [2].

2.2 Content-Based Recommender System

The system learns to recommend items that are similar to the ones that the user liked in the past. The similarity of items is calculated based on the features associated with the compared items [1]. Content-Based Recommender Systems thus uses the item's details as a source to provide recommendations. For example, if a user likes electronics goods from some e-commerce website then items from the same category will be recommended to that user further.

Content-based Information Filtering (IF) systems need proper techniques for representing the items and producing the user profile, and some strategies for comparing the user profile with the item representation. The high level architecture of a content-based recommender system is depicted in Figure 2.1. The recommendation process is performed in three steps, each of which is handled by a separate component:

• CONTENT ANALYZER – 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 [6], which are able to infer a model of user interests starting from itemsliked or disliked in the past. For instance, the PROFILE LEARNER of a Web page recommender can implement a relevance feedback method [6] 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 [7]), 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 [1].

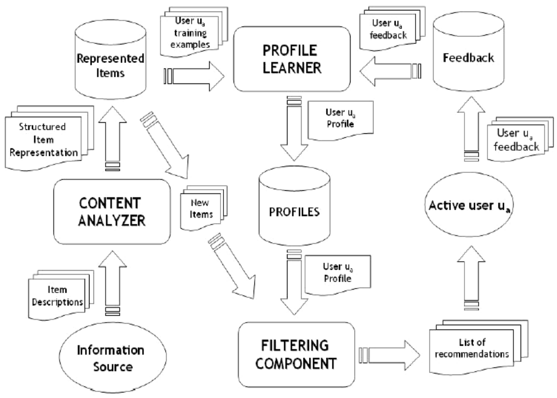

Fig : 2.1 : High level architecture of a Content-based RecommenderSystem. [1]

2.3 Demographic Recommender System

Demographic recommender systems aim to categorize the user based on personal attributes and make recommendations based on demographic classes. An early example of this kind of system was Grundy (Rich, 1979) that recommended books based on personal information gathered through an interactive dialogue. The users' responses were matched against a library of manually assembled user stereotypes. Some more recent recommender systems have also taken this approach. [2]

A demographic recommender system uses the personal information of the user to make predictions regarding their likes. Krulwich (1997), for example, uses demographic groups from marketing research to suggest a range of products and services. A short survey is used to gather the data for user categorization. In other systems, machine learning is used to arrive at a classifier based on demographic data (Pazzani 1999). [2]
The demographic information in a user model can be represented in various ways. Demographic techniques form "people-to-people" correlations like collaborative ones, but use different data. The benefit of a demographic approach is that it may not require a history of user ratings of the type needed by collaborative and content-based techniques.[2]

2.4 Knowledge-Based Recommender System

Knowledge-based systems recommend items based on specific domain knowledge about how certain item features meet users needs and preferences and, ultimately, how the item is useful for the user. Notable knowledge based recommender systems are case-based [5].

In Knowledge-Based Recommender systems a similarity function estimates how much the user requirements match the recommendations. In such systems, the similarity score can be directly used as the utility of the recommendation for the user.

Knowledge-based systems tend to work better than others at the beginning of their deployment but if they are not equipped with learning components they may be surpassed by other shallow methods that can exploit the logs of the human/computer interaction.[1]

2.5 Community Based Recommender System

This type of system recommends items based on the preferences of the users friends. This technique follows the epigram "Tell me who your friends are, and I will tell you who you are". [4]

Evidence suggests that people tend to rely more on recommendations from their friends than on recommendations from similar but anonymous individuals [5]. This observation, combined with the growing popularity of open social networks, is generating a rising interest in community-based systems or, as or as they usually referred to, social recommender systems [6]. This type of Recommender Systems models and uses the information about the social relations between the users and the preferences of their friends and family. The recommendation is such systems are based on the ratings provided by the user's friends and family members. These kinds of Recommender Systems are following the rise of social-networks and enable a simple and comprehensive acquisition of data related to the social relations of the users.[1]

2.6 Hybrid Recommender System

The Hybrid Recommender Systems are based on the combination of the above mentioned techniques. A hybrid system combining techniques A and B tries to use the advantages of A to fix the disadvantages of B. For instance, Collaborative Filtering methods suffer from new-item problems, i.e., they cannot recommend items that have no ratings. This does not limit content-based approaches since the prediction for new items is based on their description (features) that are typically easily available. Given two (or more) basic RSs techniques, several ways have been proposed for combining them to create a new hybrid system.[1]

III. COMPARISONS OF RECOMMENDATION TECHNIQUES

All recommendation techniques have strengths and weaknesses discussed below and summarized in Table I.Perhaps the best known is the "ramp-up" problem (Konstan, et al. 1998). This term actually refers to two distinct butrelated problems.

New User: Because recommendations follow from a comparison between the target user and other users based solely on the accumulation of ratings, a user with few ratings becomes difficult to categorize.

New Item: Similarly, a new item that has not had many ratings also cannot be easily recommended: the "new item" problem. This problem shows up in domains such as news articles where there is a constant stream of new items and each user only rates a few. It is also known as the "early rater" problem, since the first person to rate an item gets little benefit from doing so: such early ratings do not improve a user's ability to match against others (Avery and Zeckhauser, 1997). This makes it necessary for recommender systems to provide other incentives to encourage users to provide ratings.[8]

The advantages and disadvantages of the Recommendation Systems are as follows:

Technique	**Advantages**	**Disadvantages**
Collaborative filtering	• Can identify cross-genre slot. • Domain knowledge not needed. • Adaptive: quality improves over time. • Implicit feedback sufficient	• New user ramp-up problem • New item ramp-up problem • "Gray sheep" problem • Quality dependent on large historical data set. • Stability vs. plasticity problem
Content-based	• Domain knowledge not needed. • Adaptive: quality improves over time. • Implicit feedback sufficient	• New user ramp-up problem • Quality dependent on large historical data set. • Stability vs. plasticity problem
Demographic	• Can identify cross-genre slot. • Domain knowledge not needed. • Adaptive: quality improves over time.	• New user ramp-up problem • "Gray sheep" problem • Stability vs. plasticity problem • Must gather demographic information
Utility-based	• No ramp-up required • Sensitive to changes of preference • Can include non-product features	• User must input utility function • Suggestion ability static (does not learn)
Knowledge-based	• No ramp-up required • Sensitive to changes of preference • Can include non-product features • Can map from user needs to Products	• User must input utility function • Knowledge engineering required.

IV. CONCLUSION

All of the learning-based techniques (collaborative, content-based and demographic) suffer from the ramp-up problem in one form or another. The converse of this problem is the stability vs. plasticity problem for such learners. Once a user's profile has been established in the system, it is difficult to change one's preferences. A steak-eater who becomes a vegetarian will continue to get steakhouse recommendations from a content-based or collaborative recommender for some time, until newer ratings have the chance to tip the scales. Many adaptive systems include some sort of temporal discount to cause older ratings to have less influence, but they do so at the risk of losing information about interests that are long-term but sporadically exercised (Billsus&Pazzani, 2000; Schwab, et al. 2001). For example, a user might like to read about major earthquakes when they happen, but such occurrences are sufficiently rare that the ratings associated with last year's earthquake are gone by the time the next big one hits. Knowledge- and utility-based recommenders respond to the user's immediate need and do not need any kind of retraining when preferences change.[8]

REFERENCES

[1] Recommender Systems Handbook Francesco Ricci · LiorRokach · BrachaShapira · Paul B. Kantor .Springer.

[2] Hybrid Recommender Systems: Survey and Experiments, Robin Burke ,California State University, Fullerton Department of Information Systems and Decision Sciences.

[3] Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommendersystems. In: The Adaptive Web, pp. 291–324. Springer Berlin / Heidelberg (2007)

[4] Bridge, D., G¨oker, M., McGinty, L., Smyth, B.: Case-based recommender systems. The

[5] Knowledge Engineering review 20(3), 315–320 (2006)

[6] Ricci, F., Cavada, D., Mirzadeh, N., Venturini, A.: Case-based travel recommendations. In: D.R. Fesenmaier, K.Woeber, H.Werthner (eds.) Destination Recommendation Systems: Behavioural Foundations and Applications, pp. 67–93.

[7] Arazy, O., Kumar, N., Shapira, B.: Improving social recommender systems. IT Professional11(4), 38–44 (2009).

[8] Ben-Shimon, D., Tsikinovsky, A., Rokach, L., Meisels, A., Shani, G., Naamani, L.: Recommender system from personal social networks. In: K. Wegrzyn-Wolska, P.S. Szczepaniak (eds.) AWIC, Advances in Soft Computing, vol. 43, pp. 47–55. Springer (2007)

[9] Burke, R.: Hybrid web recommender systems. In: The AdaptiveWeb, pp. 377–408. Springer Berlin / Heidelberg (2007)

