JOURNAL OF INFORMATION, KNOWLEDGE AND RESEARCH IN COMPUTER ENGINEERING

Hybrid Recommender systems: survey and experiments

PROF. VIPUL VEKARIYA
Research Scholar,
Suresh Gyanvihar University,Jaipur.
India
vmvekariya@yahoo.co.in

DR.G.R.KULKARNI
Principal,
D.I.E.T. , Sajjangad Road
Satara, Maharashtra
grkulkarni29264@rediffmail.com

Abstract

Recommender systems represent user preferences for the purpose of suggesting items to purchase or examine. They have become fundamental applications in electronic commerce and information access, providing suggestions that effectively prune large information spaces so that users are directed toward those items that best meet their needs and preferences. A variety of techniques have been proposed for performing recommendation, including content-based, collaborative, knowledge-based and other techniques. To improve performance, these methods have sometimes been combined in hybrid recommenders. This paper surveys the landscape of actual and possible hybrid recommenders, and introduces a novel hybrid, system that combines content-based recommendation and collaborative filtering to recommend restaurants.

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' [1]. The term now has a broader connotation, describing any system that produces individualized 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. Recommender systems are now an integral part of some e-commerce sites such as Amazon.com and CDNow [2]. It is the criteria of 'individualized' and 'interesting and useful' that separate the recommender system from information retrieval systems or search engines. The semantics of a search engine are 'matching': the system is supposed to return all those items that match the query ranked by degree of match. Techniques such as relevance feedback enable a search engine to refine its representation of the user's query, and represent a Simple form of recommendation. The next-generation search engine Google1 blurs this distinction, incorporating 'authoritativeness' criteria into its ranking (defined recursively as the sum of the authoritativeness of pages linking to a given page) in order to return more useful results. One common thread in recommender systems research is the need to combine recommendation techniques to achieve peak performance. All of the known recommendation techniques have strengths and weaknesses, and many researchers have chosen to combine techniques in different ways. This article surveys the different recommendation techniques being researched analyzing them in terms of the data that supports the recommendations and the algorithms that operate on that data C and examines the range of hybridization techniques that have been proposed. This analysis points to a number of possible hybrids that have yet to be explored.

II.RECOMMENDATION TECHNIQUES

Recommendation techniques have a number of possible classifications [1][2]. Of interest in this discussion is not the type of interface or the properties of the user's interaction with the recommender, but rather the sources of data on which recommendation is based and the use to which that data is put. Specifically, recommender systems have (i) background data, the information that the system has before the recommendation process begins, (ii) input data, the information that user must communicate to the system in order to generate a recommendation, and (iii) an algorithm that combines background and input data to arrive at its suggestions. On this basis, we can distinguish eve different recommendation techniques as shown in Table I. Assume that I is the set of items over which recommendations might be made, U is the set of users whose preferences are known, u is the user for whom recommendations need to be generated, and i is some item for which we would like to predict u's preference. Collaborative recommendation is

Technique	Background	Input	Process
Collaborative	Ratings from U of items in I.	Ratings from u of items in I.	Identify users in U similar to u, and extrapolate from their ratings of i.
Content Based	Features of items in I.	u's ratings of items in I.	Generate a classifier that fits u's rating behavior and use it on i.
Demographic	Demographic information about U and their ratings of items in I.	Demographic information about u.	Identify users that are demographically similar to u, and extrapolate from their ratings of i.
Utility – based	Features of items in I.	A utility function over items in I that describes u's preferences.	Apply the function to the items and determine i's rank.
Knowledge Based	Features of items in I. Knowledge of how these items meet a User's needs.	A description of u's needs or interests.	Infer a match between i and u's need.

Table – I: Recommendation Technique

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. Some systems used time-based discounting of ratings to account for drift in user interests [3]. In some cases, ratings may be binary (like/dislike) or real-valued indicating degree of preference. Some of the most important systems using this technique are [4] Tapestry (Goldberg et al., 1992) and Recommender (Hill et al., 1995). These systems can be either memory-based, comparing users against each other directly using correlation or other measures, or model-based, in which a model is derived from the historical rating data and used to make predictions (Model-based recommenders have used a variety of learning techniques including neural networks (Jennings & Higuchi, 1993), latent semantic indexing (Foltz, 1990), and Bayesian networks (Condliff et al., 1999).

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] call this 'people-to-people correlation.

'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 user's responses were matched against a library of manually assembled user stereotypes. Some more recent recommender systems have also taken this approach. 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 [6]. The representation of demographic information in a user model can vary greatly. Rich's system used hand-crafted attributes with numeric confidence values. Pazzani's model uses Winnow to extract features from users' home pages that are predictive of liking certain restaurants.

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.

Content-based recommendation is an outgrowth and continuation of information filtering research (Belkin & Croft, 1992). In a content-based system,

the objects of interest are defined by their associated features. For example, text recommendation systems like the newsgroup filtering system NewsWeeder (Lang, 1995) uses the words of their texts as features. A content-based recommender learns a pro¢le of the user's interests based on the features present in objects the user has rated. [2] call this 'item-to-item correlation.' The type of user profile derived by a content-based recommender depends on the learning method employed. Decision trees, neural nets, and vector-based representations have all been used. As in the collaborative case, content-based user profiles are long-term models and updated as more evidence about user preferences is observed.

Utility-based and knowledge-based recommenders do not attempt to build long-term generalizations about their users, but rather base their advice on an evaluation of the match between a user's need and the set of options available. Utility-based recommenders make suggestions based on a computation of the utility of each object for the user. Of course, the central problem is how to create a utility function for each user. The user profile therefore is the utility function that the system has derived for the user, and the system employs constraint satisfaction techniques to locate the best match. The benefit of utility-based recommendation is that it can factor non-product attributes, such as vendor reliability and product availability, into the utility computation, making it possible for example to trade off price against delivery schedule for a user who has an immediate need.

Knowledge-based recommendation attempts to suggest objects based on inferences about a user's needs and preferences. In some sense, all recommendation techniques could be described as doing some kind of inference. Knowledge-based approaches are distinguished in that they have functional knowledge: they have knowledge about how a particular item meets a particular user need, and can therefore reason about the relationship between a need and a possible recommendation. The user profile can be any knowledge structure that supports this inference. In the simplest case, as in Google, it may simply be the query that the user has formulated. In others, it may be a more detailed representation of the user's [7].

III. COMPARING RECOMMENDATION TECHNIQUES

All recommendation techniques have strengths and weaknesses discussed below and summarized in Table II. Perhaps the best known is the 'ramp-up' problem [8]. This term actually refers to two distinct but related problems.

Technique	Advantage	Disadvantage
Collaborative filtering (CF)	A. Can identify cross-genre niches. B. Domain knowledge not needed. C. Adaptive: quality improves over time. D. Implicit feedback sufficient	I. New user ramp-up problem J. New item ramp-up problem K. 'Gray sheep' problem L. Quality dependent on large Historical data set. M. Stability vs. plasticity problem
Content-based (CN)	B, C, D	I, L, M
Demographic (DM)	A, B, C	I, K, L, M
Utility-based (UT)	E. No ramp-up required F. Sensitive to changes of preference G. Can include non-product features	O. User must input utility function P. Suggestion ability static (does not learn)
Knowledge-based (KB)	E, F, G H. Can map from user needs to products	P Q. Knowledge engineering required.

Table-II. Tradeoffs between recommendation Techniques

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 (This makes it necessary for recommender systems to provide other incentives to encourage users to provide ratings.

IV. AN APPROACH HYBRID COLLABORATIVE FILTERING AND CONTENT-BASED FILTERING

The proposed hybrid filtering transparently creates and maintains user preferences. It assists users by providing both collaborative filtering and content-based filtering, which are updated in real time whenever the user changes his/her current page using any navigation technique. The WebBot uses the link provided in the restaurant dataset to download restaurant content from database. WebBot keeps track of each individual user and provides that a user online assistance. The assistance includes two lists of recommendations based on two different filtering paradigms: collaborative filtering and content-based filtering. WebBot updates the list each time the user changes his/her current page. Content-based filtering is based on the correlation between the content of the pages and the user preferences. The collaborative filtering is based on a comparison between the user path of navigation and the access patterns of past users. Hybrid filtering may eliminate the shortcomings in each approach. By making collaborative filtering, we can deal with any kind of content and explore new domains to find something interesting to the user. By making content-based filtering, we can deal with pages un-seen by others. Fig. 1 is the system overview for hybrid filtering.

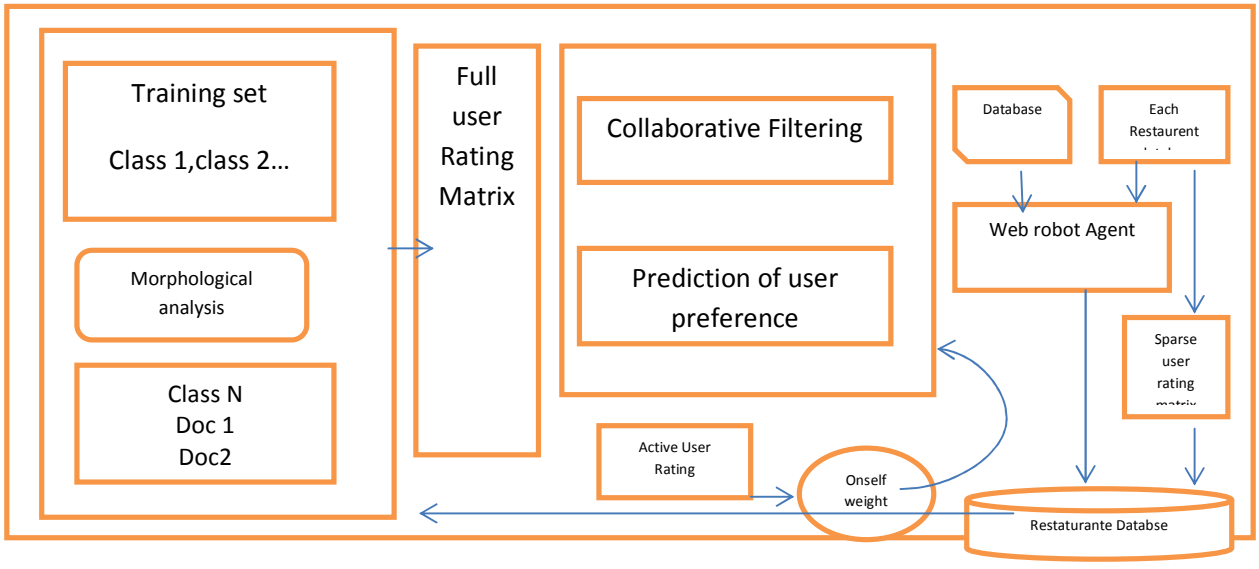

Fig. 1 is the system overview for hybrid filtering.

To overcome the problem of stateless connection in HTTP, WebBot follows users through tracking their IP address. To track user presence, a timeout mechanism is used to delete user's session information after a predetermined amount of idle time. So that, a connection after the specified period having the same IP is identified as a new user. This method is fairly easy to implement. Consequently, the IP of a proxy server may represent two or more people who are accessing the same web site simultaneously in their browsing sessions, causing an obvious conflict. However, the reality is that many large sites use this method and have not any clashes. The restaurant dataset also provides the user-ratings matrix; which is a matrix of users versus items, where each cell is the rating given by a user to an item. We will refer to each row of this matrix as a user-rating vector. The user-ratings matrix is very sparse, because most users have not rated most items. The content-based predictor is trained on each user-ratings vector and a pseudo user-ratings vector is created. A pseudo user-ratings vector contains the user's actual ratings and content-based filtering for the un-rated items. All pseudo user-ratings vector put together from the pseudo ratings matrix, which is a full matrix. Now given an active user's ratings, filtering predictions are made for a new item using Collaborative filtering on the full pseudo ratings matrix.

V.EXTRACTING INFORMATION FROM WEB ROBOT AGENT AND BUILDING A DATABASE

Our current prototype system, WebBot: Web Robot Agent uses a database of movie content information extracted from web page at restaurant database. Therefore, the system's current content information

about restaurant name consists of textual metadata rather than the actual text of the items themselves. A restaurant database subject search is performed to obtain a list of restaurant-description of broadly relevant titles. Web- Bot then downloads each of these data and uses a simple pattern based information extraction system to extract data about each restaurant. Information extraction is the task of locating specific pieces of information from a document, thereby obtaining useful structured data from unstructured text. A WebBot follows the restaurant link provided for every restaurant in the restaurant dataset and collects information from the various links off the main database. We represent the content information of every restaurant as a set of features. Each feature is represented simply as a bag of words. Database produces the information about related facilities and restaurant names using collaborative filtering: however, WebBot treats them as additional content about the restaurant. The text in each feature is then processed into an un-ordered bag of words and the examples represented as a vector of bags of words.

VI. PERFORMANCE EVALUATION

We used a subset of the restaurant dataset. This dataset contains 3,291 randomly selected users and 728 restaurant for which content was available from database. To evaluate various approaches of filtering, we divided the rating dataset in test-set and training-set. The rating database is used a subset of the ratings data from the restaurant dataset. The training-set is used to predict ratings in the test-set using a commonly used error measure. The metrics for evaluating the accuracy of a prediction algorithm are used mean absolute error(MAE) and rank scoring measure(RSM) [9].

For evaluation, this paper uses the following methods: The proposed hybrid collaborative filtering and content-based filtering (HMW_HF), a collaborative filtering (P_Corr), the recommendation method using only the content-based filtering (Content), and a naïve combined approach (N_Com). The naïve combined approach takes the average of the ratings generated by the collaborative filtering and the content based filtering. The various methods were used to compare performance by changing the number of clustering users. Also, the proposed method was compared with the previous methods in section 1 that use both collaborative filtering and content-based filtering method by changing the number of user evaluations on items. The aforementioned previous method includes the Soboroff method [10] that solved the sparse rating problem, the Fab method that solved the first-rater problem, and the Pazzani method [6] that solved both the sparse rating problem and the first-rater problem.

Fig. 2 shows the MAE and RSM of varying the number of users. Fig. 2, as the number of users increases, the performance of the HMW_HF, and the P_Corr also increases, whereas the method using content shows no notable change in performance. In terms of accuracy of prediction, it is evident that method HMW_HF, which uses both collaborative filtering and content-based filtering, is more superior to method N_Com.

Fig. 3 is used to show the MAE and RSM when the number of user's evaluations is increased. In Fig. 3, the Soboroff method, which has the first-rater

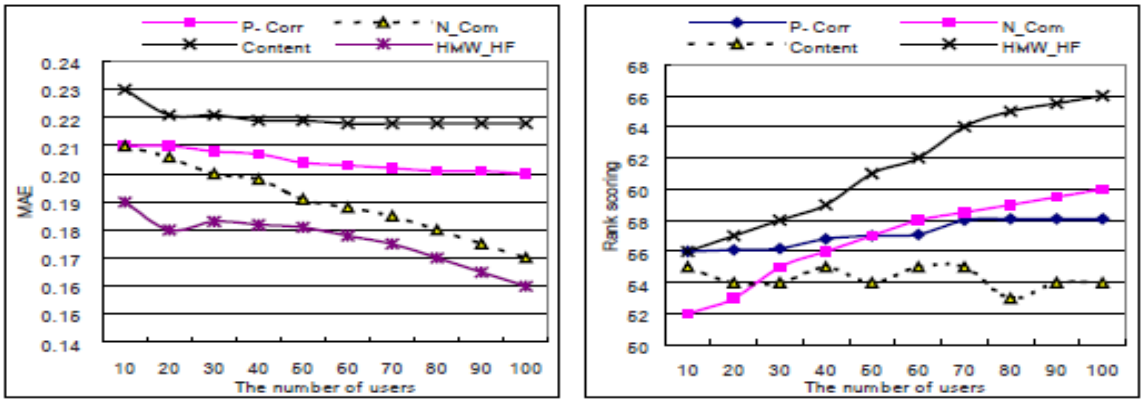

Fig 2. MAE, ranking scoring Measure at varying the number of user

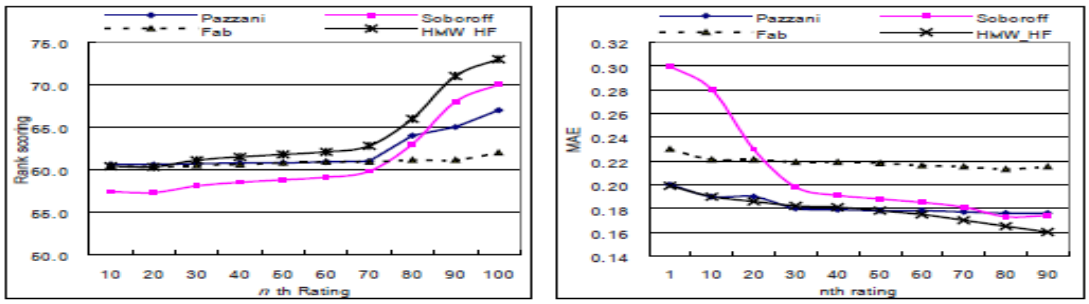

Fig 3. MAE, ranking scoring measure at nth rating

Problem, shows low performance when there are few evaluations; the other methods outperform the Soboroff method. Although the Pazzani method, which solved both the sparse rating problem and the first-rater problem, along with the HMW_HF show high rates HMW_HF shows Highest accuracy of all methods.

Since we use a pseudo ratings matrix, which is a full matrix, we eliminate the root of the sparse rating problem and the first-rater problem. Pseudo user-ratings vectors contain ratings for all items; and hence all users will be considered as potential neighbors. This increases the chances of finding similar users. The original user-ratings matrix may contain items that have not been rated by any user. In a collaborative filtering approach these items would be ignored. However in HMW_HF, these items would receive a content-based prediction from all users. Hence these items can now be recommended to the active user, thus overcoming the first-rater problem.

VII. CONCLUSION

Hybrid collaborative filtering and content-based filtering can significantly improve predictions of a recommender system. In this paper, we have shown how hybrid collaborative filtering and content-based filtering performs significantly better than collaborative, content-based, and combined filtering approach. The proposed hybrid filtering exploits content-based filtering within a collaborative framework. It overcomes the disadvantages of collaborative filtering and content-based filtering, collaborative filtering with content and vice versa. Further, due to the nature of the approach, any improvements in collaborative filtering or content-based filtering can be easily exploited to build a powerful improved recommender system.

VIII. REFERENCES

[1] Resnick, P. and Varian, H. R.: 1997, 'Recommender Systems'. Communications of the ACM, 40 (3), 56^58.

[2] Schafer, J. B., Konstan, J. and Riedl, J.: 1999, 'Recommender Systems in E-Commerce'. In: EC '99: Proceedings of the First ACM Conference on Electronic Commerce, Denver, CO, pp. 158^166.

[3] Billsus, D. and Pazzani,M.: 2000. 'User Modeling for Adaptive News Access'. User-Modeling and User-Adapted Interaction 10(2^3), 147^180.

[4] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J.: 1994, 'GroupLens: An Open Architecture for Collaborative Filtering of Netnews'. In: Proceedings of the Conference on Computer Supported Cooperative Work, Chapel Hill, NC, pp. 175^186.

[5] Jennings, A. and Higuchi, H.: 1993, 'A User Model Neural Network for a Personal News Service.' User Modelingand User-Adapted Interaction, 3, 1^25.

[6] Pazzani, M. J.: 1999, 'A Framework for Collaborative, Content-Based and Demographic Filtering'. Artificial Intelligence Review, 13 (5/6), 393^408.

[7] Towle, B. and Quinn, C.: 2000, 'Knowledge Based Recommender Systems Using Explicit User Models'. In Knowledge-Based Electronic AAAI Technical Report WS-00-04. pp. Markets, Papers from the AAAI Workshop,74^77. Menlo Park, CA: AAAI Press.

[8] Konstan, J. A., Riedl, J., Borchers, A. and Herlocker, J. L.: 1998, 'Recommender Systems: A GroupLens Perspective.' In: Recommender Systems: Papers from the 1998 Workshop (AAAI Technical Report WS-98-08). Menlo Park, CA: AAAI Press, pp. 60^64

[9] J. S. Breese, et al., "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," In Proc. of the Conference on Uncertainty in Artificial Intelligence, pp. 43-52, 1998.

[10] Soboroff, C. Nicholas, "Combining Content and Collaboration in Text Filtering," In Proc. of the IJCAI'99 Workshop on Machine Learning in Information Filtering, pp. 86-91, 1999.