Recommending HTML-documents using Feature Guided Automated Collaborative Filtering

Gabriela Polčicová[1]

[1] Comenius University, Faculty of Mathematics and Physics, Institute of Informatics,
Mlynská dolina
842 15 Bratislava, Slovakia
polcicova@fmph.uniba.sk

Abstract. We proposed the system that utilizes Feature Guided Automated Collaborative Filtering for recommending relevant HTML-documents to the users. While browsing the World Wide Web, user expresses his opinions on documents by rating them. The system "learns" user's opinions and searches for like-minded users in order to recommend him unseen relevant documents of interest.

1 Introduction

Exponential growth of the number of information sources accessible through the Internet makes searching for relevant information complicated. One of the approaches how to tackle this problem is Information Filtering. Filtering systems that learn user's preferences and recommend to the user only information suitable for his needs are also termed Recommender systems. The method for recommending appropriate information using opinions of users with similar preferences is called Automated Collaborative Filtering (ACF). The recommender system - described in this paper - uses a particular type of the ACF method called Feature Guided Automated Collaborative Filtering (FGACF) for recommending HTML-documents to the users.

2 Feature Guided Automated Collaborative Filtering

Information Filtering (IF) is a term used to describe a variety of processes involving the delivery of information to the people who need it [2]. One type of automated IF is Automated Collaborative Filtering. ACF leverages the collective intelligence of the people in a community to assist individuals in finding personally relevant information. This filtering is based on the hypothesis that users who had similar preferences in the past will probably have similar preferences again [7]. Recommender systems using ACF operate by collecting user's opinions on a set of items via ratings of these items. By comparing ratings they search through the community of users to detect users with similar opinions. Opinions of those like-minded users about items that particular user with similar interest has not seen are used to predict his ratings. Items with positively predicted ratings are recommended to the user.

An important advantage of this approach, as compared to other filtering techniques, is that items being filtered need not to be analyzed by a computer. These systems are therefore able to recommend also items with relevant content that are formally different. Moreover the items do not need to be of the same type. Finally, recommendations are based on the item's subjective quality evaluation, rather than on some objective measurable properties.

With regard to the assumption that agreement of preferences in one topic (e.g. in literature) does not imply agreement of preferences in another topic (e.g. in sport), it is necessary to recommend items of each topic separately. According to content items are classified into categories (topics). This method of recommendation is called Feature Guided Automated Collaborative Filtering (FGACF) [4].

A variety of algorithms and systems for ACF have been reported in the literature. Most of the systems, such as the system Tapestry [6] that filters e-mails, Ringo [8] recommending music albums and artists, GroupLens [7] using Usenet newsgroups as a domain, or Fab [1] recommending HTML-documents, make use of correlation coefficient. Other approaches for ACF, e.g. method based on finding like-minded users by clustering them according to the filtered items content (system Yenta [5]), or method utilizing neural networks [3], have been reported.

3 Recommender system

We have proposed recommender system for the WWW that utilizes FGACF method. The system consists of users' agents (communication and recommendation agent) and servers (Fig. 1). For each user the agents perform the following tasks:

- collect user's ratings of HTML-documents (communication agent),
- find like-minded users (recommendation agent),
- use preferences of like-minded users to recommend HTML-documents to the user (recommendation agent)

All this is done for each topic represented by a category. Both types of agents acquire necessary information from servers. Both servers and agents can request information from Yahoo! search engine. Fig.2 shows communication among agents, servers and Yahoo!. The system was implemented in Java.

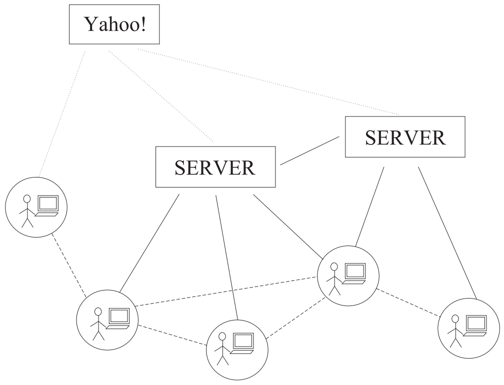

Fig. 1. The schema of the recommender system. Circles represent users' agents. Dashed lines represent communication between recommendation agents through profile documents, solid lines represent communication between servers and between servers and agents. Dotted lines represent communication between servers or agents with Yahoo!.

3.1 Communication agent

Communication agent performs two functions. The first one is to collect user's ratings and the second one is to offer recommended documents to the user. Both functions are accessible through GUI. While browsing users rate documents by writing URL of the rated document and the value of rating. Value of rating is a number from 7-point scale, where 7 represents the best and 1 the worst rating. A category the document belongs to has to be determined. This can be done either by the system or by the user. In the former case agent requests server to determine the category (*send me the category for the document* message). If the server does not know the category for the document, it has to be determined by the user. In the latter case agent sends to the server pair consisting of URL of the document and the category (*the document belongs to the category* message). If there is no server available, the agent requests Yahoo! only by sending the former type of message. The communication agent writes all ratings to the user's profile (HTML-document) accessible through the Internet. Profile is divided into categories. Each category consists of three parts:

- *a name* of the category,
- *links* – a list of ratings and rating predictions for the documents classified into the category each represented either by pair – document URL, the value of rating - or by the triplet – document URL, rating prediction and the word "prediction" ,
- *similar profiles* - a list of like-minded users' profiles for the category represented by the pair - URL of like-minded user's profile and degree of similarity expressing how much are ratings written in the profile similar to the user ratings.

List of predictions and similar profiles, as well as the list of recommendations (Fig. 2) is generated by the recommendation agent (RA).

A list of recommendations containing URL of the document, predicted rating and the category the document belongs to is offered to the user by communication agent.

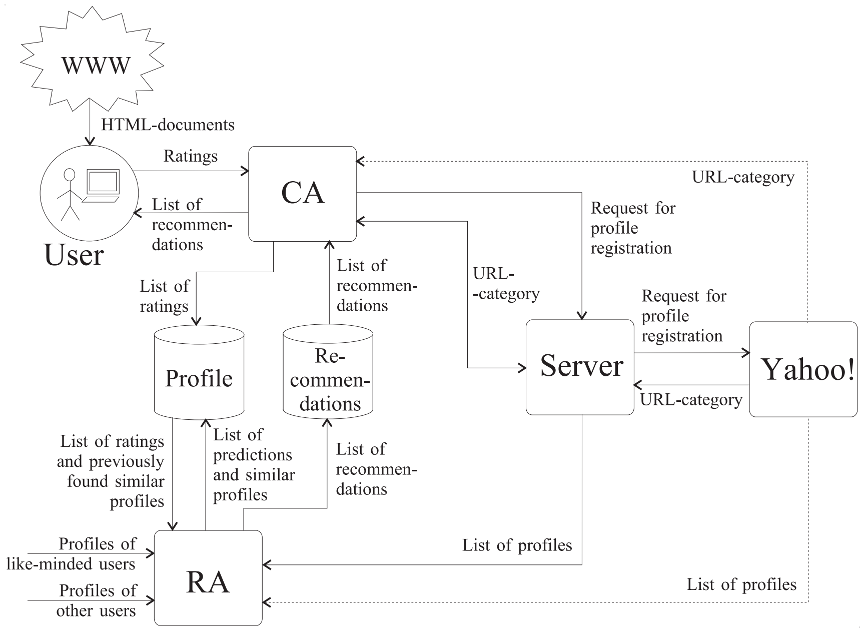

Fig. 2. Communication among user, communication agent (CA), recommendation agent (RA), server and Yahoo! search engine (solid lines). In the case that no server is available, CA and RA communicate directly with Yahoo! (dashed lines).

3.2 Recommendation agent

Recommendation agent compares profile of its user with profiles of other users for each category in which its user has rated. This process consists of three tasks:

1. reading the profile of the agent's user - it reads ratings, previously computed predictions and it also creates a list of previously found like-minded users,
2. reading profiles of the like-minded users and comparing them with user’s profile,
3. reading profiles of other users with unclear preferences and comparing them with user’s profile - agent acquires URLs of other users profiles by sending message to the server (*send me URLs of the profiles*). If there is no server available, recommendation agent can acquire list of registered profiles by requesting them from Yahoo!.

The amount of time tasks 2 and 3 take is determined by the user. Profiles comparison is done by degree of similarity computation for each category both users have rated. The degree of similarity is determined using Pearson correlation coefficient [8]

$$k_{xy} = \frac{\sum_{j \in I_{xy}} \left(r_{xj} - \bar{r}_x \right) \left(r_{yj} - \bar{r}_y \right)}{\sqrt{\sum_{j \in I_{xy}} \left(r_{xj} - \bar{r}_x \right)^2} \sqrt{\sum_{j \in I_{xy}} \left(r_{yj} - \bar{r}_y \right)^2}}, \qquad \textbf{(1)}$$

where I_{xy} is a set of documents rated by users x and y, r_x (r_y) is the rating of document j by the user x (the rating of document j by the user y) and $\bar{r}_x$ ($\bar{r}_y$) is average of user x (y) ratings.

Ratings predictions for the documents unrated by the user x are computed from the ratings of the other users y and their degree of similarity (k_{xy}) with the user x. Rating prediction (p_{xj}) for document j is given by [7]

$$p_{xj} = \bar{r}_x + \frac{\sum_{y \in U_j} \left(r_{yj} - \bar{r}_y \right) k_{xy}}{\sum_{y \in U_j} \left| k_{xy} \right|}, \qquad \textbf{(2)}$$

where U_j is the set of users who rated document j and whose profiles are used for prediction computation. This is done for each category.

User may set two options for predictions computation. The first one is whether the predictions are computed only from the ratings of like-minded users (all users y with degree of similarity to the user x holding $|k_{xy}| > 0.5$) or from the ratings of all users. The second option is whether for computing of predictions only ratings or both ratings and predictions will be used.

Ratings predictions are written into the user's profile. Documents with predicted rating higher then the threshold (e.g. 4 mean value of the scale) are written to the file *Recommendations* (Fig. 2). CA uses this file for recommending documents to the user.

3.3 Server

The server maintains two lists – list consisting of URLs of the agents profiles and list of pairs - URL of the rated document and the category the document belongs to. The function of the server is to react to the following agents' messages:

- *register new profile* - it adds the URL of the profile to the list and sends a request for profile registration to Yahoo! search engine,
- *send me the category for the document* - if pair - URL of the document, document category – exists in the list maintained on the server or if Yahoo! determined the category, server sends the category to the CA,

- *the document belongs to the category* - it adds the pair - URL of the document, document category - to the list,
- *send me URLs of the profiles* - it sends the list of profiles URLs to the RA.

Servers can also communicate with each other (Fig. 1) in order to complete their lists (*send me your lists* message).

4 Conclusions

Ambition of the proposed recommender system is to help the user in orientation in the number of information sources available through the Internet on basis of recommendations from like-minded users. The method used by the system enables it to make recommendations for particular categories. Documents categorization is performed either by Yahoo! search engine or by the users according to the topic. Documents recommendation is accomplished using ratings predictions based on user similarity.

An important feature of the system is that for the computation of new predictions in addition to ratings previously computed predictions may be used. This is an attempt to eliminate the well-known limitation of the ACF - profiles of the users who did not rate the same documents are not comparable. It enables the system to generate more recommendations (even less accurate). This might be helpful at the beginning of system operation when only a few documents are rated.

The system consists of agents and servers. Agents form preference profiles and generate recommendations for each user. That is why the system can work (with above mentioned constraints) even without servers that just maintain and provide information for agents. This may be considered as an advantage of this recommender system.

As the data acquisition is very time consuming, till now we have done only some preliminary studies with a small number of users and much work remains for further experiments. Requesting Yahoo! for profile registration also remains to be implemented. The future work will be aimed at solving a problem of documents categorization that is in the current system mostly done by the users.

References

1. Balabanovic, M., Shoham, Y.: Fab: Content-based, Collaborative Recommendation. Communications of the ACM, 40(3):29—38 (1997)
2. Belkin, N.J., Croft, B.W.: Information filtering and information retrieval: Two sides of the same coin? Communications of the ACM, 35 (12):29—38 (1992)
3. Billsus, D., Pazzani, M.: Learning Collaborative Information Filters. Proceedings of the International Conference on Machine Learning. Morgan Kaufmann Publishers. Madison (1998) accessed November 1998, url: http://www.ics.uci.edu/AI/ML/MLPapers.html

4. Firefly. Collaborative filtering technology: An overview, A White Paper (1997) accessed December 1997, url: http://www.firefly.net/products/CollaborativeFiltering.html
5. Foner, L.: Yenta: A multi-agent, referral based matchmaking system. The First International Conference on Autonomous Agents (Agents '97), (1997)
6. Goldberg, D., Nichols, D., Oki, B., Terry, D.: Using collaborative filtering to weave an information tapestry. Communication of the ACM, 35 (12): 61--70, (1992)
7. Resnick, P., Iacovou, N., Sushak, M., Bergstrom, P., Riedl, J.: GroupLens: An open architecture for collaborative filtering of netnews. Proceedings of the 1994 Computer Supported Cooperative Work Conference, New York, ACM. (1994)
8. Shardanand, U., Maes, P.: Social information filtering: Algorithms for automating "word of mouth". Proceedings of the 1995 ACM Conference on Human Factors in Computing Systems, 210--217, New York, ACM (1995)