RECOMMENDATION PROCESS IN SR1 WEB DOCUMENT RECOMMENDER SYSTEM

Dan MUNTEANU

"Dunărea de Jos" University of Galati
Faculty of Computer Science
Department of Computers and Applied Informatics
111 Domnească Street, 800201-Galati, Romania
Phone/Fax: (+40) 236 460182; (+40) 236 461353
E-mail: dan.munteanu@ugal.ro

Abstract: This paper presents a recommender system for web documents (given as bookmarks). The system uses for classification a combination of content, event and collaborative filters and for recommendation a modified Pearson-r algorithm. The algorithm for recommendation is using not only the correlation between users but also the similarity between classes. Some experimental results that support this approach are also presented.

Keywords: web document recommender system, content filtering, collaborative filtering, vector space model.

1. INTRODUCTION. RECOMMENDER SYSTEMS

Recommender systems were introduced as a computer-based intelligent technique to deal with the problem of information and product overload.

Recommender systems make a recommendation for a specific object by using evaluations for that object made by other users with similar interests. Examples of such systems are movie recommender systems like Moviefinder, MovieLens and Movie Critic, music recommender systems like CDNow's Album Advisor, Launch and book recommender systems like Amazon's Recommendation Center, Barnes and Noble's Recommended Reads. These systems ignore any information that can be extracted from the content.

The two basic entities which appear in any Recommender System are the user and the item. A user is a person who utilizes the Recommender System providing his opinion about various items and receives recommendations about new items from the system.

The goal of Recommender Systems is to generate suggestions about new items or to predict the utility of a specific item for a particular user.

This paper tries to present a recommender system that combine content filtering, collaborative filtering and agent technology. Every user has a personal agent which helps him to classify the information found on Internet and the information he had on his personal computer and also helps at recommending the documents to other users with similar interests. The agent suggests a classification of a document and extracts ratings for every document by analyzing user's actions (accept, reject, and modify agent's suggestion).

This paper was recommended for publication by Cornelia Novac-Ududec

2. SR1 - WEB DOCUMENT RECOMMENDER SYSTEM

The goal of the system is to assist the user in the process of classifying web documents and to automatically recommend them to other user with similar interest.

The system contains a database with bookmarks and references to local documents for each user and an agent that monitors the user's actions. When a document is registered, the agent suggests a classification in a category by analyzing the content of the document and user's profiles. The user can confirm the suggestion or choose another category which he considers to be better. In the meantime the agent checks to see if there are new bookmarks and recommends them to other users.

This system has two major components: one for classification and the other for recommendation. For classification it will use a text classification algorithm based on Rocchio's algorithm (Salton and Buckley, 1990). The difference is that the keywords used for representing the domain can be added and modified. The classifier uses relevance feedback (Douglas and Jinmook, 1998) when a document is added to the database by using implicit evaluation of the document. For updating the classifiers (that are used in the process of classification) the system uses the information gain measure to select the most informative keywords. The keywords will be words and roots of the words that are obtained using the Porter's stemming algorithm (Porter, 1980). A text classifier contains a number of keywords (128) that are manually selected (28) and the rest are extracted from the well classified documents.

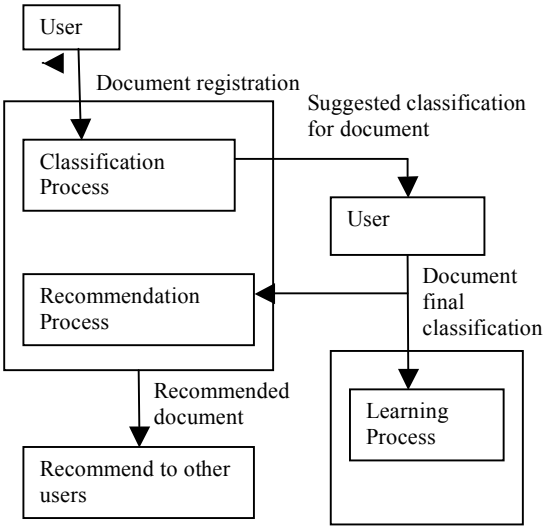

Fig. 1. System Architecture.

Documents and user profiles are represented using keywords vectors for comparing and learning. For a specific user, processing a lot of relevant documents correctly classified and irrelevant documents from a domain can lead to identify the relevant terms for that domain.

The system has a number of *n* categories to classify a document. From here the term category is considered to be similar with class, topic. In the same way document will represent web page, web document and bookmark.

3. RECOMMENDATION PROCESS IN SR1

The recommendation process in SR1 uses a modified *Pearson-r* algorithm (Breese, 1998), computing the correlation between users and modifying by adding the correlation between categories. The Pearson correlation coefficient was first defined in the context of the GroupLens project (Resnick *et al*., 1994) as the basis for the weights.

The agent constructs user-category matrix which will be used in the process of recommendation. The user-category matrix (M_{mxn}, m number of users and n number of categories) is constructed automatically counting for each user when a document is classified correctly in a class. This matrix is initialized with the categories chosen in the process of user registration.

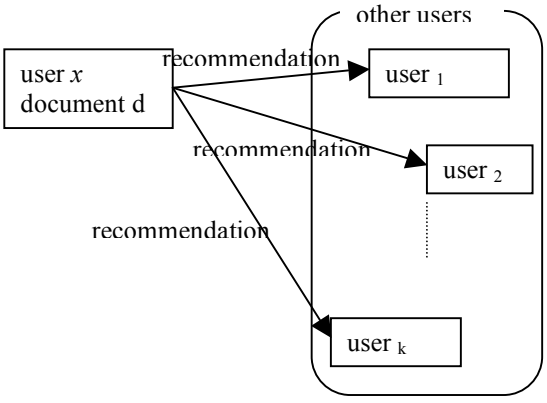

Fig. 2. Recommendation process

M	user 1	user 2	...	user k
category 1	7	7	...	3
category 2	4	3	...	2
...	...	...	...	...
category n	2	5	...	7

Fig. 3. User-category matrix

For the selection of the users who will receive recommendations the correlation between user *x* and the users from 1 to *k* must be computed. This way for every category it is computed the number of correctly classified documents. Using these values the correlation between users can be obtained.

User-category matrix is used to compute the correlation between user u_x and the rest of the users using *Pearson-r* algorithm and the users with the highest correlation are selected for recommendation.

$$corel(u_x,u_r)=\frac{\sum_{i=1}^{n}(u_{x,i}-\bar{u}_{x,i})(u_{r,i}-\bar{u}_{r,i})}{\sqrt{\sum_{i=1}^{n}(u_{x,i}-\bar{u}_{x,i})^2\sum_{i=1}^{n}(u_{r,i}-\bar{u}_{r,i})^2}} \quad (1)$$

in which

$$\bar{u}_{j,i}=\frac{\sum_{i=1}^{n}u_{j,i}}{n};\ j\in\{x,r\} \quad (2)$$

The problem with the above equation is that does not take into account the relation between categories. The agent may recommend a document to some users just because they are correlated with the initial user not because they are interested in the subject of the document and this is not a good recommendation. That's why the agent will increase the weight of the correlation between users interested in categories correlated with the class of the document. It is calculated the similarity between two classes for the user u_i, at the moment:

$$rel_i^t(c_m,c_n)=\frac{2\times\left|Q_{i,m}^t\cap Q_{i,n}^t\right|}{\left|Q_{i,m}^t+Q_{i,n}^t\right|} \quad (3)$$

in which $\left|A\cap B\right|$ is the number of common terms and $\left|A\right|$ is the number of terms from A.

Given class c_j of the document, class similarity vector is:

$$\vec{R}_j=<rel_i^t(c_j,c_1),rel_i^t(c_j,c_2),...,rel_i^t(c_j,c_n)>$$

where *n* is the number of classes.

This vector is multiplied by the user-category matrix and the result is a weighted user-category matrix (WM).

$$wm_{q,p}=rel_p^t(c_j,c_q)\cdot m_{q,p} \quad (4)$$

WM

		u_1	…	u_p	…	u_m
	c_1	$wm_{1,1}$	…	$wm_{1,p}$	…	$wm_{1,m}$
	…	…	…	…	…	…
c_j	c_q	$wm_{p,1}$	…	$wm_{q,p}$	…	$wm_{q,m}$
	…	…	…	…	…	…
	c_n	$m_{n,1}$	…	$a_{n,p}$	…	$a_{n,m}$

Using this new matrix it is computed the weight between user u_x (which recommends) and other users u_i (which may receive recommendation) as the correlation between them, using a threshold value of 0.5.

$$weight(u_x,u_i)=corel(u_x,u_i) \quad (5)$$

The agent also checks if the document isn't already in the database so the multiple recommendation of the same document to be avoided.

4. OTHER METHODS THAT COMBINE COLLABORATIVE AND CONTENT BASED TECHNIQUES

Other two different hybrid methods found in the literature that combine collaborative filtering techniques with content-based filtering are presented below:

First method is Content-Boosted Collaborative Filtering (Melville, *et al*., 2001) and the the idea is to use a content-based predictor to enhance existing user data, expressed via the user-item matrix, *M*, and then provide personalized suggestions through collaborative filtering. The content-based predictor is applied on each row from the initial user-item matrix, corresponding to each separate user, and gradually generates a pseudouser-item matrix, *PM*. At the end, each row, *i*, of the pseudo user-item matrix *PM* consists of the ratings provided by user u_i, when available, and those ratings predicted by the content-based predictor, otherwise. The pseudo

user-item matrix, *PM*, is a full dense matrix and collaborative filtering can be performed using *PM* instead of the original user-item matrix *M*.

The second approach is Combining Content-Based and Collaborative Filters and combines different filtering methods by first relating each of them to a distinct component and then basing its predictions on the weighted average of the predictions generated by those components (Claypool, *et al.*, 1999).. In its simplest version, it includes only two components: one component generates predictions based on content-based filtering while the second component is based on the classic collaborative filtering algorithm. At the beginning, when the number of user ratings is limited and thus, adequate neighbourhoods of similar users cannot be created, the content-based component is weighted more heavily. As the number of users is increased and more user opinions on items are collected, the weights are shifted more towards the collaborative filtering component, improving the overall accuracy of the prediction.

5. EXPERIMENTAL RESULTS

The experiment involved 5 users who used the system for a week and kept web pages with content relevant to their current research interests. At the end there were 65 bookmarks in 10 different classes. Out of 65 registered bookmarks only 42 were unique, which means that 23 of them (35,38% of all the bookmarks) actually came from recommendations. This indicates that intelligent information sharing and collaborative filtering occurred in high degree.

Table 1 Recommendation Acceptance Rate

User	Accepted	Rejected	Total	Accuracy (%)
1	11	2	13	84.6
2	7	3	10	70.0
3	9	1	10	90.0
4	6	2	8	75.0
5	1	2	3	33.3

As we can see in the above table, the overall acceptance rate was quite high for the majority of the users. In total, there were 44 recommendations, 34 (77.2%) of which were accepted.

6. CONCLUSIONS

This paper has presented a recommender system for web documents (given as bookmarks). The system uses for classification a combination of content, event and collaborative filters and for recommendation a modified Pearson-r algorithm.

This paper presented also an algorithm for recommendation in which not only the correlation between users is used but also the similarity between classes.

Some experimental results that support this approach were also presented. In the future SR1 recommender system should be tested with more users and should be compared it with other similar systems and also improve the efficiency of recommendation processes.

REFERENCES

Breese, J., D. Heckerman and C. Kadie, (1998). *Empirical Analysis of Predictive Algorithms for Collaborative Filtering*. In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*.

Claypool, M., A. Gokhale, T. Miranda, P. Murnikov, D. Netes and M. Sartin (1999). *Combining content-based and collaborative filters in an online newspaper*, in ACM SIGIR *Workshop on Recommender Systems-Implementation and Evaluation*. Berkeley, CA.

Douglas W. O. and K. Jinmook (1998). *Implicit Feedback for Recommender Systems*. Digital Library Research Group, College of Library and Information Services, University of Maryland

Melville, P., R. Mooney and R. Nagarajan (2001). *Content-boosted collaborative filtering*. ACM SIGIR *Workshop on Recommender Systems*. New Orleans, LA.

Porter, M.F. (1980). *An Algorithm For Suffix Stripping In*. In: Program 14 (3), pp. 130-137.

Resnick, P., N. Iacovou, M. Sushak, P. Bergstrom and J. Riedl (1994). *GroupLens: An Open Architecture for Collaborative Filtering of Netnews*. In the Proceedings of the *CSCW 1994 conference*.

Salton, G. and C. Buckley (1990). *Improving retrieval performance by relevance feedback*. In *Journal of the American Society for Information Science* Vol. 41, pp. 288-297