

METIOREW: An Objective Oriented Content Based and Collaborative Recommending System

David Bueno¹, Ricardo Conejo¹, Amos A. David²

¹ Department of Languages and Computer Science, University of Málaga,
29071, Málaga, Spain.

{bueno, conejo}@lcc.uma.es

² LORIA, BP 239, 54506 Vandoeuvre, FRANCE.
adavid@loria.fr

Abstract. The size of Internet has been growing very fast and many documents appear every day in the Net. Users find many problems to obtain the information that they really need. In order to help users in this task of finding relevant information, recommending systems were proposed. They give advice using two methods: the content-based method that extracts information from the already evaluated documents by the user in order to obtain new related documents; the collaborative method that recommends documents to the user based on the evaluation by users with similar information need. In this paper we will present our approach through the employment of a user model and analyze some existing Web recommending systems and identify some problems that we try to solve in our system METIOREW. Some of the problems in document recommendation are: a) how to begin with document recommendation to users at the beginning of interaction when there is little or no knowledge on the user, b) how to make document recommendation to the user with changing information needs (objectives) without employing the general preferences of all the users but employing explicit individualized user model that integrates the user's objectives, c) how to provide access to the user's past history in order to review interesting documents related to specific objectives. The algorithms that we propose for calculating the degree of relevance of documents based on our user model is also explained.

1 Introduction

The size of Internet has been growing very fast and many documents appear every day in the Net. Users find many problems to obtain the information that they really need. In order to help users in this task of finding relevant information *recommending systems* are proposed. Due to this great amount of information that comes from everywhere, recommending systems are needed to filter junk e-mail [27], to obtain only the relevant news from Usenet, like GroupLens [25] or URN [10], to get only the interesting World News for the user [7] and probably the most important, to find information in the WWW. In this article we will concentrate on this last type of systems.

In the next subsections we will present the three methods that exist to make recommendations on the Web. In section 2 we will see the work that has been done in this area focusing on the advantages and disadvantages of each. In section 3 we will describe our Web Recommending system METIOREW that tries to solve most of the identified problems in the systems of section 2. Section 4 describe the state of development of this system and the lines that we are following.

1.1 Content-Based Recommendations

In order to recommend a document to a user some systems use only the content of the documents. To do so, documents are represented with a set of features like title, author, keyword, etc. When a user evaluates a document as interesting this set of features is used to look for similar documents. A user model or profile is constructed from different evaluations that lead the systems to know more about the user preferences. This model has been constructed only using the features of the documents. That's why they are called *content-based recommending systems*.

1.2 Collaborative Recommendation

The activities of many users on an Information Retrieval System (IRS) are often very similar because they have similar preferences or related interest. This means that the difficulties to find interesting information are repeated for each of these users. A possible solution to avoid work already done is to share the result of a user between other users with "similar interest".

A pure collaborative recommending system offers documents to the user not because of its content but because there is a similar user who has evaluated them as interesting. This means that in this case the similarity is between people that evaluated in the same way some documents without taking into account the content of those documents.

The problem with this kind of systems relates to new documents. Until somebody evaluates them the system has no information regarding their relevance and the system will not be able to recommend them. Another important problem relates to the number of users of these systems. The fewer the number of users the lower the probability of evaluating the documents with the same interest. In this case, the system will not be efficient.

1.3 Hybrid Solution

The content-based method has been used in IR area with interesting results. But the idea of completing the recommended documents, that has been evaluated by the user, with other documents that has been retrieved by other users with similar

characteristics looks like very promising. This hybrid solution is more efficient than each method applied separately.

2 Related Work

In this section we will analyze the most representative recommending systems that apply some of the methods explained above. We will concentrate on the main aspects that differentiate the systems from each other.

WebWatcher¹. [1] This system makes only content-based recommendations. The user expresses what he's looking for using keywords that define his goal. The goals are restricted to technical reports and the keywords can be on author, title, etc. The user navigates through the Web under the supervision of WebWatcher that will assist him by highlighting the links that are closer to the keywords of the goal. To calculate the degree of relevance they use the methods of Winnon [20], Wordstat, TFIDF [26] and Random.

Letizia. [19] This is another content-based system that recommends web documents. The user doesn't need to enter information about his information need. Letizia supervises his actions and uses some heuristics to determine what's interesting for the user. For example, if a user makes a bookmark of a document, it means that he's interested in it. Other less strong heuristic is that if a user analyses the links of a document, the document is most likely related to his information need. Documents are represented by list of keywords.

Syskill & Webert. [23] Using content-based recommendation this system predefines some topics that can be the possible goals of the users. An index for each topic has been manually created. When the user evaluates some documents of this index the system can recommend the most related pages with the pages already evaluated. The algorithm to select relevant documents is a Bayesian classifier. Also LYCOS' queries can be constructed by the system.

FAB. [2-4] FAB is an adaptive collaborative web recommending system. It has different kind of agents: collection agents (to look for new information of a limited number of topics), selection agents (one for user who has a model of him in order to recommend the most interesting documents) and a central router (who send pages obtained from the collection agents to selection agents of users with a similar profile to the content of the pages). The user regularly receives a list of pages to evaluate. This information is used to update the original collection agent (that is not attached to this user) and his selection agent. This agent uses the TFIDF [26] to obtain the keywords of the document and the cosine similarity measure to calculate the

¹ WebWatcher <http://www.cs.cmu.edu:8001/afs/cs.cmu.edu/project/theo-6/web-agent/www/project-home.html>

similitude between the user profile and the document. The best-evaluated documents are sent to other users with similar profile.

PTV. [12] This system recommends Television programs through the WWW and the WAP. There is a user profile composed of the channels, keywords, programs, etc., that interest the user. He can update the model but the best way is through relevant feedback. The system selects the k users most similar to the actual one and offers the r best programs for this user. When a recommendation is asked by the user a list of programs is shown, some of them selected from those r programs and others from content recommendations.

MOVIELENS². [13] It recommends films to the users. For this, it recommends films using the information of other users with similar video preferences and also information based on the user's previous evaluations. They use different agents to collect information using different methods and combine them to obtain better results. In their experiment they make comparison when using only content-based information or possible combination with one or more agents, and they conclude that the best solution is the mixture of several agents and the information based on the user feedback.

Casper/Jobfinder. [8; 24] Casper helps to find a new job. It works making case based reasoning. It evaluates each possible new job comparing it with the jobs already evaluated and proposes it if it is the most similar to one that has interested to the user. With this idea they restrict the selection problem to a classification one. They use a standard weighted-sum metric to calculate the similarity, and as features they use the kind of work, salary, experience, etc. Casper is also collaborative because it makes recommendations from similar user, where the calculation of similarity is done using the number of different jobs that they have evaluated in common.

GASS. [5] It pretends that a group of people with the same goal looks for information in the Web, and that this information will be shared between them. For this they need to have a group model besides a user model.

WebCobra. [29] It's also a recommending system where initially the user evaluates a set of documents from where a vector of keywords is extracted that will be used to identify this user. This vector is sent to a server that uses the simple cosine method to calculate the similarity and assigns a user to a group. When the user evaluates other documents he selects which of them are the best to send to the partners of the group. The subjects for the groups are concentrated in very specific domains to facilitate the task of a group. The user can ask for recommendations and he will receive the documents marked as interesting by other partners.

We will resume here some limitations of these systems and present how we have tried to solve the problems in our system METIOREW.

² MovieLens <http://movielens.umn.edu/>

The first problem relates to the creation of a model of a user on which the system has little or no information. This is a typical problem in User Modeling. One of the solutions to solve this problem is through the use of stereotypes [17] [15]. Letizia, Syskill, Webert and Casper have difficulties at the beginning to give advice to the user because the model is empty. In MOVIELENS, a little training is needed by the system that offers a list of films to evaluate to create the initial model. In FAB, at the beginning, a list of documents is given to the user to evaluate. FAB has an ‘amalgamated profile’ with the documents that are most interesting for the actual population of the system and from which n documents are offered to the user to begin to create the model. WebWatcher and WebCobra begin to create the model using some initial keywords that nearly fix the model to them.

Other important problem that we have found in those systems is the global vision of the people’s interests. It looks like if somebody will always want to find the same information in the web. Letizia, FAB, PTV, MOVIELENS or WebCobra don’t define the concept of goal or objective. But even the systems that define it are very restrictive or let the user to have only one goal. For example WebWatcher allows one goal restricted to technical reports, Syskill & Webert allows the selection of a limited number of topics for which some index has been manually constructed, GASs supposes that many people have the same one single goal.

The last important concept is related to the manipulation of the history. Everybody has bookmarks in his Web browser with the most relevant URLs. Many users group them by topics. But why can’t we find the documents that we have already evaluated as FAB does? From the systems analysed only³ MOVIELENS allows the review of the evaluated documents.

3 METIOREW description

METIOREW is a collaborative and content-based Web recommending system. It recommends documents to the user by trying to solve the problems presented in section 2. The first aspect of METIOREW is that it is objectives oriented where an objective expresses an information need. We can relate users and objectives in the following manners in an Information Retrieval System a) a user’s information need can evolve, b) the user can have the same information need at different times and c) different information needs can be related. Related information needs does not have the same degree of similarity. The same methodology is used in METIORE [11].

METIOREW allows the user to review the documents already evaluated through the user’s history. The user also has the possibility to modify the evaluations attached to the documents. This can be interpreted as an “intelligent bookmark” organized by

³ The information we have from these systems is based on the content of their publications, perhaps there are other features that haven’t been documented

user when needed. The *keyword agent* does a description of documents content in order to know the real relevance of these documents for the user.

Keyword Agent. This agent receives a Web page and generates a set of keywords that describe it. In 3.4 we have the algorithm to select the features that represent a document.

Collaborative Agent. Its goal is to offer relevant information to the user taking as reference documents that have already been evaluated by other users with similar objective with the current user's objective. Only documents with degree of similarity superior to a predefined threshold are proposed. The agent searches for the most similar objectives by comparing the models and for each of the objective retrieves a list of pages that will be sorted according to the degree of relevance.

Mail Agent. There is one mail agent for each user. It's activated with a timer defined by the user (for example once day or once week). Its mission is to examine the list of recommendations generated by the collaborative and search agent and send the N best links for each objective to the user through mail. This lets the user define different objectives, improve their model in different sessions, thus allowing the system to retrieve relevant documents without the user's interaction.

3.2 Functionality

When the user begins a session with a new objective the *personal agent* asks the user to insert a textual description of his current objective and a list of initial keywords that will describe it. Then a *search* and a *collaborative agent* are initiated to look for related pages. The user can also begin to navigate freely on the Web in a supervised way. If he finds by himself relevant documents he will give a feedback that is used to update the model. Whenever the user asks for recommendation the *personal agent* will look for the new documents that have been found for this objective and they are proposed in a list.

As the initial model (real model) is only restricted to a list of keywords, METIOWE tries to improve it using the model (external model) of the user with the most similar objective to the current one. The possible recommendations of the two models are used to make new recommendations. Each relevant feedback serves to improve the real model. The external model is used until the real one is enough independent (at least 10 positive feedback). Also the possibilities of disable or change the external model for another one are contemplated. This will be use when a high percentage of the recommendations of the external model are rejected.

After evaluating some documents the user model will be refined and composed of the initial keywords (that will have an important weight because they have been directly selected by the user) and new ones obtained from the documents evaluated by the user. The *search* and *collaborative agents* use the current information of the model to search for new related documents that are kept in a repository for this user.

The *personal* and *mail agents* consult it to generate recommendations. The documents of the repository are sorted by degree of relevance to the objective.

3.3 User Model

The user model keeps all the information needed to personalize the interactions with the user. In METIOREW we keep diverse information that we resume in Table 1. The model is objective oriented. This means that for each user we can have several models depending on the different information needs. With this representation the same user will be able to work in different sessions with different objectives, but having the possibility of review past sessions through the information acquired by the system.

Documents Revised	URL, keywords, evaluation	Used to regenerate the keyword synthesis, and for the review of the user's history
Keyword synthesis	Keyword, ev1, ev2, ev3, ev4	Each relevant keyword and the frequency for each of the four kinds of evaluations are kept in the system
Related objectives	User, objective, % similitude	List of objectives found by the collaborative agent as similar to the current objective
Documents to recommend	URL, keywords, %similitude	Documents obtained by the search and collaborative agents that hasn't been evaluated

Table 1. Content of the user model

The user's relevant feedback is fundamental to make the personalized recommendations. In METIOREW we use four kinds of feedback. *OK* (The document is interesting for the user), *KNOWN* (The document is interesting for the user but he already knew it), *BOF* (With the current knowledge of the user, he can not determine if this document is interesting or not), *ERROR* (This document is not relevant to user's objective).

3.4 Calculation of the degree of relevance

In this section we describe briefly the methods used to calculate the degree of similarity between a document and the user model (keyword synthesis), between two objectives and how we extract the relevant keywords from the documents and the model.

Obtaining relevant keywords. The keyword agent will obtain the features to describe the web page. It makes it using the Term Frequency TF [14] by applying some heuristics such as "remove the most and least repeated words". It is expected that this will provide the best *m* words that describe the document.

Classifying new pages in the user model. After each evaluation by the user, the synthesis of the keywords in user model is updated. Increasing by one the frequency of the evaluation for each keyword that represents the document evaluated does this. When a new page arrives the system must predict how the user will evaluate it. To do that we compare the vector of features of this document with the user model for the

current objective using an adaptation of the Naive Bayes [18] that has been proved to be a good classifier in [18] [28] [21] [16] [30].

Objectives similarity. To find similar models is needed to compare different objectives. For this we use the Pearson Correlation [6] that we adapt to the representation of our synthesis model. In the eq. 1 $w(a,i)$ is the similitude between the objectives a and i . $v_{i,j}$ is the probability that the user with objective i (u_i) evaluates as interesting the element j . Where interesting means classify as *ok(c1)* or *known(c2)* and I_i is the set of features on which the user has given a feedback⁴.

$$w(a,i) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - \bar{v}_i)^2}} \quad (1)$$

$$v_{i,j} = P(c1, c2 / u_i \wedge j) \quad (2) \quad \bar{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j} \quad (3)[9]$$

Obtaining representative features for searching. To create a vector that represents the model of the current objective we use the eq. 2. It gives the probability of evaluating as correct each feature. Sorting this we obtain the n best keywords to be used by the *search agent*.

4 Future Work

The system presented here is in the phase of development and we are planning its experimentation in a real situation. This experimentation will be composed firstly by the analysis of information collected by the system, basically efficiency in recommendations and percentage of correct prediction of feedback. Besides that, we elaborate a questionnaire to be filled by the users in order to make a correlation between the system's proposal and the user's opinion. We are also working on the improvement of the *personal agent* as a 3D agent in the style of Pazzani [7] that makes the user feel a more human interaction with the computer.

5 References

1. Armstrong, R., Freitag, D., Joachims, T., & Mitchell, T. (1995). "Webwatcher: A learning apprentice for the world wide web". AAAI Spring Symposium on Information Gathering from Heterogeneous Distributed Environments
2. Balabanovic, M. (1997). "An Adaptive Web Page Recommendation Service". Proceedings of the First International Conference on Autonomous Agents Marina del Rey, CA

⁴ The user doesn't gives feedback for features but for documents. The main features (or keywords) of a document inherit its feedback.

3. Balabanovic, M., & Shoham, Y. (1995). "Learning Information Retrieval Agents: Experiments with Automated Web Browsing". AAAI Spring Symposium on Information Gathering, Stanford, CA
4. Balabanovic, M., & Shoham, Y. (1997). "Combining Content-Based and Collaborative Recommendation". *Communications of the ACM*, 40(3)
5. Barra, M. (2000). "Distributed Systems for Group Adaptivity on the Web". *Adaptive Hypermedia and Adaptive Web-Based Systems* Springer-Verlag
6. Billsus, D., & Pazzani, M. (1998). "Learning Collaborative Information Filters". *Proceedings of the International Conference on Machine Learning* Madison, Wisc. Morgan Kaufmann Publishers
7. Billsus, D., & Pazzani, M. (1999). "A Hybrid User Model for News Story Classification". *Proceedings of the Seventh International Conference on User Modeling (UM '99)* Banff, Canada
8. Bradley, K., Rafter, R., & Smyth, B. (2000). "Case-Based User Profiling for Content Personalisation". *Adaptive Hypermedia and Adaptive Web-Based Systems* Springer-Verlag
9. Breese, J., Heckerman, D., & Kadie, C. (1998). "Empirical Analysis of Predictive Algorithms for Collaborative Filtering". *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* Madison, WI. Morgan Kaufmann Publisher
10. Brewer, R. S., & Johnson, P. M. (1994). "Toward Collaborative Knowledge Management within Large, Dynamically Structured Information Systems". University of Hawaii, Dpt. of Information and Computer Science. Honolulu:
11. Bueno, D., & David, A. A. " METIORE: A Personalized Information Retrieval System". 8 International Conference on User Modeling. UM2001
12. Cotter, P., & Smyth, B. (2000). "WAPing the Web: Content Personalisation for WAP-Enabled Devices". *Adaptive Hypermedia and Adaptive Web-Based Systems* Springer-Verlag
13. Good, N., Schafer J. , Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., & Riedl, J. (1999). "Combining collaborative filtering with personal agents for better recommendations". In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*
14. Joachims, T. (1997). "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization". *Proc. of the 14th International Conference on Machine Learning ICML97* (pp. 143-151).
15. Kay, J. (1995). "Vive la difference! Individualised interaction with users". *IJCAI95*
16. Keogh, E., & Pazzani, M. (1999). "Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches.". *Uncertainty 99, 7th. Int'l Workshop on AI and Statistics*, (pp. 225-230). Ft. Lauderdale, Florida
17. Kobsa, A. (1993). "User Modeling: Recent Work, Prospects and Hazards".
18. Kononenko, I. (1990). "Comparison of Inductive and Naive Bayesian Learning Approaches to Automatic Knowledge Acquisition". *Current Trends in Knowledge Acquisition*, 190-197
19. Lieberman, H. (1995). "Letizia: An Agent That Assists Web Browsing". *International Joint Conference on Artificial Intelligence* Montreal, CA.
20. Littlestone, N. (1988). "Learning quickly when irrelevant attributes abound". *Machine Learning*, 2:4, pp. 285-318. Notes: Pedido a biblioteca
21. Mitchell, T. M. (1997). "Machine Learning". The McGraw-Hill Companies, Inc.
22. Nwana, H. (1996). "Software Agents: An Overview". *Knowledge Engineering Review*
23. Pazzani, M., Muramatsu, J., & Billsus, D. (1996). "Syskill & Webert: Identifying interesting web sites". *AAI Spring Symposium on Machine Learning in Information Access*. URL= <http://www.parc.xerox.com/istl/projects/mlia/papers/pazzani.ps>.

24. Rafter, R., Bradley, K., & Smyth, B. (2000). "Automated Collaborative Filtering Applications for Online Recruitment Services". Adaptive Hypermedia and Adaptive Web-Based Systems ItalySpringer-Verlag
25. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). "GroupLens: An Open Architecture for Collaborative Filtering of Netnews". Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work (pp. 175-186).
26. Rocchio, J. (1971). "*Relevance Feedback in Information Retrieval*". The SMART Retrieval System: Experiments in Automatic Document Processing, 313-323Notes: No lo tengo aun.
27. Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). "A bayesian approach to filtering junk e-mail". AAAI-98 Workshop on Learning for Text Categorization
28. Singh, M., & Provan, G. M. (1996). "Efficient learning of selective Bayesian network classifiers". . Proceedings of the 13th International Conference on Machine Learning
29. Vel, O., & Nesbitt, S. (1997). "A Collaborative Filtering Agent System for Dynamic Virtual Communities on the Web". URL= <http://citeseer.nj.nec.com/de-collaborative.html>.
30. Versteegen, L. (2000). "The Simple Bayesian Classifier as a Classification Algorithm". URL= <http://www.cs.kun.nl/nsccs/artikelen/leovv.ps.Z>.