Recommendation System Based on the Discovery of Meaningful Categorical Clusters

Nicolas Durand[1], Luigi Lancieri[1], and Bruno Crémilleux[2]

[1]France Telecom R&D
42 rue des Coutures - 14066 Caen Cédex 4 - France
{nicola.durand, luigi.lancieri}@francetelecom.com
[2]GREYC CNRS-UMR 6072, Université de Caen
Campus Côte de Nacre - 14032 Caen Cédex - France
bruno.cremilleux@info.unicaen.fr

Abstract. We propose in this paper a recommendation system based on a new method of clusters discovery which allows a user to be present in several clusters in order to capture his different centres of interest. Our system takes advantage of content-based and collaborative recommendation approaches. The system was evaluated by using proxy server logs, and encouraging results were obtained.

1 Introduction

The search of relevant information on the World Wide Web is still a challenge. Even if the indexing methods get more efficient, search engines stay passive agents and do not take into account the context of the users. Our approach suggests an active solution based on the recommendation of documents. We propose in this paper a hybrid recommendation system (*collaboration via content*). Our method allows to provide recommendations based on the content of the document, and also recommendations based on the collaboration. Recommendation systems can be installed on the user's computer (like an agent for recommending web pages during the navigation), or on particular web sites, platforms or portals. Our approach can be applied to systems where users' consultations can be recorded (for example: a proxy server, a restricted web site or a portal).

Our system is based on recent KDD (Knowledge Discovery in Databases) methods. In [2], we defined a new method of clusters discovery from frequent closed itemsets. In this paper, we create a recommendation system by taking advantage of this method. We form clusters of users having common centres of interest, by using the keywords of the consulted documents. Our method relates to content-based filtering (the identification of common keywords) but uses some technics of collaborative filtering (i.e. clustering of users). Moreover, we discover a set of clusters and not a strict clustering (i.e. a partition) like the recommendation systems based on clustering. This means that our approach enables a user to be in several clusters. We can retrieve a user with several kinds of queries corresponding to his different centres of interest. Our system is autonomous, it does not need the intervention of users. Indeed, we use logs (consultations of documents for several users) that are a good source of information to indicate what the users want [13]. We can perform both content-based recommendations and collaboration-based recommendations. For a new document arriving in the system, we can recommend it to users by comparing the keywords of the document to the clusters. We can also recommend to a user of a cluster, documents that other users of the cluster have also consulted (collaborative approach).

The rest of the paper is organized as follows: in the next section, we present related work. Then, we detail our method of clusters discovery (called Ecclat). In Section 4, we explain our recommendation system based on the discovered clusters. Then, we describe our experimentations and give some results. We conclude in Section 6.

2 Related Work

Recommendation systems are assimilated to information filtering systems because the ideas and the methods are very close. There are two types of filtering: content-based filtering and collaborative filtering.

Content-based filtering identifies and provides relevant information to users on the basis of the similarity between the information and the profiles. We can quote the *Syskill&Webert* system [12] which produces a bayesian classifier from a learning database containing web pages scored by the user. The classifier is then used to establish if a page can interest the user. *SiteHelper* [9] recommends only the documents of a web site. It uses the feedback of the user. *Letizia* [6] is a client-side agent which searches web pages similar to the previous consulted or bookmarked ones. *WebWatcher* [4] uses the proxy server logs to do some recommendations. Mobasher et al. [7] propose a recommendation system based on the clustering of web pages from web server logs. This system determines the URLs which can interest the user by matching the URLs of the user current session with the clusters.

Collaborative filtering finds relevant users who have similar profiles, and provides the documents they like to each other. Rather than the similarity between documents and profiles, this method measures the similarity between profiles. *Tapestry* [3] and *GroupLens* [5] allow users to comment the Netnews documents, and to get the ones recommended by the others. *Amalthaea* [8] is an agent which allows to create and modify the user profile. In these systems, users have to specify their profiles. Among the autonomous collaborative approaches, we have some methods based on the clustering and the associations of items. Wilson et al. [14] use the frequent associations containing two items (TV programs) in order to determine the similarity between two user profiles.

There are also some hybrid approaches. Pazzani [11] showed by some experimentations that the hybrid systems use more information, and provide more precise recommendations. Pazzani talk about *collaboration via content*, because the profile of each user is based on the content, and is used to detect the similarity among the users. *Fab* [1] implements this idea in a similar way. In *Fab*, some agents (one per user) collect documents and put them in the central repository (to take advantage of potential overlaps between user's interests) in order to recommend them to users. We can also cite *OTS* [15] which allows a set of users to consult some papers provided by a publication server. The users are grouped according to their profile. These profiles are defined and based on the content of papers. Contrary to *OTS*, our system can provide recommendations on documents not consulted by users yet, and our method of cluster discovery does not use defined profiles.

3 Clusters Discovery with Ecclat

We have developed a clustering method (named Ecclat [2]) for the discovery of interesting clusters in web mining applications i.e. clusters with possible overlapping of elements. For instance, we would like to retrieve a user (or a page) from several kinds of queries corresponding to several centres of interest (or several points of views). Another characteristic of Ecclat is to be able to tackle large data bases described by categorical data. The approach used by Ecclat is quite different from usual clustering techniques. Unlike existing techniques, Ecclat does not use a global measure of similarity between elements but is based on an evaluation measure of a cluster. The number of clusters is not set in advance. In the following discussion, each data record is called a *transaction* (a user) and is described by *items* (the consulted keywords).

Ecclat discovers the frequent closed itemsets [10] (seen as potential clusters), evaluates them and selects some. An itemset X is frequent if the number of transactions which contains X is at least the frequency threshold (called $minfr$) set by the user. X is a closed itemset if its frequency only decreases when any item is added. A closed itemset checks an important property for clustering: it gathers a maximal set of items shared by a maximal number of transactions. In other words, this allows to capture the maximum amount of similarity. These two points (the capture of the maximum amount of similarity and the frequency) are the basis of our approach of selection of meaningful clusters.

Ecclat selects the most interesting clusters by using a cluster evaluation measure. All computations and interpretations are detailed in [2]. The cluster evaluation measure is composed of two measures: *homogeneity* and *concentration*. With the *homogeneity* value, we want to favour clusters having many items shared by many transactions (a relevant cluster has to be as homogeneous as possible and should gather "enough" transactions). The *concentration* measure limits the overlapping of transactions between clusters. Finally, we define the *interestingness* of a cluster as the average of its *homogeneity* and *concentration*.

Ecclat uses the *interestingness* to select clusters. An innovative feature of Ecclat is its ability to produce a clustering with a minimum overlapping between clusters (which we call "*approximate clustering*") or a set of clusters with a slight overlapping. This functionality depends on the value of a parameter called M. M is an integer corresponding to a number of transactions not yet classified that must be classified by a new selected cluster. The algorithm performs as follows. The cluster having the highest *interestingness* is selected. Then as long as there are transactions to classify (i.e. which do not belong to any selected cluster) and some clusters are left, we select the cluster having the highest *interestingness* and containing at least M transactions not classified yet.

The number of clusters is established by the algorithm of selection, and is bound to the M value. Let n be the number of transactions, if M is equal to 1, we have at worst $(n - minfr + 1)$ clusters. In practice, this does not happen. If we increase the M value, the number of clusters decreases. We are close to a partition of transactions with M near to $minfr$.

4 Recommendation System

In this section, we present the basis of our recommendation system. It is composed of an off-line process (clusters discovery with ECCLAT) and an on-line process realizing recommendations. The on-line process computes a score between a new document and each of the discovered clusters. For a document and a cluster, if the score is greater than a threshold, then the document is recommended to the users of the clusters. We can also use the collaboration and recommend the documents that the users of a cluster have consulted to any user of a cluster. At this moment, we concentrate ourselves on the first type of recommendations.

The score between a document and a cluster is computed as follows. Let D be a document and K_D be the set of its keywords. Let C_i be a cluster, C_i is composed of a set of keywords K_{C_i} and a set of users U_{C_i}. We compute the covering rate :

$$CR(D, C_i) = \frac{|K_D \cap K_{C_i}|}{|K_D|} * 100$$

Let *mincr* be the minimum threshold of the covering rate. If $CR(D, C_i) \geq$ *mincr*, then we recommend the document D to the users U_{C_i}.

Let us take an example, a document K_D={`fishing hunting england nature river rod`}, and the following clusters:

- K_{C_1}={`fishing hunting internet java`}, CR=33%.
- K_{C_2}={`fishing england`}, CR=33%.
- K_{C_3}={`fishing hunting england internet java programming`}, CR=50%.
- K_{C_4}={`fishing`}, CR=16%.
- K_{C_5}={`internet java`}, CR=0%.

In this example, we have the following order: $C3 > C1, C2 > C4$, and $C5$ is discarded. Let us remark that the used measure (CR) is adapted to the problem, because in a cluster, keywords can refer to different topics. For instance, if a set of users are interested in `fishing` and `programming`, it is possible to have a corresponding cluster like C_3. This point does not have to influence the covering rate. For this reason, we select this measure which depends on the common keywords between the document and the cluster, and on the number of the keywords of the document. CR does not depend on the number or the composition of the keywords set of the cluster. The other classical measures like Jaccard, Dice, Cosine, are not adapted to our problem. The possible mixing of topics does not influence the recommendations, but *mincr* does not have to be too high, because the number of keywords for a cluster is free, and for a document, it is fixed. Another remark, if a user is very interested in `C++` and if he is the only one, we do not detect this. We take into account the common interests shared by the group.

5 Experimentation

In order to evaluate recommendations, we used proxy server logs coming from France Telecom R&D. This data contains 147 users and 8,727 items. Items are keywords of the HTML pages browsed by 147 users of a proxy-cache, over a period of 1 month. 24,278 pages were viewed. For every page, we extracted a maximum of 10 keywords with an extractor (developed at France Telecom R&D) based on the frequency of significant words.

Let L be the proxy server log. For a document D in L, we determine the users interested by D (noted $UsersR(D)$), by using the previous discovered clusters. Then, we check by using the logs, if the users who have consulted the document (noted $Users(D)$) are present in $UsersR(D)$. Let us remark that we do not use a web server where the sets of documents and of keywords are known and relatively stable over time. For a proxy server, the set of documents and especially the set of keywords can be totally different between two periods. So we used the same period to discover the clusters and the recommendations for a first evaluation without human feedbacks.

We use the following measures to evaluate the results:

$$failure(D) = \frac{|Users(D) - UsersR(D)|}{|Users(D)|}$$

$$r\_hit(D) = \frac{|UsersR(D) \cap Users(D)|}{|UsersR(D)|}$$

The *failure* rate evaluates the percentage of users who consulted a document that has not been recommended. The *r_hit* value (recommendation hit) measures the percentage of users indicated in the recommendations of a document from those who really consulted it.

We set $minfr$ to 10%. It corresponds to a minimal number of 14 users per cluster. The number of frequent closed itemsets is 454,043. We set M to 1 in order to capture the maximum of different centres of interest (overlapping between clusters), we find 45 clusters (the average number of users per cluster is 21). Let us note that here, our aim is not to study the impact of the parameters. This has already been done in [2].

The choice of the *mincr* value is not easy. The *mincr* value influences especially the number of recommended documents. The higher the *mincr* value is, the lower the number of recommended documents is. Too many recommendations make the system unpractical. We need to have a compromise between the number of recommended documents and, as we could guess, the quality of the system. For the evaluation, we did not really perform recommendations to users, we just evaluated the accuracy of our recommandations. So we used a relatively low value of *mincr* in order to have a lot of recommendations. We set *mincr* to 20%. The system has recommended 11,948 documents (49.2% of the total).

In Figure 1, we remark that 80% of the documents (among 11,948) are well recommended, and we have only 16% of failure. We ranked the documents according to the *r_hit* values and we obtained Figure 2. We can deduce from the *r_hit* measure that the number of users who are in the results and have not consulted the document is not null. We found more users, maybe they would have been interested, but we cannot verify it. It would be necessary to have human feedbacks.

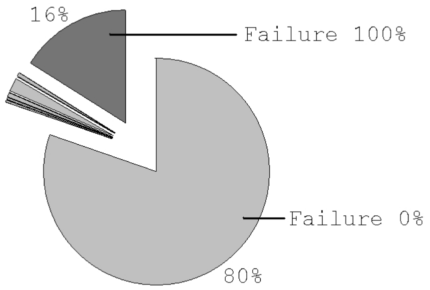

Fig. 1. Distribution of the documents according to the failure rate, *mincr*=20%.

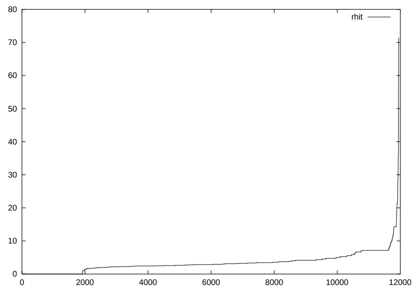

Fig. 2. *r_hit* value according to the rank of the documents, *mincr*=20%.

6 Conclusion

We have presented a recommendation system based on the discovery of meaningful clusters of users according to the content of their consulted documents. Our method of clusters discovery allows to capture the various centres of interest for the users because of the possibility to have a user in several clusters and so retrieve him with several kinds of queries. We provided recommendations of documents using the discovered clusters. We evaluated our method on proxy server logs (not usually done in this application), and we obtained good results, that is encouraging for other experiments (with human feedbacks) and the development of our system. In future works, we will evaluate the second type of possible recommendations i.e. based on the collaboration. We will also look for an incremental version of ECCLAT in order to propose a system in pseudo real-time.

References

1. M. Balabanovic. An Adaptive Web Page Recommendation Service. In *the 1st International Conference on Autonomous Agents*, pages 378–385, Marina del Rey, CA, USA, February 1997.
2. N. Durand and B Crémilleux. ECCLAT: a New Approach of Clusters Discovery in Categorical Data. In *the 22nd Int. Conf. on Knowledge Based Systems and Applied Artificial Intelligence (ES'02)*, pages 177–190, Cambridge, UK, December 2002.
3. D. Goldberg, D. Nichols, B.M. Oki, and D. Terry. Using Collaborative Filtering to Weave an Information Tapestry. *Communication of the ACM*, 35(12):61–70, 1992.
4. T. Joachims, D. Freitag, and T. Mitchell. WebWatcher: A Tour Guide for the World Wide Web. In *the 15th Int. Joint Conference on Artificial Intelligence (IJCAI'97)*, pages 770–775, Nagoya, Japan, August 1997.
5. J.A. Konstan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, and J. Riedl. GroupLens: Applying Collaborative Filtering to Usenet News. *Communication of the ACM*, 40(3):77–87, March 1997.
6. H. Lieberman. Letizia: An Agent that Assists Web Browsing. In *the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 924–929, Montral, Qubec, Canada,, August 1995.
7. B. Mobasher, R. Cooley, and J. Srivastava. Creating Adaptive Web Sites through Usage-Based Clustering of URLs. In *IEEE Knowledge and Data Engineering Exchange Workshop (KDEX99)*, Chicago, november 1999.
8. A. Moukas. Amalthaea: Information Discovery and Filtering Using a Multi-Agent Evolving Ecosystem. *International Journal of Applied Artificial Intelligence*, 11(5):437–457, 1997.
9. D.S.W. Ngu and X. Wu. SiteHelper : A Localized Agent that Helps Incremental Exploration of the World Wide Web. In *the 6th international World Wide Web Conference*, pages 691–700, Santa Clara, CA, 1997.
10. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient Mining of Association Rules Using Closed Itemset Lattices. *Information Systems*, 24(1):25–46, Elsevier, 1999.
11. M Pazzani. A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review*, 13(5):393–408, 1999.
12. M Pazzani, J. Muramatsu, and D. Billsus. Syskill & Webert: Identifying interesting web sites. In *Proceedings of the 13th National Conference on Artificial Intelligence*, pages 54–61, Portland, Oregon, 1996.
13. M. Spiliopoulou. Web Usage Mining for Web site Evaluation. *Com. of the ACM*, 43(8):127–134, August 2000.
14. D. Wilson, B. Smyth, and D. O'Sullivan. Improving Collaborative Personalized TV Services. In *the 22nd Int. Conf. on Knowledge Based Systems and Applied Artificial Intelligence (ES'02)*, pages 265–278, Cambridge, UK, December 2002.
15. Y.H. Wu, Y.C. Chen, and A.L.P. Chen. Enabling Personalized Recommendation on the Web Based on User Interests and Behaviors. In *the 11th Int. Workshop on Research Issues in Data Engineering (RIDE-DM 2001)*, pages 17–24, Heidelberg, Germany, April 2001.