A DISTRIBUTED INFORMATION FILTERING : STAKES AND SOLUTION FOR SATELLITE BROADCASTING

Sylvain Castagnos, Anne Boyer, François Charpillet
LORIA - INRIA Lorraine
Campus Scientifique, B.P. 239
54506 Vandoeuvre-lès-Nancy Cedex, France

Keywords: Collaborative filtering, user profiling, distributed algorithm, internet broadcasting per satellite, privacy.

Abstract: This paper is a preliminary report which presents information filtering solutions designed within the scope of a collaboration between our laboratory and the company of broadcasting per satellite SES ASTRA. The latter have finalized a system sponsored by advertisement and supplying to users a high bandwidth access to hundreds of web sites for free. This project aims at highlighting the benefits of collaborative filtering by including such a module in the architecture of their product. The term of collaborative filtering (Goldberg et al., 2000) denotes techniques using the known preferences of a group of users to predict the unknown preference of a new user. Our problem has consisted in finding a way to provide scale for hundreds thousands of people, while preserving anonymity of users (personal data remain on client side). Thus, we use an existing clustering method, that we have improved so that it is distributed respectively on client and server side. Nevertheless, in the absence of numerical votes for marketing reasons, we have chosen to do an innovative combination of this decentralized collaborative filtering method with a user profiling technique. We have also been submitted to constraints such as a short answer time on client side, in order to be compliant with the ASTRA architecture.

1 INTRODUCTION

The ASTRA company[1], located in Luxembourg, has conceived a service of Web sites broadcasting per satellite called Sat@once. This service is sponsorised by advertisement and free for the users, providing that they are equipped with a DVB receiver. The satellite bouquet contains hundreds of Web sites which are sent to about 118.000 persons[2] via a high bandwidth and one-way transmission.

Users who have a standard internet connection, in addition to the satellite reception, can select (through the client application called Casablanca) Web sites that particularly interest them, either by choosing items in a list provided by the server or by suggesting new contents. This means, the users receive above all those sites of the bouquet for which they have shown a particular interest. The users votes are also sent to server, in order to do a classification of the most popular sites. These ones will then be included in the bouquet[3] during the next update which takes place every week.

This approach however presents two major drawbacks for users. On the one hand, the amount of available sites is very important. This makes it difficult for the users to consult all of these documents (also called ”items”) in order to spot, in a reasonable amount of time, pieces of information which preoccupy them. Consequently, votes are often based on some presumed interest for an item, and not according to an experience feedback. Results provided by such a system are thus not always satisfying enough. On the other hand, people generally don't take the time to skim through the whole list of items and restrict possibilities from the beginning.

In order to cope with these problems, our goal consists in designing both the client and the server modules which provide users with documents likely to interest them but that they shouldn't have consulted spontaneously. These processes of investigation require techniques of collaborative filtering. In practical

[1] http://www.ses-astra.com/

[2] This estimation has been done in November 2004, according to number of persons who have downloaded the client application.

[3] The bandwidth of satellites being limited, it is not possible to include all the sites. Hence the need to select the most popular ones.

Castagnos S., Boyer A. and Charpillet F. (2005).
A DISTRIBUTED INFORMATION FILTERING: STAKES AND SOLUTION FOR SATELLITE BROADCASTING.
In *Proceedings of the First International Conference on Web Information Systems and Technologies*, pages 299-304
DOI: 10.5220/0001231302990304

terms, it amounts to identifying active user to a set of persons having the same tastes and, that, in function of his/her preferences and his/her past readings. This system starts from the principle that users having appreciated the same documents have the same topics of interests. Thus, it is possible to predict pieces of data likely to live up users' expectations by taking advantage of experience of a similar population.

The common feature of most of existing collaborative filtering methods is to be centralized. Even if the research of nearest neighbors among some thousands of candidates in real time is no longer a problem, the transition to hundred thousands of users or more remains an open issue. According to (Breese et al., 1998), the bottleneck due to a large user population of potential neighbors in conventional collaborative filtering algorithms is problematic. (Sarwar et al., 2001) have paved the way by proposing an alternative: they suggest to compute recommendations by identifying items that are similar to other items the user has liked. They suppose that the relationships between items are relatively static. Nevertheless, this approach is unlikely to work in the context investigated in this paper, since the number of users is there far more important than the number of items. Moreover, the bouquet (that is to say items it contains) can change radically from one week to the next. Therefore, we have chosen to explore ways to distribute computations.

Furthermore, centralization of data is in contradiction with the agreement of 28 January 1981 of the Council of Europe and with instructions of the ***C**ommission **N**ationale de l'**I**nformatique et des **L**ibertés*[4] (CNIL), unless users are handled with anonymity. As a matter of fact, the confidentiality of any information related to the users constitutes an european legal obligation. In France, it is the CNIL organization that is responsible for the protection of private life and for the preservation of personal data.

In order to distribute the model, we have thus decided to split a clustering collaborative filtering method into client and server parts. However, this is not enough to solve all the problems ASTRA is confronted with, since this filtering method requires to have explicit numerical or boolean votes. For marketing reasons[5], this kind of votes is not suitable, because it underpins some negative valuations of items by users. We will show, in section 3, how to bypass this difficulty with an assistance function to the votes. We then present the clustering algorithm in section 4. Part 5 is dedicated to a discussion about the advantages and drawbacks of the model. At last, Part 6 presents our perspectives of research. Beforehand, we would like to familiarize reader with the global architecture of our information filtering system in the following section.

2 ARCHITECTURE

The architecture of our information filtering system is shown on figure 1. This model associates a user profiling method based on the Chan formula (Chan, 1999) (cf. infra, 3 Assistance to votes, p. 3) and a new version of the hierarchical clustering algorithm, also called RecTree (Chee et al., 2001) (cf. infra, 4 Clustering algorithm, p. 3). This new version presents the advantage to be distributed.

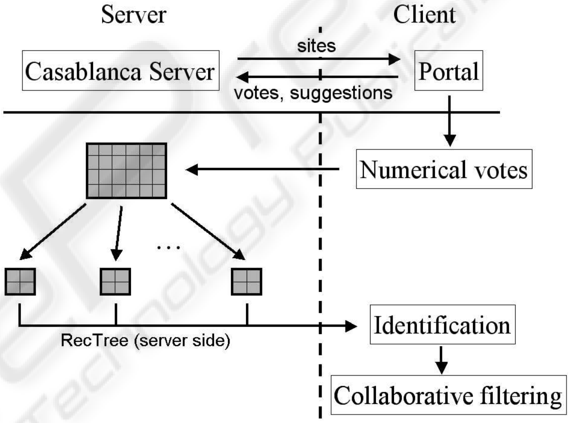

Figure 1: Architecture of information filtering module.

Web sites are sent via satellites from the Casablanca server to the client. Moreover, users who also have a standard internet connection can sent non-numerical votes (cf. infra, 3 Assistance to votes, p. 3) and suggestions for new contents to the server. This system interfaces itself with our information filtering module thanks to DLL files.

In order to distribute the system, the server part has been separated from the client side. The assistance function to the votes determines numerical votes for the items according to the users' actions. Then, these numerical votes are sent to the server, together with the non-numerical ones. The server thus has at its disposal, as input parameters, the matrix of users votes and a database including sites and descriptors. In this way, the server has no information about the population, but anonymous votes. Users preferences are stored in the profile on clients. Thus, the confidentiality criterion is duly respected.

The RecTree algorithm aims at reducing quantity of data that needs to be processed. The offline computations of RecTree allow to build typical users pro-

[4] http://www.cnil.fr

[5] ASTRA doesn't want that users could positively reject items for which companies have paid the inclusion in the bouquet.

Adapted Chan formula:

$$Interest(item) \ = \ Frequency(item)\ .\ (1 \ + \ IsFavorite(item) \\ + \ Duration(item) \ + \ Recent(item) \ + \ PourcentVisitedLinks(item))$$

$$\text{With: } Duration(item) \ = \ max_{visited\ items}\Big(\frac{time\ spent\ on\ pagesofitem}{size\ of\ the\ item}\Big)$$

$$\text{And: } Recent(item) \ = \ \frac{date(last\ visit) - date(log\ beginning)}{date(present) - date(log\ beginning)} \quad (1)$$

$Interest(item)$ must be normalized to correspond to scale of votes.
$IsFavorite(item)$ equals 1 if the item has been voted by the user (non-numerical vote) and 0 else.
At last, $PourcentVisitedLinks(item)$ corresponds to the number of visited pages divided by the number of pages on the item.

files. In this way, it is no longer necessary to consider the whole votes matrix, but only the votes of those persons belonging to the group of the active user. This not only reduces the number of people that need to be considered, but also the number of items: it is pointless to keep documents that none of the group members has read. In this way, we avoid the problem of bottleneck of collaborative filtering on client side: the active user can very quickly be assigned to one of the typical users groups.

3 ASSISTANCE TO THE VOTES

In the Casablanca client application, users have the possibility to check boxes corresponding to the sites that interest them most among those contained in the bouquet. However, we can't describe these non-numerical votes as boolean. Indeed, we can't differentiate in the system items which don't interest the active user (negative votes) from those he/she doesn't know or has simply omitted to check. This kind of votes is not sufficient to do relevant predictions with collaborative filtering methods.

For this reason, we have chosen to determine numerical marks without any rating[6] from the users. An other advantage of this method is to increase the number of votes in the matrix. In order to do that, we have chosen to develop an assistance function to the votes based on the formula of Philip CHAN (Chan, 1999). We have adapted this formula so that it can deal with items (cf. infra, formula 1, p. 3). Whereas the original formula was designed for Web pages, the items we are focusing on correspond to Web sites, that is to say sets of pages. The duration of consultation for a specific item thus corresponds to the cumulative time spent on each of its pages for example.

This assistance function undertakes to estimate marks that user is likely to allocate for different sites from implicit criteria (such as time or frequency that user takes to consult a page[7]). The system analyses log files of the active user to get back useful data. But all pieces of informations consulted in these log files remain on client side. Only numerical votes which have been deduced from this process are sent anonymously to the server. They are required for the use of RecTree clustering algorithm.

4 CLUSTERING ALGORITHM

The hierarchical clustering algorithm, also called RecTree (Chee et al., 2001), tries to divide the set of users into cliques. The algorithm of Chee *et al* was purely centralized, such as most of existing collaborative filtering methods. Our contribution consists in distributing this process: in this section, we explain how to build typical users profiles on server side and how to identify the active user to a group. From now on, this second step takes place on client side. Moreover, the identification phase has been optimized so that answer time is very short. the client part thus provides predictions in real time.

Table 1 proposes an example of votes characterized by integers going from 1 to 10. This graduation is arbitrary. The precision of this scale must be chosen by the designer of the system, providing that users could make the distinction between items they like, they don't like or those whose let them indifferent.

Figure 2 illustrates organization forms of groups in

[6] Ideally, the numerical votes should be submitted to their approval for checking.

[7] These are pieces of information easily and legally salvageable in Web browser of client.

Pearson correlation coefficient:

$$w(u_i, u_k) = \frac{\sum_{r \in R_i \cap R_k} (eval(u_i, r) - v)(eval(u_k, r) - v)}{\sqrt{\sum_{r \in R_i \cap R_k} (eval(u_i, r) - v) \sum_{r \in R_i \cap R_k} (eval(u_k, r) - v)}} \quad (2)$$

With: $w(u_i, u_k)$ the distance between u_i and u_k;
$eval(u_i, r)$ the valuation of r by u_i;
v the average mark of the resource;
R_i the items marked by u_i;

Table 1: Example of matrix of votes.

	R1	R2	R3	R4	R5
User 1		6	7	6	
User 2			4	7	7
User 3		7	6	5	
User 4	6	6			7

the above example. Users are divided up in clusters according to a distance calculated between them.

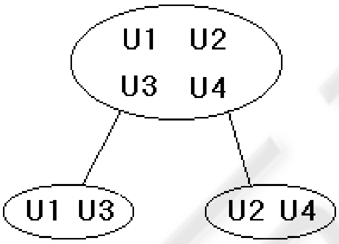

Figure 2: Hierarchical organization of users.

The RecTree algorithm is a model-based approach, described as a clustering method. However, it is managed as a memory-based approach because all the pieces of information are required for similarity computation. It allows, within the scope of our architecture, to limit the number of persons considered in the prediction computations. Thus, processing time for the sorting of site (cf. infra, 6 Perspectives, p. 6) will be shorter and results will be potentially more relevant since observations will be about a group closer to active user (Ungar and Foster, 1998). A way to popularize this process amounts to considering that active user asks a group of persons having same tastes as him/her for their opinions[8]. Each leaf of the RecTree tree corresponds to a profile of typical users.

The first step consists in associating the global matrix including users votes for each resource to the root of the tree. Afterwards, the set of users is divided up into two sub-groups using the nearest neighbours method, also called K-means (Herlocker et al., 1999). The latter consists, firstly, in choosing randomly k centers in the users/items representation space. In our case, the number k equals 2, since we must subdivide population into two sub-sets. Then, each user is positioned in the cluster of nearest center (figure 3).

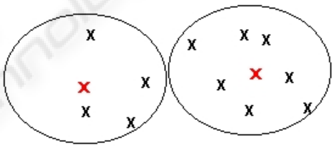

Figure 3: Users/items representation space.

Metrics used to determine distance in comparison with these centers is the Pearson correlation coefficient (Resnick et al., 1994) (cf. infra, formula 2, p. 4).

Literature shows that Pearson correlation coefficient works well (Shardanand and Maes, 1995), because it only takes into account items commonly valuated by compared users and disregards missing data.

Once groups of persons have been formed as previously mentioned, the position of isobarycentre is recalculated for each cluster and this operation is repeated from the beginning until we have obtained a stable state (where centers no longer move after recalculation of their position). The nearest neighbours algorithm complexity is in $o(k^2 n)$ for k clusters and n users. Once this first subdivision has been done, operation is renewed on each of the two obtained subgroups until we have reached the wished tree depth. Thus, the more we go down in the structure and the more clusters are specific to a certain group of similar users. Consequently, the more we glance through the tree in depth, the more persons share the same opinion concerning the assignment of a certain mark for a

[8] The computer process is obviously transparent for users.

given article. The whole complexity of the construction of the tree yields $o(n.log_2 n)$, where n is the number of users.

Subsequently, the identification phase of the active user to one of cliques is in $o(2p)$ on client side, where p corresponds to depth of the tree. The latter is built so that cliques hold just about the same number of persons for a given depth.

5 DISCUSSION

The novelty of our model lies in the fact that we have mixed a distributed collaborative filtering with a user profiling technique. In this way, we tackle problems of scale and of anonymity in information filtering systems. Furthermore, we have adapted to constraints of satellite broadcasting in order to highlight the benefits of such a system for industry.

To assure that answer time on client side will be short is essential too. But it is difficult to conform to this requirement if database contains a great number of users. Indeed, in order to secure confidentiality of data concerning active user, the identification phase of those to a group must be done on client side. That means we must retrieve pieces of data relating to population on clients. To avoid an overloading of user terminal by sending all of the votes matrix, typical users profiles are created with the help of RecTree algorithm (cf. supra, 4 Clustering algorithm, p. 3). In a first time, the server undertakes to split all of users in cliques from the votes matrix (algorithm in $o(n.log_2 n)$). Then, the tree structure is sent to client where the identification phase of the active user to a group takes place (algorithm in $o(2p)$). Thus, the number of persons taken in consideration is limited, because only those belonging to the same clique than active user are retained[9].

The figure 4 shows the answer time comparison done in (Chee et al., 2001) between correlation-based collaborative filter, called CorrCF (Resnick et al., 1994), and RecTree with different partition sizes b. The data for this study was drawn from the EachMovie database[10].

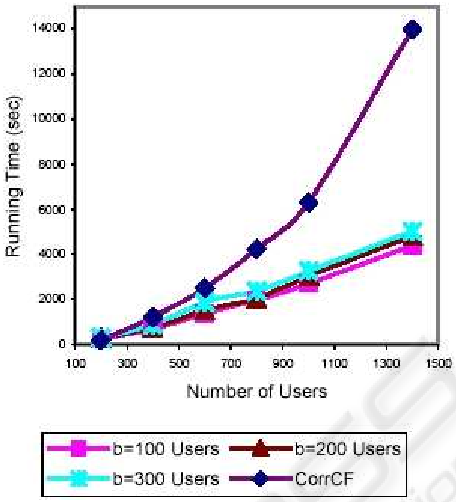

Figure 4: Offline performance of RecTree (Chee et al., 2001).

We can note that answer time increases linearly in function of the number of users. If we extend this straight line to 120,000 users, we will obtain in theory around four days and half of computations. This estimation seems huge. Nevertheless, this value has been calculated by considering that the matrix of votes was almost full[11]. But statistics of ASTRA show there are only around 6,000 regular voters and 50,000 occasional voters. Moreover, the estimated answer time remains much lower than a standard method (CorrCF) and is less than the period of bouquet update. Renewing periodically the server side computations can of course augur for slight differences between last votes and preferences taken into account in prediction computation. But these differences should be minimal because of the great number of users. This way to proceed also assures that the system is stable in case of addition of documents, because these ones will only be considered after reiteration of server side computation. Furthermore, RecTree algorithm has two major advantages:

- this method was easily divisible in two parts, respectively runnable on client and server side;
- the online part of computations, that is to say identification of user to a group, is in $o(2p)$. Thus, answer time on client side won't be penalized by the use of this algorithm. This part of computations has been optimized in our model, in comparison with the centralized RecTree algorithm of Chee *et al*. The online part of Chee's algorithm was in $o(b)$, where b was the number of users in each partition. Users had consequently to wait for a few seconds. In our version, the complexity of client

[9] We call back that it is possible to choose depth of the tree: if the latter is not excessive, studied groups will be smaller than the entire population but still disparate and consequently able to suggest novelties to user.

[10] http://www.research.digital.com/SRC/eachmovie/

[11] Indeed Chee *et al* have selected randomly, in the EachMovie database, users who have marked at least 100 items which is approximately the number of sites included in the ASTRA satellite bouquet. We can also notive that these tests have been done in 2001 and that, according to Moore's law, the performance of microprocessors doubles approximately every 18 months.

part only depends on the depth of the tree and the answer time will be much faster.

6 PERSPECTIVES

We have presented an algorithm that carries out distributed collaborative filtering on large databases and in relatively short time. This project is still in progress and we are currently working on validating the assistance function to the votes. There is to check if the user profiles built with this function are in agreement with the marks that users would chose for the same items if they were authorized to do it. This evaluation requires the use of the system by real users.

Moreover, thanks to our collaboration with Jean-Charles Lamirel and Randa Kassab (Kassab et al., 2005), we are considering combining this distributed collaborative filtering model with content-based filtering techniques to sort items in increasing order of importance for active user on client side. Indeed, the cache size of the client application is limited and it can be useful to favour the most relevant items if there is not enough space. Content-based techniques will also allow us to manage suggestions of new contents in the bouquet: for the moment, when a user suggests a site, nobody has voted for it and we can't include it in the `RecTree` computations.

ACKNOWLEDGEMENT

We want to thank the company `ASTRA` and the research center `Henri Tudor` which have encouraged this work.

REFERENCES

Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the fourteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 43–52, San Francisco, CA.

Chan, P. (1999). A non-invasive learning approach to building web user profiles. In *Workshop on Web usage analysis and user profiling, Fifth International Conference on Knowledge Discovery and Data Mining*, San Diego.

Chee, S. H. S., Han, J., and Wang, K. (2001). Rectree : An efficient collaborative filtering method'. In *Proceedings 2001 Int. Conf. on Data Warehouse and Knowledge Discovery (DaWaK'01)*, Munich, Germany.

Goldberg, K., Roeder, T., Huptan, D., and Perkins, C. (2000). Eigentaste : a constant time collaborative filtering algorithm. Technical Report M00/41, IEOR and EECS Departments, UC Berkeley.

Herlocker, J. L., Konstant, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *In Proceedings 1999 Conference of Research and Development in Information Retrieval*, pages pages 230–237, Berkeley, CA.

Kassab, R., Lamirel, J.-C., and Nauer, E. (2005). Novelty detection for modeling user's profile. In *International Conference FLAIRS 2005*, Clearwater Beach, Florida, USA.

Resnick, P., Iacovou, N., Suchak, M., Bergstorm, P., and Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina. ACM.

Sarwar, B. M., Karypis, G., Konstan, J. A., and Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *World Wide Web*, pages 285–295.

Shardanand, U. and Maes, P. (1995). Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1, pages 210–217.

Ungar, L. and Foster, D. (1998). Clustering methods for collaborative filtering. In *Proceedings of the Workshop on Recommendation Systems*, Menlo Park California. AAAI Press.