From implicit to explicit data: a way to enhance privacy

Sylvain CASTAGNOS
LORIA – Université Nancy 2
B.P.239, 54506 Vandoeuvre-lès-Nancy Cedex, France
Sylvain.Castagnos@loria.fr

Anne BOYER
LORIA – Université Nancy 2
B.P.239, 54506 Vandoeuvre-lès-Nancy Cedex, France
Anne.Boyer@loria.fr

ABSTRACT

In this paper, we describe a collaborative filtering model which has been implemented within the context of satellite website broadcasting. We particularly focus on the way to preserve intimacy of users, while exploiting the knowledge of the population to compute predictions. Our model relies on a client/:server architecture. It exploits the user behaviors on the system, also called implicit preferences, to compute ratings, that is to say explicit data. Implicit data remains on client side. Only the explicit data are sent anonymously on server side.

Author Keywords

Decentralized Collaborative Filtering, Privacy, User actions, Explicit votes.

ACM Classification Keywords

H3.3. Information Information Search and Retrieval.

INTRODUCTION

With the development of information and communication technologies, the size of information systems all over the world has exponentially increased. Consequently, it becomes more and more difficult for users to identify interesting items in a reasonable time, even if they use a powerful search engine. To cope with this problem, more and more companies as Amazon or Yahoo choose to integrate a recommender system in their products. The goal is then to provide users with resources likely to interest them, instead of waiting that they ask for them. These processes of investigation may be provided by collaborative filtering techniques. In practical terms, it amounts to identifying active user to a set of persons having the same tastes and, that, based on his/her preferences and his/her past readings. This system relies on the principle that users who liked 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.

Nevertheless, collaborative filtering algorithms tend to be more and more intrusive in the private life of users. For example, personal pieces of data are sometimes used in intelligent systems to compute recommendations. Users are not always aware of that phenomenon. Some governmental organizations are consequently in charge of the preservation of privacy and each software must comply with rules. Thus, centralization of data is not compliant with the European Council directive of 28 January 1981 and with instructions of the French Data Protection Commission[1] (CNIL), unless users are handled anonymously. As a matter of fact, the confidentiality of any information related to the users constitutes an European legal obligation.

We argue that a good way to cope with this problem is to use distributed models. In this paper, we will shortly introduce a client/server architecture which has been implemented within the framework of satellite website broadcasting[2].

DISTRIBUTED COLLABORATIVE FILTERING APPROACH

More and more researchers investigate various means to preserve privacy, as Peer-to-Peer architectures ([2], [6]), secure servers [7] and/or profile decentralization [1]. We choose to explore the distributed approaches within a client/server context, because it is mostly used for satellite broadcasting or for e-commerce applications at times.
In order to define the degree of privacy of a recommender system, we refer to the four axes of personalization as in [5]. We assume that an ideal system should be based on an explicit data collection method, transient profiles, user initiated involvment and non-invasive predictions.

ARCHITECTURE FRAC+

The architecture of our information filtering system is shown on Figure 1. It has been implemented within a website broadcasting software, where items are sent by satellites and votes are sent with a standard internet connection. This model associates a user modeling method based on the Chan formula [4] and a hierarchical clustering algorithm, called FRAC [3]. This architecture has the advantage to be distributed so that privacy criterion is duly fullfilled.

The function of user modeling determines numerical votes (explicit data) for items according to implicit user actions. We store information in log files such as: the frequency of consultations or the percentage of visited links for each website, the time spent to read items, the list of favorite websites which have been explicitly chosen by the

[1] ***C**ommission **N**ationale de l'**I**nformatique et des **L**ibertés* (http://www.cnil.fr).

[2] SES ASTRA (http://www.ses-astra.com/)

active user, the time spent since the last consultation, etc. The log files are of the same type as those of an Apache server, but are generated on client side. The implicit pieces of data remains locally and are not persistent, that is to say stored for a limited duration. This collection of actions is called “user model”. Then, we use the Chan formula to convert these parameters in ratings. The whole set of numerical votes for each user is called “user profile”. The profile can be sent to the server and this action is initiated by the users. Thus, the server aggregates the profiles so that to use, as input parameters, the matrix of user votes and the database including sites and descriptors. In this way, the server has no information about the population, except anonymous votes. Indeed, the standard internet connection is only used from clients to server. Each client has a unique ID (generated by the client software) which is associated to the active user’s profile. The server uses these IDs to guarantee that each user is only stored once in the matrix. There is no need to keep the IP addresses of users and IDs are not sufficient to find the corresponding users. The anonymity is thus performed by the fact that server has only the list of IDs. It does not know the link between IDs and identities.

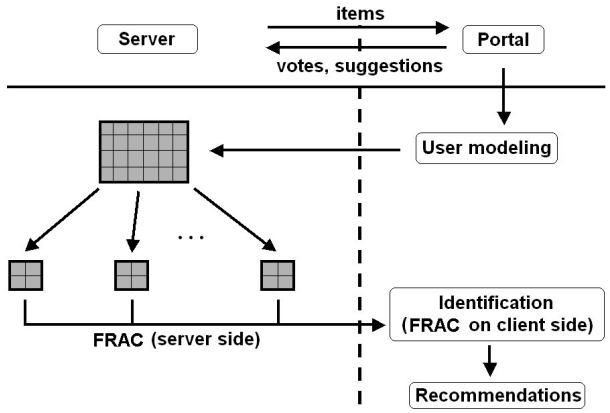

Figure 1. Architecture of the information filtering module.

Once the clusterization has been made on server side, an average profile of each group is sent to clients. We call them “typical profiles” and they are sent by satellites in the same way as items (by broadcasting on multicast IP addresses). The identification phase consists in comparing the local active user profile with group ones. Even when the active user did not want to share his/her preferences, it is possible to do predictions since they are made on client side. The user IDs are also used to prevent the system from malicious users, since valid IDs are generated by the client software. It is harder for hackers to flood the server with wrong profiles.

DISCUSSION AND CONCLUSION

As explained in [5], we reduced the intrusion in privacy by converting implicit actions into explicit ratings. The user profiles only allow the server to know preferences, while user models store the consultation details (duration, date, frequency, etc.). User models are transient. Moreover, the system is designed in such a way that profiles are sent anonymously on the initiative of users.

Alan Westin defines privacy as "*the claim of individuals to determine for themselves when, how, and to what extent information about them is communicated to others*". We proposed a client/server collaborative filtering model which faces with the two first aspects. Nevertheless, we encountered difficulties to implement the third point within our industrial framework for marketing reasons. Because our industrial partners have defined a business model relying on advertising, they do not allow people to explicitly build their profiles. Consequently, we wonder how it would be possible to combine user expectations in term of privacy and business constraints.

Moreover, countries have different laws about privacy. How is it possible to guarantee that pieces of software always fullfilled the national laws in these conditions? At last, we wonder how we can be sure to have enough data to do good predictions, since users decide when to send information?

ACKNOWLEDGMENTS

We thank reviewers who provided helpful comments on previous versions of this document.

REFERENCES

1. Berkovsky, S., Eytani, Y., Kuflik, T. and Ricci, F. (2005). *Privacy-Enhanced Collaborative Filtering.* In *proc. of CHI Workshop on Privacy-Enhanced Personalization* (PEP05).
2. Canny, J. (2002). *Collaborative Filtering with Privacy.* In *IEEE Symposium on Security and Privacy*. Oakland, CA, 2002.
3. Castagnos, S. and Boyer, A. (2006). *FRAC+: A Distributed Collaborative Filtering Model for Client/Server Architectures*. In *proc. of the International Conference on Web Information Systems and Technologies* (WebIST06). Setùbal, Portugal, 2006.
4. 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*.
5. Cranor, L. F. (2005). *Hey, That's Personal!* Talk in the *International User Modeling Conference* (UM05).
6. Miller, B. N., Konstan, J. A., and Riedl, J. (2004). *Pocketlens: Toward a personal recommender system*. In *ACM Transactions on Information Systems*, volume 22, pages 437–476.
7. Polat, H. and Du, W. (2004). *SVD-based Collaborative Filtering with Privacy.* In *proc. of ACM Symposium on Applied Computing*. Nicosia, Cyprus, 2004.