Modeling Preferences in a Distributed Recommender System

Sylvain Castagnos, Anne Boyer

HAL Id: inria-00171802
https://inria.hal.science/inria-00171802

Submitted on 13 Sep 2007

Modeling Preferences in a Distributed Recommender System

Sylvain Castagnos and Anne Boyer

LORIA - Université Nancy 2, Campus Scientifique - B.P.239
54506 Vandoeuvre-lès-Nancy Cedex, France
{sylvain.castagnos, anne.boyer}@loria.fr

Abstract. A good way to help users finding relevant items on document platforms consists in suggesting content in accordance with their preferences. When implementing such a recommender system, the number of potential users and the confidential nature of some data should be taken into account. This paper introduces a new P2P recommender system which models individual preferences and exploits them through a user-centered filtering algorithm. The latter has been designed to deal with problems of scalability, reactivity, and privacy.

1 Introduction

Usual search engines provide too many results to ensure that the active user will identify the most relevant items in a reasonable time. As a result, the scientific community is rethinking the existing services of searching and accessing information under the designation "Web 2.0".

A solution consists in providing each user with items that are likely to interest him/her. To do this, we first build his/her model by collecting his/her preferences. Our approach is based on an analysis of usages. Nevertheless, it is not always possible to collect data about the active user quickly enough. Collaborative filtering techniques [1] are a good way to cope with this difficulty. There are several fundamental problems when implementing a collaborative filtering algorithm. In this paper, we pay attention to significant problems such as scalability, reactivity, and respect of privacy.

Our algorithm relies on a distributed user-based collaborative filtering technique. It has been integrated in our document sharing system called "SofoS".[1] We state the hypothesis that documents are transient on the platform, whereas human-computer interactions are long-standing. We assume that, in this case, a user-based approach may be more appropriate than an item-based one. If a significant proportion of resources is constantly removed or added, correlations between users will potentially need fewer updates than an item correlation model.

Our P2P model offers the advantage of being fully distributed. It collects data about the preferences of users, and takes advantage of an "Adaptive User-centered Recommender Algorithm" called AURA. The latter provides a service

---

[1] SofoS is the acronym for "Sharing Our Files On the System".

which builds a virtual community of interests, centered on the active user by selecting his/her nearest neighbors. AURA is an anytime algorithm which furthermore requires very little computation time and memory space.

2 Related Work

A way to classify collaborative filtering techniques is to consider user-based methods in opposition to item-based algorithms. For example, Miller *et al.*[3] show the great potential of distributed item-based algorithms. They propose a P2P version of the item-item algorithm, and thus address the problems of portability (even on mobile devices), privacy, and security with a high quality of recommendations. On the contrary, we explored a distributed user-based approach within a client/server context in [2]. In this model, implicit criteria are used to generate explicit ratings. These votes are anonymously sent to the server. An offline clustering algorithm is then applied, and group profiles are sent to clients. The identification phase is done on the client side in order to cope with privacy. This model also deals with sparsity and scalability. We highlighted the added value of a user-based approach in the situation where users are relatively stable, whereas the set of items may often vary considerably.

In this paper, we introduce a new user-based collaborative filtering technique (AURA), distributing profiles and computations. It has been integrated in the SofoS platform and relies on a P2P architecture.

3 SofoS

3.1 Construction of Preference Models

SofoS is our new document platform, using a recommender system to provide users with content. Once it is installed, users can share and/or search documents, as they do on P2P applications like Napster. The goal of SofoS is also to assist users to find the most relevant sources of information in the most efficient way. In order to reach this objective, the platform exploits the AURA recommender module. The performance of this module crucially depends on the accuracy of the individual user preference models.

The first step when modeling preferences of users consists in choosing an efficient way to collect data. Proposing a series of questions to users is an efficient way to do accurate preference elicitation. Such an approach however would require asking hundreds of questions, and most users are generally not willing to take enough time to carry through such a lengthy process. This is why we prefer to let users explicitly rate the items they want, without order constraints.

However, an explicit data collection may be insufficient. Psychological studies [4] have shown that people construct their preferences while learning about the available items. This means that *a priori* ratings are not necessarily relevant. Unfortunately, few users provide a feedback about their consultations. We assume that, despite the explicit voluntary completion of profiles, there are a

lot of missing data. We consequently add a user modeling function based on the Chan formula [2]. This function relies on an analysis of usages. It temporarily collects information about the action of the active user (frequency and duration of consultations for each item, etc.) and transforms them into numerical votes. In order to preserve privacy, all data related to the user's actions remain on his/her peer. The explicit ratings and the estimated numerical votes constitute the active user's personal profile.

3.2   The AURA Algorithm

The personal preference-based profiles are used by AURA, in order to provide each user with the content that most likely interests him/her. AURA relies on a Peer-to-Peer architecture.

Each user on a given peer of the system has his/her own profile and a single ID. The session data remain on the local machine in order to enhance privacy. There is no central server required since sessions are only used to distinguish users on a given peer.

For each user, we use a hash function requiring the IP address and the login in order to generate his/her ID on his/her computer. In this way, an ID does not allow identification of the name or IP address of the corresponding user. The communication module uses an IP multicast address to broadcast the packets containing addressees' IDs.

Users can both share items on the platform and integrate a feedback about their preferences. Each item has a profile on the platform. In addition to the available documents, each peer owns 7 pieces of information: a personal profile (cf. section 3.1), a public profile, a group profile and 4 lists of IDs (list "A" for IDs of peers belonging to its group, list "B" for those which exceed the minimum-correlation threshold as explained below, list "C" for the black-listed IDs and list "O" for IDs of peers which have added the active user to their group profile).

The public profile is the part of the personal profile that the active user u_a accepts to share with others. The algorithm also has to build a group profile. It represents the preferences of a virtual community of interests, and has been especially designed to be as close as possible to the active user's expectations. In order to do that, the peer of the active user asks for the public profiles of all the peers it can reach through the platform. Then, for each of these profiles, it computes a similarity measure with the personal profile of the active user. The active user can indirectly define a minimum-correlation threshold which corresponds to the radius of his/her trust circle.

If the similarity is lower than this fixed threshold, which is specific to each user, the ID of the peer is added to the list "A" and the corresponding profile is included in the group profile of the active user, using the procedure of table 1.

We used the Pearson correlation coefficient to establish a similarity measure. Of course, if this similarity measure is higher than the threshold, we add the ID of the peer to the list "B". The list "C" is used to systematically ignore some peers. It enables to improve trust – i.e. confidence that users have in the

recommendations – by identifying malicious users. The trust increasing process will not be considered here.

When his/her personal profile changes, the active user has the possibility to update his/her public profile p_a. In this case, the active peer has to contact every peer[2] whose ID is in the list "O". Each of these peers re-computes the similarity measure. If it exceeds the threshold, the profile p_a has to be removed from the group profile, using the procedure of table 1. Otherwise, p_a has to be updated in the group profile, that is to say the peer must remove the old profile and add the new one.

Proc AddToGroupProfile(profile of u_n)	**Proc** RemoveToGroupProfile(old profile)
$W = W + |w(u_a, u_n)|$	$W = W - |w(u_a, u_n)|$
for each item i **do**	**for** each item i **do**
$(u_{l,i}) = (u_{l,i}) * (W - |w(u_a, u_n)|)$	$(u_{l,i}) = (u_{l,i}) * (W + |w(u_a, u_n)|)$
$(u_{l,i}) = ((u_{l,i}) + w(u_a, u_n) * (u_{n,i}))/W$	$(u_{l,i}) = ((u_{l,i}) - w(u_a, u_n) * (u_{n,i}))/W$
end for	**end for**

$(u_{l,i})$ the rating for item i in the group profile;
$(u_{n,i})$ the rating of user n for item i;
W the sum of $|w(u_a, u_i)|$, which is stored;
$w(u_a, u_n)$ the correlation coefficient between the active user u_a and u_n.
Table 1. Add or remove a public profile.

4 Discussion

In our system, the users have complete access to their preferences. They have an effect on what and when to share with others. Only numerical votes are exchanged and the logs of user actions are transient. Even when the active user does not want to share his/her preferences, it is possible to do predictions, since public profiles of other peers are temporarily available on the active user device. Each user has a single ID, but the anonymity is ensured by the fact that there is no table linking IDs and identities.

As regards scalability, our model no longer suffers from limitations since the algorithms used to compute group profiles and predictions are in $o(b)$, where b is the number of commonly valuated items between two users, since computations are made incrementally in a dynamic context. In return, AURA requires quite a lot of network traffic. This is particularly true if we use a random discovery architecture. Other P2P structures can improve communications [3].

We evaluated our model in terms of prediction relevancy by computing the *Mean Absolute Error* (`MAE`) on the GroupLens test set[3]. We simulated arrivals of peers by progressively adding new profiles. As shown on figure 1, we got predictions as good as using the PocketLens algorithm [3]. PocketLens relies on a distributed item-based approach. This comparison demonstrates that AURA provides as relevant results as an efficient item-based collaborative filtering.

---

[2] A packet is broadcasted with an heading containing peers' IDs, the old profile and the new public profile.
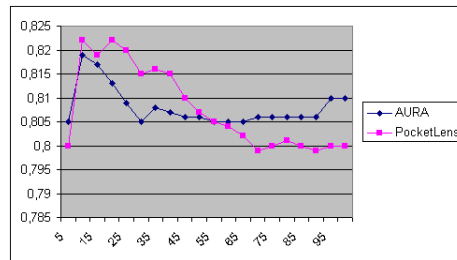
[3] http://www.grouplens.org/

Fig. 1. MAE as neighborhood size grows.

We also conducted some tests in order to measure computation time. They highlight the fact that AURA allows to do real-time predictions. It does not need to do offline computations, since we can take into account 10,000 profiles with 150 items in less than 0.5 second. For 100,000 users, we need about 3 seconds. The system does not have to wait until all similarity measures end. As the algorithm is incremental, we can stop considering other peers at any moment.

5 Conclusion

SofoS is a new document sharing platform including a recommender system. To cope with numerous problems specific to information retrieval, we proposed a Peer-to-Peer collaborative filtering model which is totally distributed. It allows real-time personalization. We show in this paper that AURA can deal with important problems such as scalability, privacy, and quality.

Our algorithm is anytime and incremental. Contrary to PocketLens, our model is user-based because we consider that the set of items can change. Even if an item is deleted, we can continue to exploit its ratings in the computation of predictions. Moreover, the dynamic context of our model allows the system to update the modified profiles instead of resetting all the knowledge about neighbors. At last, our model has very low memory requirements because it does not need to store any neighbors' ratings, similarity matrix, dot product matrix, etc. It only requires the sum of the Pearson coefficients and four lists of user IDs.

References

[1] J.S. Breese, D. Heckerman, C. Kadie: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. Proceedings of UAI-98. San Francisco, CA (July 1998).
[2] Sylvain Castagnos, Anne Boyer: A Client/Server User-Based Collaborative Filtering Algorithm: Model and Implementation. Proceedings of the 17th European Conference on Artificial Intelligence (ECAI2006). Riva del Garda, Italy (August 2006).
[3] Bradley N. Miller, Joseph A. Konstan, John Riedl: PocketLens: Toward a Personal Recommender System. ACM Transactions on Information Systems **22** (July 2004).
[4] J.W. Payne, J.R. Bettman, E.J. Johnson: The Adaptive Decision Maker. Cambridge University Press (1993).