A Novel Multi-agent Community Building Scheme Based on Collaboration Filtering*

Yu Sun[1], Peng Han[2], Qian Zhang[1], and Xia Zhang[1]

[1] National Engineering Research Center for Computer Software, Northeastern University, Shenyang 110004, China
{sun.yu, zhangqian, zhangx}@neusoft.com
[2] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200030, China
phan@mail.sjtu.edu.cn

Abstract. Research on e-learner community building has attracted much attention for its effectiveness in sharing the learning experience and resources among geographically dispersed e-learners. While collaborative filtering proves its success as one of the most efficient methods in finding similar users in e-commerce domain, it does meet special challenges in e-learning areas. In this paper, we incorporate multi-agent techniques into collaborative filtering and propose a novel community building scheme. By doing so, we manage to collect useful information from the learner behaviors and thus increase the scalability and flexibility of traditional collaborative filtering methods. The experiment on a standard benchmark shows that our scheme has reasonable community building quality and e-learners can make better recommendations to each other inside the community.

1 Introduction

Recently, the research on *e-learner community building* has attracted much attention which tried to group learners with similar background and interests into communities so that they can share their learning resources and experiences efficiently. In this paper, we propose a novel e-learner community building scheme by integrating the collaborative filtering [1-3] with multi-agent architectures. The experiment shows that our community building scheme enables the learners to locate potential neighbors efficiently and eventually self-organize similar users into learning communities.

The rest of this paper is organized as follows. In Section 2, some basic concepts and algorithm framework on CF are presented and discussed. In Section 3, we give the design and key features of our e-learner community building scheme and present the experimental results in Section 4.Finally we conclude the paper and provide an outlook on future research work in Section 5.

* Supported by the National High-Tech Research and Development Plan of China under Grant No 2003AA4Z3020.

R.W.H. Lau et al. (Eds.): ICWL 2005, LNCS 3583, pp. 221 – 225, 2005.

2 Memory-Based Collaborative Filtering

Generally, the task of CF is to predict the votes of active users based on the data in the user database which consists of a set of votes corresponding to the vote of user *i* on item *j*. The memory-based CF algorithm calculates this prediction as a weighted average of other users' votes on that item using the following formula:

$$P_{a,j} = \overline{v_a} + \kappa \sum_{i=1}^{n} \varpi(a,j)(v_{i,j} - \overline{v_i}) \tag{1}$$

Where $P_{a,j}$ denotes the prediction of the vote for active user *a* on item *j* and *n* is the number of users in user database. $\overline{v_i}$ is the mean vote for user *I* and I_i is the set of items on which user *i* has voted. The weights $\varpi(a,j)$ reflect the similarity between active user and users in the user database. κ is a normalizing factor to make the absolute values of the weights sum to unity.

3 Strategy of Learner Community Self-organization

3.1 Learner Profile Generation

Describing the interest and intention of learners is the first and vital step of e-learner community building. Here, we divide the interest into *explicit* interests and *implicit* interests. In this paper, we name the set of explicit interest as Int^{e} and the set of implicit interest as Int^{i}. So for each resource the learner accessed, we can generate a tuple $< \mathrm{u_i}, \mathrm{Int_i^e}, \mathrm{Int_i^i} >$. Here $u_i \in U$ is the identity of the resource accessed, Int_i^e is the explicit interests and Int_i^i is the implicit interests. Each tuple has either Int_i^e, Int_i^i or both depend on their availability. In order to decrease the complexity of matching and avoid the traffic overload, we further merge the Int_i^e and Int_i^i into a single $\mathrm{Int_{u_i}}$ as following:

$$\mathrm{Int_{u_i}} = \mathrm{g}(\mathrm{f^e}(\mathrm{Int_j^e}), \mathrm{f^i}(\mathrm{Int_j^i})) \tag{2}$$

Where f^{e} and f^{i} are the uniform functions for the explicit and implicit interest respectively while g is the function to combine the two kinds of interests. We implemented these functions as a weighted arithmetic average where each attributes has a weighted assigned.

3.2 Distributed Learner Profile Management Scheme

In order to find similar learners using collaborative filtering algorithm, the LAs should share the profile they generate for learners to each other. So we propose a distributed learner profile management scheme by introducing another kind of agent called Group Agent (GA) which serves as the broker for LAs and responsible for forwarding this information to potential neighbor learners.

Distributed learner profile management has two key steps: *Division* and *Location*. In our scheme, we wish to divide the whole learner profile space into fractions which are called *bucket* in the following of this paper. We make each bucket hold a group of learners' records who has a particular <*Unit_ID, Int*> tuple. It means that learners in the same bucket have the same interest on at least one unit. Figure1 illustrates our division strategy:

Each GA will be responsible to store one or more buckets and later when the LA wants to make prediction for a particular user, we only need to contact special GA to retrieve those buckets which the active user's profile is in. This strategy is based on the heuristic that learners with similar interests will at least rate one item with similar votes. As we can see in section 4.2.1, this strategy has a very high hitting ratio.

3.3 Community Building Scheme

In this section, we provide formal definitions on which we will rely upon for describing our community building scheme presented later.

Let **G** and **L** be disjoint sets of GAs and LAs.

Definition 1: A learner agent l is a tuple A_l=<*Learner_ID*, ***Unit_Int*, *Local_Neighbor_list***>, where *Learner_ID* is the uniform ID of l and **Unit_Int** is the vote vectors of l as described in section 3.2. ***Local_Neighbor_list*** is the list of similar neighbors with the form of <*Learner_ID, Trust_award*>, where *Trust_award* is the evaluation of interest similarity between l and the learner in the local neighbor list.

Definition 2: A group agent g is a tuple A_g=<*Local_Learner_List, Unit_Int_List, Neighbor_List*>, where *Local_Learner_List* is the LAs list registered on and managed by g. *Unit_Int_List* maintains the <*Unit_ID, Int*> tuples cashed in g, *Neighbor_List* contains the *bucket* related to the <*Unit_ID, Int*> in the *Unit_Int_List.*

When a LA generates a new <*Unit_ID, Int*> for the e-learner it monitors, it will send a notification message to the GA which is in charge of storing the bucket corresponding to the tuple. By doing so, the LA can retrieve the profiles in the buckets back which then can be used to make recommendations by CF algorithms. Still, the GA can register the LA in its Local_Learner_List and inform other LA in the list about the updating. The other users can then use this information to update their neighbor list so that later they can make recommendation directly to the LA in their neighbor list.

4 Experimental Results

4.1 Data Set and Metrics

We use EachMovie data set [4] to evaluate the performance of improved algorithm. The EachMovie data set is provided by the Compaq System Research Center as a standard benchmark on the evaluation of collaborative filtering algorithms and contains 2,811,983 <*Unit_ID, Int*> tuples from 72,916 users on 1,628 resources.

We use Mean Absolute Error (MAE), a statistical accuracy metrics, to report prediction experiments for it is most commonly used and easy to understand:

$$MAE = \frac{\sum_{a \in T} | v_{a,j} - p_{a,j} |}{| T |} \tag{3}$$

Where $v_{a,j}$ is the interests given to item j by user a, $p_{a,j}$ is the predicted value of user a on item j, T is the test set, $|T|$ is the size of the test set.

We select 5000 users and choose one user as active user per time and the remainder users as his candidate neighbors, because every user only makes self recommendation locally. We use ALL-BUT-ONE strategy [1] and the mean prediction accuracy of all the 5000 users as the system's prediction accuracy.

4.2 Experimental Result

We design several experiments for evaluating our algorithm and analyze the effect of various factors by comparison. All our experiments are run under Windows 2000 on an Intel Pentium 4 PC with a CPU speed of 1.8 GHz and 512 MB of RAM.

We compare the prediction accuracy of traditional CF algorithm and our Multi-agent based CF algorithm and the results are shown as Figure 1. We can see that our algorithm has better prediction accuracy than the traditional CF algorithm.

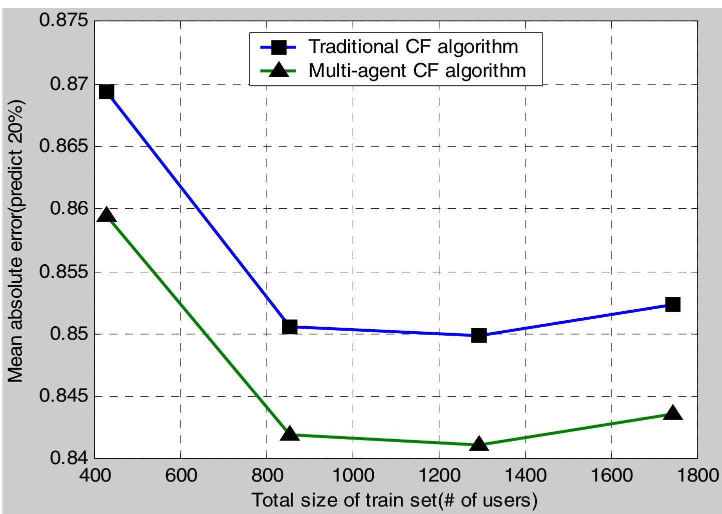

Fig. 1. Multi-agent based CF vs. Traditional CF

5 Conclusion

In this paper, we propose a novel e-learner community building scheme by integrating the collaborative filtering and multi-agent techniques. By using the intelligent agents, are able to monitor the whole dynamic learning behaviors of e-learners and automatically learn the interest of knowledge-oriented resources, then generate the learner profile which can be used by collaborative filtering algorithm. The agents can also accelerate the profile sharing in the distributed environment. Based on this, we extend the traditional collaborative filtering algorithm to make it operational decentralized by proposing a distribute user profile management scheme. The experiment shows that our community building scheme enables the learners to locate potential neighbors efficiently and eventually self-organize similar users into learning communities.

Acknowledgement

The work described in this paper is supported partially by National High-Tech Research and Development Plan of China under Grant No 2003AA4Z3020

References

1. Breese, J., Heckerman, D., and Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, p.43-52, 1998.
2. J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl.: An algorithmic framework for performing collaborative filtering. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, p.230-237, 1999.
3. G. Linden, B. Smith, and J. York, Amazon.com Recommendations Item-to-item collaborative filtering, IEEE Internet Computing, Vo. 7, No. 1, p.7680, Jan. 2003.
4. Eachmovie collaborative filtering data set.: http://research.compaq.com/SRC/eachmovie