Proceedings of the 6th WSEAS International Conference on Applied Informatics and Communications, Elounda, Greece, August 18-20, 2006 (pp250-253)

E-Learner Community Exploiting Based on Collaboration Filtering

ZHI-MEI WANG[1], PENG HAN[2], FAN YANG[2]
[1] Department of Computer
Wenzhou Vocational & Technical College
Wenzhou, 325035, Zhejiang
CHINA

[2]Faculty of Mathematics and Information Technology
Feruniversität in Hagen
Universitat Str.27, 58094, Hagen
GERMANY

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. This paper incorporates multi-agent techniques into collaborative filtering and proposes a novel community building scheme. By doing so, this paper manages to collect useful information from the learner behaviors 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.

Key-Words: - E-learning, E-learner Community, Collaborative Filtering, Community Exploiting

1 Introduction

While web-based learning provides learners with unprecedented flexibility and abundance in learning styles and resources, it also brings together thousands of geographically dispersed individual learners who are eager of useful recommendation and collaboration from their online buddies. As a result, the research on e-learner community building has attracted much attention. Learners with similar background and interests are grouped into communities so that they can share their learning resources and experiences efficiently. As one of the most efficient personalized recommendation techniques in e-commerce, collaborative filtering (CF) [1] seems to be quite a good choice for this task because of its strong mathematical foundation and excellent performance in describing and calculating the similarities between users. Up to now, many effective similarity measures for CF have been proposed, and many successful applications have been built in e-commerce domain [2, 3].

In this paper, a novel e-learner community building scheme is proposed which can integrate the collaborative filtering with multi-agent architectures. By using intelligent agents, it is able to monitor the whole dynamic learning behaviors of e-learners and automatically learn their interest of 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. The traditional collaborative filtering algorithm is also extended to make it operational decentralized by proposing a distribute user profile management scheme which increase its scalability. 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, the design and key features of our e-learner community building scheme is discussed. The referential implementation of our scheme is described in Section 4 and the experimental results are also presented in Section 5. Finally a brief conclusion of the paper and an outlook on future research work is given in Section 6.

2 Memory-Based Collaborative Filtering

Proceedings of the 6th WSEAS International Conference on Applied Informatics and Communications, Elounda, Greece, August 18-20, 2006 (pp250-253)

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:

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

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* as:

$$\overline{v_i} = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j} \qquad (2)$$

Where 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, this paper divides the interest into explicit interests and implicit interests. Explicit interests are expressed explicitly by the learner through rating on resources. Implicit interests are captured automatically by the system as a result of users' dynamic learning behavior. The collection of the implicit interest information is out of the scope of this paper, more detailed description could be found in [5].

In this paper, it names the set of explicit interest as Int^e and the set of implicit interest as Int^i. So for each resource the learner accessed, the system 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 depends on their availability. In order to decrease the complexity of matching and avoid the traffic overload, the system further merges the Int_i^e and Int_i^i into a single $\mathrm{Int}_{\mathrm{u_i}}$ as following:

$$\mathrm{Int}_{\mathrm{u_i}} = \mathrm{g}(\mathrm{f^e}(\mathrm{Int_j^e}), \mathrm{f^i}(\mathrm{Int_j^i})) \qquad (3)$$

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. These functions are implemented 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. However, in the distributed web-based learning environment, it will be difficult to for a LA to obtain all the others user's profiles which may cause traffic overload and decrease efficiency as the number of learners becomes large. So in this section, it proposes 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, the system divides the whole learner profile space into fractions (for concision, it will call such fractions by the term *bucket* in the following of this paper) in such a manner that potential neighbors can be put into the same bucket. So later it can access only several buckets to fetch the useful user profiles. Here, it solves the first problem by proposing a division strategy which makes 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, the system only needs 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 shown in section 4.2.1, this strategy has a very high hitting ratio.

3.3 Community Building Scheme

In this section, it provides formal definitions on which the system will rely upon for describing our community building scheme presented later.

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

Definition1: 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.

Definition2: 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

In this paper, the EachMovie data set [4] is used 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:

$$MAE = \frac{\sum_{a \in T} | v_{a,j} - p_{a,j} |}{| T |} \qquad (4)$$

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.

5000 users is selected and one user is chosen as active user per time and the remainder users as his candidate neighbors, because every user only makes self recommendation locally. In the experiment, a ALL-BUT-ONE strategy [1] is used which take the mean prediction accuracy of all 5000 users as the system's prediction accuracy.

Several experiments is designed 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.

4.2 The Efficiency of Neighbor Choosing

A data set of 500-5000 users is used and shown among the users chosen by Return-All neighbor searching scheme, how many of the top-100 users have the most similarities with active users. Figure 1 shows a comparison between the results calculated by the traditional memory-based CF algorithms and after it is enhanced by the multi-agent architecture for profile sharing. We can see from the data that when the user number rises above 1000, more than 80 users who have the most similarities with the active users are chosen by our Neighbor choosing scheme

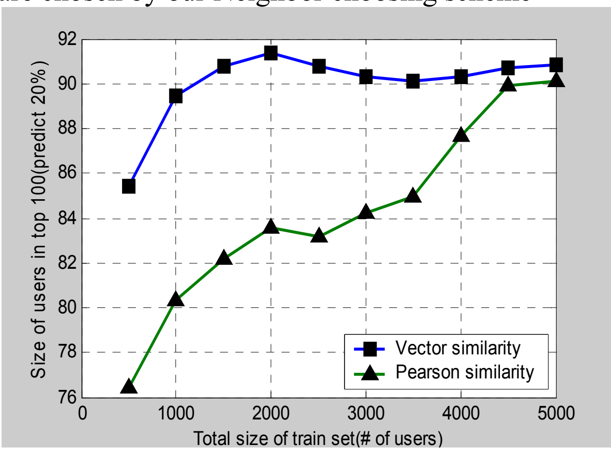

Fig. 1. How Many Users Chosen by DHT-based CF Fall in Traditional CF's Top 10

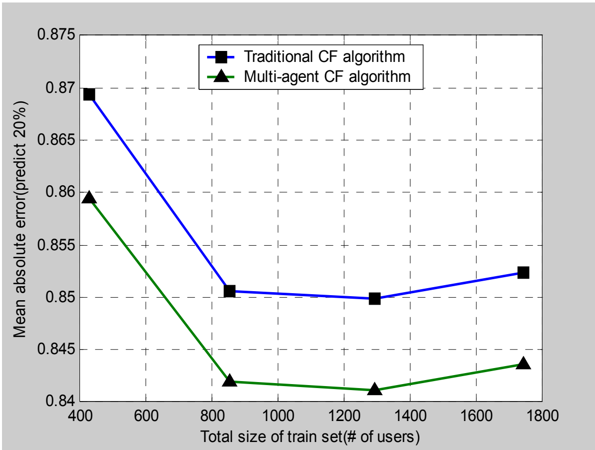

Fig. 2. Multi-agent based CF vs. Traditional CF

4.3 Prediction Performance

The prediction accuracy of traditional CF algorithm is compared with our Multi-agent based CF algorithm. And the results are shown as Figure 2 which show that our algorithm has better prediction accuracy than the traditional CF algorithm.

5 Conclusion

In this paper, it proposes a novel e-learner community building scheme by integrating the collaborative filtering and multi-agent techniques. By using the intelligent agents, the system is 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, the traditional collaborative filtering algorithm is extended 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.

6 Acknowledgement

The authors would like to thank the Wenzhou Science&Technology Bureau that awarded the first author allowing her to deepen her research on innovative personalized recommendation and advanced e-learning technologies. The work of the second author was supported in part by the Deutsche Forschungsgemeinschaft (the German Research Foundation) under code number 44200719, by the EU under contract number EC-US U8041. And third author was supported by the state government of North-Rhine Westfalia with a Lise-Meitner fellowship.

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, 1998, pp.43-52.

[2] Herlocker, J. L., Konstan, J. A., Borchers, A. and Riedl, J.: An algorithmic framework for performing collaborative filtering. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 1999, pp.230-237.

[3] G. Linden, B. Smith, and J. York, Amazon.com Recommendations Item-to-item collaborative filtering, IEEE Internet Computing, Vol. 7, No. 1, Jan. 2003, pp.76-80.

[4] Eachmovie collaborative filtering data set.: http://research.compaq.com/SRC/eachmovie

[5] Yang fan, Han peng, Shen Ruimin, Tong Ren, Construction and Application of the Learning Behavior Analysis Center based on Open E-Learning Platform", ICWL2005, HongKong, pp.34-43.

[6] Han Peng, Xie Bo, Yang Fan, Shen Ruimin, "A Scalable P2P Recommender System Based on Distributed Collaborative Filtering", Expert systems with applications, Vol.27, No2, Sep 2004, pp. 123-142.