A Dynamic Framework for Maintaining Customer Profiles in E-Commerce Recommender Systems

Choochart Haruechaiyasak Chatchawal Tipnoe Sarawoot Kongyoung
Chaianun Damrongrat Niran Angkawattanawit

Information Research and Development Division (RDI)
National Electronics and Computer Technology Center (NECTEC)
Thailand Science Park, Klong Luang, Pathumthani 12120, Thailand
Email: choochart.haruechaiyasak@nectec.or.th

ABSTRACT - Recommender systems have been successfully applied to enhance the quality of service for customers, and more importantly, to increase the sale of products and services in e-commerce business. In order to provide effective recommendation results within an acceptable response time, a recommender system is required to have the scalability to handle a large customer population in real time. In this paper, we propose a new recommender system framework based on the incremental clustering algorithm in order to dynamically maintain the customer profiles. Using the incremental clustering technique, the dynamic changes in the number of customers and products purchased could be handled effectively. Experiments on real data sets showed that the proposed framework helps to reduce the recommendation time, while retaining accuracy.

1. Introduction

Recommender systems have recently gained much attention as a new business intelligence tool for e-commerce business [5]. Applying a recommender system for an online retailer store helps to enhance the quality of service for customers and increase the sale of products and services. The recent commercial success of recommender systems has been demonstrated in many online stores including Amazon.com, CDNow.com, Barnes&Noble.com, and MovieFinder.com.

Recommender system framework generally applies the information filtering technique called *collaborative* or *social-based* filtering [1, 3]. The typical collaborative filtering approach relies on the rating system where each customer is asked to rate the purchased items based on his/her preference. These ratings are recorded in a database which can be viewed as a two-dimensional array of customer by rating. To generate a recommendation list for a particular customer, collaborative filtering algorithm first searches for a group of *neighbors* by comparing their similarity or correlation values. This is known as the *neighborhood-based* algorithm. The ratings for new products are predicted by analyzing the neighbor's ratings. Then the final ranked products are presented to the user as the recommendation. The *neighborhood-based* collaborative filtering suffers from the poor scalability problem, i.e., the time to search for a group of neighbors is proportional to the number of all customers in the system.

Clustering techniques have been applied to reduce the neighbor searching time. However, most previous works only focused on applying clustering algorithms as a dimensional reduction technique [4, 6]. The issue of dynamic changes of customers and products in the system was not considered. Recommender systems are operated in the dynamic Web environment in which at anytime a user may decide to purchase some products and become a new customer, or existing customers may purchase more products and thus alter their profiles. To provide effective recommendation results including customer satisfaction, a recommender system is required to have, not only, the ability to handle a large number of users, i.e., good scalability, but also the ability to adapt to the dynamic changes of customers and products in the system.

In this paper, we proposed a framework to improve the scalability, and at the same time dynamically maintain the customer profiles. The framework is based on the proposed Incremental Hierarchical Agglomerative Clustering (IHAC) algorithms. IHAC is modified from the original Hierarchical Agglomerative Clustering (HAC) algorithms to handle the addition and deletion of both customers and products in real time. Based on the experiments on real data sets, the results showed that by applying our framework, the recommendation time was greatly reduced while the accuracy was relatively retained.

The remainder of this paper is organized as follows. In the next section, the proposed dynamic framework for implementing a recommender system is presented. Section

This paper is reprinted from a paper published in The 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, pp. 768-771, 2005

3 describes the incremental clustering algorithms considered for our framework. Experiments with performance evaluation are given in Section 4. The paper concludes in Section 5.

2. The Proposed Dynamic Recommender System Framework

Figure 1 illustrates overall framework which consists of three major components: task manager, processing modules and databases. Task manager includes the user interface which receives the request from the user and decides the appropriate actions to take. The request in the scope of e-commerce includes making the first purchase and become a new customer, purchasing more products, canceling orders, etc. The processing modules are composed of the incremental clustering and collaborative filtering modules. The incremental clustering is based on the Incremental Hierarchical Agglomerative Clustering (IHAC) algorithm and the collaborative filtering relies on a *neighborhood-based* algorithm. The database components consist of customer profiles and cluster structure. Customer profile contains ratings which are assigned to items by a customer.

The following example describe a situation where the system is used for an online store. Suppose a new user decides to purchase some products from the store, he/she would be required to register and fill out necessary information such as address and telephone number. This process is performed by the customer database manager which is connected to the recommender system. The content searching and browsing are performed via the content manager. Once the order is placed, a new customer profile will be created and the cluster structure must be updated to reflect this new customer. This step is performed through the incremental clustering component. Once the update is finished, the task manager would signal the collaborative filtering component to perform the recommendation task by using the updated user profiles and the cluster structure. The list of recommended products will then be generated and presented to the customer.

3. Incremental Hierarchical Agglomerative Clustering (IHAC)

Hierarchical clustering algorithms have been studied and successfully applied to organize documents in Information Retrieval (IR) system. There are two types of hierarchical clustering techniques: *agglomerative* and *divisive*. Of these two approaches, the Hierarchical Agglomerative Clustering (HAC) is more commonly used. Three approaches of HAC have been proposed: single link, complete link and group average link [2]. Each of these approaches are different in the way the similarity between clusters is determined. All of the HAC methods are known to have the time complexity of $O(n^2)$[2]. However, in the recommender system, the initial clustering construction can be performed as an off-line process, thus does not effect the recommendation response time.

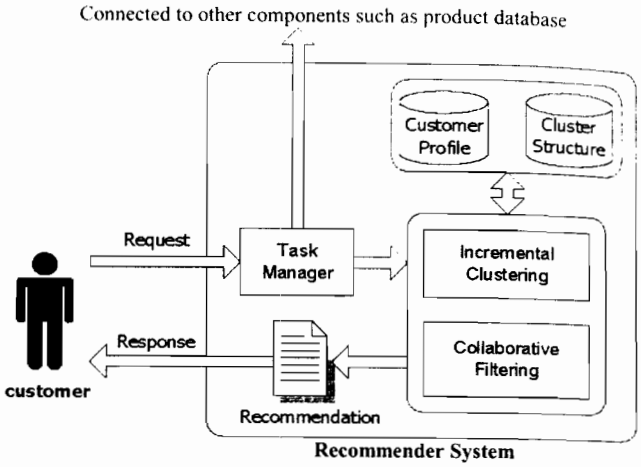

Figure 1. The proposed dynamic recommender system framework.

Typical recommender systems contain a large number of users and items. Since the HAC algorithms have an exponential running time complexity, it is therefore impractical to reconstruct the cluster structure as the system progresses. A dynamic system such as the recommender system implies that the users and items are changing on a regular basis. Therefore re-clustering the data set is prohibited as it would degrade the system response time and defeat the purpose of applying a clustering algorithm itself.

The incremental clustering refers to the technique of maintaining the cluster structure such that the overall performance of the cluster is well preserved. In this paper, we adopt a simple incremental strategy for HAC algorithms. When a new object, i.e., a set of user ratings, is presented to the cluster, the similarity of this object is compared to all other existing objects in the cluster. The definition of similarity between clusters are used according to the underlying HAC algorithms. For example, if the single link algorithm is used, the similarity between the new object and a cluster is the maximum similarity measures (or shortest distance) of the object to all items within the cluster. The new object is added to the same node which is the most similar.

4. Experiments and Discussion

The main data set used to perform our experiments is the movie rating database obtained from the MovieLens project Web site [7]. Specifically, the *100K MovieLens Dataset* is chosen. This data set consists of *100,000* ratings, with the scale of one to five, from *943* users on *1,682* movies. This data set is split into five training (base) and test sets. Each is split into 80% training and 20% test sets. The purpose is for performing five-fold cross validation in which the final results are averaged on these five sets.

4.1 Experimental Results

We used two performance metrics, averaged recommendation time per user and Mean Absolute Error (MAE), for evaluating the quality of different recommendation approaches [1, 4]. A set of experiments are performed to evaluate the performance among four approaches for recommendation:

- Recommendation without clustering (NC),
- Recommendation via single link clustering (SL)
- Recommendation via complete link clustering (CL)
- Recommendation via group average link clustering (GAL)

Consider the averaged MAE under all five data sets, the comparison plots are shown in Figure 2. From the figure, it can be seen that CL has the lowest MAE, i.e., highest recommendation accuracy, among all four approaches. SL has the worst MAE result. The interesting result is that CL yields better MAE result than NC for all user percentages. This means that applying a clustering method could help to reduce the overall recommendation error.

Consider the averaged recommendation time under all five data sets, the comparison plots are shown in Figure 3. From the figure, it can be obviously observed that NC has much higher recommendation time than all other clustering approaches. In addition, the increasing rate of the recommendation time for the NC approach is much faster than those of the clustering approaches. This indicates that without applying a clustering algorithm, the recommendation would suffer from poor scalability, i.e., high system response time. Both SL and GAL have the similar response times for all user sizes, whereas CL has the lowest response time and the lowest increasing rate, i.e., best scalability.

The next experiment evaluates the recommendation approaches via different IHAC algorithms. The incremental clustering is denoted by "INC" in front of the HAC algorithms. Again, all five movie data sets are used during the experiments. However, the variation is based on the ratio of percentages between *base* users versus *incremental* users. The main purpose is to analyze how well each of the incremental HAC algorithms could maintain the performance under the dynamic increase of participating users in the system. Within each of the data sets, users are separated into *base* users and *incremental* users. The base users are those who previously exist in the system before the experimental time, *t*. The incremental users are those who were added to the system after *t*. For our experiments, the portions of *base* users and *incremental* users are varied as follows: 20/80, 40/60, 60/40, and 80/20. For example, 20/80 means that 20% of all users are first randomly selected to form the cluster structure. Thereafter, the rest of 80% users are merged into the cluster structure.

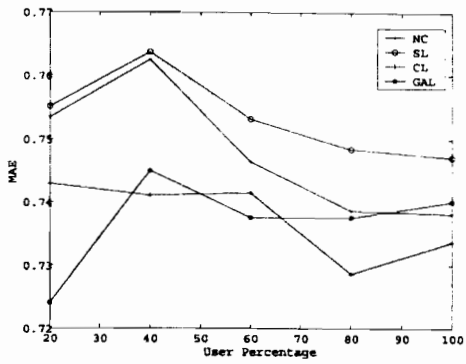

Figure 2. Mean Absolute Error (MAE) by varying user percentage

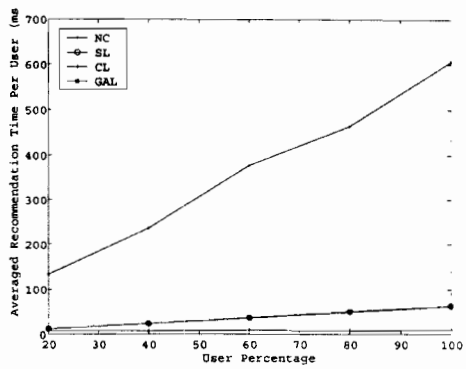

Figure 3. Averaged recommendation time by varying user percentage

Consider the averaged MAE under all five data sets, the comparison plots are shown in Figure 4. From the figure, INC-CL has a decreasing rate of recommendation errors. INC-CL even outperforms CL when the ratio of base and incremental users is 80/20. It is observed that CL has a noticeable smaller MAE than the original recommendation approach, NC. Finally, INC-SL has the highest MAE and does not show any significant changes in MAE.

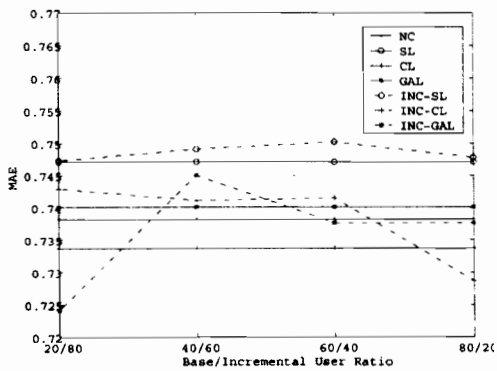

Figure 4. Mean Absolute Error (MAE) by adjusting base/incremental user percentage

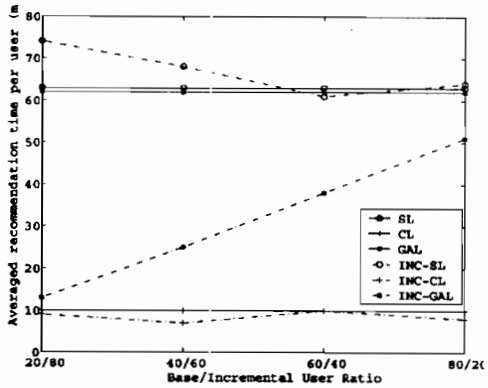

Figure 5. Averaged recommendation time by adjusting base/incremental user percentage

Consider the averaged recommendation time under all five data sets, the comparison plots are shown in Figure 5. The figure does not include the plot for NC approach. Since its value, i.e., 606 milliseconds, is much higher than all the other approaches. Thus, it can be implied that applying the clustering techniques for the recommendation process helps reducing the system response time. CL and INC-CL have the lowest averaged recommendation time and INC-CL is insensitive to the incremental changes in users. INC-GAL, on the other hand, is sensitive to the incremental changes. The recommendation time increases as the base users increases. This implies that merging additional users into the cluster structure provides better structure for finding the similar users. SL and INC-SL has the longest recommenda tion time. This is due to the typical characteristic of single link algorithm which tends to result in a long and staggering chain of clustering tree.

5. Conclusions

Typical recommender systems suffer from poor scalability and the lack of ability to handle dynamic changes in the user profiles. In this paper, a new dynamic recommender system framework was proposed to address these two issues. Our framework is based on the proposed Incremental Hierarchical Agglomerative Clustering (IHAC) algorithms. According to the results obtained from the experiments, applying IHAC helped reducing the averaged recommendation time per user up to five times, while the Mean Absolute Error (MAE) remained comparable to the traditional HAC approaches and the method without using any clustering algorithm.

References

[1] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, "An Algorithmic Framework for Performing Collaborative Filtering," Proc. of the 1999 Conf. on Research and Development in Information Retrieval, Aug. 1999.

[2] E. Rasmussen, "Clustering algorithms," in W. Frakes and R. Baeza-Yates (eds.), Information Retrieval: Data Structures and Algorithms, Prentice Hall, 1992.

[3] P. Resnick, N. Iacovou, M. Sushak, P. Bergstrom, and J. Riedl, "Grouplens: An open architecture for collaborative filtering of Netnews," Proc. of ACM 1994 Conf. on Comput. Supported Cooperative Work, pp. 175–186, Oct. 1994.

[4] B. M. Sarwar, G. Karypis, J. Konstan, and J. T. Riedl, "Application of dimensionality reduction in recommender system–A case study," Proc. of ACM WEBKDD, pp. 82–90, 2000.

[5] J. B. Schafer, J. A. Konstan, and J. Riedl, "Ecommerce recommendation application," Data Mining and Knowledge Discovery, 5(1/2):115–153, 2001.

[6] L. H. Ungar and D. P. Foster, "Clustering methods for Collaborative Filtering," Proc. of the AAAI-98 Workshop on Recommender Systems, pp. 112–125, 1998.

[7] MovieLens Project Web site, http://movielens.umn.edu