Hybrid Product Recommendation for e-Commerce : A Clustering-based CF Algorithm

Do Hyun Ahn[a], Jae Sik Kim[b], Jae Kyeong Kim[c] and Yoon Ho Cho[d]

[a,b,c] School of Business Administration, KyungHee University
1, Hoeki-dong, Dongdaemoon-gu, Seoul, 130-701, South Korea
Tel: +82-2-961-9355, Fax: +82-2-967-0788, E-mail: jaek@khu.ac.kr

[d] Department of Internet Information, Dongyang Technical College
62-160, Gochuck-dong, Guro-gu, Seoul, 152-714, South Korea
Tel: +82-2-2610-1918, Fax: +82-2-2610-1859, E-mail: yhcho@dongyang.ac.kr

Abstract

Recommender systems are a personalized information filtering technology to help customers find the products they would like to purchase. Collaborative filtering (CF) has been known to be the most successful recommendation technology. However, its widespread use in e-commerce has exposed two research issues, sparsity and scalability. In this paper, we propose several hybrid recommender procedures based on web usage mining, clustering techniques and collaborative filtering to address these issues. Experimental evaluation of suggested procedures on real e-commerce data shows interesting relation between characteristics of procedures and diverse situations.

Keyword: *Product recommendation, Web usage mining, Clustering, Collaborative Filtering, Personalization*

1. Introduction

The rapid expansion of e-commerce forces existing recommender systems to deal with a large number of customers and products [Melville, et al., 2001]. Collaborative Filtering (CF) [Hill, et al., 1997] has been known to be the most successful recommendation technique that has been used in a number of different applications. However, CF based recommender systems suffer from two fundamental problems, sparsity and scalability. To overcome these problems, we propose hybrid recommender procedures and show experimental results of their performance.

The characteristics of our suggested procedures are as follows: (1) Clustering techniques are applied to improve scalability of recommender systems. (2) Products are recommended to target customers according to Web usage mining based CF to address sparsity issue.

To compare the effect of clustering and web usage mining, the procedures are evaluated with real Internet shopping mall data.

The remainder of the paper is organized as follows. Section 2 reviews the past research works related to our research. Section 3 provides our research framework. Section 4 describes experimental works. Section 5 finally provides conclusions and future works.

2. Backgrounds

2.1 Recommender Systems

Recommender systems are changing the face of e-commerce on the Internet by enabling Web sites to help their customers find products they will be interested in buying. These systems apply data analysis techniques to the problem of helping customers find the products they would like to purchase at e-commerce sites by producing a prediction score or a list of top-N recommended products for a given customer. For instance, a recommender system on Amazon.com (www. amazon. com) suggests books to customers based on other books the customers have told

Amazon they like. Recommendations can be based on demographics of the customers, overall top selling products, or past buying habit of customers as a predictor of future products [Sarwar et al., 2001]. In essence, these techniques try to personalize the e-commerce space for the customers. Among the different approaches applied to achieve personalization in e-commerce, Collaborative Filtering (CF) is arguably the most successful technique deployed in commercial applications as well as in academic research [Goldberg et al., 1992].

Recommender systems increase e-commerce sales in three ways. First, recommender systems help to convert browsers into buyers by providing personalized recommendations on a variety of products. Second, recommender systems improve cross-sell by suggesting additional products for the customer to purchase. Third, recommender systems improve loyalty by creating a value-added relationship between the e-commerce site and the customer. Numerous recommender systems have been built for both research and practice. Although the algorithms behind these systems vary, most are based on one or more of two classes of technology.

2.2 Collaborative Filtering Algorithm

Collaborative Filtering (CF) presents an alternative information evaluation approach based on the judgments of human beings. It attempts to automate the "word of mouth" recommendations that we regularly receive from family, friends, and colleagues. In essence, CF allows everyone to serve. This inclusiveness circumvents the scalability problems and it becomes possible to review millions of books (Sarwar, 2001). Automated CF systems use a machine learning approach called the nearest neighbor algorithm to provide a computer implementation of this technique. Such systems maintain a database containing the ratings that each user has given to each item that each user has evaluated (e.g. in the form of a score from 1 to 5). For each user in the system, the recommendation engine computes a neighborhood of other users with similar options; this neighborhood is usually based on a proximity measure such as correlation. To evaluate other items for this user, the system forms a normalized and weighted average of the opinions of the user's neighbors.

A common interface to CF systems is the Recommender System (Resnick & Varian, 1997). Recommender Systems provide several application interfaces. An application may add ratings when a user has provided explicit or implicit ratings for an item. The application may also request a prediction of a user's interest level in a specific item or request a set of recommendations of items the user would likely prefer. Finally, some recommender system interfaces allow the application to retrieve a user's neighbors to form affinity groups. Several recommender systems based on automated CF have been developed.

2.3 Web Usage Mining

Web usage mining is the process of applying data mining techniques to the discovery of behavior patterns based on Web log data, for various applications. In the advance of e-commerce, the importance of Web usage mining grows larger than before. The overall process on Web usage mining is generally divided into two main tasks; data preprocessing and pattern discovery. Mining behavior patterns from Web log data needs the data preprocessing tasks that include data cleansing, user identification, session identification, and path completion. Data cleansing performs merging Web logs from multiple servers, removing irrelevant and redundant log entries with filename suffixes such as gif, jpeg, map, count.cgi, and so on, and parsing of the logs. To track individual user's behaviors at a Web site, user identification and session identification is required. For Web sites using session tracking such as URL rewriting, persistent cookies or embedded session IDs, user and session identification is trivial. Web sites without session tracking must rely on heuristics. Path completion may also be necessary because of local or proxy level caching. Cooley, Mobasher and Srivastava (1999) presented a detailed description of data preprocessing methods for

mining Web browsing patterns. The pattern discovery tasks involve the discovery of association rules, sequential patterns, usage clusters, page clusters, user classifications or any other pattern discovery method [Mobasher et al., 2000]. Usage patterns extracted from Web data can be applied to a wide range of applications such as Web personalization, system improvement, site modification, business intelligence discovery, usage characterization, and so on [Srivastava et al., 2000].

There have been several customer behavior models for e-commerce, which have different analysis purposes. Menascé et al. (1999) have presented a state transition graph, called Customer Behavior Model Graph (CBMG) to describe the behavior of groups of customers who exhibit similar navigational patterns. VandeMeer, Dutta and Datta (2000) have developed a user navigation model designed for supporting and tracking dynamic user behavior in online personalization. The model supports the notion of a product catalog, user navigation over this catalog and dynamic content delivery. Lee et al. (2001) have provided a detailed case study of clickstream analysis from an online retail store. A part of Lee et al's model is adapted to our research, because they focus the online retailer that is our consideration as well. They have analyzed the shopping behavior of customers according to the following four shopping steps; product impression, click-through, basket placement, and purchase. And they have applied micro-conversion rates (e.g., click-to-buy rate) computed for each adjacent pair of these steps in order to measure the effectiveness of efforts in merchandising.

2.4 Clustering Techniques

Clustering techniques have been studied extensively in statistics, pattern recognition, and machine learning. Current clustering techniques can be broadly classified into two categories: partition and hierarchical. Given a set of objects and a clustering criterion, partitional clustering obtains a partition of the objects into clusters such that the objects in different clusters. The popular k-means and k-medoids methods determine k cluster representative and assign each object to the cluster with its representative closest to the object such that the sum of the distances squared between the objects and their representative is minimized.

Clustering is also dimensionality reduction technique. Clustering techniques and cluster analysis have been applied to a wide range of disciplines such as multivariate statistics, demography, ecology, clinical diagnosis, and market research to name but a few. We, however, restrict our focus on multivariate analysis and clustering algorithms applied in collaborative filtering and related fields. A discussion of single-link clustering algorithms such as nearest-neighbor and minimum spanning tree based algorithms is given in [Konstan, et, al, 1997]. Other classical clustering algorithms such as K-Nearest Neighbor (KNN or K-means) and E-M algorithms are discussed in [Ungar & Foster, 1998]. The authors mention that current collaborative filtering methods use K-NN algorithm for the neighborhood formation. Earlier collaborative filtering research conducted in the Usenet domain [Konstan, et, al, 1997] reported the benefits of partitioning. In particular, they found improved prediction quality with partitioned newsgroups compared to the whole Usenet. This result prompted researchers to investigate further. In an attempt to partition the database using clustering techniques researchers have shown that partitioning movie database by genre and by clustering, in fact, improved the quality of predictions. The authors used a publicly available graph partition program to perform the clustering operation.

3. Methodology

3.1. Overall Procedure

The overall procedure of our methodology is shown in Figure 1. Our research suggests 4 different procedures as follows.

Method 1: CF with Purchase data + No Clustering

Method 2: CF with Web log data + No Clustering

Method 3: CF with Purchase data + Clustering

Method 4: CF with Web log data + Clustering

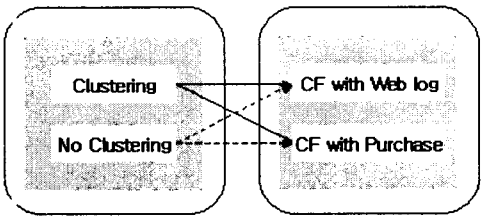


Figure 1- Overall procedure of Hybrid Procedures

3.2. Clustering Phase

We consider the application of clustering techniques to improve scalability of recommender systems. Earlier studies [Konstan et al., 1997] indicate the benefits of applying clustering in recommender systems. Therefore, we use the *k*-means method that has been shown to be effective in producing good clustering results for many practical applications.

The *k*-means algorithm proceeds as follows. First, it randomly selects *k* of the objects, each of which initially represents a cluster mean or center. For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean. It then computes the new mean for each cluster. This process iterates until the criterion function converges. We briefly describe the *k*-means clustering algorithm as follows [Han & Kamber, 2001]. Let the k prototypes $(w_1,\ldots, w_k)$ be initialized to one of the n input patterns $(i_1,\ldots, i_n)$. Therefore,

$$w_j = i_l, \quad j \in \{1,\ldots,k\}, \; l \in \{1,\ldots,n\}$$

C_j is the j^{th} cluster whose value is a disjoint subset of input patterns. The quality of the clustering is determined by the following error function:

$$E = \sum_{j=1}^{k} \sum_{i_l \in C_j} | i_l - w_j |^2$$

Using the above *k*-means clustering algorithm, we group customers with demographic and behavior data such as age, gender, job, frequency, recency, duration, click-through, basket- placement and purchase information for improving the scalability of recommender systems. Generally, the types of data can be divided into three basic types: demographic data, behavior data, psychographic data.

3.3. CF Phase

A CF algorithm is composed of profile creation, neighborhood formation, and generation of recommended products.

Step 1. Profile Creation

A profile is a collection of information that describes a user. One of the important issues in the profile creation is what information should be included in a user profile.

Ratings based on Purchase data: Ratings based on purchase data are collections of historical purchasing transaction of n customer on m products. It is usually represented as an m x n customer-product matrix, R, such that ri,j is one if the *i*th customer has purchased the jth product, and zero, otherwise [Sarwar, et al., 2000].

Ratings based on Web data: The customer profile is constructed based the following three general shopping steps(click-through, basket placement, and purchase) in online stores modified from works of Lee et al. (2001):

1. *click-through*: the click on the hyperlink and the view of the Web page of the product,
2. *basket placement*: the placement of the product in the shopping basket,
3. *purchase*: the purchase of the product – completion of a transaction.

A basic idea of measuring the customer's preference is simple and straightforward. The customer's preference is measured by counting only the number of occurrence of URLs mapped to the product from clickstream of the customer. In general Internet shopping malls, products are purchased in accordance with the three sequential shopping

steps, so we can classify all products into four product groups such as purchased products, products placed in the basket, products clicked through, and the other products. This classification provides an *is-a* relation between different groups such that purchased products *is-a* products placed in the basket, and products placed in the basket *is-a* products clicked through. From this relation, it is reasonable to obtain a preference order between products such that {products never clicked} π {products only clicked through} π {products only placed in the basket} π {purchased products}. Hence, it makes sense to assign the higher weight to occurrences of purchased products than those of products only placed in the basket. Similarly, the higher weight is given to products only placed in the basket than those of products only clicked through, and so on

Let p_{ij}^{c} be the total number of occurrences of click-throughs of a customer i across every products in a grain product class j. Likewise, p_{ij}^{b} and p_{ij}^{p} are defined as the total number of occurrences of basket placements and purchases of a customer i for a grain product class j, respectively. p_{ij}^{c}, p_{ij}^{b} and p_{ij}^{p} are calculated from clickstream data as the sum over the given time period, and so reflect individual customer's behaviors in the corresponding shopping process over multiple shopping visits.

From the above terminology, we define the customer preference matrix $\mathbf{P} = (p_{ij})$, $i = 1, \Lambda, M$ (total number of customers), $j = 1, \Lambda, N$ (total number of grain product classes, i.e., $|G|$), as follows:

$$p_{ij} = \left(\frac{p_{ij}^{c} - \min_{1 \le j \le N}(p_{ij}^{c})}{\max_{1 \le j \le N}(p_{ij}^{c}) - \min_{1 \le j \le N}(p_{ij}^{c})} + \frac{p_{ij}^{b} - \min_{1 \le j \le N}(p_{ij}^{b})}{\max_{1 \le j \le N}(p_{ij}^{b}) - \min_{1 \le j \le N}(p_{ij}^{b})} + \frac{p_{ij}^{p} - \min_{1 \le j \le N}(p_{ij}^{p})}{\max_{1 \le j \le N}(p_{ij}^{p}) - \min_{1 \le j \le N}(p_{ij}^{p})} \right) \times \frac{1}{3} \quad (1)$$

Please note that the weights for each shopping step are not the same although they look equal as in Equation (1). From a casual fact that customers who purchased a specific product had already not only clicked several Web pages related to it but placed it in the shopping basket, we can see that Equation (1) reflects the weight difference.

Step 2: Neighborhood Formation

The goal of neighborhood formation is to find, for each customer u, and ordered list of l customers N = {N1, N2, ····, Nl} such that u Ï N and sim(u, N1) is maximum, sim(u, N2) is the next maximum and so on [Sarwar et al., 2000].

Pearson correlation: Proximity between two users a and b is measured by computing the Pearson correlation $corr_{ab}$, which is given by

$$corr_{ab} = \frac{\sum_i (r_{ai} - \bar{r}_a)(r_{bi} - \bar{r}_b)}{\sqrt{\sum_i (r_{ai} - \bar{r}_a)^2 (r_{bi} - \bar{r}_b)^2}}$$

Cosine: In this case two customers a and b are thought of as two vectors in the m dimensional product space. The proximity between them is measured by computing the cosine of the angle between the two vectors, which is given by

$$\cos(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|_2 \cdot \|\vec{b}\|_2}$$

Step 3: Generation of Recommendation list

The final phase of our methodology is to ultimately derive the top-N recommendation from the neighborhood of customers. We suggest three different techniques for generating a recommendation list for a given customer.

Recommendation of the most frequently purchased product (MFP): This technique, adopted from of the study of Sarwar, et al. (2000), looks into the neighborhood and for each neighbor, scans through a sales database and counts the purchase frequency of the products.

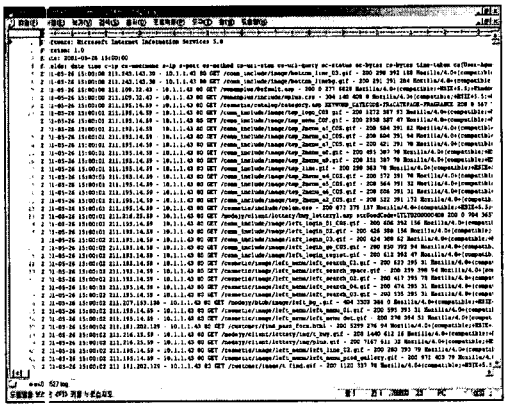
4. Experimental Evaluation

4.1. Data Preparation

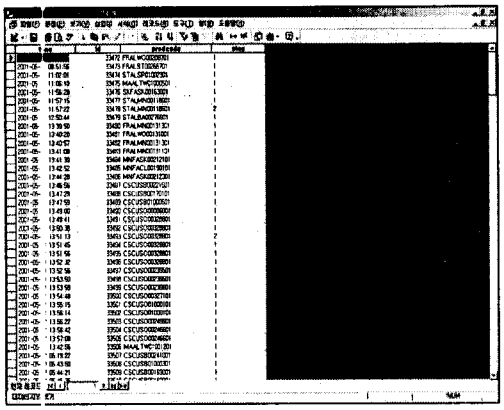
For our experiments, we use Web log data and product data from the S Internet shopping mall that sells women's

supplies.

Web log data. The 110 log files was collected from four IIS Web servers during period between 1st May 2001 and 7th June 2001. The total size of log files is about 25,360MB, and total number of HTTP requests is about 510,000,000,000. For an application to our experiments, the preprocessing tasks such as data cleansing, user identification, session identification, path completion, and URL parsing were applied to the log files. Finally, we obtained a transaction database in the form of *<time, customer-id, product-id, shopping-step>* which the shopping-step represents one of the click-through step, the basket-placement step and the purchase step. This database contains transactions of 49597 customers on 278 products. In total, the database contains 428,510 records that consist of 781 purchase records, 5,350 basket-placement records, and 422,379 click-through records. Figure 2 provides raw Web log data and the corresponding transaction database.



(a) Raw web log data



(b) Transaction database

Figure 2 - Web log preprocessing

We set the period between 1st May 2001 and 24th May 2001 and the period between 25th May 2001 and 7th June 2001 as the training period and the test period, respectively. And then, as the target customers, we selected 130 customers who have purchased one more products in the training period and clicked one more products for the test period. Finally, the training set consists of 6,331 transaction records created by the target customers for the training period, and the test set consists of 677 click-through records created by them for the test period.

Product data. S Internet shopping mall deals with 7513 products. Table 1 shows products managed in S Internet shopping mall.

Table 1- example of product data set

prodcode	prodname	classcode	classname
MWCACD00H02901	니트점퍼형가디건	MWCACD	가디건
MWCACT99H64701	라이트다운코트	MWCACT	코트
MWCACT99H85801	맨스노우롱다운	MWCACT	코트
MWCAJK00B73501	크링클자켓	MWCAJK	자켓
MWCAJK00B73601	스웨이드자켓	MWCAJK	자켓
MWCAJP00A42001	롤리버시블점퍼	MWCAJP	점퍼
MWCAJP00A55401	남성양면점퍼	MWCAJP	점퍼
MWCAJP00B37001	힙합후드점퍼	MWCAJP	점퍼
MWCASH00A41501	서큐넥버튼셔츠	MWCASH	셔츠
MWCASH00A41510	서큐넥버튼셔츠	MWCASH	셔츠
MWSUJK00F93801	3버튼자켓	MWSUJK	자켓
MWSUJK00F93901	기획정장자켓	MWSUJK	자켓
MWCASW00A46903	라운드스웨터	MWCASW	스웨터
MWCASW00A46914	라운드스웨터	MWCASW	스웨터

4.2. Evaluation Metrics

Recommender systems research has used a number of different measures for evaluating the success of a recommender system. Main research objective of this paper is to develop new procedures for making recommendations that has better quality and more speed compared to previously studied approaches. Therefore, two evaluation metrics are employed for evaluating our procedures in terms of quality and performance requirements.

4.2.1. Quality evaluation metric

With the training set and the test set, our 4 methods work on the training set first, and then it generates a set of recommended products, called recommendation set, for a given customer. To evaluate the quality of the

recommendation set, *recall* and *precision* have been widely used in the recommender system community [Sarwar et al., 2000]. Recall is defined as the ratio of the number of products in both test set and recommendation set to the number of products in test set.

Precision is defined as the ratio of the number of products in both test set and recommendation set to the number of products in recommendation set. Recall means how many of all the products in the actual customer purchase list are recommended correctly whereas precision means how many of the recommended products belong to actual customer purchase list. These measures are simple to compute and intuitively appealing, but they are often in conflict since increasing the size of recommendation set tends to increase recall but at the same time decrease precision [Sarwar et al., 2000]. Hence, a widely used combination metric called *F1 metric* that gives equal weight to both recall and precision is employed for our evaluation, and computed as follows:

$$F1 = \frac{2 \times recall \times precision}{recall + precision}$$

4.2.2. Performance evaluation metric

To evaluate the scalability issue, we use a performance evaluation metric in addition to the quality evaluation metric. The *response time* are employed to measure the system performance. The response time defines the amount of time required to compute all the recommendations for the training set per second.

4.3. Experiment Results

In this section, we present a detailed experimental evaluation of the different procedures.

4.3.1. Experiments with neighborhood size

The size of the neighborhood has significant impact on the recommendation quality [Sarwar et al, 2000]. To determine the sensitivity of neighborhood size, we performed an experiment in which we varied the number of neighbors and computed the corresponding *F1* metric. Figure 3 shows our experimental results. Looking into the results, we can see that the size of the neighborhood does affect the quality of *top-N* recommendations.

In general, the quality increases as we increase the number of neighbors, but, after a certain peak, the improvement gains diminish and the quality becomes worse This reason may be that choosing too many neighbors result in too much noise for those who have high correlates. In the case of Web data, the peak is reached in the 15, whereas in case of Purchase data is reached in the 24. Hence, we used a neighborhood of size 15 for the Web data and that of 24 for the Purchase data as our ideal choice of neighborhood size.

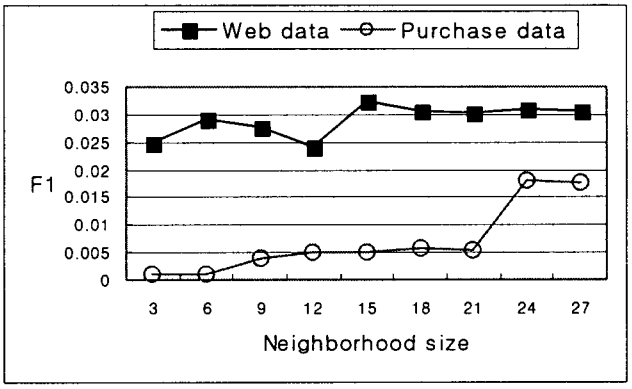


Figure 3 - Impact of neighborhood size on recommendation quality

4.3.2. The effect of Web log data

Given the optimal values of the parameters, we compare CF with Purchase data with CF with Web log data. Our results are shown in Figure 4. It can be observed from the chart that CF with Web log data works better than CF with Purchase at all the number of recommended products. The recommendation with Web log data results better performance than that of Purchase data only.

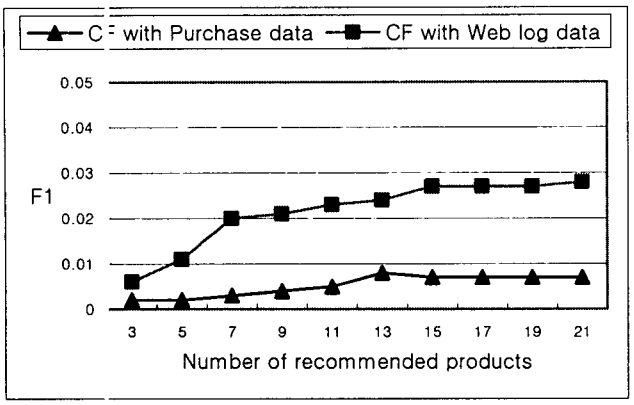


Figure 4 – The effect of Web log data

4.3.3. The effect of Clustering Technique

We also compare No Clustering based CF with Clustering based CF. Our results are shown in Figure 5. We can see that the quality of Clustering based CF is better than that of No Clustering based CF. However, using the Clustering technique is not robust performance, especially at a few number of recommended products.

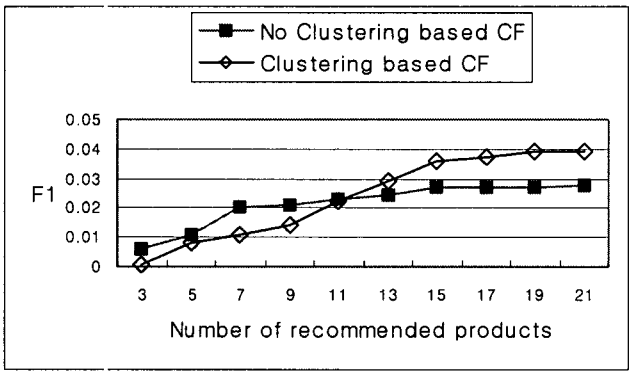


Figure 5 – The effect of Clustering Technique

4.3.4. Comparison of four Methods

With the number of recommended products from 3 to 21, Figure 6 shows the comparison of method 1, 2, 3 and 4. It can be observed from the charts that Method 2 and Method 4 work better than Method 1 and Method 3 at all the number of recommended products. This implies that Web usage mining gives better results. Furthermore, we can see that the quality of Clustering based CF is better than that of No Clustering based CF. However, the application of clustering did not always give better performance.

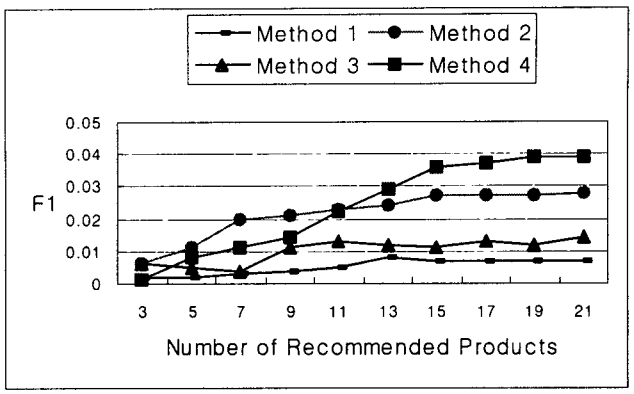


Figure 6 – Comparison of four Methods

4.3.5. Performance comparison of hybrid algorithms

To compare the performance of our hybrid procedures with that of the benchmark CF algorithm, we performed an experiment in which we measure the *response time* of each procedure. The response time means the amount of time required to compute all the recommendations for the training set per second. Table 2 shows the response time provided by the three algorithms. Looking into the results shown in Table 2, we can see that the performance of [Method 3, Method 4] is better than that of other methods. We believe this is due to the effect of Clustering technique.

Table 2 - Performance comparison of our hybrid algorithms

Hybrid Procedures	Response time(sec.)
Method 1: CF with Purchase data + No Clustering	405.2
Method 2: CF with Web log data + No Clustering	1173.5
Method 3: CF with Purchase data + Clustering	23
Method 4: CF with Web log data + Clustering	84

5. Conclusion

5.1. Summary

We suggested hybrid recommender procedures based on web usage mining, clustering and collaborative filtering. We experimentally evaluated our hybrid procedures on real

e-commerce data and compared the effect of each approach.

Based on the experiments, we compared the quality of CF based on web usage mining with that of CF based on purchase data and then evaluated the effect of Clustering technique. Our experiments presented that the quality of CF with Web log data) better than CF with Purchase data. However, the application of clustering did not always give better performance.

5.2. Contributions

The research work presented in this paper makes the following contributions to the recommender systems related research community.

(1) Application of the k-means clustering algorithm to improve scalability of recommender systems.
(2) Development of a clickstream analysis technique to capture implicit ratings by tracking customer shopping behavior on the Web and its application to reduce the sparsity.
(3) Development of a methodology (clustering + CF based on Web usage mining) to apply data mining techniques for enhancing collaborative recommendations, in which Web usage mining and clustering algorithm is applied to address sparsity, scalability issues together.
(4) Suggestion of methodologies and evaluation of them with the real Internet shopping mall data to compare the effect of each approach.

5.3. Future Works

While our experimental results suggest that the proposed methodology are promising new recommendation methodology, these results are based on studies limited to the particular e-commerce site that has small customers, products, and transactions. Therefore, it is required to evaluate our methodologies in more detail using data sets from a variety of large e-commerce sets. As future works, it will be interesting to compare our suggested methodologies with one of outstanding approaches to reduce the dimensionality of recommender system databases in the aspect of recommendation performance. And it will be also an interesting research area to conduct a real marketing campaign to customers using our methodologies and to evaluate their performance.

References

[1] Alsabti, K., Ranka, S., & Singh, V. (1998). An Efficient K-Means Clustering Algorithm. Proc, First Workshop on High-Performance Data Mining.

[2] Bradley, P. S., Fayyad, U. M. and Reina, C. (1998). Scaling Clustering Algorithms to Large Databases. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD98), 9-15.

[3] Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 43-52.

[4] Cho, Y.H., Kim, J,K., and Kim, S.H. (2002). A Personalized Recommender System based on Web Usage Mining and Decision Tree Induction. Expert Systems With Applications, Vol. 23(3)

[5] Cooley, R., Mobasher, B., and Srivastava, J. (1999). Data preparation for mining World Wide Web browsing patterns. Journal of Knowledge and Information Systems, 1.

[6] Dutta, K.; and Datta, A. (2000). Enabling scalable online personalization on the Web. In *Proceedings of ACM E-Commerce Conference*, 185-196.

[7] Goldberg, D., Nichols, D., Oki, B. M. & Terry, D. (1992). Using Collaborative Filtering to Weave an

[8] Information Tapestry. Communications of the ACM, 35(12), 61-70.

[9] Han, J. and Kamber, M. (2001). Data mining: concepts and techniques, Morgan Kaufmann Publishers.

[10] Hill, W., Stead, L., Rosenstein, M., and Furnas, G. (1995). Recommending and evaluating choices in a virtual community of use. In Proceedings of the 1995 ACM Conference on Human Factors in Computing Systems, 194-201.

[11] Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R. and Riedl, J. (1997). GroupLens: applying collaborative filtering to usenet news. Communications of the ACM, 40, 3, 77-87.

[12] Lee, T., Podlaseck, M., Schonberg, E. and Hoch, R. (2001). Visualization and analysis of clickstream data of online stores for understanding web merchandising. *Data Mining and Knowledge Discovery*, 5, 1-2, 59-84.

[13] Melville, P., Mooney, R.J, and Nagarajan, R. (2001). Content-Boosted Collaborative Filtering. In the Proceedings of the SIGIR-2001 Workshop on Recommender Systems.

[14] Menascé, D.A., Almeida, V.A., Fonseca, R. and Mendes, M.A. (1999). A methodology for workload characterization of e-commerce sites. In *Proceedings of ACM E-Commerce Conference*, 119-128.

[15] Mobasher, B., Dai, H., Luo, T., Sun, Y. and Zhu, J. (2000). Integrating Web usage and content mining for more effective personalization. In *Proceedings of the EC-Web* 2000, 165-176.

[16] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of netnews. In Proceedings of the ACM 1994 Conference on Computer Supported Cooperative Work, 175-186.

[17] Sarwar, B., Karypis, G., Konstan, J. and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithm. In Proceedings of The Tenth International World Wide Web Conference, 285-295.

[18] Srivastava, J., Cooley, R., Deshpande, M. and Tan P. (2000). Web usage mining: discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1, 2, 1-12.