韓國經營科學會誌
第29卷 第2號
2004年 6月

재구성된 제품 계층도를 이용한 협업 추천 방법론 및 그 평가

조윤호* · 박수경** · 안도현** · 김재경**

Collaborative Recommendations using Adjusted Product Hierarchy : Methodology and Evaluation

Yoon Ho Cho* · Su Kyung Park** · Do Hyun Ahn** · Jae Kyeong Kim**

■ Abstract ■

Recommendation is a personalized information filtering technology to help customers find which products they would like to purchase. Collaborative filtering works by matching customer preferences to other customers in making recommendations. But collaborative filtering based recommendations have two major limitations, sparsity and scalability. To overcome these problems we suggest using adjusted product hierarchy, grain. This methodology focuses on dimensionality reduction and uses a marketer's specific knowledge or experience to improve recommendation quality. The quality of recommendations using each grain is compared with others by several experimentations. Experiments present that the usage of a grain holds the promise of allowing CF-based recommendations to scale to large data sets and at the same time produces better recommendations. In addition, our methodology is proved to save the computation time by 3~4 times compared with collaborative filtering.

Keyword : Recommendation, Collaborative Filtering, Product Hierarchy, CRM

논문접수일 : 2003년 11월 5일　　논문게재확정일 : 2004년 5월 10일
* School of e-Business, Kookmin University, Seoul, Korea
** School of Business Administration, Kyung Hee University, Seoul, Korea

1. Introduction

Many companies offer millions of products and have become to be harder to survive in competitions. Customers also face a difficult situation to choose a particular product among products. Especially the amount of information and products in the e-business world are increasing more quickly than our ability to process them.

Recommender systems are a personalized information filtering technology, used to identify a set of top-N products that will be of interest to a target customer. One of the most successful recommender technologies is collaborative filtering (CF)[Hill et al., 1997 ; Konstan et al., 1997 ; Resnick et al., 1994 ; Shardanand et al., 1995 ; Sarwar et al., 2000]. CF works by building a database of preferences for products by customers. CF-based recommendations use historical information to identify a neighborhood of people that have exhibited similar behavior in the past and then analyze this neighborhood to identify new products that will be preferred by the user. However, despite of their success, CF-based recommender systems have two major limitations. The first limitation is related to sparsity. The amount of historical information for each user and for each product is often quite limited. As a result, CF-based recommendations cannot accurately compute the neighborhood and identify the products to recommend. The second limitation is related to scalability. Algorithms finding nearest neighbor usually require the computation time that grows linearly with both the number of customers and the number of products. With millions of customers and products, existing CF-based recommendations suffer serious scalability problems.

To address these problems, a variety of techniques have been developed and used [Han et al., 1994; Kim et. al, 2002; Kim & Cho, 2003 ; Lu et al., 1997; Sarwar et al., 2001b]. Among these techniques it is not easy to find CF methodologies using expert marketers' domain-specific knowledge. However, the marketer's knowledge plays an important role in the product recommendation process. One of such knowledge is represented as hierarchies or taxonomies over data primitives. The use of a product hierarchy can be a useful way to solve the limitations of collaborative filtering.

In this paper, we suggest a CF-based recommendation methodology to use dimensionality reductionby proposing an adjusted product hierarchy. An adjusted product hierarchy is defined as a reconstructed product hierarchy based on the criterion of purchased amount of a product or a marketer's knowledge and experience. The focus of this paper is three-fold. First, we would like to enhance the quality of collaborative filtering by using adjusted hierarchies. Second, we introduce a new measure to the step of neighborhood formation to overcome the problem of scalability and find true neighbors of a target customer. Third, we analyze the effectiveness of recommendations and newly devised procedure on actual customer data from H department.

Our suggested methodology is faster in the performance than previous ones. Furthermore, we can find the most efficient product hierarchy among adjusted product hierarchies after evaluating their performance. And the participation of a marketer results in providing better performance in recommendations.

The rest of the paper is organized as follows. The next section provides a brief review of some related research work. The overall procedure is described in detail at a following section. Next it is described the experimental evaluation on actual customer data. The final section provides concluding remarks and discussions.

2. Background

2.1 Recommendation Algorithm

Recommendation has been developed to propose the product to the customer using statistic and knowledge exploration. The underlying techniques used in today's recommendation systems fall into two distinct categories : content-based filtering and collaborative filtering(CF)[Billsus et al., 1997]. Content-based methods require textual descriptions of the items to be recommended and draw on results from both information retrieval and machine learning research. In general, a content-based filtering analyzes a set of documents rated by an individual user and uses the content of these documents, as well as the provided ratings, to infer a profile that can be used to recommend additional items of interest. In contrast, collaborative filtering recommends items based on aggregated user ratings of those items, i.e. these techniques do not depend on the availability of textual descriptions.

Collaborative filtering has many significant advantages over traditional content-based filtering, primarily because it does not depend on error-prone machine analysis of content [Herlocker et al., 1999]. The advantages include the ability to filter any type of content (e.g. text, art work, music), the ability to filter based on complex and hard to represent concepts, such as taste and quality, and the ability to make serendipitous recommendations. It is important to note that collaborative filtering does not necessarily compete with content-based filtering. In most cases, they can be integrated to provide a powerful hybrid filtering solution.

However, in this paper, only collaborative filtering is considered in our research to make recommendations. Although collaborative filtering has been shown to produce high quality recommendations, the performance degrades with the number of customers and products. Therefore, many techniques have been studied to reduce the amount of informationof customers and products, which means dimensionality reduction. Among several techniques, dimensionality reduction using a marketer's knowledge and experience is being stressed from researchers. We use this dimensionality reduction technique, especially via the product hierarchy representing her/his knowledge.

In the following we briefly review some of the research literature related to our work in the field of collaborative filtering and dimensionality reduction using a product hierarchy.

2.2 Collaborative filtering

Collaborative filtering is the most successful recommendation method to date and used in many of the recommender systems. This recommends products to a target customer based on the opinions of other customers. Sarwar et al.(2000) investigate several techniques for analyzing large-scale purchase data for the purpose of producing useful recommendations to customers. They applied a collection of algorithms such as traditionaldata mining, nearest-neighbor collaborativefiltering, and dimensionality reduction on two different data sets. The CF-based techniques are reported to do better than the traditional rule-based approach. In nearest-neighbor collaborative filtering, the center-based technique does better compared to the aggregated neighborhood.

Tapestry is one of the earliest implementations of collaborative filtering based recommender systems [Goldberg et al., 1992]. This system relied on the explicit opinions of people from a close-knit community. But recommender systems for large communities cannot depend on each person knowing the others. Another type of dimensionality reduction, Latent Semantic Indexing(LSI), is widely used in information retrieval community [Berry et al., 1995 ; Deerwester et al., 1990]. LSI uses singular decomposition(SVD) to factor the original space into three matrices and the process of dimensionality reduction is performed by reducing the singular matrix. Sarwar et al.(1999) investigated the use of dimensionality reduction for recommender systems using SVD technique and presented the effectiveness of reduced matrix. Also they suggested that the SVD algorithms fitted well with the collaborative filtering data and neighborhoods formed in the reduced dimensional space were better than their high dimensional counterparts. A special issue of Communications of the ACM presents a number of different recommender systems [Resnick et al., 1997]. Despite their success, the widespread use has exposed some limitations such as sparsity and scalability and so on. Sparsity problem in recommender systems has been addressed in [Good et al., 1999; Sarwar et al., 2001a] and the problem associated with scalability has been discussed in Sarwar et al.(2001b) and Kim & Cho (2003). To overcome these problems this research suggests using adjusted product hierarchy, grain as a dimensionality reduction method.

2.3 Usage of a marketer's knowledge

There have been few research work done in the area of a marketer's specific knowledge. However, in data mining field many researchers have emphasized the usage of this knowledge [Adomavicius & Tuzhilin, 2001 ; Berry & Linoff, 1997; Lawrence et al., 2001]. In particular, Adomavicius & Tuzhilin(2001) have shown that an explicit participation of the marketer is required to build more personalized profiles which can lead to the improvement of recommendation quality. The formal use of hierarchies as the most important background knowledge in data mining is introduced by Han, Cai, and Cercone(1993). Lu(1997) also deals with hierarchies as one of the most important background knowledge in the context of data mining. Brew(1991) and Mellish(1991) have shown that hierarchies are important in knowledge representation and reasoning. As the size of hierarchies increases, there is a growing need to represent them in a form that is amenable to performing operations efficiently. This paper presents superior advantages of hierarchies at a primitive level.

3. A Collaborative Recommendation Methodology

3.1 Overall Procedure

The entire procedure of our suggested CF-based recommendation procedure is composed of three sub-tasks; data representation, neighborhood formation, and recommendation generation as shown in [Figure 1].

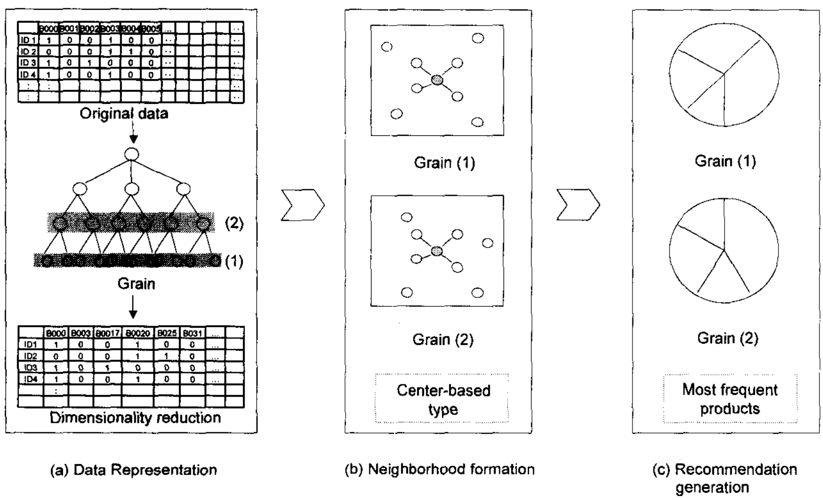
First, the representation task deals with customers' purchase data and a database with descriptions of products extracted from an off-line real store. Based on a general architecture of product hierarchy, four types of grain are suggested considering marketer's knowledge. Second, the neighborhood formation task is conducted using the reduced dimensional data. Third, the recommendation generation task focuses on the problem of finding the top-N recommendedproducts from the neighborhood of customers. Finally, the evaluation task of performance is made to determine more efficient grain. Detailed description of each process is given at the following subsections.

3.2 Data Representation

The input data is a collection of purchase transactions of n customers on m products in CF-based recommendations. This is represented as an $n \times m$ customer-product matrix, R, such that r_{ij} is one if the i^{th} customer has purchased the j^{th} product, or zero, otherwise. Let's term R as original data representation.

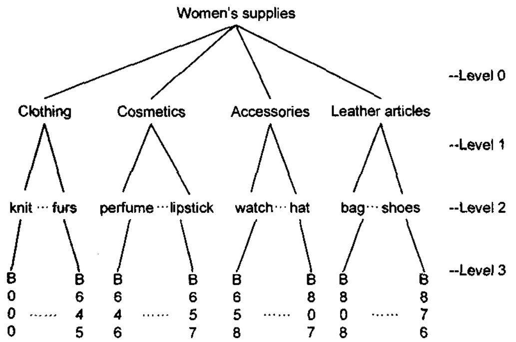
As discussed in many related work, the original data representation has some problems for nearest-neighbor recommendation procedure, such as sparsity and scalability. In this paper we introduce a method of dimensionality reduction based on product hierarchy. An example of product hierarchy is shown at [Figure 2].

A product hierarchy plays a fundamentally important role in the knowledge discovery process since it represents a particular store



[Figure 1] Overall procedure

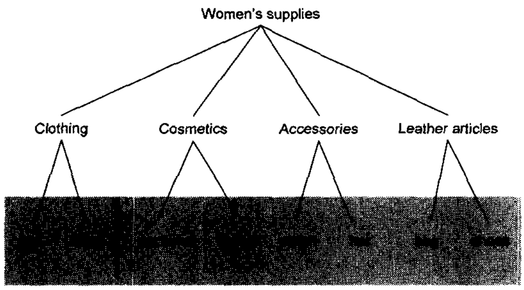
dependent knowledge. Further, it's very useful since choosing the right levels of the product hierarchy based on a marketer's knowledge leads to improve the quality of recommendations.



[Figure 2] A product hierarchy

Several terms related to a product hierarchy, such as a leaf node, non-leaf node, root node, parent, etc., are used under theiroriginal meanings (Adomavicius et al., 2001 ; Han et al., 1994 ; Lu et al., 1997 ; Sarwar et al., 2001b). In Figure 3, "Women's supplies" is a root node and "Clothing" and "Knit" are non-leaf nodes. A parent of "B645" which is a leaf nod is "furs". A level number can be assigned to each node in the product hierarchy. The level of a root node is zero, and the level of any other node is one plus the level of its parent. The product hierarchy of [Figure 2] has four levels, referred to as level 0, 1, 2, and 3.

We suggest a concept of "grain", which refers the bottom band of a newly adjusted product hierarchy. The concept of "cut"proposed by Adomavicius & Tuzhilin (2001) is a similar to our "grain" concept. They proposed it as a flexible way for a marketer to examine multiple rules at a time by grouping similar rules together. However, we generate a grain in a different way by grouping leaf-nodes into an integrated leaf node using a marketer's knowledge or experience. The grain generating process is performed at the original product hierarchy. In this paper, four types of grains are presented as follows; two types of flat-level grain, termed 'Flat 1' and 'Flat 2' respectively and two types of cross-level grain, termed 'Cross 3' and 'Cross 4'. The number of grains may be varied depending on the domain. It is discussed in detail below.



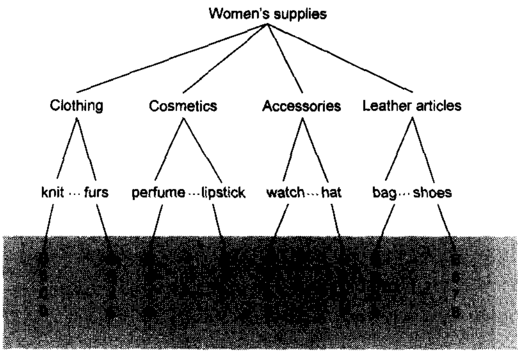
[Figure 3] 'Flat 1' grain

3.2.1 Grains at flat-level

In this case, a grain is generated on a particular level of product hierarchy and consists of nodes belonged to that level. Therefore, this grain is presented in the shape of a flat band.

Flat-level 1 : In this case we consider non-leaf nodes at the level 2 as leaf-nodes. The number of leaf-nodes in our product hierarchy is 24 which represent classes. Therefore, a newly generated customer-product matrix contains the number of products, 24, leading to the reduced dimensional space compared with the original matrix. [Figure 3] shows the generated Flat-level 1 grain. We term this 'Flat 1' in our experiments and it is marked with the shaded box in [Figure 3].

Flat-level 2 : In this case all the leaf-nodes of the original product hierarchy are considered as a grain since constructing a product hierarchy itself, leading to the dimensionality reduction. In our case, 876 products are contained. However, this matrix is used as an original matrix in the experiments only when we evaluate the performance of recommendations among grains. This grain is termed with 'Flat 2' in the experiments and marked with the shaded box in [Figure 4].



[Figure 4] 'Flat 2' grain

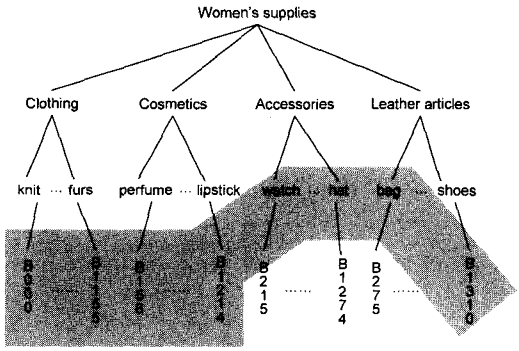
3.2.2 Grains at cross-level

Unlike grains at flat-level, this grain is formed on the two kinds of levels of product hierarchy. This grain is presented in the shape of cross-leveled band.

Cross-level 3 : Compared to grains at flat-level, the cross-leveled grains are constructed to represent a marketer's knowledge and experience more actively.

Cross-leveled grains are generated at the level 3, the leaf-nodes of product hierarchy and at the level 2, the class level of the leaf-nodes. This process is divided into two steps. First, amarketer chooses nodes at the level 3 of product hierarchy. So this grain is formed with nodes, products preferred by customers. And then to obtain more reduced customer-product matrix, a marketer adopts a not-leaf node with 3~5 leaf-nodes as the leaf-node of that grain, based on his/her knowledge or experience. With this process, the shape of a grain may be cross-leveled.

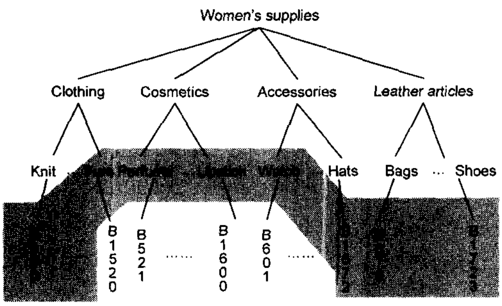
To construct a cross-leveled grain, a marketer analyzes the frequency that is the total number of products purchased by a customer in the database. After generating all grains, we evaluate the performance of each grain based on F1 measure, which is explained in detail at section 4.2 and choose the best grain. The best grain is compared with other grains. Please refer section 4.4.2 for more detailed explaination. We term the best grain selected in the experiment 'Cross 3'. Various shape of each grain can be generated like [Figure 5] in the product hierarchy.



[Figure 5] 'Cross 3' grain

Cross-level 4 : In this case, a grain is generated in a different way compared with the cross-level 3. Whereas the Cross 3 is made for the target of all customers, Cross-level 4 is made for a particular target group. To generate this grain we first select a particular target customer group and here it is assumed that this group is limited to the persons who purchased luxurious clothing. And then we analyze the propensity to costume and generate a grain fitted with this inclination. For this group, it can be defined as the higher class. Therefore, this grain consists of leaf-nodes or non-leaf nodes which represent very expensive articles such as luxurious watches, jewelry, furs and so on. This grain is also generated by using a marketer's knowledge or experience like Cross-level 3 and termed with 'Cross 4' in the experiments and marked with the shaded box in [Figure 6].

In this specification, the proper grain level depends on product, on its relative importance for making recommendations and on its frequency in the transaction data. Recent data mining research has shown that data mining algorithm usually produce the best results when product-related transactions are evenly occurred (Berry & Linoff, 1997 ; Han & Fu, 1994). In our research, the even transaction distribution among nodes in the grain is expected to prevent preference ratings from being dominated by the most frequent products. Hence, one of heuristic methods of grain generation is to reorganize or adjust the existing product hierarchy as a set of nodes with relatively even transaction distribution.



[Figure 6] 'Cross 4' grain

3.3 Neighborhood Formation

In the previous subsection, we have constructed adjusted matrices using each grain. The next step, the most important step in CF-based recommendations, is that of computing the similarity between customers as it is used to form a proximity-based neighborhood between a target customer and a number of like-minded customers. This process is learning process for a recommender system algorithm. The goal of neighborhood formation is to find, for each customer u, an ordered list of l customers $N = \{N_1, N_2, \cdots, N_l\}$ such that $u \notin N$ and $sim(u, N_1)$ is the highest, $sim(u, N_2)$ is the next highest and so on (Sarwar et al., 2000).

There are two different aspects of neighborhood formation, the proximity measure and neighborhood formation algorithm.

Proximity Measure

There are two types of proximity between two users a and b. One is the Pearson correlation and the other is Cosine measure (Sarwar et al., 2001b).

These two measures are well known in most collaborative filtering based recommender systems.

However, there is a problem in these measures. If the size of neighborhood is small and the neighbor of a target customer is matched perfectly or very similarly, there are few products to recommend. For example, let's suppose that a customer-product matrix in <Table 1> is as follows.

〈Table 1〉 customer-product matrix

CID \ Pro.	A	B	C	D
101	1	0	0	1
102	0	0	1	1
103	1	0	0	1
104	0	1	1	1

According to the correlation and cosine, a target customer, CID (customerID) 101, has a perfect neighbor, CID 103. However, if you intend to recommend any product that isn't purchased by a target customer, you can find nothing. Rather, CID 102 and 104 are true neighbors of a target customer since they have products that are not purchased by her/him.

So we introduce a new measure. In this measure proximity between a target customer, *a* and other customer, *b* $prox_{ab}$ is measured as follows.

$$prox_{ab} = \frac{P_a \cap P_b}{P_a} \times \frac{P_b - (P_a \cap P_b)}{T - (P_a \cap P_b)}$$

Here P denotes the number of products a customer purchased and T the number of total products.

In the experiments we compare the performance and computation time of the new measure $prox_{ab}$ and those of the Pearson correlation and Cosine.

Neighborhood Types

This task is to form the neighborhood after computing the proximity between customers. There are several types for neighborhood formation. Among them, we use the center-based neighborhood formation. The center-based type forms a neighborhood of size *k*, for a target customer *c*, by simply selecting the l nearest other customers (Kim & Cho, 2003 ; Sarwar et al., 2001b). Using several matrices generated by grains in Section 3.2, we compute similarity matrices and neighborhood.

3.4 Recommendation Generation

In this subchapter, we produce top-N recommendations for a target customer. This recommendation list is derived from the neighborhood of a target customer. There are several types of recommendation generation, such as the most frequently purchased product, the latest product recommendation, association rule-based recommendation, and so on. Among them, we use the most frequently purchased product. This assumes that the more popular product means the more buyable product. This is commonly used in collaborative filtering based recommendations.

In this step we have two sub-tasks; choosing top-L classes and producing top-N recommendation list. First, we look into the neighborhood N and then choose top-L classes that are most frequently purchased by neighborhood. Second, we check their purchase data and then perform a frequency count of the products in each class. After all neighbors are accounted for, this sorts the products according to their frequency count and simply returns the N most frequent products as recommendation that have not yet been purchased by the target customer (Sarwar, et al., 2000). Namely, this recommendation list consistsof most frequently purchased products in each top-L class. Each top-class returns

2~3 products as recommendation products. In this way this recommendation list can provide more various products to a target customer and can be another type of cross-selling.

4. Performance Evaluation

4.1 Data Set

We used data from the H department store. We restrict our experiments to the customers who purchased only women's supplies in May 2000 and April 2001 and only consider women's supplies to construct a product hierarchy.

This data set contains purchase information of 50,000 customers on 876 products. We randomly selected enough users to obtain 11,414 purchase-records from the database.

Purchase-record in this context is defined to be a triplet <customer, product, purchase amount>. We divided the purchase records into training set and a test set by using the same 60%/40% train/test ratio. This value indicates that we divide the 11,414 purchase data set into 6,848 training cases and 4,566 test cases. The training data was converted into a customer-product matrix R that had 1,833 rows (i.e., 1,833 customers) and 557 columns (i.e., 557 products that were purchased by at least one of the customers). This matrix is used for the grains 'Flat 1', 'Flat 2', and 'Cross 3'. However, 'Cross 4' using this matrix has a different matrix that had 616 rows and 503 columns because of its generation characteristics. In this experiment we use purchase amount rating. So each entry $r_{i,j}$ represented the rating of the ith customer on the jth product. We also take another factor into consideration, the sparsity level of this data set. For the data matrix R this is defined as 1 (nonzero entries/total entries). The sparsity level of this data set is 0.9888.

4.2 Evaluation metrics

Recommendation procedure research has used several types of measures for evaluating the quality of recommendation. Among several measures there are two metrics used in the information retrieval (IR) community, recall and precision to evaluate top-N recommendation (Kowalski, 1997). We start by dividing our data set into the training data set and the test data set. These algorithms work on the training set and generated a set of recommendations, the top-N set. Our goal is to look into the test set and match products with our top-N set. Products that appear in both sets are members of a special set called hit set.

These two metrics are computed as follows (Sarwar et al., 2000).

Recall : For recommendation experiments, we define recall as the ratio of hit set size to the test size, i.e., recall = size of hit set/size of test set which can be written as followes.

$$Recall = \frac{size\ of\ hit\ set}{size\ of\ test\ set} = \frac{test \cap top - N}{|test|}$$

Precision : In the context of the recommendation precision is defined as the ration of hit set size to the top-N set size, i.e., precision = size of hit set/size of top-N set which can be written as follows.

$$\Pr ecision = \frac{size\ of\ hit\ set}{size\ of\ top\ N\ set} = \frac{test \cap top - N}{|N|}$$

But these two measures are often conflicting in nature (Sarwar et al., 2000). For example, increasing the number of N tends to increase recall but decreases precision. Therefore we introduce the standard F1 metric (Yang et al., 1999) that gives equal weight to both of them. This formula is computed as the following.

$$F1 = \frac{2 * Recall * Precision}{Recall + Precision}$$

We compute F1 for each individual and calculate the average value to use our metric.

4.3 Experimental methodology

4.3.1 Experimental platform

For our experiments Window 98 based PC having Intel Pentium Ⅲ 450 was used. Its CPU is 450MHz and RAM 128MB. The program to perform our experiments was programmedby Visual Basic 6.0. Under these conditions all the experiments were performed.

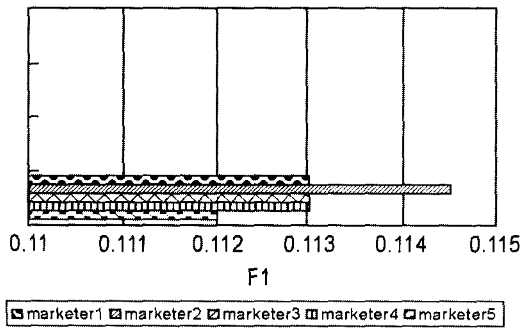
Data set was converted to be used in the program as follows. Four tables among many ones are presented; a 'transaction' table of all customers, 'taxonomy' table to construct a product hierarchy, a 'rating' table showing purchase amount, and 'a marketer's grain' table to construct 'Cross 3′, adjusted product hierarchy. Using these kinds of tables, we performed all the experiments.

4.3.2 Experimental steps

In this section several terms of experiments are presented before we evaluate the performance of all grains and recommendation quality. The ratio of training set is fixed at 60% and running counts of each experiment is 20 times. The evaluation metric, F1, is computed for each customer and then we use the average value as our metric. $prox_{ab}$ is used to form neighborhoods.

Before the main experiments are performed, two experiments are conducted as follows. We first evaluate the performance of each grain generated by the process of 'Cross 3' and fix the number of neighbors at 250 and choose the best grain among them.

Here the threshold value of each grain termed with 'marketer' is 20, 30, 50, 100 and 150 respectively for marketer 1, 2, 3, 4, and 5. After performing the generation process of 'Cross 3', the leaf-nodes of each grain were 44, 44, 43, 40, and 35 respectively. As we can see from [Figure 7], 'marketer4' having the threshold value of 100 produces the highest quality in recommendations. This indicates that the purchase pattern or preference is represented well in this grain. So we use this grain as 'Cross 3'. For the rest of the experiments 'marketer4' is used as 'Cross 3'.

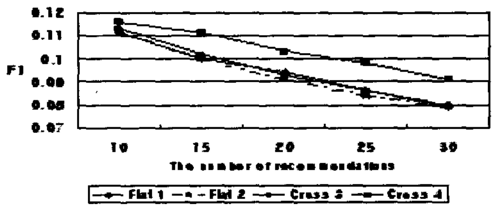


[Figure 7] Recommendation quality using 'Cross 3'

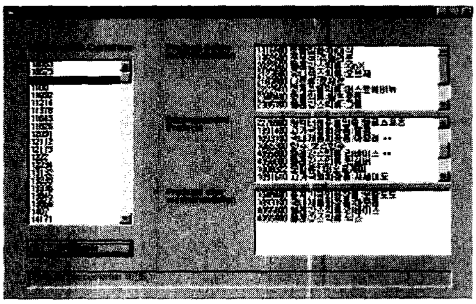
Second, we are interested in determining the number of recommendations which enhances the quality of recommendations. We fixed the size of neighborhood at 250. As we can see from [Figure 8](a), the peak of the number of recommendations in each grain is 10. Unlike our expectation, the quality of recommendations becomes worse after 10.

Here we can observe similar shapes of each grain from [Figure 8](a). However, the grain, 'Cross 4' representing a marker's knowledge and experience is better than others in F1 value. This means that the usage of a marketer's knowledge produces better performance in recommendations.

We can also know that what products a customer bought from the recommendation list in [Figure 8](b). After you select a certain customer and press the button of 'Analysis' in the program window, the results, such as the number of hit set and the detail of products. For example, choosing and analyzing a customer who has an ID 10726, we can see she/he bought 3 products from the recommendation list. These products are marked with "**".



(a)



(b)

[Figure 8] Results with the number of recommendations

Using the number of recommendations, 10, we perform all the experiments in the following.

4.4 Results

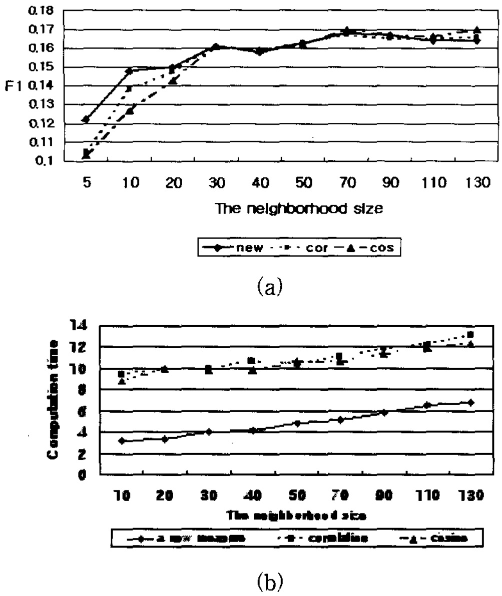
We present several experimental results using our adjusted product hierarchy to compare these performances with that of original product hierarchy at the level 3 termed with 'Flat 2'.

4.4.1 Results with the performance of each measure

So far the Pearson correlation and cosine measure are well known and preferred in the proximity measure of the neighborhood formation. These measures make recommendations at the level of products in the hierarchy but our algorithm at the level of classes of products. In addition, these measures cannot solve the problem of perfect or very similar matching between a target customer and another customer.

These characteristics required another measure for our experiments and therefore we introduced a new measure $prox_{ab}$ mentioned in Section 3. Here we look into the efficiency of $prox_{ab}$ among them. We experimented to investigate the performance of each measure and the computation time taken to find neighbors. We used a grain, 'Cross 3' and fixed the number of recommendations at 10. The results are below.

As our expectation, a new measure is better in the performance than other measures and reaches its peak earlier than others for the neighborhood size up to 30. After a value of 30, its performance is almost the same with



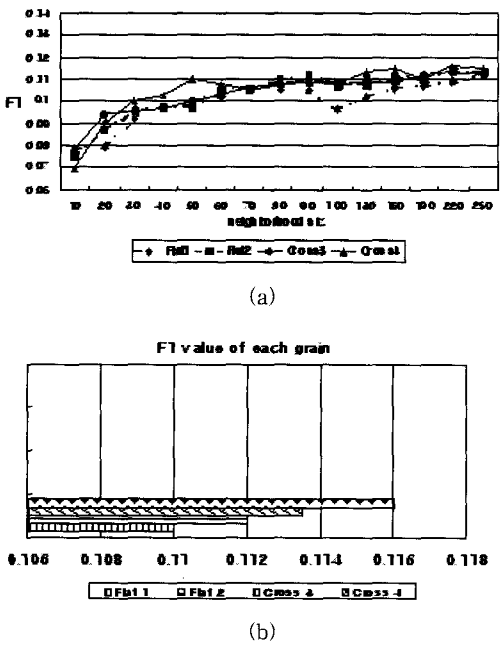
[Figure 9] Results with the performance of each measure

others. Namely, in this range of neighborhood size there isn't a great difference among them. In addition, considering very fast computation time, the performance of a new measure is much better. Let's see the computation time of the [Figure 9](b). The unit of computation time is a minute and the value of each measure in computation time indicates the average minute it takes to recommend products to all the customers. This shows the efficiency of our new measure and this efficiency is shown in all grains. In particular, it takes much less computation time than other measures in the entire range of neighborhoods. The computation time of this measure grows with an increasing number of neighborhood size like others but it is two or two and a half times faster than those of others. This also leads to inexpensive experiments and can deal with numerous data sets under the condition of this measure.

4.4.2 Results with the neighborhood size

As the size of neighborhood affects the performance of recommendations, we conducted an experiment with varying neighborhood size. These results are shown in [Figure 10].

[Figure 10](a) indicates that the size of neighborhood has impact on the recommendation quality. The F1 value continually increases as we increase the number of neighborhood. In particular, it rapidly increases in the range of 10~50. However, the increment in F1 value becomes dull and maintains a little constant value after a certain point. In general, each grain's peaks are reached in the range of 220~250. But each grain shows a little different aspect in reaching itspeak. Consequently this experiment indicates that the size of neighborhood affects the quality of recommendations.



[Figure 10] Results with the neighborhood size

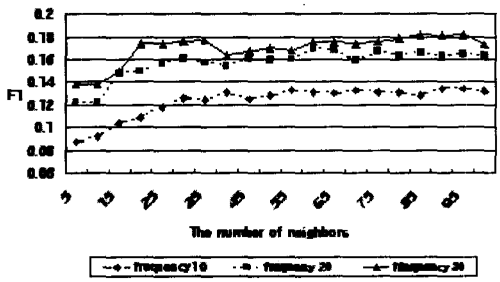
As we can see from [Figure 10](b), there is a difference among grains. Grains of Cross 3 and 4 have a higher value. In particular, the grain of 'Cross 4' is better than others in F1 value. This tells us that the usage of a marketer's knowledge can improve the quality of recommendations and represents customers' preferences well. It also leads to very reduced dimensional matrix so that it takes less time to perform this experiment. In this case we fixed the size of neighborhood at 220.

4.4.3 Results with the level of sparsity

As we discussed in chapter 1, one of the problems in collaborative filtering algorithm is sparsity. Sparsity can result in poor quality of recommendations and the high level of sparsity requires much computation time. To overcome this problem, we varied the minimum number of purchased products, that is, frequency. We fixed the number of frequency at 10, 20, and 30 respectively. Here the frequency 10 means that it considers persons who purchased above 10 products and this is the same with the frequency 20 and 30. In this experiment we used a grain, 'Cross 3' as this grain was higher in F1 value and required less time to compute F1 value than other grains. And we fixed the number of recommendations at 10 and used our new measure. We now present the results of these experiments as follows.

This graph indicates that F1 value increases as we increase the number of frequency and the graph of frequency 30 is better in the quality of recommendations than others. In this result, the level of sparsity at frequency 10, 20, and 30 is 0.9784(679 customers and 520 products), 0.9654(262 customers and 453 products), and 0.9505(111 customers and 379 products) respectively. Therefore, [Figure 11] shows that the lower the level of sparsity is, the higher the F1 value is.

Namely, in the lower level of sparsity a recommendation procedure can make better recommendations for a particular customer. Furthermore, it can save much computation time. For the future experiments using the matrix made at the frequency 20 is recommended, considering customers at the level of frequency 30 are too small, namely, 111 out of 1883 customers to form enough neighbors of a target customer. Therefore, the frequency at 30 cannot give a target customer a wide range of recommendations. Furthermore, in the aspect of scalability the computation time at frequency 20 is comparable to that of frequency 30.



[Figure 11] Results with the level of sparsity

4.4.4 Result summary and suggestions

We reached the following facts through the results of experiments. First, the more the number of recommendations is, the less the quality of recommendations. Second, a new measure $prox_{ab}$ is more efficient than the Pearson correlation and cosine for our experiments. Third, the neighborhood size affects the performance of recommendations. The F1 value continually increases as we increase the number of neighborhood. However, the increment F1 value becomes dull and maintains a little constant value after a certain point. Fourth, the level of sparsity is also a major factor affecting the quality of recommendations. In the lower level of sparsity a recommendation procedure can make better recommendations and save much computation time. Fifth, comparing the performance of grains with that of original product hierarchy, grains produce better performance in recommendations. In particular, 'Cross 4' has higher value than others in F1 value. This shows that a marketer's participation improves the quality of recommendations and requires less time to perform experiments.

Unlike our expectation, the usage of adjusted product hierarchy, namely cross-level grains cannot product much better performance in recommendations. There seems to be many factors in causing these results. Among them, the following factors may lead to a little unsatisfying result. First, only transaction data is included in the experiments. Transaction data plays the most important role in recommending certain products to someone but it can't represent all the characteristics of a customer.

Second, the total number of customers used in this experiment is not enough to gain more accurate information and the distribution of customers highly concentrates in the clothing field. This aspect may result in some poor quality of recommendations. Third, it is required to have more running counts than twenty times. The more running counts, the more accurate results.

Based on our experiments, CF algorithm using other dimensionality reduction methods is necessary to be developed, and will be interesting to be compared with our suggested algorithm. The experiments will be performed at diverse situations with numerous product domains, large customer set and long time range. Furthermore, considering counts of product purchase, Web data, and demographic information is also necessary to improve the performance of CF algorithm and to lessen the sparsity problem.

5. Conclusion

Recommender systems are a powerful new technology for extracting additionalvalue for a business from its customer database. These systems can help customers find products they want to buy from a store. One of the most successful recommender technologies is collaborative filtering. However, the limitations of collaborative filtering may lead to the poor quality of recommendations.

In this paper we identified the problems of collaborative filtering techniques and showed how these problems can be addressed through our methodology. Our results show that the usage of a grain hold the promise of allowing CF-based algorithms to scale to large data sets and at the same time produce high quality in recommendations. This paper stresses a marketer's participation in the recommendation procedure since she/he as an expert in the marketing field has specific domain knowledge or experience and the usage of a new measure $prox_{ab}$ in collaborative algorithms.

The performance of adjusted product hierarchy was better than the original hierarchy, especially in a marketer's grain, 'Cross 4'. And also the usage of a new measure $prox_{ab}$ leads to very fast computation time. This measure may process numerous data sets and save much computation time and cost in experiments.

As we can see the above, dimensionality reduction using adjusted product hierarchy is proven to be very effective technique, which leads to improvement of recommendation quality. In particular, a marketer's participation in this procedure affects the recommendation result and produces much better performance in recommendations. The suggested methodolgy is based on the product data which is represented by a hierarchy. However, all data is not represeted by a hierarchical structure. For example, if the dependancy of product data is so strong, our suggested methodology is not applicable without modification. Therefore, the methodology is effective on the domain that is suitable to be represeted by a hierarchical structure. To obtain more effective and efficient recommendation methodology, it is necessary to compare this methodology with others, such as content-based filtering and association rules and to conduct a real campaign to customers using it and evaluate the performance. We hope that our suggested methodology may be a useful application to many cases and our experimental results will help develop better ideas for more effective use of this information.

References

[1] Adomavicius, G. and Tuzhilin, A., Expert-driven validation of rule-based user models in personalization applications, *Data Mining and Knowledge Discovery*, Vol.5, No.1-2(2001), pp.33-58.

[2] Agarwal, R.C., Aggarwal, C. and Prasad, V., A Tree Projection Algorithm for Generation of Frequent Itemsets, *Journal of Parallel and Distributed Computing*, 2000.

[3] Berry, J.A. and Linoff, G., *Data mining techniques : for marketing, sales, and customer support*, New York : Wiley, 1997.

[4] Billsus, D., and Pazzani, M., Learning collaborative information filters, *In Proceedings of Recommender Systems Workshop. Tech. Report WS-98-08. AAAI Press*, 2000.

[5] Brew, C. Systemic classification and its efficiency, *Computational Linguistics*, Vol. 17, No.4(1991), pp. 375-408.

[6] Cho, Y.H., Kim, J.K., and Kim, S.H., A Personalized Recommender System based on Web Usage Mining and Decision Tree Induction, *Expert Systems with Applications*, Vol.23(2002), pp.329-342.

[7] Deerwester, S. Dumais, S.T., Furnas, G.W., Landauer, T.K., and Harshman, R., Indexing by Latent Semantic Analysis . *Journal of the American Society for Information Science*, Vol.41, No.6(1990), pp. 391-407.

[8] Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., Advances in Knowledge Discovery and Data Mining, *AAAI press/MIT press*, 1996.

[9] Goldberg, D., Nichols, D., Oki, B.M., and Terry, D., Using Collaborative Filtering to Weave an Information Tapestry, *Communications of the ACM*, December, 1992.

[10] Good, N., Schafer, B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., and Riedl, J., Combining Collaborative Filtering With Personal Agents for Better Recommendations, *In Proceedings of the AAAI-99 conference*, 1999, pp.439-446.

[11] Han, J., and Fu, Y., Dynamic Generation and Refinement of Concept Hierarchies for Knowledge Discovery in Databases, *AAAI 94 Workshop on Knowledge Discovery in Databases*, 1994, pp.157-168.

[12] Han, J., Cai, Y., and Cercone, N., Data-driven discovery of quantitative rules in relational databases, *IEEE Tran. on Knowledge and Data Engineering*, Vol.5, No.1 (1993), pp.29-40.

[13] Hill, W., Stead, L., Rosenstein, M., and Furnas, G.., Recommending and Evaluating Choices in a Virtual Community of Use, *In Proceedings of CHI 95*, 1995.

[14] Karypis, G., Evaluation of item-based top-n recommendation algorithms, *Communications of the ACM*, 43(2000), pp.121-132.

[15] Kim, J.K., Cho, Y.H., Kim, W.J., Kim, J.R., and Suh, J.H., A Personalized Recommendation Procedure for Internet Shopping Support, *Electronic Commerce Research and Applications,* Vol.1(2002), pp.301-313

[16] Kim, J.K. and Cho, Y.H., Using Web Usage Mining and SVD to Improve E- commerce Recommendation Quality, *Lecture Notes in Artificial Intelligence 2891*, 2003, pp.86-97.

[17] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L, and Riedl, J., GroupLens : Applying collaborative filtering to usenet news, *Communications of the ACM*, Vol. 40, No.3(1997), pp.77-87.

[18] Lu, Y., Concept hierarchy in data mining : specification, generation and implementation. Simon Fraster University, 1997.

[19] Mellish, C. The description identification problem, *Artificial Intelligence*, Vol.52, No. 2(1991), pp.151-167.

[20] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J., GroupLens : An Open Architecture for Collaborative Filtering of Netnews, *In Preceedings of CSCW 94, Chapel Hill, NC*, 1994.

[21] Resnick, P., and Varian, H.R., Recommender Systems, *Special issue of Communications of the ACM*, Vol.40, No.3(1997).

[22] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J., Analysis of recommendation algorithms for e-commerce, *In Proceeding of ACM E-Commerce 2000 Conference*, 2000.

[23] Sarwar, B., Karypis, G., Konstan, J. and Riedl, J., Item-based collaborative filtering recommendation algorithm, *In Proceeding of the Tenth International World Wide Web Conference*, 2001a.

[24] Sarwar, B., Karypis, G., Konstan, J. and Riedl, J., Applicatin of Dimensionality Reduction in Recommender System A Case Study, *In Proceeding of the Tenth International World Wide Web Conference*, 2001b.