Enhanced Recommendation Algorithm using Semantic Collaborative Filtering : E-commerce Portal

Shohel Ahmed
Computer Science and Information Engineering, Inha University
(shohel@inhaian.net)

Jongwoo Kim
Computer Science and Information Engineering, Inha University
(skytolove@inha.edu)

Sanggil Kang
Computer Science and Information Engineering, Inha University
(sgkang@inha.ac.kr)

This paper proposes a semantic recommendation technique for a personalized e-commerce portal. Semantic recommendation is achieved by utilizing the attributes of products. The semantic similarity of the products is merged with the rating information of the products to provide an accurate recommendation. The recommendation technique also analyzes various attitudes of the customer to evaluate the implicit rating of products. Attitudes are classifies into three types such as "purchasing product", "adding product to shopping cart", and "viewing the product information." We implicitly track customer attitude to estimate the rating of products for recommending products. Also we implement a session validation process to identify the valid sessions that are highly important for giving an accurate recommendation. Our recommendation technique shows a high degree of accuracy as we use age groupings of customers with similar preferences. The experimental section shows that our proposed recommendation method outperforms well known collaborative filtering methods not only for the existing customer, but also for the new user with no previous purchase record.

Received : June 27, 2011 Revision : July 11, 2011 Accepted : July 19, 2011
Type of Submission : APJITM Special Corresponding author : Sanggil Kang

1. Introduction

E-commerce portals have become one of the primary means used by customers for purchasing various kinds of products. Swift evolution of e-commerce has caused product overload and usually makes the customer's mind vulnerable to effectively choose his/her preferred product. In order to solve the problem, personalization of e-commerce portals have become a key

* This work was supported by an Inha University Research Grant.

factor to raise customer satisfaction and the chance of revisit. To achieve a high degree of customer satisfaction, a personalized e-commerce portal utilizes recommender systems (Resnick and Varian, 1997; Huang et al., 2007) to provide the customer with a personalized list of products that matches his/her preference pattern. Usually, recommender systems are designed to allow customers to determine their preferred products quickly and to avoid possible product overload.

A commonly approved technique to implement recommender system is collaborative filtering (CF) (Panagiotis et al., 2007; McLauglin and Herlocher, 2004; Herlocker et al., 2004). Various commercial and non-commercial recommender systems (e.g., www.amazon.com and www.movielens.org) have been utilizing CF to recommend personalized product lists to their customers. The goal of CF is to forecast the relevant products for a customer based on his/her preference and customers who have similar preferences. Usually in CF method, customers who share similar preference patterns to the target customer are grouped together and the preference of these like-minded customers is used to predict the recommendation. However, Scalability and Cold Starting (Liu and Deng, 2008) has become the two most remarkable problems of CF. Scalability highlights the vast amount of products and customer information that increases the computational complexity. Cold Starting problem highlights the incapability of giving a recommendation to a new customer. Along with these problems, the rating information of product supplied to the recommender systems is not sufficient in most cases. Most recommender systems collect product ratings by explicitly asking their customers. It is obvious that customers are less interested in spending their time rating products. Due to the lack of product rating information the performance of recommender system also degrades.

Our research focuses on addressing these problems. This paper presents the following contributions towards the improvements of the traditional CF based recommender systems:

- We introduce a hybrid recommender system which uses both user similarity and product similarity. Evaluating product similarity semantically is one issue that improved our recommendation accuracy.
- We present a new implicit rating method that can alleviate the problems with explicit rating. Also a new session validation algorithm is proposed, which can smartly filter out the noisy and incomplete session from Customer Browsing Log.
- We use the age of the customers to find customers of similar preference. This specific method reduces the overhead of processing huge amounts of customer data and eventually reduces the scalability problem.

As every user must belong to some age group, we provide a recommendation to a new customer by using his/her age group preference. The experimental section shows that this method of recommendation for a new customer provides a more improved result than other CF methods for the same issue.

In the experimental section we compare our method with well known CF based methods and hybrid CF method using both user-based and item-based CF methods.

The rest of this paper is organized as follows : Section 2 contains a brief overview of various related work on collaborative filtering methods. Then in the next section (Section 3), we introduce our system where we explain the session validation process for filtering the customer browsing log and the proposed recommendation methodology. The method of finding the semantic similarity of products is also discussed. Section 4 presents our experimental work. We describe our data collection and evaluation metrics and compare the performance of our method with other methods. The conclusion and some future direction of our research are included in Section 5.

2. Related work

This section covers various research literatures related to CF-based recommendation system. Goldberg et al. (1992) first introduces the term CF and Resnick et al. (1994) presented an automated CF technique using a neighbourhood-based algorithm. Since then, numerous research works that have published can be categorized as a user-based CF and an item-based CF and a user-based CF (Ziegler et al., 2005; Massa et al., 2006; Herlocker et al., 1999; Breese et al., 1998) which relies on user-user similarity. In a user-based CF, customers of similar preference are grouped together and their rating information is analyzed to predict a recommendation for the target customer. However, one obvious challenge of this technique is the scalability issue presented by Sarwar et al. (2000). In Sarwar et al. (2000), it is presented that computation complexity increases as the number of customer and item increases. Numerous optimization strategies such as similarity indexing Aggarwal et al. (1999) and Latent Semantic Indexing (LSI) Sarwar et al. (2000) have been proposed to alleviate the problem of user-based CF method. These strategies reduce the scalability problem but degrade the performance of recommender system in terms of accuracy.

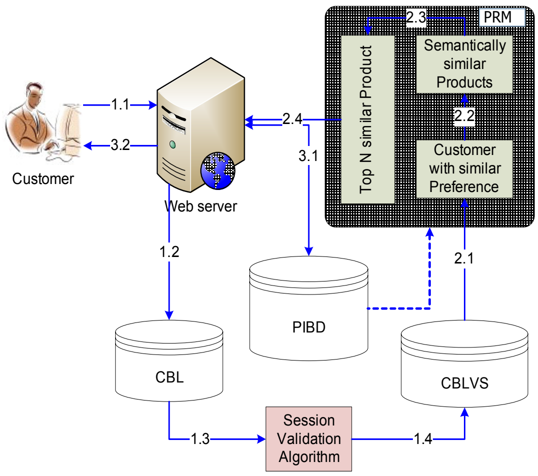
To deal with the scalability problem and to increase the recommendation accuracy, various item-based CFs, Sarwar et al. (2001), Karypis (2001), Deshpande and Karypis (2004), Hofmann (2004) have been introduced. In these literatures, CF is achieved by learning the similarity of the products rather than customers which eventually alleviates the scalability problem that exists in a user-based CF method. The underlying assumption in an item-based CF method is that, a customer who prefers one product has a high possibility to prefer another product from a similar product list. For an item-based CF, various machine learning techniques (Billsus and Pazzani, 1998; Li et al., 2007; Zhang and Iyengar, 2002; Heckerman et al., 2000) have been practiced. In Billsus and Pazzani (1998), a neural network classifier is presented to achieve CF (Li et al., 2007;

Zhang and Iyengar, 2002; Heckerman et al., 2000; Mobasher et al., 2004). introduces probabilistic method, linear classifier, dependency networks and latent semantic analysis on semantic attributes respectively. All these techniques are capable to achieve an improved recommendation accuracy that is comparable to or even better than a user-based CF method. However, item-based CF methods are suffering from the lack of ability to provide a recommendation or prediction for a new customer properly. Very few authors also reported the recommendation with a hybrid CF (Basu et al., 1998; Li et al., 2005). Literature (Basu, 1998) presents a hybrid recommendation using both content-based information and CF method that outperforms the user- based CF method. Meanwhile, a user-based CF and an item- based CF are combined to achieve higher recommendation accuracy in literature (Li et al., 2005). Both of the work faces a major difficulty in recommendation when the number of rated products is small.

In this paper, we propose a hybrid recommendation technique which is a kind of collaborative filtering based on the user and the semantic product similarity. Semantic product similarity is gained by using the attributes of the products. Our method shows considerable improvement over other CF method and provides one remarkable advantage. For a new customer, when no previous product rating information is available, the system can provide reasonable recommendation using the semantic similarity of the product.

3. System Architecture

<Figure 1> shows the overall architecture of our proposed system for a personalized e-commerce portal.



<Figure 1> Overall System Architecture

In the figure there are three parts in the system such as Customer Browsing Log (CBL), Product Recommendation Module (PRM), and Product Information Database (PIDB). When a customer requests any web page (1.1), the server starts to capture his/her navigation patterns and stores them to the CBL (1.2). This log information includes all the navigation patterns of each customer with their ID. For example, if a customer clicks on a product, a new record is created in the CBL with a new timestamp and corresponding product ID for the customer. <Table 1> shows the structure of CBL where C_ID and P_ID represents the Customer and Product ID respectively and CL_time, CL_page and CL_order stands for the click time, clicked page and click order respectively.

<Table 1> Structure of CBL

C_ID	*CL_time*	*CL_page*	*P_ID*	*CL_order*
201	5May08 : 11 : 09 : 31	home	-	1
201	5May08 : 11 : 11 : 01	home	-	2
201	5May08 : 11 : 12 : 51	LG laptop9	P310	3
201	5May08 : 11 : 17 : 30	LG laptop1	X110	4
201	5May08 : 11 : 22 : 54	LG laptop2	X120	5
201	5May08 : 11 : 29 : 11	LG battery	B134	6
201	5May08 : 11 : 33 : 18	LG battery	B127	7

A huge amount of clicking issued by the customers is captured by the server. However, not all the clicking of customers is a valid during each session. For example, a customer clicks to purchase a product and he/she leaves the portal without completing the purchasing process. This kind of information is noted as incomplete or noisy information in a session. In order to filter out the incomplete and noisy clicking information, we use our session validation process (1.3) to store only the valid sessions (1.4) in the customer browsing log with valid session (CBLVS). The successive discussion will give a detailed understanding of the session validation process.

3.1 Session Validation Process

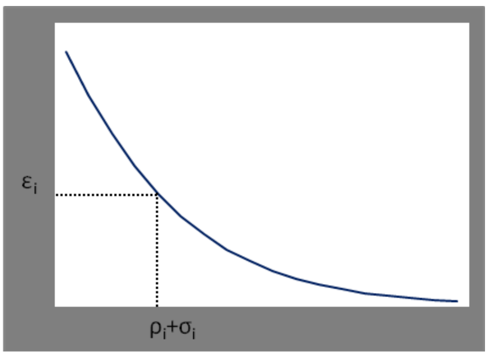
A session is a collection of activities performed by a user from the moment she enters a server site to the moment she leaves the site (W3C, 1999). Like any other portal, it is difficult to tell when a customer has left the e-commerce portal as there is no record of users leaving unless he/she logged out in a proper manner. Thus, we need to think about the situation where the difference between login and logout time is too big. However, a 30 minute session duration is proposed in Kang (2006) and used commonly in many applications. But, to estimate a valid session, we modify the standard reading time of each page which was developed in our previous work (Kang, 2006).

In the session validation process, we consider the real session form the CBL which is shown in <Table 1>. The k^{th} session of user j is defined as $S_k = \{<p_i, t_{i,j}>, <p_{i+1}, t_{i+1}, j>, \cdots, <p_n, t_{n,j}>\}$, where, p_i represents the i^{th} page and $t_{i,j}$ represents the arrival time of customer j on page i. The validation process justifies whether the session S_k is a valid one or not. To do this, we remodel the definition of standard reading time in Kang (2006), where it is calculated by converting the contents of that page to time. But our assumption is that, not all users are interested in the full contents of a page. For this, we consider the average reading time and the stand-

```
for each customer j;
for each session S_k;
set an empty session as S_k,valid
for each page I visited in session S_k;
if [τ_i /(μ_i + σ_i)] <= 1+ε_i
then  record all click information during τ_i to S_k,valid
increase i by 1
else
declare S_k,valid as invalid session
end loop;
increase k by 1;
end loop;
     Increase j by 1;
end loop;
```

<Figure 2> Session Validation Algorithm

ard deviation of reading time of a page to estimate the standard reading time. Average reading time, ρ_i is calculated by taking the average of the reading time of the customers who have already visited page *i*. The standard deviation of reading time of page *i*, σ_i, is added with the ρ_i to calculate the standard reading time of page *i*, where the assumption is that the reading time of page *i* by different customers is normally distributed. As a large number of different customers are visiting an e-commerce site every day, we will have a large number of reading time measurements for a page. It is known that under certain conditions, the distribution of a large number of small independent measurements is close to be normal (Sheskin, 2004). So, it is reasonable for us to assume that the distribution for reading time is normally distributed. Also, there may be some pages with a small amount of browsing information. For those pages we find the similarity between web pages by applying an iterative reinforcement algorithm presented in Shenghua et al. (2007) and calculate the standard reading time of those pages using a page



<Figure 3> Function Used for the Value of ϵ_i

similar to them. Beside this, our assumption is that, pages with a small amount of browsing information will have very low contribution in the recommendation process.

At this phase, we use a session validation algorithm to find the valid sessions from the set of all sessions S. <Figure 2> shows the session validation algorithm.

The session validation algorithm checks each session, denoted as *k*, for each customer, denoted as *j*, and find the validity of a session by using Eq. (1).

$$[\tau_i/\rho_i+\sigma_i)] \le 1+\epsilon_i \qquad (1)$$

In Eq. (1), τ_i is the time to move form page p_i to p_{i+1} which is determined using Eq. (2).

$$\tau_i = |(t_{t,j} - t_{i+1,j})| \qquad (2)$$

where $t_{i,j}$ and $t_{i+1,j}$ is the arrival time of customer *j* on page *i* and *i+1* respectively. The term ϵ_i in Eq. (1) is a tolerance factor for the session validity that gives some extra convenience to a session. The value of ϵ_i depends on the value of the standard reading time, which value is $(\rho_i+\sigma_i)$. Larger value of $(\rho_i+\sigma_i)$ in Eq. (1) tends to satisfy the condition for a session to be valid. In such case we use a smaller value for ϵ_i. And for smaller value of $(\rho_i+\sigma_i)$, using a larger value for ϵ_i gives much convenience to a session to be valid. The function used to evaluate the value of ϵ_i is shown in <Figure 3>, which is a natural logarithmic function. This function assures a higher value for ϵ_i when the value of is $(\rho_i + \sigma_i)$ lower, and vice versa.

3.2 Product Recommendation Module (PRM)

Generally, customers to an e-commerce portal often look over the site without purchasing anything. That is why a Product Recommendation Module (PRM) is used to help the customer to find a product they wish to purchase. As shown in <Figure 1>, the PRM starts its function by analyzing the data presented in CBLVS (2.1) and finds a group of customers using an age group similar to the target customer. Then the PRM produces a list of relevant products (2.2) based on semantic similarity of the target customer by searching the profiles of the customers with similar preference and recommends top N products for the customer (2.3). After getting the recommended product list from the PRM (2.4), the server collects the product information from PIDB (3.1)and supplies a web page with the recommended product information (3.2) to the target customer. PIDB is a relational database where product information is stored with a unique product ID for each product. The following sections will give detailed knowledge about how to find customers with similar preferences and generate similar product lists.

3.2.1 Customers of Similar Preference

The PRM first tries to find the customer with a similar preference pattern that matches with the target customer. For this the PRM checks the age of the target customer. Then it starts analyzing the CBLVS to generate a group of customers who are of the same age group of the target customer. We define five age groups (0～14, 15～24, 25～34, 35～44, and 45～) for this research work. We strongly believe that a similar age group has a similar kind of attitude to the products. Next, the PRM finds the customers whose preference pattern on the products is similar to the target customer by evaluating the group with a similar age. It is actually done by considering the implicit rating calculated from customers browsing attitude to the products.

The set of customers with same preference as the target customer is denoted as $U_{ck} = \{U_{1k}, U_{2k}, \cdots, U_{mk}\}$. The U_{ck} represents that customer c has the similar preference pattern as customer k. The preference similarity of customer c and k, denoted as $P(c, k)$, is measured in terms of cross correlation coefficient Aarts et al. (2002) as seen in Eq. (3).

$$P(c,k) = \frac{\sum_{i=1}^{n}(r_{ic}-\mu_c)(r_{ik}-\mu_k)}{\left(\sqrt{\sum_{i=1}^{n}(r_{ic}-\mu_c)^2}\right)\left(\sqrt{\sum_{i=1}^{n}(r_{ik}-\mu_k)^2}\right)} \quad (3)$$

where n is the number of product preferred by the customer c and k. The r_{ic} and r_{ik} is the rating of product i estimated from the attitude of customer c and k, respectively. Also, μ_c and μ_k is the average of r_{ic} and r_{ik}, respectively. To calculate the rating of a product i, we analyze each customer's browsing pattern and evaluate the rating of the product according to customer attitude. The customer attitudes are categorized into three types such as purchasing product (PP), adding products to the shopping cart (APSC), and viewing the product information (VPI). Each attitude has a different weight in the sense of significance of the attitude. That is, PP contributes a higher value than APSC and APSC contributes higher than VPI in the evaluation of product rating. This is because PP is the acute indication of customer interest on a product. The overall rating of product i by customer k is obtained by summing customer attitudes to the product i as seen in Eq. (4).

$$r_{ik} = \alpha_{ik} + \beta_{ik} + \gamma_{ik} \tag{4}$$

where α_{ik}, β_{ik}, and γ_{ik} is the weight of PP, APSC, and VPI of product i for customer k respectively. The value of α_{ik}, β_{ik}, and γ_{ik} is estimated from the distribution of customer attitude as seen in <Figure 4>.

<Figure 4> Distribution of Customer Attitude

The value at the top of each bar indicates the frequency of corresponding attitude for each customer. Now we estimate the value of α ik, β ik, and γ ik by taking the inverse of the frequency of corresponding attitude. And then Eq. (4) can be modified as seen in Eq. (5).

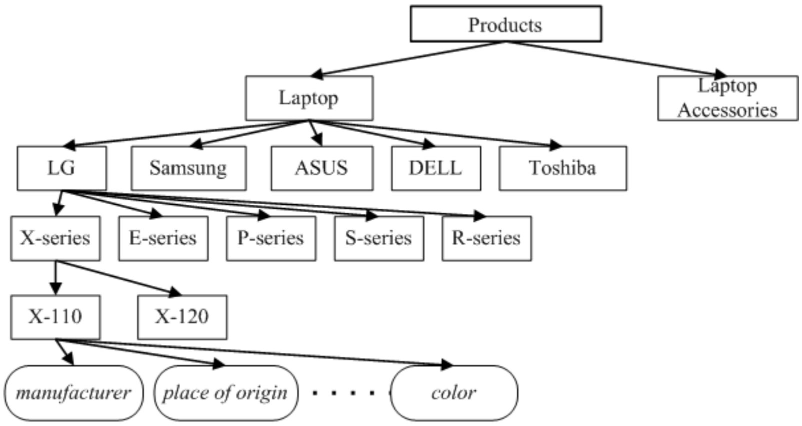
$$r_{i,k} = \frac{1}{f_{ik,Ap}} + \frac{1}{f_{ik,Aa}} + \frac{1}{f_{ik,Av}} \tag{5}$$

where $f_{ik,Ap}$ is the frequency of purchasing the product i by customer k and $1/\mathrm{f}_{\mathrm{ik,Ap}} = \alpha_{\mathrm{ik}}$. We believe that purchasing a product is a stronger indication of customer interest than adding a product to the shopping cart which is a stronger indication than viewing product information. Also, in general, the rate of a purchasing product is less than the adding a product to the shopping cart and viewing the information of a product is much higher than purchasing the product and adding the product to the shopping cart.

3.2.2 Semantic Similarity of Products

After finding the customers with similar preference by using the preference similarity value in the previous section, a list of products is generated that are rated by the customers having similar preferences. Then we find the semantic similarity among the products and finally we recommend top N similar product to the target customer.

In order to obtain the semantic similarity between products we consider the attributes describing a product. Each product in the Product Information Database (PIDB), shown in <Figure 1>, has a unique product ID and can be de-



<Figure 5> An Example of the Hierarchy of Products with Their Attributes

scribed with a set of attributes, denoted as A. Also, a product can have l number of attributes such as A_1, A_2, $\cdots$, A_l, and each attribute A_i can have m number of values such as a_{i1}, $a_{i2}, \cdots$, a_{im}. <Figure 5> shows an example of the hierarchy of products with their attributes. In <Figure 5>, *Laptop* and *Laptop Accessories* are the classes and all the descendent nodes are subclasses except the leaf nodes and the parents of leaf nodes. The leaf nodes are the attributes of products, parent nodes of the leaf nodes are the products. We consider all attributes that compel a customer to select a laptop. A set of attributes considered for this example is A = {*manufacturer, place of origin, price range, processor type, processor speed, cache, HDD capacity, RAM size, display size, battery backup, WLAN, weight, colour*}. Each attribute can have numerous values. For example, the *colour* attribute has the value <*black, white, red, silver, pink*>. The value of each attribute is numbered from 1 to q, where q is the number of value that the attribute may have. The *colour* attribute can be numbered as <*1, 2, 3, 4, 5*> where 1 represents *black*, 2 represents *white* and so on. And then each product P can be represented with an attribute-value vector that will contain 13 elements. For instance, a product *X110* has the attribute value like {*LG, Korea, $550-$600, Intel ®ATOMTM, 1.6 GHz, 512KB, 160 GB, 1024MB, 10 inch, 4hours, 802.11b/g, 1.9kg, white*}. It can be coded as {*2, 11, 3, 2, 6, 3, 3, 6, 4, 3, 2, 4, 2*}. In this fashion we can generate an attribute-value vector, denoted as V, for each product that is rated by the customer of similar preference.

To find the similarity between two products, we use their attribute-value vector such as V_i and V_j. In a real situation, V_i and V_j may represent the product from two different classes. Then the attribute-value vector will have all the attributes that belong to those classes. For example, V_i has the attribute set like $\{A_1, A_2, \cdots, A_l\}$ and V_j has the attribute set like $\{B_1, B_2, \cdots, B_r\}$. In such situation, we define a new attribute set for both V_i and V_j as $\{A_1, A_2, \cdots, A_l\} \cup \{B_1, B_2, \cdots, B_r\}$ Also the attribute that does not have any value for a particular class will set to be 0 in the attribute-value vector. The cardinality of the attribute-value vector will be $|V_i \cup V_j|$.

Now, we form a cluster of the attribute-value vectors for the products that are rated by customers of similar preference. Also we measure a mean attribute-value vector, denoted as V_μ, for the cluster. Then we estimate the distance between the attribute-value vector V_i and V_μ by using Mahalanobis distance measure McLachlan (1999) as shown in Eq. (6).

$$D(V_i, V_\mu) = \sqrt{(V_i - V_\mu)^T \sum^{-1} (V_i - V_\mu)} \qquad (6)$$

where $\sum$ is the covariance matrix and $D(V_i, V_\mu)$ is the Mahalanobis distance between V_i and V_μ. For our case, the covariance matrix is calculated using all the attribute-value vectors of the products that are rated by the customer of similar preference. Highly rated products are more likely to be purchased by the customer. We merge the rating of the products to Mahalanobis distance as the rating information has an extensive importance along with the semantic relation among the products. We named this new measure as Modified Mahalanobis Distance (MMD) measure. Eq. (7) illustrates the MMD of the attribute-value vector V_i and V_μ, denoted as $D_{\tilde{w}}(V_i, V_\mu)$.

$$D_{\tilde{w}}(V_i, V_\mu) = \tilde{w}_i \times D(V_i, V_\mu) \qquad (7)$$

where $\tilde{w}_i$ is the weight factor which is calculated using the rating of the product i. To evaluate the value of $\tilde{w}_i$ we calculate the average rating of product i as shown in Eq. (8).

$$\tilde{r}_i = \frac{1}{N_R} \sum_{k=1}^{N_R} r_{ik} \qquad (8)$$

where r_{ik} is the rating of product i by customer k, $\tilde{r}_i$ is the average rating of product i and N_R is the number of customer with similar preference who rated product i. The value of $\tilde{r}_i$ can be large if N_R is very small. For example, a product is rated by one customer with a high value such as 5, and another product is rated by 12 customers and average rating is 3.5. The latter product should have more influence in the weighting factor even though it has a lower average rating. For this we included the number of customers who rated the product when we evaluated the value of the weight factor $\tilde{w}_i$. Eq. (9) is used to evaluate the value of $\tilde{w}_i$.

$$\tilde{W}_i = \frac{N_R}{\mathcal{M}} \times \tilde{r}_i \qquad (9)$$

where $\mathcal{M}$ is the total number of customer with similar preference. By plug the value of $\tilde{w}_i$ in Eq. (9) to Eq. (7) we can measure the MMD of any attribute-value vector V_i with the V_μ.

A lower value for MMD indicates greater similarity between the attribute-value vectors. Based on these values the PRM generates the list of top N similar products for recommending them to the target user.

4. Experiment

4.1 Experimental Data

In order to investigate the effectiveness of our recommendation method, we compared our proposed recommendation method with well known CF methods, such as user-based Sarwar et al. (2000), item-based Sarwar et al. (2001) and hybrid recommendation (Li et al., 2005) method. For this, we developed an e-commerce portal for selling laptop and laptop accessories. This portal includes 1,805 laptop and laptop accessories for sale. We collected the customer's browsing information from April 2008 to December 2008. We used the same portal interface using a different CF method to collect data. For recommendation, each method was used for duration of two months. During this period, we found 285 common customers who logged on our portal and used all the CF based recommendations along with our method. Most of our customers were the students and faculty members of INHA University in Korea. We also found 4,878 sessions and among them 3,672 sessions were declared as valid using the session validation process described in Section 3.1. For our methods, we calculated the rating of the products implicitly by analyzing the valid sessions. For other CF based method we collected the explicit rating of a product using our portal to compare our method with other methods. The complete information of the dataset used for our experiment is shown in <Table 2>.

All experiments were performed using computers with windows operating system, Intel® Core2 Duo processor running at 2.66GHz and 2GB of memory. The communication between the server and the customers was implemented by using the method proposed in Wang et al. (2006). Customers and the server communicate with each other through HTTP by exchanging HTML files. The database connectivity with the server is maintained using Open Database Connectivity (ODBC) which provides a common set of Application Programmer's Interfaces (API) to communicate with the database using SQL.

<Table 2> Information of Experimental Dataset

		Value
Number of common customer		285
Age group	0～14	17
	15～24	114
	25～34	88
	35～44	39
	45～	27
Total number of products		1,805
Number of product rated explicitly		1,175
Number of product rated implicitly		851
Number of valid sessions		3,672

4.2 Evaluation Metric

To evaluate the top-N recommendation, we chose the widely used metric in information retrieval system (IR), known as precision and recall (Ziegler 2005; Yang and Liu, 1999; Kowalski, 1997).

Usually this metric is used to judge the recommendation quality of the recommender system in e-commerce. Eq. (10) and Eq. (11) shows the definition of the precision and recall metric respectively.

$$precision = \frac{numberofcorrectrecommendation}{numberofallsimilarproduct} \tag{10}$$

$$recall = \frac{numberofcorrectrecommendation}{numberofallsimilarproduct} \tag{11}$$

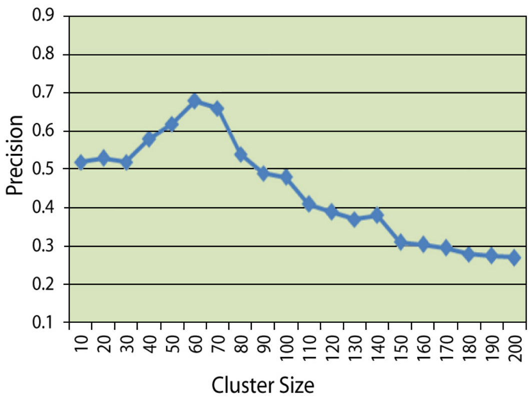
A higher value of precision metric indicates that most of the recommendation is correct and a higher value of *recall* indicated that most of the similar products were recommended. However, *precision* and *recall* metric is inversely related (i.e. increase of *recall* value decreases the value of *precision* and vice versa). To overcome this short coming of *precision* and *recall* metric, *F1 measure* is used. Eq. (12) is the definition of *F1*measure which is the harmonic average of *precision* and *recall*.

$$F1 = \frac{2\times(precision\times recall)}{(precision\times recall)} \tag{12}$$

4.3 Experimental Result and Discussion

4.3.1 Cluster Size Selection

The recommendation accuracy of our method highly depends on the cluster size of the attribute-value vector stated in Section 3.2.2. To compare our method with other methods, we first chose the cluster size for which our method can recommend more accurately. <Figure 6> shows the precision of our recommendation method for different cluster sizes. A cluster size of 60 is used for all other experiments as we get the highest precision of our method for this cluster size.



<Figure 6> Cluster Size Selection

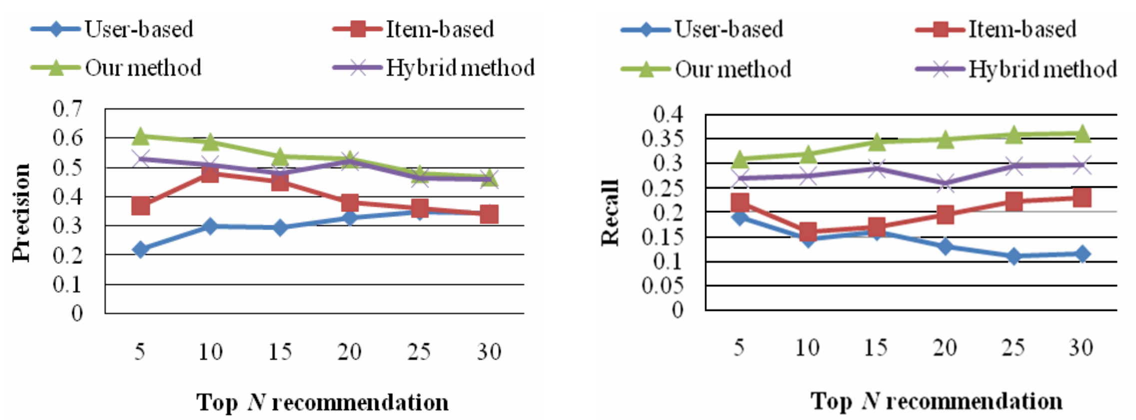
4.3.2 Comparison with Other Methods

<Figure 7> shows the comparison of well-known CF methods with our method for top N recommendation in terms of precision and recall. As seen in <Figure 7>(a) <Figure 7>(b), our method outperforms all the existing CF based methods. For example, at N = 5, user-based CF method had a precision of 0.22, item-based CF method had a precision of 0.48, hybrid method had a precision of 0.53 and our method has a precision of 0.61.

The poor performance of the user-based CF is due to the lack of sufficient rating information on the items recommended. In other words, the target customer has very few similar customers who have rated a product. The item-based CF shows a slightly improved result than the used-based CF method because the recommendation is made based on the items that are similar to other items the customer has liked. Also, it is possible to retain only a small subset of items to provide reasonably good prediction quality and similarities between items that are relatively static. As the size of the recommended item increases, the performance degraded. This is because of the involvement of less similar items in the recommendation process. The hybrid method we chose for comparison assures better performance than a user-based and item-based CF method as the method uses a user-item matrix to determine the similar user group for a similar set of items.

The use of semantic similarity of product leads our method to achieve an improved result when compared to other methods. When the number of recommended products is low our method performs the best (i.e. the highest value for precision is 0.61 when N = 5). The attribute-value vector of products for the customers of similar preference is clustered based on the product rating. Thus highly similar products are recommended to the customer. But when the number of recommended products increases, we found that the inclusion of less similar product to the recommendation list degrades the performance slightly. In particular, for our method, *precision* is 22.9%, *recall* is 19.9% improved than user-based CF method. Also, for our method, *precision* is 13.9%, *recall* is 14.1% improved than item-based CF method and precision is 4.2%, and recall is 5.9% improved than the hybrid method.

<Table 3> shows the *F1measure* of different methods for different top N recommendations. In general, *F1 measure* depends on the value of corresponding *precision* and *recall* values of top N recommendations. The higher values of our method for *F1 measure* implies that



(a)Precision (b)Recall

<Figure 7> Performance comparison of different methods with our method based on (a) Precision, (b) Recall

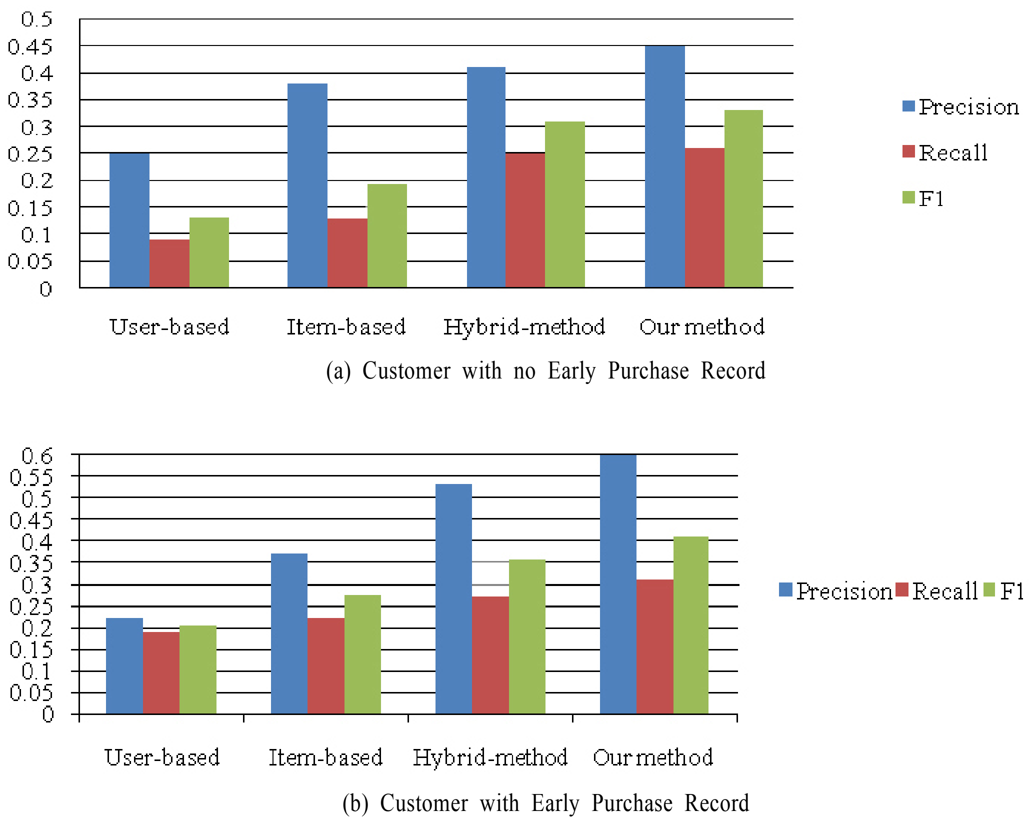
the proposed method can provide better recommendation than the other methods.

<Table 3> F1 measure for Top N Recommendation

Method	*Top 5*	*Top10*	*Top15*	*Top20*	*Top25*	*Top30*
User-based	0.203	0.195	0.207	0.186	0.167	0.172
Item-based	0.275	0.240	0.246	0.257	0.274	0.274
Hybrid method	0.357	0.357	0.361	0.346	0.360	0.361
Our method	0.411	0.414	0.421	0.421	0.411	0.408

4.3.3 New Customer vs. Earlier Customer

<Figure 8>(a) <Figure 8>(b) presents the performance comparison of various methods with our method for customers with no early purchase record (i.e. new customers) and customers with an early purchase record. This experiment was carried out over 10 randomly chosen customers who had no early purchase record and 10 randomly chosen customers who had an early purchase record. It was observed that our recommendation method performed better than the other CF based recommendations to recommend products for a new customer. All other methods use the rating information to recommend products for a new customer. In general, new cus-



<Figure 8> Performance Comparison of Different CF Methods with Our Method for (a) Customer with no Purchase Record (b) Customer with Previous Purchase Record

tomers do not have any previous rating information or have little rating information. In such cases, the accuracy of recommendation dramatically decreases for new customer. We use the age information of the new customer and provide the recommendation to the customer based on the semantic product similarity of that age group.

In particular, we observed that our method was 20% better than user-based, 7% better than item-based method and 4% better than the hybrid method in terms of precision. Similarly we obtained improved results for the customers with an early purchase record (e.g. our method shows 39% better performance than user-based, 24% better performance than item-based and 8% better performance than hybrid method in terms of precision).

4.3.4 Recommendation Accuracy of Age Groups

The results displayed in <Table 4> exhibits the recommendation accuracy of our method for different age groups based on precision, recall, and F1 measure. The results are for top N recommendation. For this experiment we chose N = 5 as we found that our method performed best when the value of N is 5.

As seen in <Table 4>, the higher recommendation accuracy belongs to the age group 15~24 and lower recommendation accuracy belongs to the age group 0~14. We found that the recommendation accuracy for any age group is influenced by the number of customers of that age group. It is observed that age group with more customers provides a more implicit rating to the product which plays an important role while calculating the semantic similarity of the products. Also little rating information has been observed for an age group with fewer customers and eventually the recommendation accuracy decreases for those groups. For instance, the age group 15~24 with 114 customers has a precision of 0.795 whereas the age group 0~14 with 17 customers has a precision of 0.42. This indicates that our method can provide more accurate recommendation for the age group with large number of customers.

<Table 4> Recommendation Accuracy for Age Groups

		Precision	Recall	F1 measure
Age group	0～14	0.420	0.350	0.381
	15～24	0.795	0.285	0.419
	25～34	0.682	0.295	0.411
	35～44	0.605	0.32	0.418
	45～	0.550	0.34	0.420

5. Conclusion

From the experimental results, we can conclude that the semantic recommendation technique for the personalization e-commerce portal is able to produce successful product recommendations. The implicit rating of products using the attitudes of the customer minimizes the existing problems with explicit product rating. Also we showed that our method is well capable of handing new customers with better accuracy than other methods. For the existing customer who has an early purchase record, our method outperforms traditional methods of recommendation.

However, the experimental results using the data collected from our portal may not be perfect for a large commercial e-commerce portal as most of our customers were from the age group 15~24 and 25~34. For practical purposes, we need to perform experiments using the data of some large commercial e-commerce portal as future work. Also, we need to evaluate the computational performance of different recommender systems based on time and memory limitation.

References

Aarts, R. M., R. Irwan, and A. J. E. M. Janssen, "Efficient tracking of the cross-correlation coefficient", *IEEE transaction on Speech and Audio Processing*, Vol.10, No.6(2002), 391~402.

Aggarwal, C. C., J. L. Wolf, and P. S. Yu, "A New Method for Similarity Indexing for Market Basket Data", *In Proceedings of the ACM SIGMOD*, Philadelphia, PA, USA, (1999), 407~418.

Basu, C., H. Hirsh, and W. Cohen, "Recommendation as Classification : Using Social and Content-based Information in Recommendation", In Proceedings of the 15th National Conference on Artificial Intelligence (AAAI), Madison, WI, 1998.

Billsus, D. and M. J. Pazzani, "Learning Collaborative Information Filters", In Proceeding of 15th Int'l Conference of Machine Learning, Madison, USA, (1998), 46~54.

Breese, J. S., D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative Filtering", In Proceedings of the 14th Conference of Uncertainty in Artificial Intelligence, Madison, USA, (1998), 43~52.

Deshpande, M. and G. Karypis, "Item-based top-n recommendation algorithms", ACM Transactions on Information Systems, Vol.22, No.1(2004), 143~177.

Goldberg, D., D. Nichols, B. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry", *Communications of the ACM*, Vol. 35, No.12(1992), 61~70.

Heckerman, D., D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie, "Dependency Networks for Inference, Collaborative Filtering, and Data Visualization", *Journal of Machine Learning Research*, Vol.1(2000), 49~75.

Herlocker, J. L., J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering", *In Proceedings of the 22nd Annual International ACM SIGIR, Berkeley*, USA, (1999), 230~237.

Herlocker, J. L., J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative Filtering Recommender Systems", *ACM Transaction on Information Systems*, Vol.22, No.1 (2004), 5~53.

Hofmann, T., "Latent Semantic Models for Collaborative Filtering", ACM Transactions on Information Systems, Vol.22, No.1(2004), 89 ~115.

Huang, Z., D. Zeng, and H. Chen, "A comparative study of recommendation algorithms for e-commerce applications", IEEE Intelligent Systems, forthcoming, 2007.

Kang, S., W. Park, and Y. Kim, "Dynamical

E-Commerce System for Shopping Mall Site Through Mobile Devices", In Proceedings of 2nd int'l workshop on Data Engineering Issues in E-Commerce and Services (DEECS), San Francisco, USA, (2006), 268~277.

Karypis, G., "Evaluation of item-based top-N Recommendation Algorithms", In Proceedings of the 10th ACM Int'l Conference on Information and Knowledge Management (CIKM), Atlanta, GA, (2001), 247~254.

Kowalski, G., "Information Retrieval Systems : Theory and implementation", Kluwer Academic publisher, 1997.

Li, M., B. Dias, W. El-Deredy, and P. J. G. Lisboa, "A probabilistic Model for item-based recommender systems", *In Proceedings of the Conference of Recommender Systems, Minneapolis*, USA, (2007), 129~132.

Li, Y., L. Lu, and L. Xuefeng, "A Hybrid Collaborative Filtering Method for Multiple-interests and Multiple-content Recommendation in E-Commerce", *Expert Systems with Applications*, Vol.28, (2005), 67~77.

Liu, J. and G. Deng, "A New-User Cold-Starting Recommendation Algorithm Based on Normalization of Preference", In Proceedings of 4th Int'l conference on Wireless Communications, Networking and Mobile Computing (WiCOM), Dalian, China, (2008), 1~4.

Massa, P. and B. Bhattacharjee, "Using Trust in Recommender Systems : An Experimental Analysis", Lecture Notes in Computer Science, Springer-Verlag, Vol.2995(2006), 221~235.

McLachlan, G. J., "Mahalanobis distance", Resonance, Springer India, Vol.4, No.6(1999), 20~26.

McLauglin R. and J. Herlocher, "A collaborative filtering algorithm and evaluation metric that accurately model the user experience", In Proceedings of the 27th annual international ACM SIGIR, Sheffield, UK, (2004), 329~336.

Mobasher, B., X. Jin, and Y. Zhou, "Semantically Enhanced Collaborative Filtering on the Web", Web Mining : From Web to Semantic Web, *Lecture Notes in Artificial Intelligence*, Vol.3209, Springer, Berlin, Germany, 2004.

Panagiotis, S., N. Alexandros, P. Apostolos, and M. Yannis, "Collaborative Filtering Based on User Trends", *Advances in Data Analysis, Springer Berlin Heidelberg*, (2007), 375~382.

Resnick, P. and H. Varian, "Recommender Systems", *Communications of the ACM*, Vol.40, No.3(1997), 56~58.

Resnick, P., N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl, "GroupLens : An Open Architecture for Collaborative Filtering of Netnews", In Proceedings of the Conference on Computer Supported Cooperative Work, Chapel Hill, USA, (1994), 175~186.

Sarwar, B. M., G. Karypis, J. A. Konstan, and J. T. Riedl, "Analysis of recommendation algorithms for E-commerce", In Proceedings of the 2nd ACM Conference on Electronic Commerce, New York, USA, (2000), 158~167.

Sarwar, B., G. Karypis, J. Konstan, and J. Reidl, "Item-based Collaborative Filtering Recommendation Algorithms", In Proceedings of the 10th Int'l Conference on World Wide Web (WWW). Hong Kong, (2001), 285~295.

Sarwar, B., G. Karypis, J. Konstan, and J. Riedl, "Application of Dimensionality Reduction in Recommender Systems-A Case Study", In Proceedings of the WebKDD 2000 Workshop at the ACM-SIGKDD Conference on Knowledge Discovery in Databases (KDD), Boston, MA, 2000.

Shenghua, B., X. Guirong, W. Xiaoyuan, Y. Yong, F. Ben, and S. Zhong, "Optimizing Web Search using Social Annotations", In Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada, (2007), 5012~1510.

Sheskin, D., "Handbook of Parametric and Non-parametric Statistical Procedures", Chapman and Hall/CRC, Boca Raton, Florida, 2004.

W3C. World Wide Web committee web usage characterization activity. W3C working Draft : Web Characterization Terminology and Definitions Sheet, Pages www.w3.org/1999/05/WCA-terms/.

Wang, G. H., W. S. Wang, T. B. Yu, and Y Gu, "Using Communication Technology for Automation Negotiation in E-commerce Environment", In Proceedings of 2nd Int'l conference on Wireless Communications, Networking and Mobile Computing(WiCOM), Wuhan, China, (2006), 35172~13520.

Yang, Y. and X. Liu, "A re-examination of text categorization methods", In Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval(SIGIR), Berkeley, CA, (1999), 42~49.

Zhang, T. and V. S. Iyengar, "Recommender Systems Using Linear Classifiers", *Journal of Machine Learning Research*, Vol.2(2002), 313~334.

Ziegler, C. N., S. M. Mcnee, J. A. Konstan, and G. Lausen, "Improving Recommendation Lists Through Topic Diversification", In Proceedings of 14th Int'l Conference on World Wide Web(WWW), Chiba, Japan, (2005), 22~32.

Abstract

전자상거래 포탈을 위한 시맨틱 협업 필터링을 이용한 확장된 추천 알고리즘

Shohel Ahmed* · 김종우* · 강상길*

우리는 개인 전자상거래 포탈에서 개인화를 위한 시맨틱 추천 방법을 제안한다. 시맨틱 추천은 제품의 특성(속성)을 이용하여 의미적 유사성 평가를 통해 이루어진다. 정확한 추천을 제공하기 위하여 제품의 시맨틱 유사성은 제품의 평점정보를 포함한다. 또한, 추천기술은 제품의 평점을 평가하여 고객의 다양한 내포된 의향을 분석한다. 고객의 의향은 "구입한 제품", "쇼핑카트에 추가한 제품", "정보를 본 제품"과 같이 세 가지 유형으로 분류 하고 있다. 우리는 제품의 추천을 위한 제품의 평점을 추정하기 위하여 고객의 내재적 의향을 추적할 수 있다. 또한 우리는 정확한 추천을 제공하기 위해 매우 중요한 유효한 세션을 식별하는 유효성 검사 프로세스 세션을 구현하였다. 우리의 추천기술은 유사한 환경의 고객의 연령별 그룹에서 높은 수준을 정확도를 보여 준다. 본 논문의 실험섹션에서 우리의 제안 추천방식은 기존 고객뿐만 아니라 이전의 구매기록이 없는 새로운 사용자에게도 기존에 잘 알려진 협업 필터링 방법보다 좋은 성능을 보여 주었다.

Keywords : 협업필터링, 시맨틱유사성, 추천방법, 전자상거래, 개인화

* 인하대학교

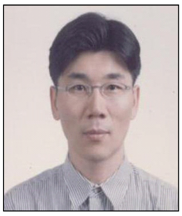
저 자 소 개

Shohel Ahmed

He received the M.S. in Computer Science and Information Engineering from INHA University, Korea in 2010. He is currently a lead engineer in Samsung Bangladesh Research and Development Center (SBRC). His research interests include Query Classification, Web Crawling, User Goal Identification, etc.

Jongwoo Kim

He received the M.S. in Computer Science and Information Engineering from INHA University, Korea in 2010. He is currently a Ph.D student in Computer Science and Information Engineering from INHA University. His research interests include Semantic Web, Artificial Intelligence, Mobile Systems, Inference Systems, Ontology Modeling, etc



Sanggil Kang

He received the M.S. and Ph.D. degrees in Electrical Engineering from Columbia University and Syracuse University, USA in 1995 and 2002, respectively. He is currently an associate professor in the Department of Computer Science and Information Engineering at INHA University, Korea. His research interests include Semantic Web, Artificial Intelligence, Multimedia Systems, Inference Systems, etc.