한국지능정보시스템학회
02 춘계학술대회 논문집, pp. 342~351

A personalized recommendation methodology using web usage mining and decision tree induction

웹 마이닝과 의사결정나무 기법을 활용한 개인별 상품추천 방법

조윤호 [a], 김재경 [b]

[a] *Graduate School of Management, Korea Advanced Institute of Science and Technology*
e-mail: yhcho@dongyang.ac.kr

[b] *School of Business Administration, Kyung Hee University*
e-mail: jaek@khu.ac.kr

Abstract

A personalized product recommendation is an enabling mechanism to overcome information overload occurred when shopping in an Internet marketplace. Collaborative filtering has been known to be one of the most successful recommendation methods, but its application to e-commerce has exposed well-known limitations such as sparsity and scalability, which would lead to poor recommendations. This paper suggests a personalized recommendation methodology by which we are able to get further effectiveness and quality of recommendations when applied to an Internet shopping mall. The suggested methodology is based on a variety of data mining techniques such as web usage mining, decision tree induction, association rule mining and the product taxonomy. For the evaluation of the methodology, we implement a recommender system using intelligent agent and data warehousing technologies.

Keywords: Product recommendation, Personalization Web usage mining, Decision tree induction, Internet shopping mall

1. Introduction

E-commerce has been growing rapidly keeping the pace with the web. Its rapid growth has made both companies and customers face a new situation. As a result, the need for new marketing strategies such as one-to-one marketing and customer relationship management (CRM) has been stressed both from researches as well as from practical affairs (Berson, Smith & Thearing, 2000; Yuan & Chang, 2001). One solution to realize these strategies is personalized recommendation that helps customers find the products they would like to purchase by producing a list of recommended products for each given customer.

Collaborative filtering has been known to be the most successful recommendation technique that has been used in a number of different applications such as recommending web pages, movies, articles and products (Hill et al., 1995). However, despite their success, their widespread use has exposed two major limitations (Claypool et al., 1999; Sarwar et al., 2000). The first is related to sparsity. The number of ratings already obtained is very small compared to the number of ratings that need to be predicted because typical collaborative filtering requires explicit non-binary user ratings for similar products. The second is related to scalability. Algorithms to find the neighborhood usually require very long computation time that grows linearly with both the number of customers and the number of products.

E-commerce data is rich and detailed compared to off-line commerce data. Through analyzing such information (i.e. web usage mining), it is possible to make a more accurate analysis of customer's interest or preference across all products than analyzing the purchase records only. Furthermore, mining association rules from clickstream provides rich and interesting relationships or associations among products.

In this paper, we propose a personalized recommendation methodology based on web usage mining. Furthermore, decision tree induction is used to minimize recommendation errors by making recommendation only for customers who are likely to buy recommended products. For the implementation of the proposed methodology, a recommender system is also developed using intelligent agent and data warehousing technology.

We begin by reviewing previous works related to our research in section 2. In section 3, the suggested recommendation methodology is explained with an illustrative example case. An agent based recommender system implemented for the evaluation is presented in section 4. Finally, we summarize our contributions with suggestions for future research in section 5.

2. Backgrounds

2.1 Web usage mining

Web usage mining is the process of applying data mining techniques to the discovery of behavior patterns based on web data, for various applications. The overall process of web usage mining is generally divided into two main tasks; data preparation and pattern discovery. Cooley, Mobasher and Srivastava (1999) presented a detailed description of data preparation methods for mining web browsing patterns. The pattern discovery tasks involve the discovery of association rules, sequential patterns, usage clusters, page clusters, user classifications or any other pattern discovery method (Mobasher, Cooley & Srivastava, 2000).

There have been several customer behavior models for e-commerce, which have different analysis purposes. But a part of Lee et al's model is adopted to our research, because they focus the online retailer which is our consideration as well. Micro-conversion rates (e.g., click-to-buy rate) used for measuring the effective of efforts in merchandising are computed for each adjacent pair of these steps. The study shows how the breakdown of clickstreams into subsegments can highlight potential problems in merchandising.

2.2 Association rule mining and product taxonomy

Given a set of transactions where each transaction is a set of items (itemset), an *association rule* implies the form $X \Rightarrow Y$, where X and Y are itemsets; X and Y are called the *body* and the *head*, respectively. The *support* for the association rule $X \Rightarrow Y$ is the percentage of transactions that contain both itemset X and Y among all transactions. The *confidence* for the rule $X \Rightarrow Y$ is the percentage of transactions that contain itemset Y among transaction that contain itemset X. The support represents the usefulness of the discovered rule and the confidence represents certainty of the rule.

In most Internet shopping malls, the product taxonomy is available. A product taxonomy T is practically represented as a tree that classifies a set of low-level products into a higher-level of a more general product. The leaves of the tree denote the *product instances*, SKUs (Stock Keeping Units) in retail jargon, and non-leaf nodes denote *product classes* obtained by combining several lower-level nodes into one parent node. Fig. 1 shows an example product taxonomy for a fashion Internet shopping mall, where *Conditioner*, *Pomade* and *Shampoo* are classified into *Hair care*, and so on.

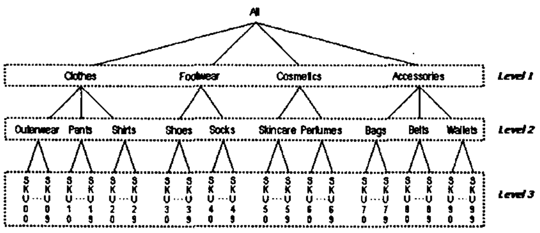

Fig. 1. Sample product taxonomy

A number called *level* can be assigned to each node in the product taxonomy. The level of the root node is zero, and the level of any other node is one plus the level of its parent. Please note that a higher-level product class has a smaller level number. The product taxonomy of Fig. 1 has four levels, referred to as levels 0 (for root), 1, 2, and 3.

Product taxonomies play an important role in the knowledge discovery process since they represent Internet shopping mall dependent knowledge and may affect the results. In many applications, strong association rules are more likely to exist at high levels of the product taxonomy but may likely repeat common knowledge.

2.3 Decision tree induction

The most popular classification method is the decision tree induction which builds a decision tree and performs classification on the given data using it (Berson, Smith & Thearing, 2000; Kim et al., 2001). A decision tree is a tree in which each nonleaf node denotes a test on an attribute of cases, each branch corresponds to an outcome of the test, and each leaf node denotes a class prediction (see Fig. 2).

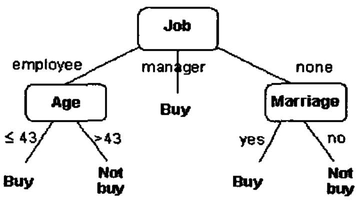

Fig. 2. A decision tree for purchase assessment

A case in the model set consists of multiple attributes (independent variables) and a known class label associated with it (dependant variables). The independent variables are represented as an attribute-value vector $\mathbf{x} = (x_1, x_2, \ldots, x_i)$. Assume that the cases can fall into j classes, that is, $C = (c_1, c_2, \ldots, c_j)$. Then, a model set can be denoted by $\mathbf{M} = \{(\mathbf{x}_m, y_m)\}$ where $\mathbf{x}_m \in \mathbf{X}$ (all possible attribute space) and $y_m \in C$ (all possible cases), $m = 1, \ldots, M$ (the size of the model set). Since all the cases in a score set have no known class levels, on the other hand, a score set is denoted by $\mathbf{S} = \{(\mathbf{x}_s, y_s)\}$ where $\mathbf{x}_s \in \mathbf{X}$ and $y_s \in \varnothing$, $s = 1, \ldots, S$ (the size of score set).

To build an effective model, the data in the model set must mimic the time frame when the model will be applied (Berry & Linoff, 2000). The time frame has three important components: *past*, *current* and *future*. Since we can predict the future through the past, the past is also divided into three time periods: the *distant past* used on the input side of the data, the *recent past* used to determine the output, and a period of *latency* used to represent the present (see Fig. 3).

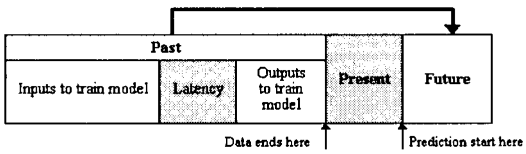

Fig. 3. The modeling time frame (Berry & Linoff, 2000)

3. Recommendation methodology

The methodology consists of five phases as shown in Fig. 4.

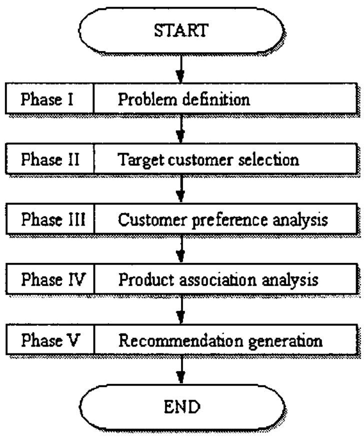

Fig. 4. The overall flow of recommendation

3.1 Problem definition

In most cases, recommendation problems in e-commerce can be classified according to 1) whether customers for whom we want recommendations (they are called target customers in this paper) are all customers or selective customers, 2) whether the objective of recommendation is to predict how much a particular customer will like a particular product (prediction problem), or to identify a list of products that will be of interest to a given customer (top-N recommendation problem), and 3) whether the recommendation is accomplished at a specific time or persistently.

This paper considers only the recommendation problem of helping selective customers find which products they would like to purchase by suggesting a list of *top-N* recommended products for each of them at the specific time.

Given the product taxonomy *T*, a recommendation problem can be denoted by $Rec(l,n,p,t)$ where *l, n, p* and *t* mean the recommendation based on level-*l* product classes over the taxonomy *T*, of presenting *n* products to each of target customers, for customers who have purchased *p* or more level-*l* product classes, and conducted at the time *t*, respectively.

Example 1. Given the product taxonomy in Fig. 1, let a recommendation problem be *Rec*(2,2,1,2001-12-1). Then, it represents the problem of recommending two products for customers, who have purchased one more level-2 product classes, on December 1 2001.

3.2 Target customer selection

To generate the model set and the score set of our recommendation problem $Rec(l,n,p,t)$, we also needs two more sets; One is the model candidate set and the other is the score candidate set which is a set of customers who form the score set.

Let $msst$, pd, pl and pr be the start time of the model set. Then, a model candidate set is defined as a set of customers who have purchased *p* or more level-l product classes between $msst$ time and $msst + pd$ time.

Example 2-1. Table 1 illustrates an example of determining the model candidate set from customer purchase records in the case that $msst$ =<May 1 2001>, pd =<four month>, pl =<one month>, and pr =<one month>. Here, we obtain as the model candidate set {101, 103, 104} since 101, 103 and 104 follow the definition of the model candidate set.

Table 1
Determining the model candidate set

CID	May	June	July	Aug.	Sep.	Oct.	Model set
101	-	-	-	Pants		-	T
102	-	-	-	-	Belts	-	F
103	Bags	Wallets	-	-		Outwear	T
104	-	Skincare, Perfumes	Socks	-		Socks	T
...	...	...	...	...		...	...

Now we discuss how to make a model set from the model candidate set. Given the model candidate set size of which is M, the model set $\mathbf{M}$ can be denoted as follows:

$$\mathbf{M} = \{(\mathbf{x}_m, y_m)\},\ m = 1, \ldots, M,$$

where $\mathbf{x}_m \in \mathbf{X}$ (all possible attribute space), and

$$y_m = \begin{cases} 1 & \text{if the customer has purchased new level-}l\text{ product classes} \\ & \text{between } msst + pd + pl \text{ time and } msst + pd + pl + pr \text{ time,} \\ 0 & \text{otherwise.} \end{cases}$$

Example 2-2. Table 2 shows the model set which can be derived from the purchase records in Table 1.

Table 2
The model set

CID	Age	Gender	Job	Purchase amount	Purchase frequency	Last visit	...	Y
101	22	M	Student	64	4	0827	...	0
103	36	F	Employee	57	6	1018	...	1
104	23	F	None	128	10	1104	...	1
...	...	...	..	...	...	...	...	...

After the decision tree has been built from the model set, the tree is applied to the score set in order to choose customers who will receive recommendation. We define the score candidate set as a set of customers who have purchased p or more level-l product classes between $t - pl - pd$ time and $t - pl$ time.

Example 2-3. Because t =<December 1 2001> from the problem definition, the score candidate set consists of the customers who have purchased one more level-*2* product classes between July and October. We can show from Table 3 that 201 and 203 satisfy the above definition and thus become the score candidate set.

Table 3
Determining the score candidate set

CID	July	Aug.	Sep.	Oct.	Nov.	Dec.	Score set
201	-	-	Skincare	Perfumes		?	T
202	-	-	-	-		?	F
203	Outwear	Shirts	Pants	-		?	T
...	...	...	...	...		...	...

The values of independent variables of the score set are also generated from records about customers who belong to the model candidate set. The next recommendation phases continue to be performed only for customers in the model candidate set who have 1 as the value of y.

Example 2-4. Given that the decision tree is applied to the score set generated in example 2-3, and the decision tree assigns 1 to the variable y of the customer 203 but 0 to the variable of y of the customer 201, only the customer 203 will receive product recommendation.

3.3 Customer preference analysis

The methodology applies the results of analyzing preference inclination of each customer to make recommendations.

A basic idea of measuring the customer's preference is simple and straightforward. If we assume that all customers in an online store buy products only in accordance with three sequential shopping steps, we can classify all products into four product groups such as purchased products, products placed in the basket, products clicked through, and the other products, as shown in Fig. 5.

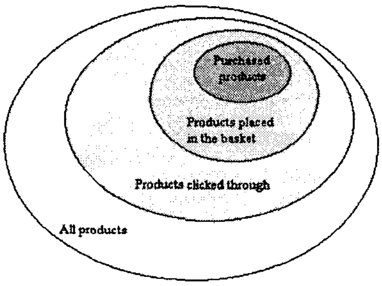

Fig. 5. Classification of products according to customer's shopping behavior

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

From the above discussions, we define the customer preference matrix $\mathbf{P} = (p_{ij})$, $i = 1, \cdots, M$ (total number of customers), $j = 1, \cdots, N$ (total number of level-l product classes, as follows:

(2)

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

Example 3. Assume that the target customers have shown their own shopping behaviors in Table 4(a) for the click-through step, Table 4(b) for the basket placement step and Table 4(c) for the purchase step, respectively. According to the Eq. (2), we obtain the customer preference matrix in Table 4(d).

Table 4
Customer preference model

(a) p_{ij}^{c}: preference in click-through step

CID	Outerwear	Pants	Shirts	Shoes	Socks	Skincare	Perfumes	Bags	Belts	Wallets
203	100	120	80	5	5	0	0	0	0	0
205	1	1	0	0	0	200	250	0	0	3
212	0	0	0	0	0	0	0	30	45	46
217	75	75	80	60	65	80	70	70	60	60
218	0	0	0	4	4	4	3	4	3	4

(b) p_{ij}^{b} preference in basket placement step

CID	Outerwear	Pants	Shirts	Shoes	Socks	Skincare	Perfumes	Bags	Belts	Wallets
203	3	3	1	0	0	0	0	0	0	0
205	0	0	0	0	0	5	4	0	0	0
212	0	0	0	0	0	0	0	1	6	6
217	0	0	0	2	2	1	0	1	1	2
218	0	0	0	2	2	2	0	2	0	2

(c) p_{ij}^{p}: preference in purchase step

CID	Outerwear	Pants	Shirts	Shoes	Socks	Skincare	Perfumes	Bags	Belts	Wallets
203	1	1	1	0	0	0	0	0	0	0
205	0	0	0	0	0	2	1	0	0	0
212	0	0	0	0	0	0	0	1	0	0
217	0	0	0	0	0	0	0	0	1	0
218	0	0	0	0	2	0	0	0	0	0

(d) p_{ij} customer preference matrix

CID	Outerwear	Pants	Shirts	Shoes	Socks	Skincare	Perfumes	Bags	Belts	Wallets
203	2.833	3	2	.042	.042	0	0	0	0	0
205	.004	.004	0	0	0	2.8	2.3	0	0	.012
212	0	0	0	0	0	0	0	1.82	1.98	2
217	.75	.75	1	1	1.25	1.5	.5	1	1.5	1
218	0	0	0	2	3	2	.75	2	.75	2

3.4 Product association analysis

In this phase, we first search for meaningful relationships or associations among product classes through mining association rules from the large transactions. The steps for mining level-l association rules from different transaction sets are as follows:

Step 1) Set the given time period as a time interval between $msst$ time and $t-1$ time.

Step 2) Gather all the transactions made in the given time period into a single transaction in the form of <customer ID, {a set of products}>.

Step 3) Find association rules among level-l product classes according to the following sub-steps:

Step 3-1) Set minimum support and minimum confidence.

Step 3-2) Replace each product in transaction set with its corresponding level-l product class.

Step 3-3) Find all frequent itemsets of size 2 using Apriori or its variants

Step 3-4) Generate association rules containing a single product class in both body and head from the set of all frequent itemsets of size 2.

Next, we calculate the extent to which each product class appeals to each customer from the discovered rules. This work results in building a model called product association model represented by a matrix. Given product class X and Y, let $X \overset{p}{\Rightarrow} Y$, $X \overset{b}{\Rightarrow} Y$, and $X \overset{c}{\Rightarrow} Y$ denote level-l association rules in purchase transaction set, in basket placement transaction set, and in click-through transaction set, respectively. Then, a product association matrix $\mathbf{A} = (a_{ij})$, $i = 1, \cdots, M$ (total number of target customers), $j = 1, \cdots, N$ (total number of level-l product classes), is defined as follows:

$$a_{ij} = \begin{cases} 1 & \text{if } i = j \\ 1 & \text{if } i \overset{p}{\Rightarrow} j \\ .5 & \text{if } i \overset{b}{\Rightarrow} j \\ .25 & \text{if } i \overset{c}{\Rightarrow} j \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The first condition of the Eq. (3) captures the association among different products in a product class: a purchase of a product in a product class implies a preference in other products in the same product class. Please note that the matrix $\mathbf{A}$ is not symmetric since the rule $j \Rightarrow i$ is not guaranteed to have minimum confidence even though the rule $i \Rightarrow j$ satisfies both minimum support and confidence.

Example 4. Assume that the association rules discovered from transactions in each shopping step are like Table 5. For simplicity of expression, basket placement rules overlapped with purchase rules and click-through rules overlapped with basket placement rules are skipped in Table 5.

Table 5

Discovered association rules

Shopping step	Rule types	Rule sets
Click-through	$X \overset{c}{\Rightarrow} Y$	Outwear ⇒ Bags, Outwear ⇒ Belts, Outwear ⇒ Wallets Pants ⇒ Bags, Pants ⇒ Wallets Shirts ⇒ Bags, Shirts ⇒ Belts, Shirts ⇒ Wallets Bags ⇒ Outwear, Bags ⇒ Pants, Bags ⇒ Shirts Belts ⇒ Outwear, Belts ⇒ Shirts Wallets ⇒ Outwear, Wallets ⇒ Pants, Wallets ⇒ Shirts
Basket placement	$X \overset{b}{\Rightarrow} Y$	Outwear ⇒ Shoes, Outwear ⇒ Socks Pants ⇒ Shoes, Pants ⇒ Socks, Pants ⇒ Belts Shirts ⇒ Shoes, Shirts ⇒ Socks Belts ⇒ Pants
Purchase	$X \overset{p}{\Rightarrow} Y$	Outerwear ⇒ Shirts Shirts ⇒ Outerwear

By applying the Eq. (3) to the discovered rules, we get a product association matrix in Table 6, each row of which represents associations among a particular product class and other product classes.

Table 6

Product association matrix

	Outerwear	Pants	Shirts	Shoes	Socks	Skincare	Perfumes	Bags	Belts	Wallets
Outwear	1		1	.5	.5			.25	.25	.25
Pants		1		.5	.5			.25	.5	.25
Shirts	1		1	.5	.5			.25	.25	.25
Shoes				1						
Socks					1					
Skincare						1	.25			
Perfumes						.25	1			
Bags	.25	.25	.25					1		
Belts	.25	.5	.25						1	
Wallets	.25	.25	.25							1

3.5 Recommendation generation

Our methodology use cosine coefficient (Mobasher, Cooly & Srivastava, 2000; Lawrence et al, 2001; Sarwar et al., 2001) to measure the similarity. The matching score s_{ij} between customer i and level-l product class j is computed as followings:

$$s_{ij} = \frac{\mathbf{P}_i \cdot \mathbf{A}_j}{\|\mathbf{P}_i\| \|\mathbf{A}_j\|} = \frac{\sum_{k=1}^{N} p_{ik} a_{jk}}{\sqrt{\sum_{k=1}^{N} p_{ik}^2} \sqrt{\sum_{k=1}^{N} a_{jk}^2}}, \quad (4)$$

where $\mathbf{P}_i$ is a row vector of the $M \times N$ customer preference matrix $\mathbf{P}$, and $\mathbf{A}_j$ is a row vector of the $N \times N$ product association matrix $\mathbf{A}$. Here, M refers the total number of customers and N

refers the total number of level-l product classes. The s_{ij} value ranges from 0 to 1, where more similar vectors result in bigger value.

Example 5-1. Applying the customer preference matrix in Table 4(d) and the product association matrix in Table 6 to the Eq. (4), we get the matching scores in Table 7.

Table 7

Matching scores

CID	Outerwear	Pants	Shirts	Shoes	Socks	Skincare	Perfumes	Bags	Belts	Wallets
203	.648	.484	.648	.009	.009	.000	.000	.392	.504	.392
205	.001	.001	.001	.000	.000	.904	.803	.001	.001	.004
212	.264	.424	.264	.000	.000	.000	.000	.498	.504	.548
217	.676	.675	.676	.296	.370	.466	.251	.441	.583	.441
218	.440	.554	.440	.391	.587	.415	.237	.359	.125	.359

We suggest three different strategies related with such a choice:

- *Recommendation of the most frequently purchased product*
- *Recommendation of product with the highest click-to-buy rate*
- *Recommendation of the latest product*

We now propose the steps for choosing recommended products from product classes using the matching score.

Step 1) *Select the choice strategy.*

Step 2) *Determine the number of recommended product classes, nc, such that nc < n and n/nc is an integer.*

For each customer:

Step 3) *Select the nc highest scored product classes.*

Step 4) *Make a recommendation list which consists of n/nc products per a selected product class, according to the selected choice strategy. Here, previously bought products are excluded from the recommendation list.*

Example 5-2. Assume that we select products with the highest click-to-buy rate and the four highest click-to-buy rates per a product class are like Table 8. If the marketer set $nc = 2$, the number of recommended products per product class become 1 because n=2 from the problem definition. Applying the above steps to Table 7 and Table 8, we finally get a recommendation list in Table 9 for each target customer.

Table 8

Click-to-buy rates

Product Class	Outerwear	Pants	Shirts	Shoes	Socks
Product	SKU02(.035)	SKU13(.031)	SKU25(.050)	SKU30(.029)	SKU48(.015)
	SKU04(.032)	SKU15(.029)	SKU20(.033)	SKU32(.028)	SKU43(.011)
	SKU03(.023)	SKU19(.025)	SKU24(.031)	SKU31(.012)	SKU42(.011)
	SKU01(.017)	SKU18(.011)	SKU22(.018)	SKU39(.011)	SKU44(.005)

Product Class	Skincare	Perfumes	Bags	Belts	Wallets
Product	SKU55(.025)	SKU63(.049)	SKU72(.054)	SKU80(.061)	SKU99(.035)
	SKU51(.023)	SKU65(.040)	SKU71(.035)	SKU81(.043)	SKU93(.025)
	SKU52(.022)	SKU62(.028)	SKU77(.027)	SKU87(.027)	SKU94(.023)
	SKU53(.021)	SKU69(.028)	SKU74(.018)	SKU85(.018)	SKU95(.020)

Table 9

Recommendation lists

CID	Purchased Products	Recommended product classes	Recommended products
203	SKU00, SKU15, SKU25	Outwear, Shirts	SKU02, SKU20
205	SKU51, SKU55, SKU69	Skincare, Perfumes	SKU52, SKU63
212	SKU72	Wallets, Belts	SKU99, SKU80
217	SKU83	Outwear, Shirts	SKU02, SKU25
218	SKU44, SKU48	Socks, Pants	SKU43, SKU13

4. Recommender system

For the implementation of the proposed recommendation methodology, a recommender system is developed using agent and data warehousing technologies.

Web log analysis agent

This agent manages web log database through periodic collecting, parsing and analyzing web server log files such as access logs, referrer logs, agent logs and cookie files.

Data transformation agent

This agent creates and manages the data mart that provides data indispensable to accomplish recommendation tasks.

Data mining agent

This agent activates and manages data mining algorithms such as the decision tree induction and the association rule mining.

Target customer selection agent

The target customer selection agent takes customer data from the data mart and creates the decision tree model from the data in cooperation with the data mining agent.

Preference analysis agent

This agent takes charge of customer preference analysis described in section 3.3.

Association analysis agent

This agent has the role of analyzing associations between product classes described in section 3.4.

Recommendation generation agent

This agent makes a personalized recommendation list for each target customer according to the steps mentioned in section 3.5.

Interface management agent

This agent provides the users with an interface whereby they can specify recommendation task, set control parameters, and show results through transparent communication with agents that are needed to accomplish the given recommendation task.

5. Conclusion

The characteristics of the suggested methodology are as follows. First, The customer preference and the product association are automatically learned from clickstream (web usage data), unlike other recommendation methodologies which learn them from purchase records only. Second, in order to avoid the poor recommendations that will lead to disappoint customers, customers who are likely to buy recommended products are selected using decision tree induction. Third, the explicit participation of the marketers and the formal usage of background knowledge such as the product taxonomy are also introduced in the recommendation process. Finally, we devise measures to choose highly business-efficient products among the candidate recommendable products.

As further researches, it will be interesting to compare our suggested methodology with a standard collaborative filtering based methodology in the aspect of recommendation performance.

References

Berson, A., Smith, K., & Thearing, K. (2000). *Building Data Mining Applications for CRM*, New York: McGraw-Hill.

Berry, J. A., & Linoff, G. (1997). *Data mining techniques: for marketing, sales, and customer support*, New York: Wiley.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. (1999). Combining content-based and collaborative filters in an online newspaper. In *ACM SIGIR'99 Workshop on Recommender Systems*, Berkeley, CA, August 1999.

Cooley, R., Mobasher, B., & Srivastava, J. (1999). Data preparation for mining world wide web browsing patterns. *Journal of Knowledge and Information Systems*, 1.

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

Lawrence, R. D., Almasi, G. S., Kotlyar, V., Viveros, M. S., & Duri, S. S. (2001). Personalization of supermarket product recommendations. *Data Mining and Knowledge Discovery*, *5* (1-2), 11-32.

Mobasher, B., Cooley, R., & Srivastava, J. (2000). Automatic personalization based on web usage mining. *Communications of the ACM*, *43* (8), 142-151.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. In *Proceedings of ACM E-Commerce 2000 conference*, 158-167.