A COMPARATIVE EVALUATION OF HYBRID PRODUCT RECOMMENDATION PROCEDURES FOR WEB RETAILERS

Do Hyun Ahn*, Jae Kyeong Kim*, and Yoon Ho Cho**

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

** School of e-Business, Kookmin University, Seoul, South Korea
E-mail: www4u@kookmin.ac.kr

ABSTRACT

A product recommendation is an enabling mechanism to overcome information overload occurred when shopping in an e-marketplace. Recommendation methods are a personalized information filtering technology to help customers find the products they would like to purchase. Collaborative filtering is the most successful recommendation technology but its application to e-commerce has exposed well-known limitations such as sparsity and scalability. We introduce several hybrid product recommendation procedures based on clustering, Web usage mining, collaborative filtering, and content-based filtering driven a bayesian model (CBBM) to overcome them. The recommendation quality of each hybrid product recommendation procedure is compared with others by several experimentations. Through the experiment with real Web retailer data, it is found that hybrid procedure using Web usage mining, and a bayesian model can perform recommendation tasks effectively, but using clustering analysis can perform efficiently.

KEYWORDS: Product recommendation, Clustering, Web usage mining, Bayesian model

1. INTRODUCTION

The rapid growth of e-commerce has made both companies and customers face a new situation. The companies in the e-commerce domain have become to be harder to survive due to more and more competitions. On the other hand, the customers have confronted with product overload where they are no longer able to effectively choose the products. As as result, the concern with new marketing strategies such as one-to-one marketing and customer relationship management(CRM) has been growing. One of the promising technologies to overcome product overload and a new marketing strategy solution is a product recommendation method.

Recommendation methods are a personalized information filtering technology to help customers find the products they would like to purchase [Basu et al., 1998; Hill et al., 1995; Lawrence et al., 2001]. Collaborative filtering (CF) has been known to be the most successful recommendation technology. CF identifies customers (neighbors) whose preferences are similar to those of a given customer and recommend products...
they have liked. However, its widespread use in e-commerce has exposed two research issues, sparsity and scalability [Cho et al., 2004; Kim et al., 2003; Hill et al., 1995; Sarwar et al., 2000]. The first is related to sparsity. In a large e-commerce site such as Amazon.com, there are millions of products and so customers may rate only a very small portion of those products. Most similarity measures used in collaborative filtering cannot work properly unless there are a minimum number of common ratings among customers [Cho et al., 2004; Kim et al., 2003; Sarwar et al., 2000]. Such a sparse situation makes CF-based recommender systems difficult to accurately compute the neighborhood, and so results in poor quality of recommendations. Many approaches have been proposed to overcome the sparsity problem. These approaches can be classified into three categories; implicit ratings, hybrid filtering and item-to-item correlation. The implicit ratings approaches attempt to increase the number of ratings through observing customers' behavior [Kostan et al., 1997]. The hybrid filtering approaches are to combine content-based filtering and collaborative filtering for augmenting sparse preference ratings [Basu et al., 1998, Cho et al., 2004; Lawrence et al., 2001; Melville et al., 2001].

The second is related to scalability. Recommender systems for e-commerce have to deal with millions of customers and products. Because these systems usually handle high-dimensional data to form the neighborhood, the nearest neighbor algorithm is often very time-consuming and scales poorly in practice [Cho et al., 2004; Kim et al., 2003; Sarwar et al., 2001]. To address the scalability problems in CF-based recommender systems, a variety of approaches have been developed. According to Sarwar (2001), these approaches can be classified into two main categories; dimensionality reduction techniques and model-based approaches. Latent Semantic Index (LSI) is a widely used dimensionality reduction technique. It uses singular value decomposition (SVD) to factor the original rating space into three matrices and performs the dimensionality reduction by reducing the singular matrix.

To overcome these problems of existing recommender systems in the e-commerce domain, we introduce several hybrid product recommendation procedures based on clustering, Web usage mining, collaborative filtering, and content-based filtering driven a bayesian model(CBBM). The characteristics of our suggested methodology are as follows: (1) We consider the application of clustering analysis to improve scalability of recommender systems. (2) We use CBBM to improve the quality of recommendation. (3) To compare the effect of each approach, we suggest several procedures and evaluate them with the real Web retailer data.

The remainder of the paper is organized as follows. Chapter 2 reviews the past research works related to recommendation techniques. Chapter 3 provides our research framework and Chapter 4 describes experimental works to verify our hybrid product recommendation procedures. Chapter 5 finally provides some conclusions and future works.

2. BACKGROUND

2.1 Collaborative Filtering Algorithm
CF is the most successful recommendation technique, which has been used in a number of different applications such as recommending movies, articles, products, Web pages, etc. [Basu et al., 1998; Cho et al., 2004; Hill et al., 1995; Kim et al., 2003; Lawrence et al., 2001]. CF-based recommender systems recommend products to a target customer according to the following steps [Sarwar, 2001]: (1) A customer provides the system with preference ratings of products that may be used to build a customer profile of his or her likes and dislikes. (2) Then, these systems apply statistical techniques or machine learning techniques to find a set of customers, known as neighbors, which in the past have exhibited similar behavior (i.e., they either rated similarly or purchased similar set of products). Usually, a neighborhood is formed by the degree of similarity between the customers. (3) Once a neighborhood of similar customers is formed, these systems predict whether the target customer will like a particular product by calculating a weighted composite of the neighbors' ratings of that product (prediction problem), or generate a set of products that the target customer is most likely to purchase by analyzing the products the neighbors purchased (top-N recommendation problem).

These systems also known as the nearest neighbor CF-based recommender systems have been widely used in practice. However, as the number of customers and that of products managed in an e-commerce site grow rapidly, its application to e-commerce has exposed two major issues: sparsity and scalability that must be addressed [Cho et al., 2004; Kim et al., 2003; Sarwar, 2001; Sarwar et al., 2000].

2.2 Web Usage Mining

Web usage mining is the process of applying data mining techniques to the discovery of behavior patterns based on Web log data, for various applications. In the advance of e-commerce, the importance of Web usage mining grows larger than before. The overall process of Web usage mining is generally divided into two main tasks: data preprocessing and pattern discovery. Mining behavior patterns from Web log data needs the data preprocessing tasks that include data cleansing, user identification, session identification, and path completion. Data cleansing performs merging Web logs from multiple servers, removing irrelevant and redundant log entries with filename suffixes such as gif, jpeg, map, count.cgi, etc., and parsing of the logs. To track individual user’s behaviors at a Web site, user identification and session identification is required. For Web sites using session tracking such as URL rewriting, persistent cookies or embedded session IDs, user and session identification is trivial. Web sites without session tracking must rely on heuristics. Path completion may also be necessary because of local or proxy level caching. Cooley, Mobasher and Srivastava (1999) presented a detailed description of data preprocessing methods for mining Web browsing patterns. The pattern discovery tasks involve the discovery of association rules, sequential patterns, usage clusters, page clusters, user classifications or any other pattern discovery method [Mobasher, Cooley & Srivastava, 2000]. The usage patterns extracted from Web data can be applied to a wide range of applications such as Web personalization, system improvement, site modification, business intelligence discovery, usage characterization, etc. [Srivastava et al., 2000]. There have been several customer behavior models for e-commerce, which have different analysis purposes. Menascé et al. (1999) presented a state transition graph, called Customer Behavior Model Graph (CBMG) to describe the behavior of groups of customers who exhibit similar navigational patterns. VandeMeer, Dutta and Datta (2000) developed a user navigation model designed for supporting and
tracking dynamic user behavior in online personalization. Lee et al. (2001) provided a
detailed case study of clickstream analysis from an online retail store. To measure the
effectiveness of efforts in merchandising, they analyzed the shopping behavior of
customers according to the following four shopping steps: product impression,
click-through, basket placement, and purchase. It has been recognized that Web usage
mining gave better recommendation quality in the CF recommendation procedures [Cho
et al., 2004, Kim et al, 2003]. Therefore, we recommend products based on Web usage
data.

2.3 Clustering Analysis

Clustering analysis has been applied to a wide range of disciplines such as data mining,
statistics, machine learning, spatial database technology, biology, and marketing. Owing
to the huge amount of data collected in databases, clustering analysis has recently
become a highly active topic [Han & Kamber, 2001]. Clustering analysis has been
studied extensively for many years, focusing mainly on distance-based clustering
analysis such as k-menas, k-medoids, and several other methods [Alsabti et al., 1998].
In data mining, efforts have focused on finding methods for efficient and effective
clustering analysis in large databases [Bradley et al., 1998]. Active themes of research
focus on the scalability of clustering methods, the effectiveness of methods for
clustering complex shapes and types of data, high-dimensional clustering techniques,
and methods for clustering mixed numerical and categorical data in large databases.
Clustering analysis has been applied to the recommendation field [Bradley et al., 1998;
Konstan et al., 1997]. Earlier collaborative filtering research conducted in the Usenet
domain [Konstan et al., 1997] reported the benefits of partitioning. In particular, they
found improved prediction quality with partitioned newsgroups compared to the whole
Usenet. Nowadays many current collaborative filtering methods use clustering analysis
for the formation of neighborhood [Ungar & Foster, 1998]. In this paper, we applied
clustering analysis to WebCF to improve the performance, especially to solve scalability
issue of traditional CF systems.

2.4 Content-based filtering driven bayesian model (CBBM)

A bayesian model can predict class membership probabilities, such as the probability
that a given sample belongs to a particular class. Studied comparing classification
algorithms have found a simple bayesian classifier known as the naïve bayesian
classifier to be comparable in performance with decision tree and neural network
classifiers. Bayesian classifiers have also exhibited high accuracy and speed when
applied to large databases [Han & Kamber, 2001]. Naïve bayesian classifiers assume
that the effect of an attribute value on a given class is independent of the values of the
other attributes. This assumption is called class conditional independence. It is made to
simplify the computations involved and, in this sense, is considered “naïve”. Bayesian
belief networks are graphical models, which unlike naïve Bayesian Classifiers allow the
proposed a bayesian methodolgy for recommender systems that incorporate user ratings,
user features, and item features in a single unified framework. They considered a dataset
with m “items”, each item having a feature vector of length p, and n “users”, each
having a feature vector of length q. Their goal was to compute P,(L=1|f), the probability
that the $i$th user will like the item represented by the feature vector $f$, where $f = (f_1, f_2, \ldots, f_p)$, and for now, $f_k \in \{0, 1\}, \ k = 1 \ldots p$. They did this for each item, and then recommend those with the highest probabilities. In this paper we also use a bayesian model to improve the quality of recommendation.

3. METHODOLOGY

3.1 Research Design

In this section, we first set up the research questions that we examine in this study. Then, we suggest several procedures designed to answer our research questions. We pose two research questions that will accomplish our research objective aforementioned:

Q1. Does clustering-based WebCF give better performance than WebCF alone?
Q2. Does WebCF with CBBM give better recommendation quality than WebCF alone?

3.2 Clustering Phase

We consider the application of clustering techniques to improve scalability of recommender systems. Clustering of users can effectively partition entire dataset. Earlier studies [Alsabi et al., 1998; Konstan et al., 1997; Sarwar, 2001; Ungar et al., 1998] indicate the benefits of applying clustering in recommender systems. The idea is to partition the users of a Web retailer using a clustering algorithm and to use the partitions as neighborhoods. We use the $k$-means method that has been shown to be effective in producing good clustering results for many practical applications [Alsabi et al., 1998; Konstan et al., 1997; Sarwar, 2001]. This method has two benefits – first, it reduces the sparsity of the dataset and second, due to the dimensionality reduction, the neighborhood formation in faster. The steps of suggested method modified from works of Sarwar (2001) is given below.

Step1. Apply the $k$-means clustering algorithm to produce partitions of users using the behavior data such as frequency, recency, duration, click-through, basket- placement and purchase information. Formally, the dataset $A$ is partitioned in $A_1, A_2, \ldots, A_p$, where $A_i \cap A_j = \phi$, for $1 \leq i, j \leq P$; and $A_1 \cup A_2 \cup \ldots \cup A_p = A$.

Step2. Determine the particular partition $A_i$ for a given customer.

3.3 Customer Profile Creation and Neighborhood Formation

A customer profile is a description of customer interests or preferences about products. Recommending products to a particular customer depends on his/her profile. This phase discovers customer’s preferences about products, and makes the customer profile based on the preferences. For this purpose, the customer profile is constructed based on the following three general shopping steps in Web retailers, which is modified from the works of Lee et al. (2001):

1. click-through: the click on the hyperlink and the view of the Web page of the
A basic idea of measuring the customer’s preference is simple and straightforward. The customer’s preference is measured by counting only the number of occurrence of URLs mapped to the product from clickstream of the customer. In general Web retailers, products are purchased in accordance with above three sequential shopping steps, so we can classify all products into four product groups such as purchased products, products placed in the basket, products clicked through, and the other products. This classification provides an is-a relation between different groups such that purchased products is-a products placed in the basket, and products placed in the basket is-a products clicked through. From this relation, it is reasonable to obtain a preference order between products such that \{products never clicked\} \(\pi\) \{products only clicked through\} \(\pi\) \{products only placed in the basket\} \(\pi\) \{purchased products\}. Hence, it makes sense to assign the higher weight to occurrences of purchased products than those of products only placed in the basket. Similarly, the higher weight is given to products placed in the basket than those of products only clicked through, and so on.

Let \(p_{ij}^c\) be the total number of occurrences of click-throughs of a customer \(i\) across every products in a grain product class \(j\). Likewise, \(p_{ij}^b\) and \(p_{ij}^p\) are defined as the total number of occurrences of basket placements and purchases of a customer \(i\) for a product \(j\), respectively. \(p_{ij}^c\), \(p_{ij}^b\) and \(p_{ij}^p\) are calculated from clickstream data as the sum over the given time period, and so reflect individual customer’s behaviors in the corresponding shopping process over multiple shopping visits. From the above terminology, we define the customer profile as the matrix of ratings \(P = (p_{ij})\), \(i = 1, \ldots, M\) (total number of customers), \(j = 1, \ldots, N\) (total number of products), as follows:

\[
p_{ij} = \left\{ \frac{p_{ij}^c - \min_{1 \leq j \leq N} (p_{ij}^c)}{\max_{1 \leq j \leq N}(p_{ij}^c) - \min_{1 \leq j \leq N}(p_{ij}^c)} + \frac{p_{ij}^b - \min_{1 \leq j \leq N} (p_{ij}^b)}{\max_{1 \leq j \leq N}(p_{ij}^b) - \min_{1 \leq j \leq N}(p_{ij}^b)} + \frac{p_{ij}^p - \min_{1 \leq j \leq N} (p_{ij}^p)}{\max_{1 \leq j \leq N}(p_{ij}^p) - \min_{1 \leq j \leq N}(p_{ij}^p)} \right\} \times \frac{1}{3} \tag{1}
\]

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

Neighborhood Formation performs computing the similarity between customers and, based on that, forming a proximity-based neighborhood between a target customer and a number of like-minded customers. The process follows the same manner as that of typical nearest-neighbor algorithms [Sarwar et al., 2000] except forming the neighborhood in the same customer segment. The details of the neighborhood formation are as follows.

Given the customer profile matrix \(P\), the similarity between two customers \(a\) and \(b\) which is contained in customer segment \(C_i\), denoted by \(\text{sim}(a, b)\), is usually measured
using either the correlation or the cosine measure. In our research, we use the following correlation measure.

**Correlation.** The similarity between two customers \(a\) and \(b\) in particular partition \(A_i\) is measured by calculating the *Pearson-r* correlation \(\text{corr}_{ab}\), which is given by

\[
\text{sim}(a, b \in A_i) = \text{corr}_{ab} = \frac{\sum_i (p_{aj} - \overline{p}_a)(p_{bj} - \overline{p}_b)}{\sqrt{\sum_i (p_{aj} - \overline{p}_a)^2 \sum_i (p_{bj} - \overline{p}_b)^2}}
\]

where \(p_{aj}\) and \(p_{bj}\) are customer \(a\) and \(b\)'s ratings on product \(j\), respectively, and \(\overline{p}_a\) and \(\overline{p}_b\) are customer \(a\) and \(b\)'s average ratings on all products, respectively.

### 3.4 Generation of Recommendation list

This step is to ultimately derive the top-N recommendation from the neighborhood of customers. For each customer \(c\), we produce a recommendation list of \(N\) products that the target customer is most likely to purchase. Previously bought products are excluded from the recommendation list in order to broaden each customer’s purchase patterns or coverage. Generally, two techniques have been used for generating a recommendation list for a given customer: Recommendation of the most frequently purchased product (MFP) and Recommendation of the most frequently referred product (MFR). In this paper, we suggest a content-based filtering driven bayesian model(CBBM) for a generating a recommendation list.

**Recommendation of the most frequently purchased product (MFP).** This technique looks into the neighborhood \(N\) and for each neighbor, scans through a sales database and counts the purchase frequency of the products [Sarwar et al., 2000]. After all neighbors are accounted for, the system sorts the products according to their frequency count and returns the \(N\) most frequently purchased products as the recommendation list. This technique assumes that the more a product is sold, the more popular it becomes.

**Recommendation of the most frequently referred product (MFR).** Unlike MFP technique based on purchase frequencies of all neighbors, this technique sorts the products according to their reference frequencies [Cho et al., 2004]. The reference frequency of the neighborhood of a particular customer \(i\) for a product \(j\) (\(1 \leq j \leq N\)), \(RF_{i,j}\), is defined below:

\[
RF_{i,j} = \sum_{\text{i.effective neighbors of customer } i} \frac{r_{ij}^c - \min_{1 \leq j \leq N} (r_{ij}^c)}{\max_{1 \leq j \leq N} (r_{ij}^c) - \min_{1 \leq j \leq N} (r_{ij}^c)} + \frac{r_{ij}^b - \min_{1 \leq j \leq N} (r_{ij}^b)}{\max_{1 \leq j \leq N} (r_{ij}^b) - \min_{1 \leq j \leq N} (r_{ij}^b)} + \frac{r_{ij}^p - \min_{1 \leq j \leq N} (r_{ij}^p)}{\max_{1 \leq j \leq N} (r_{ij}^p) - \min_{1 \leq j \leq N} (r_{ij}^p)}
\]

where \(N\) is the number of products, and \(r_{ij}^c\), \(r_{ij}^b\) and \(r_{ij}^p\) is the total number of occurrences of click-throughs, basket placements and purchases of a customer \(i\) for a product \(j\), respectively. This method follows from the hypothesis that the more a product is referred, the higher the possibility of product’s purchase becomes. The reference frequency is computed using clickstream data as in building the customer profile.
Recommendation of Content-based filtering driven bayesian model (CBBM). Our goal is to compute the probability that given customer will like the product represented by the feature vector $X$, where $X = (X_1, X_2, ..., X_n)$. We do this for the products purchased by all neighbors and then recommend Top $N$ products with the highest probabilities. For this purpose, the naïve Bayesian Classifier is constructed based on following steps modified from works of Han & Kamber (2001):

**Step 1.** Each dataset is represented by an $n$-dimensional feature vector $X$, where $X = (X_1, X_2, ..., X_n)$, from the Product Feature Matrix as shown in Table 1 which is a source of content information. We assigned four features to each product as shown in Table 1. The choice of these features was advised with a fashion specialist. We assume that for each customer the product features are conditionally independent.

**Step 2.** This research considers 2 classes, $C_1$ and $C_2$. If a product is purchased, it is labeled as $C_1$, otherwise it is $C_2$. Given an unknown dataset, $X$ (i.e., having no class label), the classifier will predict that $X$ belongs to the class having the highest posterior probability, conditioned on $X$. Thus we maximize $P(C_i|X)$. The class $C_i$ for which $P(C_i|X)$ is maximized is called the maximum posterior hypothesis. By Bayes theorem,

$$P(C_i | X) = \frac{P(X | C_i)P(C_i)}{P(X)}.$$  

**Step 3.** As $P(X)$ is constant for all classes, only $P(X|C_i)P(C_i)$ need be maximized.

Note that the class prior probabilities are estimated by $P(C_i) = \frac{s_i}{s}$, where $s_i$ is the number of training datasets of class $C_i$, and $s$ is the total number of training datasets.

**Step 4.** In order to reduce computation in evaluating $P(X|C_i)$, the naïve assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the dataset, that is, there are no dependence relationships among the attributes. Thus,

$$P(X | C_i) = \prod_{k=1}^{4} P(x_k | C_i).$$

For example, we are to predict the class label of an unknown dataset using bayesian model, given CID0001’s training dataset as in Table 2. We assign 4 features Image, Price, Fashion Trend, and Material to each product for calculation conveniently. The class label attribute, buy_product, has two distinct values (namely, \{Yes, No\}). Let $C_1$ correspond to the class buy_product = “Yes” and $C_2$ correspond to the class buy_product = “No”. The unknown dataset $X(PID15)$ in Table 5 we wish to classify is

$$X(PID15) = (\text{Image} = \text{“Classic”}, \text{Price} = \text{“Medium”}, \text{Fashion Trend} = \text{“Trend”}, \text{Material} = \text{“Wool”}).$$
### Table 1. An example of Product Feature Matrix

<table>
<thead>
<tr>
<th>CID 00001</th>
<th>Product1</th>
<th>Product2</th>
<th>Product3</th>
<th>Product4</th>
<th>Product5</th>
<th>Product6</th>
<th>Product7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Image</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>① classic</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>② modern</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>③ casual</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Price</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>① high</td>
<td>*</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>② middle</td>
<td></td>
<td>*</td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>③ low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>3. Fashion Trend</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>① trend</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>② basic</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Material</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>① cotton</td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>② linen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>③ silk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>④ wool</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>⑤ polyester</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>⑥ rayon</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>⑦ spantex</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>⑧ knit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Class: buys_product 1 0 1 0 0 0 1

### Table 2. An example of CID0001’s training dataset

<table>
<thead>
<tr>
<th>PID</th>
<th>Image</th>
<th>Price</th>
<th>Fashion Trend</th>
<th>Material</th>
<th>Class: buys_product</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Classic</td>
<td>High</td>
<td>Basic</td>
<td>Wool</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>Classic</td>
<td>High</td>
<td>Basic</td>
<td>Cotton</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>Modern</td>
<td>High</td>
<td>Basic</td>
<td>Wool</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>Casual</td>
<td>Medium</td>
<td>Basic</td>
<td>Wool</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Casual</td>
<td>Low</td>
<td>Trend</td>
<td>Wool</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>Casual</td>
<td>Low</td>
<td>Trend</td>
<td>Cotton</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Modern</td>
<td>Low</td>
<td>Trend</td>
<td>Cotton</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>Classic</td>
<td>Medium</td>
<td>Basic</td>
<td>Wool</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>Classic</td>
<td>Low</td>
<td>Trend</td>
<td>Wool</td>
<td>Yes</td>
</tr>
<tr>
<td>10</td>
<td>Casual</td>
<td>Medium</td>
<td>Trend</td>
<td>Wool</td>
<td>Yes</td>
</tr>
<tr>
<td>11</td>
<td>Classic</td>
<td>Medium</td>
<td>Trend</td>
<td>Cotton</td>
<td>Yes</td>
</tr>
<tr>
<td>12</td>
<td>Modern</td>
<td>Medium</td>
<td>Basic</td>
<td>Cotton</td>
<td>Yes</td>
</tr>
<tr>
<td>13</td>
<td>Modern</td>
<td>High</td>
<td>Trend</td>
<td>Wool</td>
<td>Yes</td>
</tr>
<tr>
<td>14</td>
<td>Casual</td>
<td>Medium</td>
<td>Basic</td>
<td>Cotton</td>
<td>No</td>
</tr>
<tr>
<td>15</td>
<td>Classic</td>
<td>Medium</td>
<td>Trend</td>
<td>Wool</td>
<td>?</td>
</tr>
</tbody>
</table>
We need to maximize $P(X|C_i)P(C_i)$, for $i = 1, 2$. $P(C_i)$, the prior probability of each class, can be computed based on the training dataset:

- $P(\text{buys\_product} = \text{"Yes"}) = 9/14 = 0.643$
- $P(\text{buys\_product} = \text{"No"}) = 5/14 = 0.357$

To compute $P(X|C_i)$, for $i = 1, 2$, we compute the following conditional probabilities:

- $P(\text{Image} = \text{"Classic"} | \text{buys\_product} = \text{"Yes"}) = 2/9 = 0.222$
- $P(\text{Image} = \text{"Classic"} | \text{buys\_product} = \text{"No"}) = 3/5 = 0.600$
- $P(\text{Price} = \text{"Medium"} | \text{buys\_product} = \text{"Yes"}) = 4/9 = 0.444$
- $P(\text{Price} = \text{"Medium"} | \text{buys\_product} = \text{"No"}) = 2/5 = 0.400$
- $P(\text{Fashion Trend} = \text{"Trend"} | \text{buys\_product} = \text{"Yes"}) = 6/9 = 0.667$
- $P(\text{Fashion Trend} = \text{"Trend"} | \text{buys\_product} = \text{"No"}) = 1/5 = 0.200$
- $P(\text{Material} = \text{"Wool"} | \text{buys\_product} = \text{"Yes"}) = 6/9 = 0.667$
- $P(\text{Material} = \text{"Wool"} | \text{buys\_product} = \text{"No"}) = 2/5 = 0.400$

Using the above probabilities, we obtain

- $P(X | \text{buys\_product} = \text{"Yes"}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$
- $P(X | \text{buys\_product} = \text{"No"}) = 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019$
- $P(X | \text{buys\_product} = \text{"Yes"}) P(\text{buys\_product} = \text{"Yes"}) = 0.044 \times 0.643 = 0.028$
- $P(X | \text{buys\_product} = \text{"No"}) P(\text{buys\_product} = \text{"No"}) = 0.019 \times 0.357 = 0.007$

Therefore, bayesian model predicts $\text{buys\_product} = \text{"Yes"}$ for the unknown dataset $X(PID15)$. We do this for the products purchased by all neighbors and then recommend Top $N$ products with the highest posterior probabilities.

### 4. EXPERIMENTAL EVALUATION

#### 4.1. Data Preparation

For our experiments, we use Web log data from the S Web retailer that sells women's supplies. S Web retailer deals with 7513 products. The 110 log files was collected from four IIS Web servers during period between 1st May 2002 and 7th June 2002. The total size of log files is about 25,360MB, and total number of HTTP requests is about 510,000,000,000. For an application to our experiments, the preprocessing tasks such as data cleansing, user identification, session identification, path completion, and URL parsing were applied to the log files. Finally, we obtained a transaction database in the form of `<time, customer-id, product-id, shopping-step>` which the shopping-step represents one of the click-through step, the basket-placement step and the purchase step. This database contains transactions of 49597 customers on 278 products. In total, the database contains 428,510 records that consist of 781 purchase records, 5,350 basket-placement records, and 422,379 click-through records.

We set the period between 1st May 2002 and 24th May 2002 and the period between 25th May 2002 and 7th June 2002 as the training period and the test period, respectively. And then, as the target customers, we selected 130 customers who have purchased one more products in the training period and clicked one more products for the test period. Finally, the training set consists of 6,331 transaction records created by the target customers for the training period, and the test set consists of 677 click-through records created by them for the test period.
4.2. Evaluation Metrics

Recommender systems research has used a number of different measures for evaluating the quality of recommendation. Main research questions of this paper are to test whether using clustering and web mining gives better quality and more speed. Therefore, two evaluation metrics are employed in terms of quality and performance requirements.

**Quality evaluation metric.** To evaluate the quality of the recommendation set, *recall* and *precision* have been widely used in the recommender system community [Cho et al., 2004; Kim et al., 2003; Hill et al., 1995; Sarwar et al., 2000]. Recall is defined as the ratio of the number of products in both test set and recommendation set to the number of products in test set. Precision is defined as the ratio of the number of products in both test set and recommendation set to the number of products in recommendation set. Recall means how many of all the products in the actual customer purchase list are recommended correctly whereas precision means how many of the recommended products belong to actual customer purchase list. These measures are simple to compute and intuitively appealing, but they are often in conflict since increasing the size of recommendation set tends to increase recall but at the same time decrease precision [Cho et al., 2004; Kim et al., 2003; Sarwar et al., 2000]. Hence, a combination metric called *F1 metric* that gives equal weight to both recall and precision is employed for our evaluation, which is computed as follows:

\[
F1 = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}
\]  \hspace{1cm} (10)

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

4.3. EXPERIMENT RESULTS

**Experiments with neighborhood size.** The size of the neighborhood has been known to give significant impact on the recommendation quality [Cho et al., 2004; Kim et al., 2003; Sarwar et al., 2001]. To determine the sensitivity of neighborhood size, we performed an experiment in which we varied the number of neighbors and computed the corresponding F1 metric. Fig. 1 shows the experimental results. Looking into the results, we can see that the size of the neighborhood affects the quality of top-N recommendations. In general, the quality increases as we increase the number of neighbors, but, after a certain peak, the improvement gains diminish and the quality becomes worse. This reason may be that choosing too many neighbors results in too much noise for those who have high correlates. In the case, the peak is reached at 15. Hence, we used a neighborhood of size 15 as our optimal choice of neighborhood size.
Effect of Clustering Analysis. To see the clustering effect, we compare No Clustering based WebCF with Clustering based WebCF. The results are shown in Fig. 3. We can see that the quality of Clustering based WebCF is better than that of No Clustering based WebCF. However, using the clustering analysis does not give robust performance result, especially at a small number of recommended products. The recommendation with clustering based WebCF gives 106% better quality result in average.

Effect of Content-based filtering driven Bayesian Model (CBBM). To see CBBM effect, we compared the relative recommendation quality of MFP, MFR, and CBBM method in the recommendation generation. The results are shown in Fig. 3. We can see that the quality of CBBM based WebCF is better than that of others, especially at a small number of recommended products. Compared to CBBM based WebCF and MFP based WebCF, MFR based WebCF achieves an average improvement of 140% and 114%, respectively.
Performance comparison. We compared the performance of our four procedures with the metric of response time. The response time means the amount of time required to compute all the recommendations for the training set per second. Table 3 shows the response time provided by the four procedures. Looking into the results shown in Table 3, we can see that the performance of [Method 1, Method 2, Method 3] is better than that of other methods. This result tells that WebCF combined clustering analysis gives about 12 times better performance result in average, so it is very efficient in performing WebCF process.

Table 3. Performance comparison of our hybrid procedures

<table>
<thead>
<tr>
<th>Hybrid Procedures</th>
<th>Response time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method 1: Clustering + MFP</td>
<td>23</td>
</tr>
<tr>
<td>Method 2: Clustering + MFR</td>
<td>84</td>
</tr>
<tr>
<td>Method 3: Clustering + CBBM</td>
<td>116</td>
</tr>
<tr>
<td>Method 4: No Clustering + MFP</td>
<td>405</td>
</tr>
<tr>
<td>Method 5: No Clustering + MFR</td>
<td>1173.5</td>
</tr>
<tr>
<td>Method 6: No Clustering + CBBM</td>
<td>1206</td>
</tr>
</tbody>
</table>

5. CONCLUSION

We suggested two research questions and several hybrid product recommendation procedures based on clustering, Web usage mining, collaborative filtering, and content-based filtering driven a bayesian model (CBBM) to test the research questions. To implement the hybrid procedures and to test the effect of clustering analysis and CBBM, we collected Web log data from S Web retailer. Preprocessing and parsing process is performed to the web log files. Based on the experiment results, we see that CBBM gives better recommendation quality (such as precision and recall) in the WebCF recommendation procedures. In contrast, the application of clustering did not always give better recommendation quality. Recommending larger number of products is more helpful in segmented customers. In the view of recommendation process time, clustering gives outstanding performance. So clustering based WebCF is proved to be very efficient, whereas CBBM based WebCF is very effective.
While our experimental results suggest several implications, these results are based on studies limited to the particular e-commerce site. Therefore, it is required to evaluate our research questions in more detail using data sets from a variety of large e-commerce sites. And it will be an interesting research area to conduct a real marketing campaign to customers using our methodologies and to evaluate their performance.

REFERENCES


