ELSEVIER

Expert Systems with Applications 26 (2004) 427–434

Expert Systems with Applications

www.elsevier.com/locate/eswa

A personalized recommender system for the cosmetic business

Yi-Fan Wang*, Yu-Liang Chuang, Mei-Hua Hsu, Huan-Chao Keh

Department of Information Management, Chang Gung Institute of Technology, 261, Wen-Hwa 1st Rd, Kwei-Shan, Taoyuan, Taiwan, ROC

Abstract

In order to have an effective command of the relationship between customers and products, we have constructed a personalized recommender system which incorporates content-based, collaborative filtering, and data mining techniques. We have also introduced a new scoring approach to determine customers' interest scores on products. To demonstrate how our system works, we used it to analyze real cosmetic data and generate a recommender score table for sellers to refer to. After tracking its performance for 1 year, we have obtained quite impressive results.

Keywords: Data mining; Association rules; Clustering; Recommender system

1. Introduction

With the advent of customer-driven marketing, most companies treat each customer as an individual in order to maximize customer satisfaction and profitability. One method used to achieve this goal is the personalized recommender system to match products to people. Over the past several years, a number of recommender systems have been proposed for different businesses. The GroupLens recommender system helps users wade through articles in Usenet news (Konstan et al., 1997; Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994). Ringo allows users to get music recommendations online and connect with other music fans (Shardanand & Maes, 1995). Fab and others guide users to web pages, news articles, and other documents online (Balabanović & Shoham, 1997). Still more systems are concerned with videos (Hill, Stead, Rosenstein, & Furnas, 1995), movie reviews (Miller, Albert, Lam, Konstan, & Riedl, 2003), online food stores (Svensson, Laaksolahti, Höök, & Waern, 2000), music suggestion (Chen & Chen, 2001), or online book stores such as Amazon.com (Linden, Smith, & York, 2003).

Most recommender systems adopt two types of techniques, the *content-based* and *collaborative filtering* approaches, to recommend products to customers (Resnick & Varian, 1997). With content-based approach, one tries to recommend items similar to those a given user has liked in the past. It is based on a comparison between their content and a user profile. Often some weighting schemes are used which give high weights to discriminating words. Once an object has been picked, which can be shown to them and feedback of some kind elicited. If the user likes the object, the weight of the object can be added to the user profile. This process is known as relevance feedback for updating user profiles. Whereas in collaborative filtering approach, one identifies users whose tastes are similar to those of the given user and recommends items they have liked. Given a set of items, users can express their ratings of items they have tried before. The recommender can then compare the user's ratings to those of other users to find the 'most similar' users based on some criteria of similarity and then recommend items that similar users have liked in the past. Scores for unseen items are predicted based on a combination of the scores known from the nearest neighbors. Collaborative filtering has several limitations. One of the most important is the startup problem (Resnick et al., 1994). When new users come along, however, the system knows nothing about them. This is called the *new user problem* for recommender systems (Avery & Zeckhauser, 1997; Balabanović & Shoham, 1997). But, Rashid et al. (2002) let us know how to learn new user preferences in recommender systems.

To discover customers' purchase behavior, we combine the data mining techniques with these two approaches, i.e. content-based, collaborative filtering, to develop a personalized recommender system. Therefore, a business can use this system to identify new business opportunities and reduce the cost of marketing campaigns to existing customers. Two types of data mining techniques,

* Corresponding author. Tel.: +886-32118999x5880; fax: +886-3-2118866.

E-mail address: yfwang@mail.cgit.edu.tw (Y.-F. Wang).

0957-4174/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2003.10.001

the *clustering algorithm* and the *association rules algorithm*, are used in our system. Clustering algorithm is a process to group together similar customers based on the variables that you have measured, while at the same time seeking to maximize the difference between the different types of customer groups it forms. The association rules algorithm is used to discover what items its customers purchase together. We segment the customers by clustering algorithm to discover different behavior groups so that customers in same group have similar purchase behavior. For each group's customers, we use the association rules algorithm to discover their purchase behavior. Then, we score each product for each customer who may be interesting in it with the collaborative filtering approach and the content-based approach. In this application, we use the association rules algorithm and clustering algorithm to identify specific products for recommendation. Consequently, we select the most appropriate products to recommend to right customers.

2. Preliminaries

2.1. Clustering

Clustering is a kind of data mining technique for discovering interesting patterns from a given database. The main idea of clustering is that given n data pointing in a m-dimensional metric space is divided into k clusters so that all the data pointing within one cluster has a closer similarity than the data within any other cluster. The clustering algorithms can be roughly classified into partitional and hierarchical (Jain and Dubes, 1988). A recent survey on clustering can be found in Jagadish, Koudas, and Muthukrishnan (1999).

The partitional clustering algorithm divides the data set into k clusters. MacQueen's (1967) algorithm is the first and most commonly used. With this algorithm, the upper k data of the data set is used as the center points for k clusters. In this paper, since some of the data is ranked and some not ranked, the k center point or points are randomly sampled. Then every record we have is allocated to the center closest to the cluster using the Euclidean distance

$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \cdots + |x_{ip} - x_{jm}|^2}$$

where $i = (x_{i1}, x_{i2}, \ldots, x_{im})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jm})$ are two m-dimensional data.

The data so obtained is then assigned to the closest cluster, the distance between it and the center of each cluster having been considered. So the data is divided into k clusters and the positions of all the points in each cluster are averaged to obtain a new center. We then reallocate each data cluster until the stop conditions are satisfied, meaning that all data points do not change the clusters to which they belong to consecutive times.

The hierarchical clustering algorithm takes the original data clusters and changes them into a multilayer hierarchical structure. An algorithm for hierarchical clustering starts with disjoint sets of clusters and places each input data point in an individual cluster. Pairs of items or clusters are then successively merged and reduced. At each step, the pair of clusters merged are the ones between which the distance is the minimum. The widely used measures for the distance between two clusters are as follows (m_i is the center for cluster C_i and n_i is the number of points in C_i)

$$d_{\mathrm{center}}(C_i, C_j) = |m_i - m_j|$$

$$d_{\min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} |p - p|$$

For example, at each step, with d_{center} as the distance measure, the pair of clusters whose centers are the closest are merged. And with $d_{\min}$, the pair of clusters merged are the ones containing the closest pair of points. All the above distance measures have a minimum variance flavor and usually yield the same results if the clusters are compact and well-separated.

2.2. Association rules algorithm: support, confidence, and lift

The association rules algorithm is mainly used to find out the relationships between items or features that occur synchronously in the database, i.e. to learn rules, there may be groceries purchased during a trip to the shopping center. For instance, 80% of the people who buy milk also buy bread as well. With such information, a decision maker can implement new strategies such as changing the positions of relevant counters or organizing related promotions. Therefore, the main purpose of implementing the association rules algorithm is to find out synchronous relationships by analyzing the random data and to use this data as reference during decision making. The association rule is defined as follows:

Make $I = \{i_1, i_2, \ldots, i_m\}$ as the item set, in which each item represents a specific commodity. D stands for a trading database in which each transaction T represents an item set. That is, $T \subseteq I$. Each item set is a non-empty sub-item set of I, and the only identifying code is TID. Each item set $X \subset I$ has a measure standard—Support, to evaluate the statistical importance of D. Support(X, D) denotes the rate of merchandising X in transaction D.

The association rule is $X \rightarrow Y$, in which $X, Y \subset I$, and $X \cap Y = \emptyset$. The rule means that if X is purchased, Y can be bought at the same time. Each rule has a measuring standard called Confidence; i.e. Confidence$(X \rightarrow Y)$ = Support$(X \cup Y, D)$/Support(X, D). In this case, Confidence$(X \rightarrow Y)$ denotes that if the merchandise including X, the chance of buying Y is relatively high. Two steps are needed to determine the association rule. The first step is to detect the large item set and the second to generate association rules, using the large item set. Such rules must satisfy two conditions:

1. Support$(X \cup Y, D) \geq$ Minsup
2. Confidence$(X \rightarrow Y) \geq$ Minconf

The Minsup and Minconf are both set by the users. In general, the numbers of the transactions that comprise X is called the support of X, denoted by sx. Make Minsup the minimum value of support. If the support of X meets the condition, $sx \geq$ Minsup, X is the large item set. As for the exploration of the association rules, many researchers usually use the Apriori algorithm (Agrawal, Imielinski, & Swami, 1993).

The judgment standard is called the lift. When all of the data employed can be classified, the simplest way to compare them is to use the lift judgment. In order to focus on a single target group from all transactions, we can use rule $X \rightarrow Y$ and find Y, applying the lift judgment. Y would be less likely to be selected if we applied any other model than the lift. For example, if we use rule $X \rightarrow Y$ to determine a customer's probability of purchasing Y, Support(Y), the product purchase multiple for Y, which is the lift target, is defined as:

$$\text{Lift} = \text{Confidence}(X \rightarrow Y)/\text{Support}(Y)$$

2.3. Recommender system

Many different recommender systems have been proposed in a wide variety of fields. In general, there are two types of methods: content-based and collaborative filtering. With content-based approach, Gauch, Gauch, Bouix, and Zhu (1999) proposed a real time video scene detection and classification. Cheng and Yang (1999) proposed a new content-based access method for video databases. Mooney and Roy (2000) proposed a content-based book recommending using learning for text categorization. Kim and Choi (2002) proposed a content-based video transcoding in compressed domain. Fleischman and Hovy (2003) proposed a natural language processing approach for recommendation without user preferences. Carenini, Smith, and Poole (2003) proposed a set of techniques to intelligently select what information to elicit from user.

Collaborative filtering is another technique used to construct recommender systems. At first, it was used to handle large volumes of e-mail (Goldberg, Nichols, Oki, & Terry, 1992). Chalmers, Rodden, and Brodbeck (1998) proposed a path-based method of collaborative filtering that allows easy access Web-based information. Schafer, Konstan, and Riedl (1999) proposed a recommender system in E-commerce. Herlocker, Konstan, and Riedl (2000) explained collaborative filtering recommendations. McDonald and Science (2000) proposed a flexible recommendation system and architecture. Kohrs and Merialdo (2001) proposed an application of collaborative filtering for user-adapted Websites. Kuo and Chen (2001) proposed a personalization technology application to Internet content provider. McNee et al. (2002) proposed the recommending of citations for research papers. Schafer, Konstan, and Riedl (2002) proposed a user-controlled integration of diverse recommendations. Lee, Kim, and Choi (2003) proposed a Web-based collaborative filtering system.

Some recommender systems combine two or more techniques, such as the decision tree and association rules. Kautz, Selman, and Shah's (1997) Referral Web combines social networks and collaborative filtering. Balabanović and Shoham (1997) presents Fab, which uses content-based and collaborative approaches. Lee, Kim, and Rhee's (2001) Web Personalization Expert combines collaborative filtering and the association rules algorithm. Sarwar, Kerypis, Konstan, and Riedl (2001) put forward an item-based collaborative filtering method. Cho, Kim, and Kim (2002) proposed a personalized recommender system which employs decision tree induction.

Also there are some specific algorithms that have been proposed. Aggarwal, Wolf, Wu, and Yu (1999) introduced a graph–theoretic approach. Kohrs and Merialdo (1999) improved collaborative filtering with multimedia indexing techniques to create user-adapting Web sites. Soboroff and Nicholas's (2000) generalized vector space model of information retrieval represents a document by a vector of its similarities to all other documents. Yu, Xu, Ester, and Kriegel (2001) proposed selecting relevant instances for efficient and accurate collaborative filtering. Domingos and Richardson (2001) proposed to model the customer's *network value*: the expected profit from sales to other customers she may influence to buy, the customers those may influence, and so on recursively. Canny's (2002) new data filtering method better protects individual privacy. Cöster and Svensson (2002) suggested an inverted file search algorithm for collaborative filtering. Fiore, Tiernan, and Smith (2002) evaluated behavioral descriptors obtained from an analysis of a large collection of Usenet newsgroup messages. Huang, Chung, Ong, and Chen (2002) proposed a graph-based recommender system for digital library. Lee, Liu, and Lu (2002) offered an intelligent agent-based recommender system for the Internet commerce as well as other businesses. Terveen, McMackin, Amento, and Hill (2002) let the user define preferences relative to their personal history.

3. Offering a recommender system

Our personalized recommender system consists of three major parts: customers clustering, products association generating, and scoring. For our present purposes, we will discuss only the scoring part, as data mining functions, i.e. clustering algorithm and association rules algorithm, have been very well addressed by scholars.

Since customer segmentation is used to discover the characteristics of customers, it is at the core of all customer centered data mining models. We use clustering algorithm to segment our customers into different groups so that each group's customers exhibit similar purchase behavior. After that, we use the association rules algorithm to analyze each segment in order to find out their purchase behavior.

$$\mathrm{Score}(x,B) = \begin{cases} 0, & |B| > 0 \\ \max(\mathrm{Score}(x,B), \mathrm{NRS}(B)), & |B| = 0 \wedge |\mathbf{A}| = 0 \\ \max\left(\mathrm{Score}(x,B), \mathrm{NRS}(B) \times \dfrac{\mathrm{Conf}[\mathbf{A} \to B]}{1 - \mathrm{Conf}[\mathbf{A} \to B]}\right), & |B| = 0 \wedge |\mathbf{A}| > 0 \end{cases} \tag{1}$$

3.1. Content-based setting: The initial score

Different recommender systems in different businesses are affected by different factors. However, in our application, customers are characterized by their spending on different product groups, and to serve as our personalized factor, we will borrow from Baragoin et al. (2001) his Normalized Relative Spend (NRS).

We use Relative Spend (RS) as a measure of customer expenditure at product level. The RS attempts to account for the significance of the purchase compared to the overall sales of the product. For example, if a customer purchases during their first shopping trip, four types of products which cost 5, 20, 25 and 50 dollars, respectively, the total amount being 100 dollars. The RS for each product item bought during the shopping trip is 0.05, 0.2, 0.25, 0.5. If items 1 and 2 were from the same group, and items 3 and 4 from another, the NRS for this product types would be changed to 0.25 and 0.75. NRS is generated by dividing the purchasing value for one product from a single transaction by the average purchasing value for that product from all transactions. For example, if the purchasing value for Product A in this transaction is 0.5 and if the average purchasing value for Product A in all transactions is 0.5, then the NRS for Product A in this transaction is 1. If the average value for all transactions is 0.1, the NRS is 5.0. This value indicates that an ordinary person is willing to spend five times the weight on this product during this transaction than all other transactions.

We use the content-based method to first set a basic score for each product. Using the NRS value that we defined above, we calculate the average NRS for each product in the cluster and use the average NRS to set the initial value for the recommender score table. If a certain type of products has a high NRS average, this means the cluster is more likely to purchase this type of product, and we should, therefore, recommend such to this cluster of customers.

3.2. Collaborative setting the recommender score

Next we use the association rules algorithm to determine rule $\mathbf{A} \to B$, where $\mathbf{A} = A_1, A_2, \ldots, A_n$, $n \geq 1$, and then assign more recommender scores to existing related products. Our method as expressed in the formula below

where x is a customer ID, NRS(B) is the NRS for product B from customer x's cluster. The $|\mathbf{A}|$ is the number of previous transactions including product $\mathbf{A}$ and $|B|$ is the number of previous transactions including product B. The Conf[$\mathbf{A} \to B$] is the confidence value of the existing association rule $\mathbf{A} \to B$ for this cluster.

When $|B| > 0$ means the customer has purchased product B, the recommender score for product B is 0. When $|B| = 0$ and $|\mathbf{A}| = 0$, it means the customer has not purchased product $\mathbf{A}$ and B, the recommender score for product B is the higher value, NRS(B) compared with Score(x,B) in this cluster. When $|B| = 0$ and $|\mathbf{A}| > 0$, it means the customer has bought product $\mathbf{A}$ but not product B, the recommender score for product B is the higher value

$$\mathrm{NRS}(B) \times \frac{\mathrm{Conf}[\mathbf{A} \to B]}{1 - \mathrm{Conf}[\mathbf{A} \to B]}$$

compared with Score(x,B).

Based on confidence characteristics, the confidence value smaller than 50 means there is a negative correlation, 50

Table 1
The NRS of cluster one

Category	Product item number	NRS
Color	1001	1.306
	1002	1.463
	1003	1.445
	1004	1.379
	⋮	⋮
Skin	3001	3.417
	⋮	⋮
	3101	3.018
	⋮	⋮
	3201	3.687
	3202	3.005
	⋮	⋮
	3301	3.712
	3302	3.573
	⋮	⋮
Other	2001	0.221

Number of Transactions = 33445
Number of Items = 36 (18 large)
Items per Transaction : Maximum = 22, Average = 3.15679
Support: Minimum = 1673 = 5%, Maximum = 33445 = 100%
Confidence: Minimum = 10%, Maximum = 100%
-------------------------------------- Rules --------------------------------------

Support	Conf	Lift	Kind	Product A	→	Product B
13.0513	69.42	1.36	+	[3301]	→	[3302]
10.4231	67.24	1.30	+	[3502]	→	[3302]
11.3231	65.98	1.27	+	[3801]	→	[3302]
6.2550	**64.93**	**1.70**	**+**	**[3202]**	**→**	**[3301]**
6.1325	64.05	1.21	+	[3202]	→	[3302]
6.9846	62.60	1.17	+	[3001]	→	[3302]
9.2032	61.04	1.13	+	[3103]	→	[3302]
9.2032	61.04	1.13	+	[3103]	→	[3302]
7.6992	57.63	1.04	+	[3901]	→	[3302]
5.0860	57.29	1.03	+	[3602]	→	[3302]
5.9022	56.00	1.36	+	[3001]	→	[3301]
13.0513	55.93	1.36	+	[3302]	→	[3301]
7.6245	54.00	1.29	+	[3103]	→	[3301]
6.4614	53.63	0.93	+	[3902]	→	[3302]
7.8517	51.88	1.21	+	[3801]	→	[3301]
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Fig. 1. Association rules 2001 transaction data with cluster one.

indicates no correlation, and greater than 50 shows a positive correlation.

4. Experiments

We used the cosmetic data from January 2001 to December 2001 as training examples and the data from January 2002 to December 2002 as testing examples. There are 41 types of the cosmetic products that fall into three categories: color, skin and other. Based on our clustering results, we divide our customers into 10 clusters, so there are a total of 10 NRS tables. One 41×1 column is set-up for new customers before they make any purchases. For different clusters, we use the association rules algorithm to calculate the cluster customers' transaction records. By combining these tables with the association rule, we created: $\mathbf{A} \rightarrow B$, each customer previous transaction record and applying the rule. Using Eq. (1) to adjust the product recommender score, we can obtain the recommender scores for customers. By ranking the scores, we can find the most appropriate products for companies to recommend to customers.

4.1. Experiment settings

We use content-based method to set a basic score for each product. Using the NRS, we defined in Section 3.1, we can calculate the average NRS for each product in the cluster. The initial value for the recommender score table is set as the average NRS for the product types in each cluster. An example of cluster one's NRS is shown in Table 1.

Using the results of association rule $\mathbf{A} \rightarrow B$, we can assign more recommender scores to existing associated products. First of all, we can categorize the customers with the clustering algorithm. We then apply the association rules algorithm to the 2001 transaction data, as shown in Fig. 1.

According to the rules in Fig. 1, we can find that those customers in cluster one who purchased **A**, but not B, from the transactions records, and then produce a recommender score table with Eq. (1) to adjust recommender scores.

For example, based on the customer in cluster one and the rule [3202] → [3301], the recommender score is

$$\mathrm{NRS}(B) \times \frac{\mathrm{Conf}[\mathbf{A} \rightarrow B]}{1 - \mathrm{Conf}[\mathbf{A} \rightarrow B]} = 3.712 \times \frac{0.6493}{1 - 0.6493}$$

$$= 6.8725$$

Thus, we can see the business potential of the customer purchasing product [3301].

The recommender score table is concluded and shown as Table 2.

4.2. Evaluation results

Table 2 was provided to people in the cosmetic business so that they could use our recommendations to sell their products. One year later, we used the association rules algorithm to run again on 2002 transaction data, as shown in Fig. 2.

As the gray areas in Figs. 1 and 2 show the confidence of this rule in 2001 was only 61.04, but in 2002 it rose to 66.34,

Table 2
Adjusted recommender score table

Customer ID	Recommender product	Recommender score
A0325672	3302	8.6578
A1473226	3302	8.1111
A1903564	3302	7.4592
A2209639	3302	7.3364
⋮	⋮	⋮
A3345027	3301	6.8725
A3588735	3301	6.9787
⋮	⋮	⋮

Number of Transactions = 88168
Number of Items = 41 (15 large)
Items per Transaction : Maximum = 19, Average = 3.04449
Support: Minimum = 4409 = 5%, Maximum = 88168 = 100%
Confidence: Minimum = 10%, Maximum = 100%
---------------------------------- Rules ----------------------------------

Support	Conf	Lift	Kind	Product A	→	Product B
12.0248	76.35	1.33	+	[3301]	→	[3302]
14.5438	76.16	1.32	+	[3304]	→	[3302]
9.5715	75.26	2.6	+	[3902]	→	[3901]
8.3477	74	1.27	+	[3502]	→	[3302]
5.6041	73.64	1.26	+	[3801]	→	[3302]
6.1757	68.01	1.13	+	[3001]	→	[3302]
8.2853	66.34	1.09	+	[3103]	→	[3302]
7.2816	66.14	1.09	+	[3102]	→	[3302]
9.5715	64.99	2.6	+	[3901]	→	[3902]
:	:	:	:	:	:	:
:	:	:	:	:	:	:
8.3477	39.65	1.27	-	[3302]	→	[3502]
8.2853	39.5	1.09	-	[3302]	→	[3103]
7.2816	**37.14**	**1.09**	-	**[3302]**	→	**[3102]**
7.1092	36.73	0.97	-	[3302]	→	[3902]
:	:	:	:	:	:	:
:	:	:	:	:	:	:

Fig. 2. Association rules of 2002 transaction data with cluster one.

so when a customer purchased product [3103], the probability of purchasing product [3302] rose by 8.68%. Moreover, we can look at rule [3202] → [1001], which was not generated by the association rules algorithm. Although these two products do not belong to the same product type, the same tracking survey can still be done, as shown in Table 3.

We can clearly see that the same customers' consumption behavior did not change significantly a year later. Those who had purchased product [3202] still did not purchase product [1001].

We can also find products that belong to the same category like [3202] → [3001] for which a rule cannot be generated with the association rules algorithm.Product [3202] did not have any association with either [1001] or [3001], but products [3202] and [3001] belong to the same product category. Interestingly, as shown in Table 4, customers who purchased product [3001] still increased 176. Therefore, setting up initial NRS values for products of the same category was useful.

Nevertheless, as rules change, timely adjustments to our recommender score table should be made so as to achieve better recommendation effects and higher sales performance. For example, as the bold-faced entry in Fig. 2 shows, the rule did not appear in 2001, but in 2002. Therefore, it is necessary to reevaluate the recommender score table every year. If a company releases new products, the database format must be updated; otherwise, recommenders for them would not be available. Even a data mining may have to be done at regular intervals in order for the seller to stay informed of customers' purchase patterns.

5. Conclusion

Data mining, which offers a shortcut to company performance, is something that every company urgently wishes to introduce to the MIS department. But data mining is not about just dumping transaction and customer data into data mining tools for calculation. To obtain useful results, rather than unintelligible data, we have to understand the characteristics of the individual industry, and then derive appropriate data mining columns. It is essential to first have domain know-how and data mining targets so that truly useful information can be produced with the data mining tools.

Determining marketing strategy requires consideration of numerous real factors; just following the data produced by the rule will not raise company performance. The right answers cannot be obtained just by complex data mining calculations. Of course, marketing experience and industry know-how are also required to achieve marketing targets.

In this paper, we divide all products into three types. If we employed demographic concepts in dividing products

Table 3
Tracking survey for [3202] → [1001]

	Customer who purchased [3202]	Customer who did not purchase [1001]
2001	3085	3030
2002	3085	3028

Table 4
Tracking survey for [3202] → [3001]

	Customer who purchased [3202]	Customer who did not purchase [3001]
2001	3085	2004
2002	3085	1828

into more types, we could gain better results. Besides, we only consider the confidence value in our scoring function; in future it might be desirable to take the lift value into account.

Sequential pattern analysis could also be incorporated into the association rules algorithm that we currently use. By adding sequential pattern analysis, we can find time related attributes and do analysis on a seasonal basis. Data can be divided into seasons and then data mining can be performed to find even better recommender systems. When we analyze the transaction data for seasonal business for the entire year, we can only find out general purchase behavior. If the data can be analyzed on a seasonal basis, we should be able to obtain a more comprehensive picture of customer purchase behavior and provide more accurate product recommendation. Instead of just recommending the right product to the right person, we hope that our system can recommend the right product to the right person at the right time.

References

Aggarwal, C. C., Wolf, J. L., Wu, K.-L., & Yu, P. S. (1999). Horting hatches an egg: a new graph–theoretic approach to collaborative filtering. *Proceedings of the KDD'99, San Diego, CA*, 201–212.

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *Proceedings of the SIGMOD'93, Washington, DC*, 207–216.

Avery, C., & Zeckhauser, R. (1997). Recommender systems for evaluating computer messages. *Communications of the ACM*, *40*(3), 88–89.

Balabanović, M., & Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, *40*(3), 66–72.

Baragoin, C., Andersen, C. M., Bayerl, S., Bent, G., Lee, J., & Schommer, C. (2001). *Mining your own business in retail*. *IBM redbooks*, pp. 54–55.

Canny, J. (2002). Collaborative filtering with privacy via factor analysis. *Proceedings of the SIGIR'02, Tampere, Finland*, 238–245.

Carenini, G., Smith, J., & Poole, D. (2003). Towards more conversational and collaborative recommender systems. *Proceedings of the IUI'03, Miami, FL*, 12–18.

Chalmers, M., Rodden, K., & Brodbeck, D. (1998). The order of things: activity-centered information access. *Computer Networks and ISDN Systems*, *30*, 359–367.

Chen, H.-C., & Chen, A. L. P. (2001). A music recommendation system based on music data grouping and user interests. *Proceedings of the CIKM'01, Atlanta, Georgia*, 231–238.

Cheng, P. J., & Yang, W. P. (1999). A new content-based access method for video databases. *Information Sciences*, *118*, 37–73.

Cho, Y. H., Kim, J. K., & Kim, S. H. (2002). A personalized recommender system based on web usage mining and decision tree induction. *Expert Systems with Applications*, *23*, 329–342.

Cöster, R., & Svensson, M. (2002). Inverted file search algorithms for collaborative filtering. *Proceedings of the SIGIR'02, Tampere, Finland*, 246–252.

Domingos, P., & Richardson, M. (2001). Mining the network value of customers. *Proceedings of the KDD'01, San Francisco, CA*, 57–66.

Fiore, A. T., Tiernan, S. L., & Smith, M. A. (2002). Observed behavior and perceived value of authors in Usenet newsgroups: bridging the gap. *Proceedings of the CHI'02, Minneapolis, Minnesota*, 323–330.

Fleischman, M., & Hovy, E. (2003). Recommendations without user preferences: a natural language processing approach. *Proceedings of the IUI'03, Miami, FL*, 242–244.

Gauch, J. M., Gauch, S., Bouix, S., & Zhu, X. (1999). Real time video scene detection and classification. *Information Processing and Management*, *35*, 401–420.

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, *35*(12), 61–70.

Herlocker, J. L., Konstan, J. A., & Riedl, J. (2000). Explaining collaborative filtering recommendations. *Proceedings of the CSCW'00, Philadelphia, PA*, 241–250.

Hill, W., Stead, L., Rosenstein, M., & Furnas, G. W. (1995). Recommending and evaluating choices in a virtual community of use. *Proceedings of the CHI'95, Denver, CO*, 194–201.

Huang, Z., Chung, W., Ong, T.-H., & Chen, H. C. (2002). A graph-based recommender system for digital library. *Proceedings of the JCDL'02, Portland, Oregon*, 65–73.

Jagadish, H. V., Koudas, N., & Muthukrishnan, S. (1999). Mining deviants in a time series database. *Proceedings of the VLDB'99, Edinburgh, UK*, 102–113.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Englewood Cliffs, NJ: Prentice-Hall.

Kautz, H., Selman, B., & Shah, M. (1997). Referral web: combining social networks and collaborative filtering. *Communications of the ACM*, *40*(3), 63–65.

Kim, T. Y., & Choi, J. S. (2002). Content-based video transcoding in compressed domain. *Signal Processing: Image Communication*, *17*, 497–507.

Kohrs, A., & Merialdo, B. (1999). Improving collaborative filtering with multimedia indexing techniques to create user-adapting web sites. *Proceedings of the Multimedia'99, Orlando, FL*, 27–36.

Kohrs, A., & Merialdo, B. (2001). Creating user-adapted websites by the use of collaborative filtering. *Interacting with Computers*, *13*, 695–716.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., & Riedl, J. (1997). GroupLens: applying collaborative filtering to Usenet news. *Communications of the ACM*, *40*(3), 77–87.

Kuo, Y.-F., & Chen, L.-S. (2001). Personalization technology application to internet content provider. *Expert Systems with Applications*, *21*, 203–215.

Lee, D.-S., Kim, G.-Y., & Choi, H.-I. (2003). A web-based collaborative filtering system. *Pattern Recognition*, *36*, 519–526.

Lee, C.-H., Kim, Y.-H., & Rhee, P.-K. (2001). Web personalization expert with combining collaborative filtering and association rule mining technique. *Expert Systems with Applications*, *21*, 131–137.

Lee, W.-P., Liu, C.-H., & Lu, C.-C. (2002). Intelligent agent-based systems for personalized recommendations in internet commerce. *Expert Systems with Applications*, *22*, 275–284.

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing: Industry Report*, http://dsonline.computer.org/0301/d/w1lind.htm.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observation. *Proceedings of the Fifth Berkeley Symposium on Mathematics, Statistics and Probability, CA*, *1*, 281–297.

McDonald, D. W., & Science, M. S. (2000). Expertise recommender: a flexible recommendation system and architecture. *Proceedings of the CSCW'00, Philadephia, PA*, 231–240.

McNee, S. M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S. K., Rashid, A. M., Konstan, J. A., & Riedl, J. (2002). On the recommending of citations for research papers. *Proceedings of the CSCW'02, New Orleans, Louisiana*, 116–125.

Miller, B. N., Albert, I., Lam, S. K., Konstan, J. A., & Riedl, J. (2003). MovieLens unplugged: experiences with an occasionally connected recommender system. *Proceedings of the IUI'03, Miami, FL*, 263–266.

Mooney, R. J., & Roy, L. (2000). Content-based book recommending using learning for text categorization. *Proceedings of the Digital Libraries, San Antonio, TX*, 195–204.

Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., & Riedl, J. (2002). Getting to know you: learning new user preferences in recommender systems. *Proceedings of the IUI'02, San Francisco, CA*, 127–134.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens: an open architecture for collaborative filtering of Netnews. *Proceedings of the CSCW'94, Chapel Hill, NC*, 175–186.

Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, *40*(3), 56–60.

Sarwar, B., Kerypis, G., Konstan, J. A., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the WWW10, Hong Kong*, 285–295.

Schafer, J. B., Konstan, J. A., & Riedl, J. (1999). Recommender systems in e-commerce. *Proceedings of the E-COMMERCE 99, Denver, CO*, 158–166.

Schafer, J. B., Konstan, J. A., & Riedl, J. (2002). Meta-recommendation systems: user-controlled integration of diverse recommendations. *Proceedings of the CIKM'02, Mclean, Virginia*, 43–51.

Shardanand, U., & Maes, P. (1995). Social information filtering: algorithms for automating word of mouth. *Proceedings of the CHI'95, Denver, CO*, 210–217.

Soboroff, I., & Nicholas, C. (2000). Collaborative filtering and the generalized vector space model. *Proceedings of the SIGIR'00, Athens, Greece*, 351–353.

Svensson, M., Laaksolahti, J., Höök, K., & Waern, A. (2000). A recipe based on-line food store. *Proceedings of the IUI 2000, New Orleans, LA*, 260–263.

Terveen, L., McMackin, J., Amento, B., & Hill, W. (2002). Specifying preferences based on user history. *Proceedings of the CHI'02, Minneapolis, Minnesota*, 315–322.

Yu, K., Xu, X., Ester, M., & Kriegel, H.-P. (2001). Selecting relevant instances for efficient and accurate collaborative filtering. *Proceedings of the CIKM'01, Atlanta, Georgia*, 239–246.