Intelligent Data Analysis 17 (2013) 899–916
DOI 10.3233/IDA-130611
IOS Press

Incorporating frequency, recency and profit in sequential pattern based recommender systems

Cheng-Lung Huang[a,*], Mu-Chen Chen[b], Wen-Chen Huang[a] and Sheng-Huang Huang[a]
[a]*Department of Information Management, National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan*
[b]*Institute of Traffic and Transportation, National Chiao Tung University, Taipei, Taiwan*

Abstract. Customers usually change their purchase interests in the short product life cycle of the e-commerce environment. Therefore, recent transaction patterns should have a greater effect on the customer preferences. From the seller's point of view, an e-commerce recommender system should focus on the profit of recommendation. This study proposes a new sequential pattern mining algorithm that incorporates the concepts of frequency, recency, and profit to discover frequent, recent, and profitable sequential patterns, called FRP-sequences. Based on the discovered sequential patterns, this study develops a collaborative recommender system to improve recommendation accuracy for customers and the profit of recommendation from the seller's perspective.
The proposed recommender system clusters customers, discovers FRP-sequences for each cluster, and then recommends items to the target customers based on their frequent, recent, and profitable FRP-sequences. In the stage of discovering FRP-sequences, the transaction patterns near the current time period and profitable items are weighted more heavily to improve profit. This study uses a public food mart database to determine the performance of the proposed approach, and compares it with traditional recommendation models. The proposed system performs better than traditional recommendation models in both recommendation accuracy and profit.

Keywords: Recommender systems, collaborative filtering, sequential patterns, profit mining, e-commerce

1. Introduction

Sequential pattern mining, which was introduced by Agrawal and Srikant [2], identifies frequently occurring ordered sequences in a sequence database [12]. A sequence database consists of a series of sequences that consist of several transactions sorted in a time ascending order. Sequential patterns suggest that a consumer who buys a new product in the current time period is likely to buy another product in the next time period. Sequential pattern mining has been applied to several fields, including sales promotions, targeted marketing, production processes, web access pattern analysis, network intrusion detection, and DNA sequence analysis. In e-commerce, sequential patterns are useful for personalizing

*Corresponding author: Cheng-Lung Huang, Department of Information Management, National Kaohsiung First University of Science and Technology, 2, Juoyue Rd., Nantz District, Kaohsiung 811, Taiwan. Tel.: +886 7601 1000 Ext 4127; Fax: +886 7601 1042; E-mail: clhuang@nkfust.edu.tw.

product recommendations and product related advertisements to improve customer satisfaction [13]. Researchers have developed recommender systems to provide personalized recommendation for product, advertisement, or content [1,20]. These software systems have been applied to many areas, including e-commerce, news, advertisement, document management, and e-learning. In e-commerce applications, recommender systems can turn browsers into buyers by providing personalized shopping information that interests the customer. The recommender systems can also be connected to the customer's activities using the push, pull, or passive e-commerce application technologies [21,22].

Sequential pattern mining identifies frequent sequential patterns whose support count exceeds a predefined minimal threshold. The frequent sequential patterns discovered by sequential pattern mining represent customers' purchase interests. In traditional sequential pattern mining, which only considers the sequence frequency, a sequential pattern with high support is more useful than one with low support. However, the design of a sequential pattern-based recommender system should account for two additional factors: Recency and profit.

(1) Recency: In an environment in which the customer gradually changes purchase patterns, transaction data close to the current temporal period is usually more important than that temporally far from the current period [13]. Therefore, a recommender system should provide higher importance for the patterns generated in more recent time periods to improve recommendation accuracy based on customer transaction data.
(2) Profit: Traditional recommender systems make recommendations based mainly on purchase probability, assuming that items with high purchase probabilities are more likely to satisfy customer needs. In addition to the purchase probability, the profitability of recommendation from the seller's viewpoint is another critical issue that a recommender system should address. Extracting highly profitable patterns or rules from customers' transactional data is called profit mining [9,24–26]. This study attempts to improve the seller's profit.

This study extends the traditional sequential patterns, which only consider frequency or support rate, by incorporating frequency, recency, and profit. The sequential patterns based on frequency, recency, and profit are called FRP-sequences in this study. These FRP-sequences are essential for designing a recommender system. Previous researchers have contributed to this area. For example, frequency and profit were combined with association rule mining in [6,16,23,28]; a recency model with time decay in sequential pattern was developed in [13]; and recency, frequency, and monetary (RFM) were considered in sequential pattern mining in [8]. Until now, no researches have discussed a collaborative system that combines sequential purchase patterns with frequency, recency, and profit. This study proposes a FRP-sequence based collaborative system to improve recommendation profit and accuracy of the e-commerce recommender system.

The rest of this paper is organized as follows. Section 2 introduces related research. Section 3 describes the proposed FRP pattern mining. Section 4 introduces the framework of the proposed FRP-sequence based collaborative recommender system. Section 5 examines the performance of the proposed system using a public retail dataset. Section 6 provides conclusions.

2. Related research

2.1. Sequential pattern mining

Agrawal and Srikant [2] were the first to introduce sequential pattern mining. A sequence consists of an ordered list of events, and each event consists of a set of items. Let $I = \{item_1, item_2, \ldots, item_m\}$

be an itemset. An event E is a subset of an itemset, i.e., $E \subseteq I$. A sequence is denoted as $S =< E_1, E_2, \ldots, E_i, \ldots, E_j, \ldots, E_k >$, where event E_i occurs before E_j. The length of a sequence is the number of itemsets in the sequence. A sequence of length k is called a k-sequence.

An itemset with minimum support is called a large itemset. The support for an itemset is defined as the fraction of customers who bought items in this itemset in a single transaction. Given a set of customer sequences for all customers, the support for a sequence S is the number of customers whose sequence contains S, denoted as *Support*(S). Sequential pattern mining finds all frequent sequences that fall within a user-specified minimum support threshold. A large sequence contains a series of large itemsets, i.e., a large k-sequence has k large itemsets.

AprioriAll [2] generates sequences that may be frequent, called candidate sequences. AprioriAll then scans the sequential database to check the support of each candidate to determine frequent sequential pattern based on the minimal support. This process includes the following steps: (i) Convert the original transaction database into a customer sequence database. (ii) Find the set of all large itemsets with minimum support. Each itemset in a large sequence must have minimum support. The support is the fraction of customers who bought the itemset in any of their transactions. (iii) Replace each transaction by the set of all large itemsets contained in a customer sequence. (iv) Use the set of large itemsets to find the desired large sequences, because any large sequence must be a list of large itemsets. (v) Find the maximal sequences among the set of large sequences.

This study builds on AprioriAll by incorporating frequency, recency, and profit to give a higher importance to the profitable patterns generated in more recent periods.

2.2. Recency and monetary considerations in sequential pattern mining

Chen and Hu [7] proposed a constraint-based sequential pattern mining that considers the recency and compactness of a pattern. Because a past purchasing pattern may not recur or the product items included in the sequential pattern may have low monetary values, Chen et al. [8] incorporated the concepts of recency, frequency, and monetary (RFM) introduced by [5] into sequential pattern mining. They first investigated the recency, frequency and monetary in sequential patterns. They defined recency as the period since a customer's last purchase; *frequency* as the number of purchases made within a certain period; and *monetary* as the amount of money that a customer spent during a certain period. They used upper and lower thresholds for recency, monetary, and frequency to constrain pattern generation. The frequency of a pattern represents the percentage of sequences that satisfy both recency and monetary constraints. A pattern is an RFM pattern if it satisfies the recency and monetary constraints and its frequency falls within the minimum and maximum support thresholds.

The proposed approach uses the profit of the product instead of the monetary, defining a profit threshold like the monetary threshold in Chen's approach [8] to filter the sequence generation. In Chen's approach, recency serves as a filter to ensure that the last transaction time of a selected sequence is between the predefined time threshold of earliest and latest times from a given starting time. For example, the last transaction of a selected pattern must occur between 200 and 270 days from a starting date. Unlike Chen's approach, this study deals with recency by weighting the frequency according to the sequence's last transaction time. In other words, the importance of a transaction sequence is proportional to its last transaction time. The underlying concept of this method is that since customers gradually change their purchase patterns (interests), transactional patterns close to the current temporal period are usually more important than those temporally far from the current period.

2.3. Profitable recommendation in recommender systems

Recommender systems often use a specific type of information filtering technique to recommend information items that are likely to interest the user. Examples of these information items include blogs, commercial products, movies, music, news, and photographs. Recommender systems make recommendations using three basic steps: acquiring preferences from the customer's input data, computing recommendations using proper techniques, and presenting the recommendations to customers [27]. Recommendation techniques include the content-based filtering approach [14], collaborative filtering approach [19] and hybrid-based recommender systems [4].

Traditional recommender systems make recommendations based mainly on purchase probability, assuming that items with high purchase probabilities are more likely to satisfy customer needs. To satisfy customer purchase preferences based on purchase frequency, a recommender system should also provide recommendations with a higher profit margin [6]. The concept of profit mining attempts to find the high profitable patterns or rules [9,24–26]. Some researchers use the term "utility mining" by incorporating the frequency and profit in association rule mining [16,23,28]. Though previous research [13] proposed a sequential pattern based recommender system, they did not consider profit mining. Unlike previous methods, this study designs a collaborative recommender system that simultaneously considers the frequency, recency, and profit in sequential pattern mining.

3. Proposed FRP-AprioriAll algorithm

To design a sequential pattern based recommender system, this study presents an enhanced FRP-AprioriAll by incorporating "recency" and "profit" with traditional frequency based sequential pattern mining. The main differences between FRP-AprioriAll and the traditional AprioriAll are the calculation of the support count, weight of recency for frequency, and the minimum profit filter. Instead of only using the frequency information, FRP-AprioriAll weights the frequency of a candidate sequence according to its transactional time, and then filters the candidate sequences by minimum support and minimum profit. The number of generated sequential patterns by FRP-AprioriAll may be less than that of the traditional sequential patterns generated by the AprioriAll because FRP-AprioriAll applies an additional profit filter. However, the sequences generated by FRP-AprioriAll are more recent and profitable than those generated by traditional sequential pattern mining.

The main difference between FRP-ArioriAll from the traditional AprioriAll is as follows. AprioriAll defines the support of an itemset as the frequency (fraction of customers who bought items in the itemset), while FRP-AprioriAll defines the support of an itemset as a combination of the frequency and recency (abbreviated as *FR*-support) of this itemset. Additionally, a large sequence with frequency, recency, and profit (called a FRP-sequence) must satisfy both the minimum *FR*-support and the predefined minimum profit. The proposed model represents a large k-sequence with frequency, recency, and profit as a k-FRP-sequence.

Like AprioriAll, FRP-AprioriAll splits mining *FRP*-sequential patterns into two phases. (i) This phase finds all frequent, recent and profitable itemsets. This phase first applies the profit threshold to each itemset to find profitable itemsets, and then calculates the *FR*-support count for each itemset to find frequent and recent itemsets. Since each itemset in a k-FRP-sequence must have a minimum threshold of profit and *FR*-support count, all frequent, recent, and profitable itemsets (also known as 1-FRP-sequences) must be found in advance. (ii) This phase finds large k-FRP-sequences, and the maximal sequences among the set of large sequences. Based on the frequent, recent, and profitable itemsets obtained in the

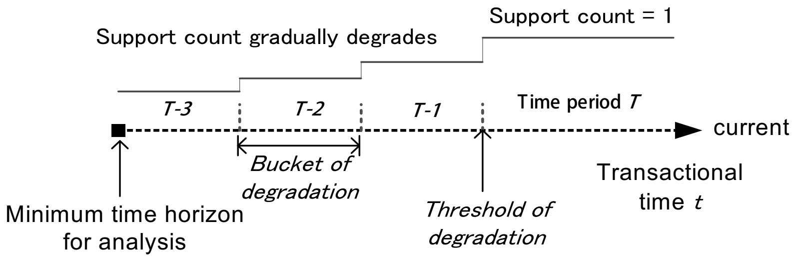

Fig. 1. Support count calculation.

previous phase, this phase finds frequent and recent k-FRP-sequences with $k \geqslant 2$ by calculating the *FR*-support.

3.1. Find all profitable itemsets

The original transaction database was converted into a database of customer sequences. To discover highly profitable sequences, profit filtering is then applied to the itemsets. Because an itemset may consist of several items, the average profit of an itemset must be calculated. The average profit for $ItemSet_i$ is represented as $ProfitOfItemSet_i$ which is calculated as follows:

$$ProfitOfItemSet_i = \frac{\sum_{Item_j \in ItemSet_i} (Price_{Item_j} - Cost_{Item_j})}{\#\,(ItemSet_i)} \tag{1}$$

where
$Price_{Item_j}$: Price of $Item_j$ in $ItemSet_i$;
$Cost_{Item_j}$: Cost of $Item_j$ in $ItemSet_i$;
$\#\,(ItemSet_i)$: Number of items in $ItemSet_i$.

An $ItemSet_i$ is profitable if $ProfitOfItemSet_i \geqslant Min\_Profit$. Because any profitable k-FRP-sequence contains a series of profitable itemsets, an itemset is removed if its profit is less than the minimum profit. The frequency and recency for itemsets are calculated after profit filtering. This stage only performs profit filtering.

3.2. Support count based on the frequency and recency for a sequence

Because the 1-FRP-sequences are all profitable itemsets, the k-FRP-sequences for $k \geqslant 2$ are also profitable. The next step simply compares the FR support count with the minimum FR support threshold. The calculation of support count is essential in sequential pattern mining. In the proposed model with frequency and recency, the support count (frequency) for a transaction sequence is proportional to its transaction time, as Fig. 1 shows. That is, the support count of a transaction sequence degrades with its transactional time. However, if the transactional time is later than a predefined threshold of frequency decay, the support count does not degrade (and remains one). Thus, the support count with frequency and recency for a sequence (or itemset), represented as *SupportCountFR*, is calculated as follows:

$$SupportCountFR = \begin{cases} 1, & \text{if } t \geqslant ThresholdOfDegradation \\ WeightOfDegradation^{\left\lceil \frac{WeightOfDegradation - t}{BucketOfDegradation} \right\rceil}, & \text{otherwise} \end{cases} \tag{2}$$

(I) **1-FRP-sequences phase**: finding all frequent, recent, and profitable itemsets (1-sequences);

For each itemset *c* in the candidate itemset table *I*, calculate $c.ProfitOfItemset$ using Eq. (1);
$I_{Profit} = \{c \in I \mid c.ProfitOfItemset \geq Min\_Profit\}$;

For each customer sequence *s* in the customers' transactional database {
  for each candidate itemset *c* in I_{Profit} that are contained in *s* {
    Increment $c.SupportCountFR$ with Eq. (2);
  }
}
Return $I_{FRP} = \{c \in I_{Profit} \mid c.SupportCountFR \geq Min\_SupportCountFR_1\}$;

(II) *k*-FRP-sequences phase: finding all frequent, recent, and profitable *k*-sequences, where $k \geq 2$;
Let $L_1 = I_{FRP}$
For (*k*=2; $L_{k-1} \neq \emptyset$; *k*++){
  C_k= New candidate sequences generated from L_{k-1};
  For each customer sequence *s* in the customers' transactional database {
    For each candidates *c* in C_k that are contained in *s* {
      Increment $c.SupportCountFR$ with Eq. (2);
    }
  }
  $L_k = \{c \in C_k \mid c.SupportCountFR \geq Min\_SupportCountFR_k\}$;
}
Return maximal sequences in $\cup_k L_k$

Fig. 2. FRP-AprioriAll algorithm.

where
ThresholdOfDegradation: Time threshold for frequency degradation;
BucketOfDegradation: Time bucket for frequency degradation;
WeightOfDegradation: Weight for frequency degradation; $0 <$ *WeightOfDegradation* < 1;
t: Transactional time of the first itemset in the sequence;
$\lceil \ \rceil$: Ceiling function.

3.3. Find large k-FRP-sequence

A frequent and recent sequence is called a *FR*-sequence. A frequent, recent and profitable sequence is called a FRP-sequence. A k-FRP-sequence is a FRP-sequence with length k. To find 1-FRP-sequences, the support counts for the profitable itemsets (obtained in previous section) are further calculated using Eq. (2) to find frequent and recent itemsets. To find k-FRP-sequences with $k \geqslant 2$, the support counts for sequences generated from previous $(k-1)$-FRP-sequences are also calculated by Eq. (2). The process of FRP-AprioriAll generates and filters the k-FRP-sequences for every possible k, as Fig. 2 shows.

Using a positive integer *Min_SuportCountFR* as the minimum support count threshold that combines the frequency and recency, a sequence S is frequent and recent in sequence database if *S.SupportCountFR* $\geqslant$ *Min_SupportCountFR*. For a sequence S to be frequent and recent, it must occur frequently and recently with a value of at least *Min_SuportCountFR*. In Eq. (2), for the 1-FRP-sequence, t is defined as the transactional time of the only itemset in the 1-FRP-sequence. For the k-FRP-sequence in which $k \geqslant 2$, t is defined as the transactional time of the first itemset in the k-FRP-sequence.

Unlike sequential pattern mining with a time window [3], or the time recency in the RFM model [8], the concept behind the proposed model is that the patterns generated close to the current period are more

Table 1
A series of transactions of four customers

Time horizon	Customer C1	C2	C3	C4
1	<(a,b),1>			<(e),1>
2		<(b),2>	<(a),2>	<(c,f),2>
3	<(e),3>			
4		<(a,c),4>	<(e),4>	
5	<(a,c),5>	<(d),5>	<(c,f,g),5>	
6	<(c,f),6>	<(c,f),6>		
7			<(c,g),7>	<(a,b,c),7>
8	<(g),8>			<(g),8>

Table 2
Profit and support count for itemset and 1-FRP-sequences

Itemset	Freq. count	FR count	Profit	Itemset	Freq. count	FR count	Avg. profit
<(a)>*	4	3.61	5	<(a,b)>	2	1.81	6.5
<(b)>	3	2.62	8	<(a,c)>	3	2.8	4
<(c)>*	4	3.8	3	<(c,f)>*	4	3.51	4.5
<(d)>	1	0.9	2	<(c,g)>	1	1	2
<(e)>	3	2.52	3	<(f,g)>	1	0.9	3.5
<(f)>*	4	3.51	6	<(a,b,c)>	1	1	5.3
<(g)>	3	3	1	<(c,f,g)>	1	0.9	3.3

important than those far from the current period. This study uses the concept of frequency degradation to give a higher importance to patterns generated in more recent periods. A recency-based pattern usually has a smaller support count than that of a traditional pattern because the proposed model degrades the frequency of a sequence that is temporally far from the current time.

3.4. *Example of FRP-sequence generation*

This section presents an example of the proposed FRP-AprioriALL algorithm. Table 1 shows a series of transactions for four customers. Customer C1 purchased items "a" and "b" at time period 1, and this transaction is represented as <(a,b),1>. Consider the candidate itemset <(a)> as an example; because all customers buy this item during these periods, its non-recency based frequency is 4 according to the traditional AprioriAll. For the recency based frequency, however, the purchase time is used to calculate the recency based frequency. The transactional time of <(a)> for the four customers are periods 5, 4, 2, and 7, respectively. Since there are two transactions (<(a,b),1> and <(a,c),5>) for customer C1, choose the latest one to compute the support count for item <(a)>.

To calculate the FRP-sequences, the system parameters were set as follows: bucket of degradation = 3, weight of degradation = 0.9, and threshold of degradation = 7. Thus the recency-based frequency of <(a)> is calculated as follows:

$$\begin{aligned} 0.9^{\lceil\frac{7-5}{3}\rceil} + 0.9^{\lceil\frac{7-4}{3}\rceil} + 0.9^{\lceil\frac{7-2}{3}\rceil} + 1 &= 0.9^1 + 0.9^1 + 0.9^2 + 1 \\ &= 0.9 + 0.9 + 0.81 + 1 = 3.61 \end{aligned} \tag{3}$$

Table 2 summarizes the corresponding new frequency for each FRP-sequence. Table 2 also illustrates the (average) profits of itemset. For example, the average profit of <(a,b)> is 6.5, which is the average profit of item "a" and "b" using Eq. (1). Let *Min_Profit* = 3, $Min\_SupportCountFR_1$ =

Table 3
2-FRP-sequences and 3-FRP-sequences

FRP-sequences	Freq. count	FR count	FR support count calculation
<(a)(c)>	3	2.52	C1(0.81) + C2(0.9) + C3(0.81) + C4(0)
<(a)(f)>	3	2.52	C1(0.81) + C2(0.9) + C3(0.81) + C4(0)
<(a)(c,f)>	3	2.52	C1(0.81) + C2(0.9) + C3(0.81) + C4(0)
<(c)(c)>	4	3.51	C1(0.81) + C2(0.9) + C3(0.81) + C4(0)
<(a)(c)(c)>	2	1.62	C1(0.9) + C2(0.9) + C3(0.9) + C4(0.81)

Table 4
2-sequences and 3-sequences by AprioriAll

Frequent sequences	Freq. count	Frequent sequences	Freq. count
<(a)(c)>	3	<(e)(f)>	3
<(a)(f)>	3	<(e)(g)>	3
<(a)(c,f)>	3	<(e)(c,f)>	3
<(c)(c)>	4	<(f)(g)>	3
<(a)(g)>	3	<(e)(c)(c)>	3
<(c)(g)>	3	<(e)(c)(g)>	4
<(c,f)(g)>	3	<(e)(c,f)(g)>	3
<(e)(c)>	3	<(e)(f)(g)>	3

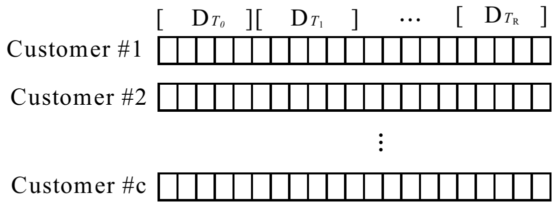

Fig. 3. Customer profile with purchase quantities of products in each periods.

3.0, $Min\_SupportCountFR_2 = 2.0$, and $Min\_SupportCountFR_3 = 1.5$. Thus, the 1-FRP-sequences are <(a)>, <(c)>, <(f)>, and <(c,f)>, and their FR support count and itemset profit are both greater than the *Min_Profit* and $Min\_SupportCountFR_1$. The transaction time of <(c)> is more recent than that of item <(f)>; therefore, the FR support count of <(c)> is greater than that of item <(f)>, although they both have the same frequency support count of four.

The calculation and comparison of FR-support count yields the 2-FRP-sequences <(a)(c)>, <(a)(f)>, <(a)(c,f)>, and <(c)(c)>, and the 3-FRP-sequences <(a)(c)(c)>, as Table 3 shows. The first itemset <(a)> in sequence <(a)(c)> occurred at time periods 1, 4, and 2 for customers C1, C2, and C3 respectively. Thus the FR-support count of <(a)(c)> was calculated by C1(0.81) + C2(0.9) + C3(0.81) + C4(0) = 2.52, as Table 3 shows. The number of 2-sequences and 3-sequences generated by AprioriAll (see Table 4) using a minimum support of 3 exceeded that of the proposed FRP model because product "g" was removed due to unprofitability and product "e" was not include due to its low FR support count. The predefined threshold of minimum profit and minimum FR support count can dominate the generating of sequential patterns.

4. Proposed collaborative recommender system

This study presents a collaborative recommender system based on the FRP-sequential patterns for e-commerce applications. To build the sequential pattern model, customers were clustered first. Then FRP-sequences were generated for each customer cluster. In the stages of customers clustering and FRP-sequential patterns generating, the product sub-classes, instead of end items, were used to reduce computation time. Because the product taxonomy is usually available in e-commerce reflects domain knowledge, it has been emphasized by many studies [10,11,13,15].

In the recommendation process, the target customers were selected from the customers to receive recommendations. Based on the target customer's transactional profile, each target customer was assigned to a specific cluster. The recommendation process includes the stages of product category prediction and product item recommending. Possible purchase categories were predicted using the FRP-sequences generated from the cluster the target customer belongs to. The product items among the possible purchase categories were then recommended to the target customer. The following sections describe these steps in more detail.

4.1. Collaborative customer clustering and FRP-sequential pattern mining

(1) Collaborative customer clustering

Customer buying behavior can be modeled by analyzing the customer's periodic transaction data, as Fig. 3 shows. The profile of a customer is defined as follows:

$$\mathit{PROFILE} = \{D_{T_0}, D_{T_1}, \ldots, D_{T_r}, \ldots D_{T_R}\} \tag{4}$$

where T is the current period, R is the number of previous periods considered, and

$$D_{T_r} = \left\{Qty_{T_r}^{P_1}, Qty_{T_r}^{P_2}, \ldots, Qty_{T_r}^{P_i}, \ldots, Qty_{T_r}^{P_N}\right\} \tag{5}$$

where $Qty_{T_r}^{P_i}$ is the quantity of product P_i that a certain customer purchased during period T_r, and N is the number of products.

$Qty_{T_r}^{P_i}$ in Eq. (5) is represented as the quantity of a product subclass. Customers are clustered via the product category (subclass) customer profile. Using the product category as the customer profile can decrease the data dimension and increase scalability. This study uses the k-means algorithm to cluster customers based on the customer profiles.

(2) FRP-sequential pattern mining for each cluster

Customers in the same cluster have similar buying behaviors. The proposed recommender system provides recommendations to target customers based on the product subclass purchased by other customers in the same group with similar buying preferences. In each cluster, the FRP-AríoriAll (see Section 3) is performed based on the customer's purchase profile to obtain the FRP-sequences for each cluster. Because the profit of a product category is required in sequential pattern mining in Eq. (1), the profit of a certain product category is calculated by averaging profits of product item in that category as follows:

$$\mathit{ProfitOfCategory} = \frac{\sum_i (\mathit{Price}_i - \mathit{Cost}_i)}{M}, i = 1, 2, \ldots, M \tag{6}$$

where

Price_i: Price of item i;

Cost_i: Cost of item i;

M: Number of product items in a certain product category.

4.2. Recommending product items to the target customer

(1) Product category prediction

This step determines the cluster that a target customer belongs to, and then finds the predicted categories by matching the target customer's purchase sequences in period T with the FRP-sequences generated from the target customer's cluster. The matched sequential patterns are then used to predict the target customer's most likely purchase items in the next time period $T + 1$ (prediction period).

The matching process is performed by subsequence determination. A sequence $SEQ_A = {<}A_1, A_2, \ldots, A_m{>}$ is a subsequence of the sequence $SEQ_B = {<}B_1, B_2, \ldots, B_n{>}$, if there exist integers $1 \leqslant j_1 < j_2 < \ldots < j_m \leqslant n$ such that $A_1 \subseteq B_{j_1}$, $A_2 \subseteq B_{j_2}, \ldots, A_m \subseteq B_{j_m}$. For example, the sequential pattern of <(20)(10,40)(50)(60,30)> matches a certain target user's purchase sequence of <(20)(50)> because the latter is a subset of the former. From this matched sequential pattern, items 60 and 30 are *candidate items* to be recommended to the target customer.

A candidate item may be generated from several sequential patterns, because these sequential patterns match the target user's purchase sequence. This candidate item's support is calculated by summarizing the FR support of all matched sequential patterns. Then the prediction score for each candidate item is calculated by combining the support values and profit. The prediction score for the predicted category is calculated as follows:

$$PredictionScore = ScaledSupport \times ScaledProfit \tag{7}$$

Both category support value and profit are scaled in advance to avoid attributes with greater numerical range dominating those with smaller numerical range. The category profit and support was scaled using the following formula:

$$ScaledX = \frac{Original\ Value\ of\ X}{Maximum\ Value\ of\ X} \tag{8}$$

where
ScaledX: ScaledSupport or *ScaledProfit*;
Maximum Value of X: Largest category support value or the largest category profit.

(2) Recommending items to target users

The previous step predicts the product categories that the target customer will probably purchase. The frequency of the item purchased by all customers in the same cluster and the item's profit are both used to measure an item's importance. The recommended item score for each item in the product category is calculated as follows:

$$RecommendationScore = ScaledPurchaseFrequency \times ScaledItemProfit \tag{9}$$

The item frequency and item profit were also scaled in advance using Eq. (8), in which the "*Maximum Value of X*" represents the largest frequency or the largest profit for an item in the target customer's cluster. Based on the recommended score, the top-N product items that the target customer will likely purchase in the target period are recommended to the target customer.

5. Experiments

5.1. Experimental design

This study uses experimental data from the Microsoft SQL Server Foodmart database to evaluate the proposed model. Three file tables in Foodmart database were used, including the product taxonomy, customer information, and purchase transaction tables. This database consists of 1,560 product items,

Table 5
Three sub-datasets from the Foodmart dataset

Dataset	Training and test period		Number of selected	Avg. profit per category		Profit per item	
	Training period	Test period	customers	Min.	Max.	Min.	Max.
DS1	Jan, Feb, Mar	Apr	175	2.42	5.74	2.31	5.92
DS2	Feb, Mar, Apr	May	195	2.53	5.23	2.38	5.77
DS3	Mar, Apr, May	Jun	191	2.48	5.68	2.44	5.68

164,558 purchase transactions, and 7,824 customers. The 1,560 product items include 102 product categories, and the product items' transaction times range from January to November 1998.

Three datasets, DS1, DS2, and DS3 were selected from the Foodmart database with different time period. Each dataset contained a three-month time horizon for training and a one-month time horizon for predicting, as Table 5 shows. For each dataset, customers were selected that made purchases at least once per month with each transaction containing at least four product categories. Because the customers who purchased frequently provided relatively sufficient information to their purchasing behavior, their sequential patterns can be identified in sequential pattern mining.

A 5-fold cross validation was conducted for each datasets to determine the performance of the proposed recommender system. All of the customers in each experimental dataset were randomly divided into five customer subgroups. Each subgroup took turns being the target customers for testing the model while the other four subgroups were used to train the model. For the customer for training model, this study clustered customers and generated sequential patterns based on the transactions of a three-month training period. The target customer's purchase sequences in the previous month of the test period were generated for matching with the sequential patterns generated from the cluster that the target customer belonged to. The target customer's purchase items in the test period were used to measure the performance of accuracy.

Three measures in evaluating the performance of the proposed method include recall, precision, and F1-measure, as Eqs (10)–(12) show, respectively. The precision is the ratio of purchase items from the recommended items for a target customer. The recall is the ratio that the recommended items successfully hit (predict) the target customer's purchased items in the test period. The F1 measure combines recall and precision with an equal weight.

$$Precision = \frac{Number\ of\ hit\ items}{Total\ number\ of\ recommend\ items} \tag{10}$$

$$Recall = \frac{Number\ of\ hit\ items}{Total\ number\ of\ items\ purchasedat\ period(T+1)} \tag{11}$$

$$F1 - Measure = \frac{Recall \times Precision}{(Recall + Precision)/2} \tag{12}$$

The total profit of recommendation to the target customers is defined as Eq. (13). The average profit performances are defined as the average profit per hit item (Eq. (14)), per hit customers (Eq. (15)), and per target customers (Eq. (16)).

$$TotalProfitOfRmd = \sum_i \sum_j Profit_{ij}, i \in HitCustomer, j \in HitItem \tag{13}$$

$$ProfitPerHitItem = \frac{TotalRecommendationProfit}{Number\ of\ hit\ items} \tag{14}$$

Table 6
Recommendation performance summary for SP-FRP, TOP-N and RND using Foodmart dataset

Model	Recall	Precision	F1-measure	Profit per customer	Profit per hit customer	Profit per hit item
SP-FRP	0.0282	0.0049	0.0079	1.26	6.56	5.09
TOP-N	0.0236	0.0047	0.0073	0.90	4.70	3.81
RND	0.0217	0.0044	0.0068	0.86	4.84	3.86

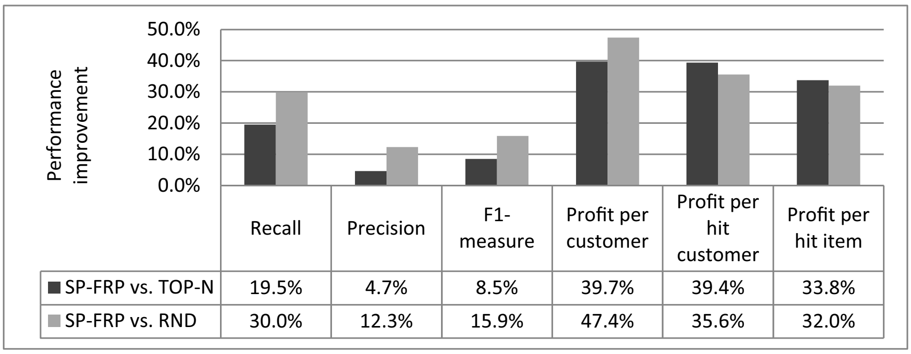

Fig. 4. Percentage of performance improvements for SP-FRP against TOP-N and RND.

$$ProfitPerHitCustomer = \frac{TotalRecommendationProfit}{Number\ of\ hit\ customers} \tag{15}$$

$$ProfitPerCustomer = \frac{TotalRecommendationProfit}{Number\ of\ target\ customers} \tag{16}$$

The target customer whose recommended items hit (or predict) any one of item in his/her purchase item in the test period is called a "*hit customer*", and the items that is successfully hit is called "*hit item*"; and $Profit_{ij}$ represents the profit for *hit item* j in *hit customer* i's purchase items in the test period.

5.2. Comparison with traditional approaches

The proposed frequency, recency, and profit sequential pattern based recommender system is abbreviated as "SP-FRP" recommendation. This study conducts a performance comparison between SP-FRP and the traditional collaborative top-N recommendation to determine the relative performance improvement of the proposed system. Additionally, to understand the base-line performance, we also conducted a performance comparison on the SP-FRP with the collaborative random recommendation. The collaborative top-N recommendation (abbreviated as "TOP-N") recommends popular (frequent) items in the target customer's cluster, which consists of the target customer's nearest neighbors, to the target customer. The collaborative random recommendation (abbreviated as "RND") recommends items that are randomly selected in the target customer's cluster to the target customer. The performance of the collaborative random recommendation was averaged from the five repetitive runs on product item for the three datasets. All the above three models are built for each customer cluster (the nearest neighbors based collaboration) with the same recommendation size N.

For the parameter setting of SP-FRP, the support rate was set to 0.05. The minimum support count in the FRP-sequence mining was set to 9. The bucket of frequency degradation was set to 8 days; the weight of frequency degradation was set to 0.8; the threshold of recency was set to 75 days; the threshold of profit was set to 3.3. The FRP-sequences were based on product category, which has 102 categories of

Table 7
Performance comparison between SP-FRP and SP-F*P

Model	Recall	Precision	F1-measure	Profit per customer	Profit per hit customer	Profit per hit item
SP-FRP	0.0282	0.0049	0.0079	1.26	6.56	5.09
SP-F*P	0.0268	0.0048	0.0076	1.24	6.67	5.19
Relative improvement	5.2%	3.8%	4.3%	2.0%	−1.8%	−1.9%

Table 8
Performance comparison between SP-FR* and SP-F**

Model	Recall	Precision	F1-measure	Profit per customer	Profit per hit customer	Profit per hit item
SP-FR*	0.0286	0.0054	0.0085	1.11	5.14	4.09
SP-F**	0.0276	0.0052	0.0081	1.07	5.31	4.13
Relative improvement	3.8%	5.0%	5.0%	4.2%	−3.2%	−0.9%

products in the Foodmart dataset. For the recommendation size, we selected the item score of the top-10 product items for each top-10 product categories, and among these 100 items, we recommended the 50 items with highest item score to the target customer. These system parameters may influence the experimental results. A thorough sensitivity analysis on the threshold of profit and frequency degradation's weight, bucket, and threshold was performed in Section 5.5.

Table 6 shows the average precision, recall, F1-measure, and profit for the three datasets; and Fig. 4 shows the relative percentage improvements on precision, recall, F1-measure and profit of the SP-FRP against that of the traditional TOP-N and the RND recommendations. For the accuracies of recommendation, the SP-FRP model was slightly better than the TOP-N recommendation as the relative percentage improvements on recall, precision and F1-measure are 19.5%, 4.7% and 8.6% respectively; however, for the profits of recommendation, the SP-FRP model was much better than the TOP-N recommendation as the relative percentage improvements on profit per customer, per hit customer and per hit item were 39.7%, 39.4% and 33.8%, respectively. The SP-FRP model significantly ($p \leqslant 0.05$) outperforms the TOP-N with on F1-measure and profit per customer, based on the paired-samples T tests.

Compared with the RND, SP-FRP was much better than the RND in terms of either accuracies or profits. Our SP-FRP model significantly ($p \leqslant 0.05$) outperforms the RND on F1-measure and profit per customer, based on the paired-samples T tests. That is, for the Foodmart dataset, the proposed model was more appropriate than both TOP-N and RND models in terms of the recommendation accuracy and profit. This implies that the proposed collaborative filtering system with SP-FRP sequential pattern can improve the recommendation performance compared to both collaborative TOP-N and RND recommendations.

5.3. Comparison between recency and non-recency models

To demonstrate the impact of recency on our proposed model, this study compares recency and non-recency models. The SP-FRP model was compared with the non-recency model, called "SP-F*P." Without considering recency, SP-F*P only generated the sequential patterns by counting the frequency and using profit filtering in the sequential pattern mining. The only difference between SP-FRP and SP-F*P was the use of frequency degradation or not. Table 7 shows that SP-FRP improves performance against SP-F*P by 5.2%, 3.8% and 4.3% for recall, precision, and F1-measure, respectively. This result indicates that the consideration of recency was essential for this dataset. For the performance of profit, the two models had similar profit performance (with significance of $p > 0.05$), because they both considered the profit factor.

Table 9
Performance comparison between SP-FRP and SP-FR*

Model	Recall	Precision	F1-measure	Profit per customer	Profit per hit customer	Profit per hit item
SP-FRP	0.0282	0.0049	0.0079	1.26	6.56	5.09
SP-FR*	0.0286	0.0054	0.0085	1.11	5.14	4.09
Relative improvement	−1.5%	−8.8%	−7.1%	13.6%	27.7%	24.5%

Table 10
Performance comparison between SP-F*P and SP-F**

Model	Recall	Precision	F1-measure	Profit per customer	Profit per hit customer	Profit per hit item
SP-F*P	0.0268	0.0048	0.0076	1.24	6.67	5.19
SP-F**	0.0276	0.0052	0.0081	1.07	5.31	4.13
Relative improvement	−2.8%	−7.8%	−6.5%	16.1%	25.8%	25.8%

Without considering the effect of profit filtering, the recency and non-recency models, called "SP-FR*" and "SP-F**", were constructed without using the profit filtering in the sequential pattern mining. The only difference between the two models was the use of frequency degradation or not. Table 8 shows the relative percentage improvement on SP-FR* against SP-F** by 3.8%, 5% and 5% for recall, precision, and F1-measure, respectively. These improvements showed that by weighting the recent sequential patterns can improve the accuracy for this dataset. For the profit performance, the two models have similar profit performances (with significance of $p > 0.05$), because the profit factor was not considered in both models.

5.4. *Comparison between profit and non-profit based models*

This section demonstrates the impact of profit on the proposed model, and compares the profit based model, SP-FRP, with the non-profit models, called SP-FR*. SP-FR* did not perform profit filtering in the sequential pattern mining. As shown in Table 9, the SP-FRP was significantly ($p \leqslant 0.05$) superior to the SP-FR* with relative percentage improvements by 13.6% for profit per customer, 27.7% for profit per hit customer, and 24.5% for profit per hit item, respectively. However, the accuracies for recall, precision, and F1-measure in SP-FRP were slightly inferior to those in SP-FR* by −1.5%, −8.8%, and −7.1%, respectively. Because some of the generated patterns were removed by the profit threshold in the sequential pattern mining, the accuracies of profit-based model may be slightly degraded. Nevertheless, the accuracy degradation was not significant in this experiment ($p > 0.05$).

For the models without considering the recency, we made a comparison between SP-F*P and SP-F**. The results in Table 10 are similar to those discussed above: the profit-based model, SP-F*P, improved the profit performance but slightly decreased the accuracy performance.

5.5. *Sensitivity analysis*

To verify the impact of parameter setting on the SP-FRP model, this section performs a sensitivity analysis on the following four parameters: weight, bucket and threshold of frequency degradation, and profit threshold. This analysis was based on one hundred frequent customers in the Foodmart dataset. Table 11 shows the parameter settings for these four factors. The previous section defines the weight of frequency degradation and the bucket of frequency degradation. For the threshold of frequency degradation and the threshold of profit, this study defines the proportion of degradation threshold and the proportion of profit threshold as follows.

Table 11
Parameter settings for sensitivity analysis

Factor	Base level	Various levels
Weight of frequency degradation	0.80	0.65, 0.80, 0.95
Bucket of frequency degradation	8	6, 8, 10
Proportion of frequency degradation threshold	0.75	0.25, 0.5, 0.75, 0.95
Proportion of profit threshold	0.25	0.25, 0.5, 0.75

Table 12
Performances of various weight of frequency degradation

Weight of degradation	0.65	0.8	0.95
Recall	0.0277	*0.0344*	0.0301
Precision	0.0086	*0.0090*	0.0083
F1 measure	0.0119	*0.0131*	0.0121
Profit per customer	1.6399	*2.1213*	1.9647

Table 13
Performances of various bucket of frequency degradation

Bucket of freq. degradation	6	8	10
Recall	0.0333	*0.0344*	0.0310
Precision	0.0089	*0.0090*	0.0083
F1 measure	0.0129	*0.0131*	0.0120
Profit per customer	1.9845	*2.1213*	1.9235

For the frequency degradation threshold, this study defines a proportion of the threshold of degradation to determine the frequency degradation threshold as follows.

$$\begin{aligned} \mathit{ThresholdOfDegradation} &= \mathit{MinTimeHorizon} + (\mathit{MaxTimeHorizon} - \mathit{MinTimeHorizon}) \\ &\quad \times \mathit{ProportionOfDegradationThreshold} \end{aligned} \tag{17}$$

This study presents the minimum and maximum time horizons for a dataset as *MaxTimeHorizon* and *MinTimeHorizon*. For example, if the proportion of degradation was set to 0.75 and the minimum and maximum time horizon were day 120 and day 20, respectively, the threshold of degradation was set to day 75.

The profit threshold was determined by the proportion of profit threshold as follows.

$$\mathit{ThresholdOfProfit} = \mathit{MinProfit} + (\mathit{MaxProfit} - \mathit{MinProfit}) \times \mathit{ProportionOfProfitThreshold} \tag{18}$$

For example, when the maximum and minimum profits were 2 and 7 (dollars) and the proportion of profit threshold was set to 0.25, the profit threshold was calculated as $2 + (7 - 2) \times 0.25 = 3.25$ (dollars).

Thirteen runs were conducted for the sensitivity analysis for the four factors experiments. A factor level is changed for each experimental run while the other three factors are set to their base level. For example, when the weight of frequency degradation was set to 0.65, 0.8, and 0.95, respectively, the other parameters were set to their base level as follows: bucket of frequency degradation = 8; proportion of degradation threshold = 0.75; and proportion of profit threshold = 0.25. This study reports preliminary results with a satisfactory outcome after several experiments with some possible values of these parameters. In practice, the setting of these parameters depends on the characteristic of application domain including the product kind, the product lifecycle, and the customer's purchase behavior.

Table 12 analyzes the weight of frequency degradation. The weight of frequency degradation was set to 0.65, 0.80, and 0.95 respectively. A weight of 0.8 produced better accuracy and average profit. The weight of frequency degradation gives transactions far from current time less impact and transactions near current more impact. Setting a proper weight is essential to achieve an adequate effect on the frequency calculation for the sequential patterns. A weight of 0.8 was good for the dataset in this study.

Table 13 analyzes the bucket of frequency degradation. The length of bucket determines the degradation on frequency calculation. The bucket was set to 6, 8, and 10 (days) respectively. A bucket of 8 produced better accuracy and profit, based on the base parameter setting.

Table 14
Performances of various frequency degradation thresholds

Proportion of frequency degradation threshold	0.25	0.5	0.75	0.95
Recall	0.0304	0.0314	*0.0344*	0.0332
Precision	0.0082	0.0081	*0.0090*	0.0086
F1 measure	0.0122	0.0122	*0.0131*	0.0130
Profit per customer	2.0688	2.0239	*2.1213*	2.0284

Table 15
Performances of various profit thresholds

Proportion of profit threshold	0.25	0.5	0.75
Recall	*0.0344*	0.0253	0.0253
Precision	0.0090	*0.0190*	*0.0190*
F1 measure	0.0131	*0.0144*	*0.0144*
Profit per customer	*2.1214*	1.5431	1.5431

Table 14 analyzes the threshold of frequency degradation. The decay threshold effects the frequency calculation. This maintains the importance of each transaction item in the current period (within the decay threshold), and reduces those outside the current period. This study sets the recency threshold at 0.25, 0.5, 0.75 and 0.95. Experimental results show that the recency threshold of 0.75 was more accurate than others.

Table 15 analyzes the profit threshold. This study sets the profit threshold at 0.25, 0.5, and 0.75. The threshold of profit determines the number of generated sequential patterns, and thus may affect recommendation accuracy. To improve profit, the profit threshold must be high; however, this may reduce the accuracy. Here, the profit threshold 0.25 was good for balancing the profit and accuracies.

6. Conclusion

Traditional sequential pattern mining only considers the frequency of transaction pattern. In designing a recommender system, however, the transaction patterns near the recent period should have a greater effect on customer preferences, especially in an environment with short product lifetimes. Traditional recommender systems focus on recommendation accuracy for target customers. From a seller's viewpoint, a recommender system should also focus on profitability. The objectives of improving prediction accuracies for the customer and increasing profits for the seller are conflicting from each other. This study incorporates user's recent interest in the sequential patterns to improve the prediction accuracy for the user's point of view. Besides, we may increase the profit and not to significantly decrease the accuracy via fine tuning the minimum profit threshold by experimentation. Also, thorough knowledge about the characteristics of the product, product lifecycle, and customer's purchase behavior may also help to fine-tune these parameters to improve the performances on both profit and accuracy.

Experimental results show that the proposed SP-FRP model performed better than the traditional collaborative recommendation system in both profit and accuracy. When considering only recency, the recency based models (SP-FRF or SP-FR*) were better than the non-recency models (SP-F*P or SP-F**) in both accuracy and profit regardless of whether profit model was included. Similarly, the profit model (SP-FRP or SP-F*P) was significantly better than the non-profit models (SP-FR* or SP-F**) in profit regardless of whether recency was included. Incorporating recency and profit into the recommender system not only maintains the accuracy of the item prediction for the customers, but also increases the potential profit of recommended item for the seller.

The suggested SP-FRP recommender system can be applied to e-commerce, in which customers often change interests gradually. Future research will focus on the following. (1) The FRP-Apriori is not efficient currently when dealing with large number of transactional data. Combining the concept of recency and profit with other efficient sequential pattern mining methods (such as [17,18]), may address this deficiency. (2) Further, the accuracy of the proposed approach can be improved by improving the recommendation process, including customer clustering in the collaborative filtering and conducting a full factorial experimental design to get optimal system parameters.

Acknowledgement

The authors would also like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract No. NSC 96-2221-E-327-008.

References

[1] G. Adomavicius and A. Tuzhilin, Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Transactions on Knowledge and Data Engineering* **17**(6) (2005), 734–749.

[2] R. Agrawal and R. Srikant, Mining sequential patterns, *Proceedings of the Eleventh International Conference on Data Engineering*, Taipei, Taiwan, (1995), 3–14.

[3] R. Agrawal and R. Srikant, Mining sequential patterns: Generalizations and performance improvements, *Proceedings of the 5th International Conference on Extending Database Technology (EDBT)*, Avignon, France, (1996), 3–17.

[4] M. Balabanovic and Y. Shoham, Fab: Content-based collaborative recommendation, *Communications of the ACM* **40**(3) (1997), 66–72.

[5] J.R. Bult and T.J. Wansbeek, Optimal selection for direct mail, *Marketing Science* **14**(4) (1995), 378–394.

[6] L.S. Chen, F.H. Hsu, M.C. Chen and Y.C. Hsu, Developing recommender systems with the consideration of product profitability for sellers, *Information Sciences* **178**(4) (2008), 1032–1048.

[7] Y.L. Chen and Y.H. Hu, Constraint-based sequential pattern mining: The consideration of recency and compactness, *Decision Support Systems* **42**(2) (2006), 1203–1215.

[8] Y.L. Chen, M.H. Kuo, S.Y. Wu and K. Tang, Discovering recency, frequency, and monetary (RFM) sequential patterns from customers' purchasing data, *Electronic Commerce Research and Applications* **8**(5) (2009), 241–251.

[9] Y. Chen, Y. Zhao and Y. Yao, A profit-based business model for evaluating rule interestingness, *Advances in Artificial Intelligence* **4509** (2007), 296–307.

[10] Y.B. Cho, Y.H. Cho and S.H. Kim, Mining changes in customer buying behavior for collaborative recommendations, *Expert System with Applications* **28**(2) (2005), 359–369.

[11] Y.H. Cho and J.K. Kim, Application of web usage mining and product taxonomy to collaborative recommendations in e-commerce, *Expert Systems with Applications* **26**(3) (2004), 234–246.

[12] J. Han and M. Kamber, *Data Mining Concepts and Techniques (Second edition)* San Francisco, USA: Morgan Kaufmann Press, (2006).

[13] C.L. Huang and W.L. Huang, Handling sequential pattern decay: Developing a two-stage collaborative recommender system, *Electronic Commerce Research and Applications* **8**(3) (2009), 117–129.

[14] K. Lang, Newsweeder: Learning to filter netnews, *Proceedings of the 12th International Conference on Machine Learning*, Tahoe City, CA, USA, (1995), 331–339.

[15] R.D. Lawrence, G.S. Almasi, V. Kotlyar, M.S. Viveros and S.S. Duri, Personalization of supermarket product recommendation, *Data mining and Knowledge Discovery* **5**(1–2) (2001), 11–32.

[16] Y. Liu, W. Liao and A. Choudhary, A fast high utility itemsets mining algorithm, *Proceedings of the 1st International Workshop on Utility-based Data Mining*, New York, NY, USA, (2005), 90–99.

[17] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal and M.C. Hsu, Mining sequential patterns by pattern-growth: The PrefixSpan approach, *IEEE Transactions on Knowledge and Data Engineering* **16**(11) (2004), 1424–1440.

[18] J. Pei, J. Han and W. Wang, Constraint-based sequential pattern mining: The pattern-growth methods, *Journal of Intelligent Information Systems* **28**(2) (2007), 133–160.

[19] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl, GroupLens: An open architecture for collaborative filtering of netnews, *Proceedings of the Computer Supported Cooperative Work Conference* Chapel Hill, NC, USA, (1994), 175–186.

[20] A.M. Rashid, I. Albert, D. Cosley, S.K. Lam, S.M. McNee, J.A. Konstan and J. Riedl, Getting to know you: Learning new user preferences in recommender systems, *Proceedings of the International Conference on Intelligent User Interfaces*, San Francisco, CA, USA, (2002), 127–134.

[21] J.B. Schafer, J.A. Konstan and J. Riedl, Recommender systems in e-commerce, *Proceedings of the First ACM Conference on Electronic Commerce*, Denver, Colorado, USA, (1999), 158–166.

[22] J.B. Schafer, J.A. Konstan and J. Riedl, E-commerce recommendation application, *Data Mining and Knowledge Discovery* **5**(1–2) (2001), 115–153.

[23] V.S. Tseng, C.J. Chu and T. Liang, An efficient algorithm for mining temporal high utility itemsets from data streams, *Journal of Systems and Software* **81**(7) (2006), 1105–1117.

[24] K. Wang and M.Y. Su, Item selection by "hub-authority" profit ranking, *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, (2002), 652–657.
[25] K. Wang, S. Zhou and J. Han, Profit mining: From patterns to actions, *Lecture Notes in Computer Science* **2287** (2002), 31–44.
[26] K. Wang, S. Zhou, Q. Yang and J.M.S. Yeung, Mining customer value: From association rules to direct marketing, *Data Mining and Knowledge Discovery* **11**(1) (2005), 57–79.
[27] K. Wei, J. Huang and S. Fu, A survey of e-commerce recommender systems, *Proceedings in of the IEEE International Conference on Service Systems and Service Management*, Chengdu, China, (2007), 1–5.
[28] J.S. Yeh, C.Y. Chang and Y.T. Wang, Efficient algorithms for incremental utility mining, *Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*, Suwon, Korea, (2008), 229–234.