The Journal of Systems and Software 86 (2013) 779–788

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Knowledge discovery of weighted RFM sequential patterns from customer sequence databases☆

Ya-Han Hu[a], Tony Cheng-Kui Huang[b,*], Yu-Hua Kao[a]

[a] Department of Information Management, National Chung Cheng University, Taiwan, ROC
[b] Department of Business Administration, National Chung Cheng University, 168, University Rd., Min-Hsiung, Chia-Yi, Taiwan, ROC

ARTICLE INFO

Article history:
Received 18 June 2012
Received in revised form 6 September 2012
Accepted 6 November 2012
Available online 23 November 2012

Keywords:
Data mining
Sequential patterns
RFM analysis
Constraint-based mining

ABSTRACT

In today's business environment, there is tremendous interest in the mining of interesting patterns for superior decision making. Although many successful customer relationship management (CRM) applications have been developed based on sequential pattern mining techniques, they basically assume that the importance of each customer is the same. Previous studies in CRM show that not all customers make the same contribution to a business, and it is indispensible to evaluate customer value before developing effective marketing strategies. Therefore, this study includes the concepts of recency, frequency, and monetary (RFM) analysis in the sequential pattern mining process. For a given subsequence, each customer sequence contributes its own recency, frequency, and monetary scores to represent customer importance. An efficient algorithm is developed to discover sequential patterns with high recency, frequency, and monetary scores. Empirical results show that the proposed method is efficient and can effectively discover more valuable patterns than conventional frequent pattern mining.

1. Introduction

Over the past decade, there has been tremendous interest in analyzing huge amounts of data through various data mining techniques. Data mining can be treated as the process of extracting implicit, previously unknown and potentially useful information from databases (Chen et al., 1996; Frawley et al., 1992). For superior decision making, the mining of interesting patterns and rules has become one of the most indispensible tasks in today's business environment. Many data mining techniques have been well-developed, such as association rules mining, sequential pattern mining, classification, clustering, and other statistical methods (Chen et al., 2011; Han and Kamber, 2006; Kim et al., 2007; Lin et al., 2006; Mannila, 1998).

Customer relationship management (CRM), which is recognized as a process to manage the interactions between companies and their customers, has become a main task in today's business environment. In the area of information technology, CRM can be seen as an integration of technologies and business processes used to fulfill customer needs during their interactions. In the past, various data mining techniques have been conducted in CRM (Shaw et al., 2001). A series of techniques has been developed and applied, including automated data collection methods, data analysis and knowledge discovery methods. These techniques help to sell more goods or services, retain old customers, and increase customer profitability.

In many CRM applications, sequential pattern mining plays a crucial role for discovering time-related purchasing behavior in sequence databases (Agrawal and Srikant, 1995; Chiang et al., 2009; Huang and Huang, 2010; Mannila et al., 1997). Consider a given set of customer purchasing sequences, called data-sequences. Each data-sequence contains a list of transactions, and each transaction contains a set of items. Given a user-specified selection criteria, sequential pattern mining finds all of the frequent subsequences in a sequence database. There are many successful CRM applications which conduct sequential pattern mining techniques. For example, sequential pattern mining techniques can be applied to discover users' browsing behavior with web servers. Interesting user access patterns can be extracted from web access logs. We can use this useful information to predict customer's next visit or enhance system performance (Pei et al., 2000; Srikant and Yang, 2001; Tao et al., 2008). As another example, the purchasing behavior of a customer can be extracted from member transaction records (Chen and Huang, 2005; Chen et al., 2003). In order to gain maximum profit, the seller/retailer can use these patterns to design a sales strategy, such as repetitive selling or cross selling.

Although there have been many successful CRM applications of sequential pattern mining techniques (Chen and Hu, 2006; Chen et al., 2009; Eirinaki and Vazirgiannis, 2003; Liu and Shih, 2005a),

☆ This paper is an extension of work originally reported in Proceedings of the Pacific Asia Conference on Information Systems (PACIS 2011).

* Corresponding author. Tel.: +886 5 2720411x34319; fax: +886 5 2720564.
E-mail address: bmahck@ccu.edu.tw (T.C.-K. Huang).

0164-1212/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2012.11.016

they basically assume that the importance of each customer is the same. That is, each customer sequence is of equal weight during the mining process. However, many studies in CRM have revealed that not all customers make the same contribution to businesses, and, to maximize business profit, it is necessary to evaluate customer value before designing effective marketing strategies. Therefore, defining customer value by examining customer purchasing sequences and incorporating customer value into sequential pattern discovery are the two most critical issues in the mining process.

Recency, frequency, and monetary (RFM) analysis is a well-known and powerful tool in database marketing and is widely used in measuring the values of customers according to their prior purchasing history. The basic idea of RFM analysis was first introduced by Hughes (1994). Valuable customers can be defined as the ones simultaneously having high recency, frequency, and monetary scores, where recency represents the time-interval between the time of the latest event (e.g. transaction) and the present, frequency represents the number of events occurring in historical records, and monetary represents the total dollar value of payments (Hughes, 1996; Huang et al., 2009; Yeh et al., 2009).

The concept of RFM analysis has recently been integrated into the mining of valuable sequential patterns (Chen and Hu, 2006; Chen et al., 2009; Cheng and Chen, 2009; Liu et al., 2009). Previous studies have taken two directions: (1) identifying the best customer cluster and performing conventional sequential pattern mining from a set of the best customers, and (2) pushing RFM constraints into the mining of sequential patterns. Although these experimental evaluations have shown promising results, they still basically assume the equal weight of customer sequences in a sequence database. As a result, some significant information cannot be revealed.

Based on the above concern, this study includes the concepts of RFM analysis in sequential pattern mining in a more reasonable way. In RFM analysis, each customer has its own recency, frequency, and monetary scores to represent customer importance. Similarly, in our proposed approach, the RFM values of each item, itemsets, and customer sequences are evaluated in the sequential pattern mining process. A sequence is called a sequential pattern if its recency, frequency, and monetary scores satisfy the corresponding minimum thresholds. More specifically, the recency score of a subsequence is determined by the time of the last occurrence in a customer sequence. The more recent the last occurrence of a subsequence is, the higher the recency score the subsequence gets. Frequency is defined as the number of occurrences of a subsequence in a customer sequence. If a subsequence repeatedly occurs in a customer sequence, the frequency score of this subsequence is considered as high. Monetary denotes the amount of money a customer spends in a subsequence. A subsequence with a high monetary score means that the customer contributes higher revenue to the business. In summary, the three definitions allow us to evaluate and accumulate the RFM values of each subsequence from each customer.

In addition, we also consider the compactness constraint, meaning that the time span between the first and the last purchases in a customer sequence must be within a user-specified threshold. This constraint can assure that the purchasing behavior implied by a sequential pattern must occur in a reasonable period. For example, a customer bought *toast* two days ago, and then buys *peanut butter* and *jelly* today. However, another customer bought *toast* three months ago, and then buys *peanut butter* and *jelly* today. Since the time span of the second sequence is too long, counting in the second sequence will reduce the significance of patterns $\langle(toast)(peanut\ butter, jelly)\rangle$.

The rest of this paper is organized as follows. Section 2 reviews previous studies related to RFM analysis and RFM-related sequential pattern mining. Section 3 formally defines the problem of mining RFM sequential patterns. To efficiently discover a complete set of RFM sequential patterns, in Section 4, we propose the RFM-PostfixSpan algorithm which is extended from the well-known PrefixSpan algorithm. Section 5 briefly depicts the scheme of our experiments. Finally, conclusions are presented in Section 6.

2. Related work

Two main approaches are proposed to integrate the RFM model into sequential pattern mining. The first is to conduct customer segmentation based on RFM analysis and then discover patterns from the best customer group (Huang and Huang, 2010; Liu and Shih, 2005a,b; Liu et al., 2009; Miglautsch, 2002). This approach is a combination of both clustering and sequential pattern mining techniques. For example, Liu and Shih (2005a) adopted weighted RFM analysis by first applying an analytic hierarchy process (AHP) to evaluate customer value. Conventional sequential pattern mining techniques have been conducted to build product recommender systems for different customer groups; Huang and Huang (2010) developed a two-stage collaborative recommender system for product recommendation. It first identifies a target customer group by utilizing RFM-like customer purchasing behavior variables and a genetic algorithm. After that, sequential patterns discovered from each customer group can be used to predict the top-M product categories and top-N products for each active customer.

Although the first approach performs well in some applications, it is limited if the items in the dataset have large variations in their characteristics. For example, a large grocery retailer contains over ten thousand kinds of goods, and the varieties of both price range and purchase frequency are huge. A customer who regularly purchases 2% milk and chocolate each week will not be classified as a valuable customer because they have a low monetary score. On the contrary, another customer who has bought a vacuum and a Blu-ray player will have a high monetary score but low recency and frequency scores. If we consider all customer transactions as a whole, both of the above customers will never be classified as the best customers in RFM analysis. Therefore, it is difficult to define the RFM value (that is, weight) of each customer. Consequently, the approach works well only when investigating single items or items having similar characteristics.

Instead of measuring the value of each customer and then discovering sequential patterns from the best customer group, the second approach is directly applies RFM features as constraints in the mining of sequential patterns. Since a sequence database records represent complete customer purchasing behavior, the concept of RFM analysis can be easily integrated into the mining process and a sequential pattern is defined to satisfy all requirements related to RFM features. This approach is far more complicated than the first since it involves redesigning the definition of a sequential pattern as well as the mining procedure. For example, Chen and Hu (2006) proposed the CFR-PostfixSpan algorithm to consider recency, frequency, and compactness constraints in sequential pattern mining. Chen et al. (2009) proposed the RFM-Apriori algorithm for sequential pattern mining with the consideration of recency, frequency, and monetary constraints. The results of the RFM-Apriori algorithm maintain the following three properties: (1) customers must spend a certain amount of money on the patterns; (2) patterns occur at least once in a specified time period; (3) patterns occur more frequently than a specified frequency threshold in the database.

The second approach can successfully solve the problem raised in the first approach. Instead of measuring the RFM values of customers and pre-filtering out valueless ones, it directly treats the RFM features as constraints in the sequential pattern mining process (Chen and Hu, 2006; Chen et al., 2009). By setting

sid	Sequence
10	<(*a*,10,5),(*c*,30,4),(*a*,40,2),(*b*,40,1),(*d*,60,3),(*e*,60,1),(*c*,80,6)>
20	<(*d*,30,2),(*a*,50,15),(*b*,50,4),(*e*,50,3),(*d*,90,6)>
30	<(*a*,30,3),(*b*,30,4),(*e*,45,2),(*c*,55,8),(*a*,60,6),(*b*,60,1),(*a*,85,3),(*e*,85,7)>
40	<(*b*,20,2),(*d*,35,3),(*b*,40,1),(*c*,40,12),(*e*,70,5),(*a*,100,130),(*d*,100,50)>
50	<(*c*,70,160),(*b*,85,7),(*d*,85,6),(*a*,90,90),(*d*,100,10)>

Fig. 1. A sequence database.

adequate recency, frequency, and monetary thresholds, the mining algorithms search the complete customer sequence database and filter out worthless patterns. However, though it does include the concept of RFM features through various constraints, the second approach still follows the basic assumption of sequential pattern mining, that is, each customer sequence has equal weight. Therefore, the sequential patterns of the best customers cannot be highlighted.

High utility mining (HUM) is an emerging technique aimed at discovering patterns with high user-defined utility, that is, the significance of a pattern from a user perspective (Hu and Mojsilovica, 2007; Yao and Hamilton, 2006). In addition to frequency, many other more suitable indicators can be utilized in various applications to evaluate the value of a pattern, such as revenue, cost, and profit. Yao and Hamilton (2006) proposed an approach for association rule mining that permits users to quantify their preferences concerning the usefulness of itemsets with utility values. Hu and Mojsilovica (2007) also addressed the same issue by proposing an algorithm for identifying high-utility frequent itemsets, i.e. the ones that contribute the most to a predefined utility, objective function, or performance metric. Including these indicators in the mining process can highlight more meaningful patterns for decision makers. Although HUM has shown its practicability and flexibility in pattern discovery, it does not satisfy the anti-monotone property, which results in a great challenge in efficiently mining utility patterns. This is also the reason why recent studies on HUM have mainly focused on the development of efficient algorithms (Ahmed et al., 2009; Chu et al., 2008; Erwin et al., 2007).

Since HUM allows a user to define the importance of a pattern in a transaction or a sequence based on significant indicators, the values of customer sequences are no longer assumed to be the same in the mining process. As a result, high-value sequences can be unveiled. Furthermore, according to our knowledge, there are no studies which draw on the concept of RFM for HUM. Therefore, this study integrates the concept of RFM features into HUM to define three new indicators for the sequential pattern mining of sequence databases. Each customer has his/her own recency, frequency, and monetary values (i.e., weights) according to his/her purchasing history.

3. Problem definition

In the past, customer data-sequences were represented as ordered lists of itemsets, with a transaction time assigned to each. A customer's data-sequence A can be represented by $\langle(a_1, t_1, q_1), (a_2, t_2, q_2), ..., (a_n, t_n, q_n)\rangle$, where (a_j, t_j, q_j) indicates that item a_j was purchased at time t_j with quantity q_j, $1 \le j \le n$, and $t_{j-1} \le t_j$ for $2 \le j \le n$. If items occur at the same time in the data-sequence, they are ordered alphabetically. Based on this format, we give the following definitions:

Definition 1. Let I denote a set of all items in the database. Give a data-sequence $A = \langle(a_1, t_1, q_1), (a_2, t_2, q_2), ..., (a_n, t_n, q_n)\rangle$ and an itemset $I_q = (i_1, i_2, \ldots, i_m)$, where $I_q \subseteq I$ and $m \le n$. We say I_q is contained in A if there are m integers $1 \le k_1 < k_2 < \cdots < k_m \le n$ such that $i_1 = a_{k_1}$, $i_2 = a_{k_2}$, ..., $i_m = a_{k_m}$ and $t_{k_1} = t_{k_2} =, \cdots, = t_{k_m}$.

Definition 2. Let $B = \langle I_1, I_2, \ldots, I_s\rangle$ be a sequence of itemsets, where each $I_q \subseteq I$, $1 \le q \le s$. Sequence B is said to be contained in A or a *subsequence* of A if the following conditions are satisfied: (1) each I_q in B is contained in A, and (2) $t_{I_1} < t_{I_2} < \cdots < t_{I_s}$ where t_{I_q} $(1 \le q \le s)$ is the time at which I_q occurs in A.

Example 1. A sequence $\langle(ab)\rangle$ is contained in data-sequence $A = \langle(b, 1, 10), (c, 3, 5), (a, 5, 40), (b, 5, 20), (d, 7, 30), (a, 8, 20), (e, 8, 10)\rangle$, because both items a and b occur in A at time 5. The sequence $\langle(ab)(ae)\rangle$ is a subsequence of A since itemsets (ab) and (ae) are contained in A at time 5 and time 8, respectively, and $t_{(ab)} < t_{(ae)}$.

Definition 3 *(Compactness constraint).* Following definition 2, assume that $B = \langle I_1, I_2, \ldots, I_s\rangle$ is a subsequence of A. Let *ms_length* be a user-specified maximum span length. B is called a compact subsequence (*c-subsequence*) of A if $t_{I_s} - t_{I_1} \le ms\_length$.

Example 2. Consider the sequence database shown in Fig. 1. Assume that $ms\_length = 25$, a sequence $B = \langle(ab)(e)\rangle$ is called a c-subsequence of sid_{10} since (1) itemsets (ab) and (e) are contained in sid_{10} at time 40 and time 60, (2) $t_{(ab)} < t_{(e)}$, and (3) $t_{(e)} - t_{(ab)} \le ms\_length = 25$.

Definition 4. A c-subsequence B is said to cyclically occur m times in A if the concatenation of B is also a c-subsequence of A and the *repetitions* of B in A are equal to m, i.e. sequence $\langle\ldots B_1 \ldots B_2 \ldots B_m \ldots\rangle$ is a subsequence of A.

Below, the concept of the RFM model is extended and the definitions of recency, frequency and monetary are defined as follows.

Definition 5 *(Frequency subsequence).* Given a data-sequence A, let B be a c-subsequence of A. Assuming that B cyclically occurs in A, the *frequency score* of B, denoted as $Fscore(B,A)$, is defined as the number of repetitions of B in A. The *total frequency score* of B in DB, denoted as $TFscore_{DB}(B)$, is defined as the sum of the frequency scores of all data-sequences containing B.

$$TFscore_{DB}(B) = \sum_{A \in DB} Fscore(B, A)$$

Example 3. Following Example 2 and considering a data-sequence sid_{30} in Fig. 1, we find that $B = \langle(ab)(e)\rangle$ is a c-subsequence of sid_{10} and sid_{30}. Besides, B occurrences two times in sid_{30}, where the first occurrence is during the 30–45 time interval and the second is during 60–85. Thus, $Fscore(B, sid_{30}) = 2$, $Fscore(B, sid_{10}) = 1$, and finally the total frequency score of B is equal to 3 (i.e., $TFscore_{DB}(B) = 1 + 2 = 3$).

Definition 6 *(Recency subsequence).* Assuming that a c-subsequence B cyclically occurs m times in A and B_r denotes the rth repetition of B in A for $1 \le r \le m$, the *recency score* of B in A, denoted as $Rscore(B,A)$, is equal to $(1-\delta)^{t_{current} - t_{I_s}^{B_m}}$, where δ is a user-specified decay speed ($\delta \in [0, 1]$), $t_{current}$ denotes the current time stamp, and $t_{I_s}^{B_m}$ denotes the timestamp of the last itemset of B_m. Given a sequence database SDB, the *total recency score* of sequence B, denoted as $TRscore_{DB}(B)$, is defined as the sum of the recency scores of all data-sequence containing B.

Item	Price
a	10
b	150
c	25
d	45
e	80

Fig. 2. A list of item unit prices.

$$TRscore_{DB}(B) = \sum_{A \in DB} Rscore(B, A)$$

Example 4. Following Examples 2 and 3, assume the current timestamp $t_{current} = 110$ and decay speed $\delta = 0.1$. Consider a data-sequence sid_{30} shown in Fig. 1, since c-subsequence $B = \langle(ab)(e)\rangle$ has two occurrences in sid_{30}, and the most recent one occurs during time interval 60–85. A c-subsequence B's *Rscore* in sid_{30} (i.e., $Rscore(B, sid_{30})$) is equal to $(1-0.1)^{110-85} = 0.0717898$. Similarly, $Rscore(B, sid_{10}) = (1-0.1)^{110-60} = 0.0051538$. Hence, $TRscore_{DB}(B) = 0.0717898 + 0.0051538 = 0.0769436$.

Definition 7 *(Monetary subsequence).* As in Definition 6, a c-subsequence B cyclically occurs m times in A. Let $P(a_i)$ denote the unit profit of item a_i, and $q_{a_i}^{B_r}$ the purchase quantity of a_i of the rth repetition of B_r in A. Then B's *monetary score* in A, denoted as $Mscore(B,A)$, is defined as follows.

$$Mscore(B, A) = \sum_{B_1}^{B_m} \sum_{a_i \in B_r} P(a_i) \times q_{a_i}$$

Moreover, the *total monetary score* of sequence B, denoted as $TMscore_{DB}(B)$, is defined as the sum of the monetary scores of all data-sequences containing B.

$$TMscore_{DB}(B) = \sum_{A \in DB} Mscore(B, A)$$

Example 5. Consider the list of item unit prices shown in Fig. 2. Following Example 3, we know that $B = \langle(ab)(e)\rangle$ occurs once in sid_{10} and twice in sid_{30}. For sid_{10} and sid_{30}, $Mscore(B,sid_{10}) = 2 \times 10 + 1 \times 150 + 1 \times 80 = 250$ and $Mscore(B,sid_{30}) = (3 \times 10 + 4 \times 150 + 2 \times 80) + (6 \times 10 + 1 \times 150 + 7 \times 80) = 1560$. Finally, the $TMscore_{DB}(B) = 1560 + 250 = 1810$.

Definition 8. Given user-specified recency, frequency, and monetary thresholds, denoted as *Rminsup*, *Fminsup*, and *Mminsup*, respectively, we say a subsequence B is an RFM-sequential pattern (RFM-SP) if $TRscore_{DB}(B) \leqq Rminsup$, $TFscore_{DB}(B) \leqq Fminsup$, and $TMscore_{DB}(B) \leqq Mminsup$.

Example 6. Given three thresholds $Rminsup = 0.05$, $Fminsup = 2$, and $Mminsup = 1500$, as in Examples 2–5, we say the sequence $B = \langle(ab)(e)\rangle$ is an RFM-SP since the sequence B satisfies $TRscore_{DB}(B) = 0.0769436 \leqq 0.05$, $TFscore_{DB}(B) = 3 \leqq 2$, and $TMscore_{DB}(B) = 1810 \leqq 1500$.

4. The RFM-PostfixSpan algorithm

An efficient algorithm, called RFM-PostfixSpan, is developed for mining all the RFM-SP from a sequence database. The RFM-PostfixSpan algorithm is developed by modifying the well-known PrefixSpan algorithm, which recursively partitions a sequence database into a number of projected databases and retrieves the RFM-SPs by exploring only the local frequent patterns in each projected database. Instead of traversing *SDB* from the prefix of a sequence, the RFM-PostfixSpan partitions *SDB* from the postfix to efficiently retrieve the recency score of a pattern. The introduction of this algorithm can also be referred to the earlier version of Hu and Kao (2011).

Below, we first define *compact postfix*, *compact projection* and *compact prefix*. We depict the RFM-PostfixSpan later.

Definition 9 *(The compact postfix).* Given a data-sequence $A = \langle(a_1, t_1, q_1), (a_2, t_2, q_2), \ldots, (a_n, t_n, q_n)\rangle$ and a sequence $B = \langle I_1, I_2, \ldots, I_s\rangle$, B is a compact postfix of A if and only if (1) B is a c-subsequence of A, and (2) $t_{I_s} = t_n$.

Example 7. Given a data-sequence $sid_{30} = \langle(a, 30, 3), (b, 30, 4), (e, 45, 2), (c, 55, 8), (a, 60, 6), (b, 60, 1), (a, 85, 3), (e, 85, 7)\rangle$ and $ms\_legth = 40$. Sequence $\langle(c)(ae)\rangle$ is a compact postfix of sid_{30} since (1) the timestamp of itemsets (c) and (ae) in sid_{30} are 55 and 85, respectively, and the distance between them is 30 (i.e., $\langle ms\_length\rangle$), and (2) itemset (ae) is the last itemset of sid_{30}.

Definition 10 *(The compact projection).* Give a data-sequence $A = \langle(a_1, t_1, q_1), (a_2, t_2, q_2), \ldots, (a_n, t_n, q_n)\rangle$, let B be a c-subsequence of A. A sequence $A' = \langle(a'_1, t'_1, q'_1), (a'_2, t'_2, q'_2), \ldots, (a'_p, t'_p, q'_p)\rangle$ of sequence A is called a compact projection of A with respect to compact postfix B if and only if (1) A' has compact postfix B, (2) A' is a c-subsequence of A, and (3) there exists no super-sequence A'' of A' such that A" is a c-subsequence of A and also has compact postfix B.

Example 8. Given a sequence $sid_{30} = \langle(a, 30, 3), (b, 30, 4), (e, 45, 2), (c, 55, 8), (a, 60, 6), (b, 60, 1), (a, 85, 3), (e, 85, 7)\rangle$, let $ms\_length = 40$. Assuming that sid_{30} is projected with compact postfix $\langle(a)\rangle$, we have the following three compact projections sid'_{30}, including $\langle(a, 30, 3)\rangle$, $\langle(a, 30, 3), (b, 30, 4), (e, 45, 2), (c, 55, 8), (a, 60, 6)\rangle$ and $\langle(e, 45, 2), (c, 55, 8), (a, 60, 6), (b, 60, 1), (a, 85, 3)\rangle$. Each of the three compact projections satisfy the following conditions: (1) it has compact postfix $\langle a\rangle$, (2) it is a c-subsequence of sid_{30} (i.e., the time-interval between the first and last items in a projection satisfies the *ms_length* constraint), and (3) appending additional items in any of the above three compact projections results in violating (1) or (2).

Definition 11 *(The compact prefix).* Let $A' = \langle(a'_1, t'_1, q'_1), (a'_2, t'_2, q'_2), \ldots, (a'_p, t'_p, q'_p)\rangle$ be a compact projection of A with respect to the compact postfix $B = \langle I_1, I_2, \ldots, I_s\rangle$. Let a'_m be the first item in itemset I_1 and t'_m be the time at which a'_m occurs in A' $(1 \leqq m \leqq p)$. Then $C = \langle(a'_1, t'_1, q'_1), (a'_2, t'_2, q'_2), \ldots, (a'_{m-1}, t'_{m-1}, q'_{m-1})\rangle$ is the compact prefix of A with respect to B.

To differentiate the compact prefixes generated from the same data-sequence, the tag $[sid{:}Mscore_{postfix}{:}start\_time{:}end\_time]$ is attached to each compact prefix, where *sid* is the identifier of the data-sequence, $Mscore_{postfix}$ is the monetary score of the compact postfix B in A, and *start_time* and *end_time* are the timestamps in A that match the first and the last itemset of B, respectively.

Example 9. Following Example 8, the compact prefixes can be easily obtained by directly removing the compact postfix from the compact projection. Since the first compact projection in sid_{30} only contains a compact postfix (i.e. $\langle(a, 30, 3)\rangle$), the compact prefix becomes *null*. The remaining two compact prefixes with respect to $\langle(a)\rangle$ are $[sid_{30}{:}60{:}60{:}60]$ $\langle(a, 30, 3), (b, 30, 4), (e, 45, 2), (c, 55, 8)\rangle$ and $[sid_{30}{:}30{:}85{:}85]$ $\langle(e, 45, 2), (c, 55, 8), (a, 60, 6), (b, 60, 1)\rangle$.

Based on Definition 7, we find the downward closure property no longer holds if *TMscore* is considered in the mining process. When we include more items or itemsets in a non-RFM-SP, its *TMscore* will increase and it is possible for its super-sequence become an RFM-SP. This property is critical in pattern-growth-based methods (i.e., PrefixSpan and RFP-PostfixSpan) because it can

Input: A sequence database *SDB*, and the maximum span length *ms_length* and three support thresholds *Rminsup*, *Fminsup* and *Mminsup*.
Subroutine: RFM-PostfixSpan(α,l, $SDB|_\alpha$)
Parameters: α is a set of RF-SP
l is the length of α
$S|_\alpha$ is the α-projected database
Output: The complete set of RF-SPs and RFM-SPs

Method:
Each item in $SDB|_\alpha$ is appended before α as α' or is added into the first itemset of α as α'
Scan the database $SDB|_\alpha$ once and calculate $TRscore_{SDB}(\alpha')$, $TFscore_{SDB}(\alpha')$, and $TMscore_{SDB}(\alpha')$ for each α'
For each α'
If $TRscore_{SDB}(\alpha') \geq Rminsup$, $TFscore_{SDB}(\alpha') \geq Fminsup$ and $TMscore_{SDB}(\alpha')+TSM_{SDB}(\alpha) \geq Mminsup$ then
Output α' as an RF-SP
If $TMscore_{SDB}(\alpha') \geq Mminsup$ then
Output α' as an RFM-SP
Construct α'-projected database $SDB|_{\alpha'}$, and call PostfixSpan(α', l+1, $S|_{\alpha'}$)
End for

Fig. 3. The RFM-PostfixSpan algorithm.

greatly decrease the search space. To solve this problem, we first give two definitions as follows.

Definition 12 *(The sequence monetary value).* Given sequence *A*, let *B* be a compact postfix of *A*. The sequence monetary value of compact postfix *B* in *A*, denoted as *SM*(*B*,*A*), is equal to the total amount of monetary gained from all compact prefixes of *A* with respect to *B*.

Example 10. Following Example 9, two compact prefixes of $B = \langle(a)\rangle$ are [sid_{30}:60:60:60] $\langle(a, 30, 3), (b, 30, 4), (e, 45, 2), (c, 55, 8)\rangle$ and [sid_{30}:30:85:85] $\langle(e, 45, 2), (c, 55, 8), (a, 60, 6), (b, 60, 1)\rangle$. Considering the list of item unit prices shown in Fig. 2, the sequence monetary values of the two compact projections are 990 and 600, respectively.

Definition 13 *(The total sequence monetary value).* The *total sequence monetary value* of compact postfix *B* is defined as the sum of all the sequence monetary values of *B* in *SDB*.

$$TSM_{SDB}(B) = \sum_{A \in SDB} SM(B, A)$$

We explain how the proposed algorithm discovers complete RFM-SPs without the downward closure property using the total sequence monetary value. Given a sequence *B*, assuming that $TMscore(B) < Mminsup$ (i.e., *B* is not an RFM-SP), the algorithm will check whether the value of $(TMscore(B) + TSM(B))$ satisfies *Mminsup* or not. If $(TMscore_{SDB}(B) + TSM_{SDB}(B)) \leqq Mminsup$, sequence *B* will be retained for further consideration since its supersets may satisfy *Mminsup*. On the contrary, if $(TMscore_{SDB}(B) + TSM_{SDB}(B)) < Mminsup$, sequence *B* will be discarded directly. In other words, $TMscore_{SDB}(B) + TSM_{SDB}(B)$ represents the maximum monetary value of any possible superset of *B* and holds the downward closure property in the whole mining process.

Moreover, when cumulating the sequence monetary value from compact prefixes, the over-counting problem may exist. This is because identical items may simultaneously exist in two or more compact prefixes. To solve this problem, the timestamp of each occurrence in a compact prefix will be recorded during the counting process. While proceeding to the next compact prefix, the algorithm will compare the timestamps of each item with those recorded in previous occurrences.

Example 11. Consider the list of item unit prices shown in Fig. 2. Following Example 10, two compact prefixes of $B = \langle(a)\rangle$ are [sid_{30}:60:60:60] $\langle(a, 30, 3), (b, 30, 4), (e, 45, 2), (c, 55, 8)\rangle$ and [sid_{30}:30:85:85] $\langle(e, 45, 2), (c, 55, 8), (a, 60, 6), (b, 60, 1)\rangle$. We find that the subsequences $\langle(e, 45, 2), (c, 55, 8)\rangle$ in both two compact prefixes are identical, so they will be counted once only. Hence, we have $SM(B,sid_{30}) = (3 \times 10 + 4 \times 150 + 2 \times 80 + 8 \times 25) + (6 \times 10 + 1 \times 150) = 1200$.

As we mentioned above, the downward closure property no longer holds in the monetary constraint, but another two constraints, recency and frequency, hold for the property. The RFM-PostfixSpan algorithm utilizes these two constraints to reduce the search space and efficiently discovers all RFM-SPs. To make it easier to explain the algorithm, we define the RF-sequential pattern.

Definition 14. Given user-specified recency and frequency thresholds, *Rminsup* and *Fminsup*, we say a sequence *B* is an RF-sequential pattern (RF-SP) if $TRscore_{SDB}(B) \leqq Rminsup$, $TFscore_{SDB}(B) \leqq Fminsup$ and $TMscore_{SDB}(B) + TSM_{SDB}(B) \leqq Mminsup$.

The RFM-PostfixSpan algorithm is shown in Fig. 3. We briefly state the main procedure as follows. Initially, we set $\alpha = null$. For a given α-projected database $SDB|_\alpha$, the algorithm first appends each item in $SDB|_\alpha$ to α to form α'. After that, it calculates $TRscore_{SDB}(\alpha')$, $TFscore_{SDB}(\alpha')$, and $TMscore_{SDB}(\alpha')$ for each α' to find complete sets of RF-SPs and RFM-SPs. Note that the super-sequences of an RF-SP can possibly be RFM-SPs. Therefore, for each RF-SP in α', we construct the projected database $SDB|_{\alpha'}$ and recursively call the procedure RFM-PostfixSpan for finding further RFM-SPs. We use the following example to illustrate the major steps of RFM-PostfixSpan in further detail.

4.1. Find 1-RF-SPs

For the sequence database *SDB* in Fig. 1, let $t_{current} = 110$, $ms\_length = 40$, $Rminsup = 0.05$, $Fminsup = 2$, and $Mminsup = 1500$. Scan *SDB* once and count the $TRscore_{SDB}$, $TFscore_{SDB}$ and $TMscore_{SDB}$ for each item to find the complete set of 1-RF-SPs (i.e., RF-SPs with

Postfix	*TSM*	*sid*	$Mscore_{postfix}$	*start_time*	*end_time*	Projected (prefix) database	*SM*
<(*a*)>	7160	10	20	40	40	<(*a*,10,5),(*c*,30,4)>	150
		20	150	50	50	<(*d*,30,2)>	90
		30	30	85	85	<(*e*,45,2),(*c*,55,8),(*a*,60,6),(*b*,60,1)>	1200
			60	60	60	<(*a*,30,3),(*b*,30,4),(*e*,45,2),(*c*,55,8)>	
		40	1300	100	100	<(*e*,70,5)>	400
		50	900	90	90	<(*c*,70,160),(*b*,85,7),(*d*,85,6)>	5320

Fig. 4. The ⟨(*a*)⟩-projected databases.

Postfix	*TSM*	*sid*	$Mscore_{postfix}$	*start_time*	*end_time*	Projected (prefix) database	*SM*
<(*a*)(*a*)>	360	30	90	60	85	<(*e*,45,2),(*c*,55,8) >	360

Fig. 5. The ⟨(*a*)(*a*)⟩-projected database.

one item only) and 1-RFM-SPs (i.e., RFM-SPs with one item only). The complete set found includes ⟨*a*⟩: (0.549867, 8, 2540, 7160); ⟨*b*⟩: (0.080289, 7, 3000, 5895); ⟨*c*⟩: (0.061061, 5, 4750, 1810); ⟨*d*⟩: (0.896466, 7, 3600, 9480); ⟨*e*⟩: (0.094583, 5, 1440, 3060), where the notation "⟨*pattern*⟩: ($TRscore_{SDB}$, $TFscore_{SDB}$, $TMscore_{SDB}$, TSM_{SDB})" represents the pattern and its associated total recency score, total frequency score, total monetary score and total sequence monetary value. Since all five items satisfy both *Rminsup* and *Fminsup* and their $TMscore_{SDB} + TSM_{SDB}$ satisfies *Mminsup*, we output all items (i.e., {*a*, *b*, *c*, *d*, *e*}) as 1-RF-SPs. In addition, since the *TMscore* of the first four items (i.e., {*a*, *b*, *c*, *d*}) also satisfy *Mminsup*, we output them as 1-RFM-SPs.

4.2. Divide and search

According to the above five 1-RF-SPs, the algorithm can partition the complete set of sequential patterns into five subsets, including (1) those with postfix ⟨*a*⟩, (2) those with postfix ⟨*b*⟩, (3) those with postfix ⟨*c*⟩, (4) those with postfix ⟨*d*⟩, and (5) those with postfix ⟨*e*⟩. Fig. 4 shows the ⟨(*a*)⟩-projected database. For sid_{10}, item *a* occurs at times 10 and 40, and we have the following two compact prefixes, [sid_{10}:50:10:10]: *null* and [sid_{10}:20:40:40]: ⟨(*a*, 10, 5), (*c*, 30, 4)⟩. Since the first compact projection is *null*, it is removed for further processing. Continuing in this manner yields the entire projected database (*a*) as well as the *SM* value of each compact projection. The *TSM* value of compact postfix ⟨(*a*)⟩ can be calculated from all of the *SM* values.

4.3. Find subsets of sequential patterns

We start to find 2-RF-SPs and 2-RFM-SPs with postfix ⟨(*a*)⟩ by calling the algorithm RFM-PostfixSpan(⟨(*a*)⟩, 2, $SDB|_{\langle(a)\rangle}$). The ⟨(*a*)⟩-projected database shown in Fig. 4 consists of six compact projections. For the first compact projection [sid_{10}:20:40:40]: ⟨(*a*, 10, 5), (*c*, 30, 4)⟩, the *Rscore*, *Fscore* and *Mscore* of the two patterns, ⟨(*c*)(*a*)⟩ and ⟨(*a*)(*a*)⟩, will be calculated. Meanwhile, the $TRscore_{SDB}$, $TFscore_{SDB}$, and $TMscore_{SDB}$ of the two patterns, ⟨(*c*)(*a*)⟩ and ⟨(*a*)(*a*)⟩ can be cumulated. At this time, the values $TRscore_{SDB}(\langle(c)(a)\rangle) = 0.0006266$, $TFscore_{SDB}(\langle(c)(a)\rangle) = 1$, $TMscore_{SDB}(\langle(c)(a)\rangle) = 20 + 4 \times 25 = 120$ (i.e., $Mscore_{postfix}$ + number of purchase item × unit price). The values $TRscore_{SDB}(\langle(a)(a)\rangle) = 0.0006266$, $TFscore_{SDB}(\langle(a)(a)\rangle) = 1$, $TMscore_{SDB}(\langle(a)(a)\rangle) = 20 + 5 \times 10 = 70$. For the second projection [sid_{20}:150:50:50]: ⟨(*d*, 30, 2)⟩, the $TRscore_{SDB}(\langle(d)(a)\rangle)$, $TFscore_{SDB}(\langle(d)(a)\rangle)$, and $TMscore_{SDB}(\langle(d)(a)\rangle)$ are increased by 0.001797, 1, and 240, respectively. For the third projection [sid_{30}:30:85:85]: ⟨(*e*, 45, 2), (*c*, 55, 8), (*a*, 60, 6), (*b*, 60, 1)⟩, the $TRscore_{SDB}$, $TFscore_{SDB}$, and $TMscore_{SDB}$ of compact postfixes ⟨(*e*)(*a*)⟩, ⟨(*c*)(*a*)⟩, ⟨(*a*)(*a*)⟩, and ⟨(*b*)(*a*)⟩ will be counted. For the forth projection [sid_{30}:60:60:60]: ⟨(*a*, 30, 3), (*b*, 30, 4), (*e*, 45, 2), (*c*, 55, 8)⟩, since it has the same *sid* as the previous one and parts of the two projections are overlapped, the algorithm omits overlapping to avoid the recounting problem. As a result, the $TRscore_{SDB}$, $TFscore_{SDB}$, and $TMscore_{SDB}$ of compact postfixes ⟨(*e*)(*a*)⟩ and ⟨(*c*)(*a*)⟩ in the forth projection will not be cumulated. Similarly, the remaining compact projections can be traversed and the scores of patterns can be counted in the same manner.

After counting each pattern in ⟨(*a*)⟩-projected database, we have ⟨(*a*)(*a*)⟩: (0.072416, 2, 160); ⟨(*b*)(*a*)⟩: (0.193366, 2, 2130); ⟨(*c*)(*a*)⟩: (0.193993, 3, 5250); ⟨(*d*)(*a*)⟩: (0.123374, 2, 1410); ⟨(*e*)(*a*)⟩: (0.420468, 2, 1890). For the first pattern ⟨(*a*)(*a*)⟩, since it satisfies *Rminsup* and *Fminsup* and $TMscore_{SDB}(\langle(a)(a)\rangle) + TSM_{SDB}(\langle(a)\rangle) = 160 + 7160 = 7320 \rangle Mminsup$, we output ⟨(*a*)(*a*)⟩ as a 2-RF-SP. The algorithm then proceeds to build the ⟨(*a*)(*a*)⟩-projected database as shown in Fig. 5 and find 3-RF-SPs and 3-RFM-SPs with postfix ⟨(*a*)(*a*)⟩ by calling RFM-PostfixSpan(⟨(*a*)(*a*)⟩, 2, $SDB|_{\langle(a)(a)\rangle}$). Continuing in this way yields the complete set of RF-SPs and RFM-SPs in *SDB* as shown in Fig. 6.

5. Experiment evaluation

This section performs a simulation study to empirically compare the proposed algorithm with the PrefixSpan algorithm (i.e., the conventional sequential pattern mining method) (Pei et al., 2004). All the algorithms are implemented in Java language and tested on the Intel core 2 Q8300-2.5 GHz Windows XP system with 4 gigabytes of main memory.

5.1. Synthetic and real-life datasets

Seven synthetic datasets are generated by applying the synthetic data generation algorithm in Chen et al. (2009). Table 1

Table 1
The parameters of synthetic datasets.

Parameters	Description
$\|D\|$	Number of customers
$\|C\|$	Average number of transactions per customer
$\|T\|$	Average number of items per transaction
$\|S\|$	Average length of maximum potential large sequences
$\|I\|$	Average size of itemsets in maximal potentially large sequences
N_S	Number of maximal potentially large sequences
N_I	Number of maximal potentially large itemsets
N	Number of items
T_I	Average length of time intervals
H_price	Average price of high price items
M_price	Average price of medium price items
L_price	Average price of low price items
H_quantity	Average purchased quantity of high price items
M_quantity	Average purchased quantity of medium price items
L_quantity	Average purchased quantity of low price items

	RF-pattern	RFM-pattern
<a>	<a>,<aa>, <ba>, <ca>, <da>, <ea>,<cba>, <aca>,	<a>,<ba>, <ca>, <da>, <ea>,<cba>
<b>	<b>,<ab>, <bb>, <cb>, <(ab)>, <a(ab)> , <c(ab)> , <acb>	<b>,<cb>, <(ab)>, <ac(ab)>
<c>	<c>,<ac>, <bc>, <dc>, <ec>,<(ab)c>, <bdc>, <aec>, <bec>	<c>,<bc>,<(ab)ec>
<d>	<d>,<ad>, <bd>, <cd>, <ed>,<(ab)d>, <cad>, <cbd>	<d>,<ad>, <bd>, <cd>, <dd>, <ed>,<(ab)d> , <cad> , <cbd>
<e>	<e>,<ae>, <be>, <ce>, <de>,<cae> , <(ab)e>, <cbe>	<be>, <ce>,<(ab)e>

Fig. 6. The complete set of RF-SPs and RFM-SPs.

lists the parameters used in the data generation algorithm. Several parameters are fixed in data generation, including $|S| = 4$, $|I| = 1.25$, $N_S = 5000$, $N_I = 25{,}000$, $N = 10{,}000$, $T_I = 10$, $H\_price = 1000$, $M\_price = 500$, $L\_price = 100$, $H\_quantity = 5$, $M\_quantity = 3$, and $L\_quantity = 1$. We vary the value of $|D|$ (from 250K to 750K), $|C|$ (from 10 to 20), and $|T|$ (from 2.5 to 4.5) to perform runtime analyses. The parameter settings of seven synthetic datasets are shown in Table 2.

We also investigate a real-life dataset in our experiments. This dataset contains all the sales data of a supermarket chain in Taiwan. The sales data, called SC-POS, recorded all the transactions from twenty branches between 2001/12/27 and 2002/12/31. Each transaction in SC-POS is a customer's shopping list, which records the purchased items, prices, and quantities. After we perform all necessary data pre-processing tasks, the dataset contains 17,685 items and 145,332 customers' data-sequences.

5.2. Performance evaluation

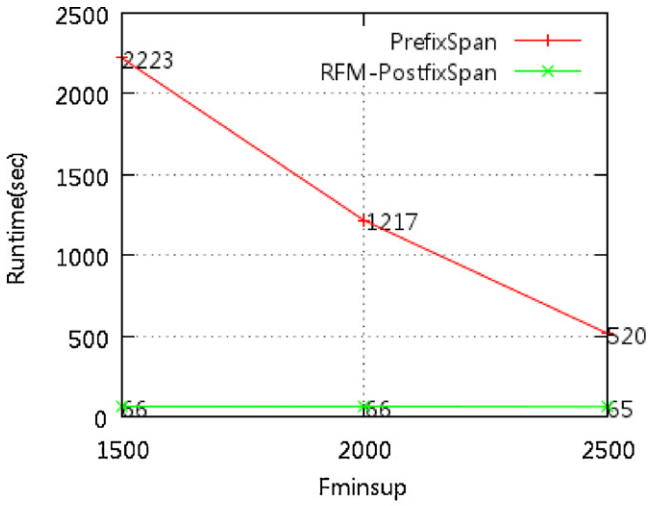
In all the following experiments, three parameters are constant: $ms\_length = 180$, $t\_current = 500$, and $\delta = 0.01$. Since traditional sequential pattern mining only considers frequency threshold, the minimum support of PrefixSpan is set as the same as *Fminsup*.

The first test evaluated the runtime of the two algorithms by varying *Fminsup*. Both the SYN-1 and the SC-POS datasets were used in the test. The parameter settings used in RFM-PostfixSpan are listed below: (1) SYN-1: *Rminsup* = 300, *Fminsup* = 1500–2500, *Mminsup* = 6,000,000; (2) SC-POS: *Rminsup* = 650, *Fminsup* = 3000–5000, *Mminsup* = 250,000. Fig. 7(a) and (b) illustrate the results of SYN-1 and SC-POS, respectively. All the results indicate that, under the same *Fminsup*, the RFM-PostfixSpan outperforms the the PrefixSpan. We observed that, although the mining of RFM-SP is more complicated than that of conventional sequential pattern, the RFM-SP requires less time to complete the pattern search procedure.

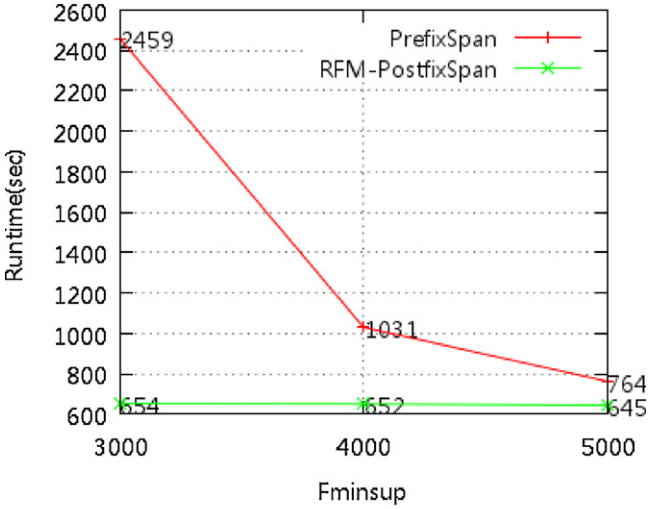
Next, the scalabilities of the two algorithms were performed based on the seven synthetic datasets. Some parameters used in the algorithms were fixed, including *Rminsup* = 300 and *Mminsup* = 6,000,000. The parameter *Fminsup* was set to $|D| \times 0.01$. In the first test, we varied the value of $|D|$ from 250K to 750K. The results in Fig. 8(a) show that both RFM-PostfixSpan and PrefixSpan scale up linearly with $|D|$. We also varied the value of $|C|$ from 5 to 15 (as shown in Fig. 8(b)) and the value of $|T|$ from 1.25 to 3.75 (as shown in Fig. 8(c)). Both two tests indicate that the runtime of both algorithms scale up exponentially with $|C|$ and $|T|$. The results satisfy our expectations. Increasing $|C|$ and $|T|$ leads to the increase of a customer sequence length, resulting in the increase in customer purchase frequencies and the total amount of money they spent.

Table 2
The parameter settings of synthetic datasets.

Name	$\|D\|$	$\|C\|$	$\|T\|$
SYN-1	250K	10	2.5
SYN-2	500K	10	2.5
SYN-3	750K	10	2.5
SYN-4	250K	5	2.5
SYN-5	250K	15	2.5
SYN-6	250K	10	1.25
SYN-7	250K	10	3.75

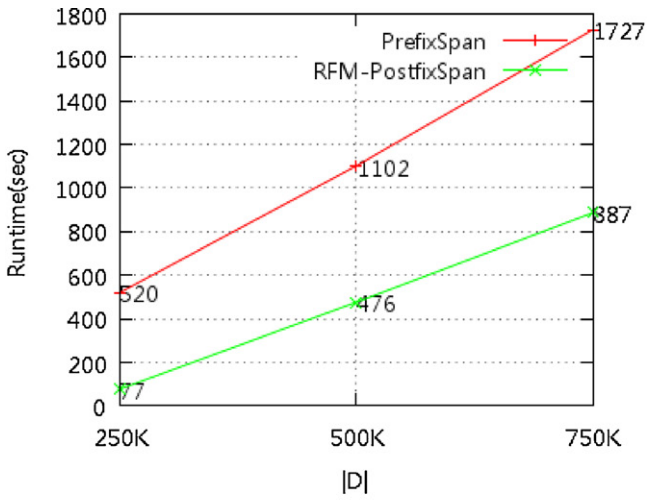


(a) *Runtime* vs. *Fminsup* (SYN-1)

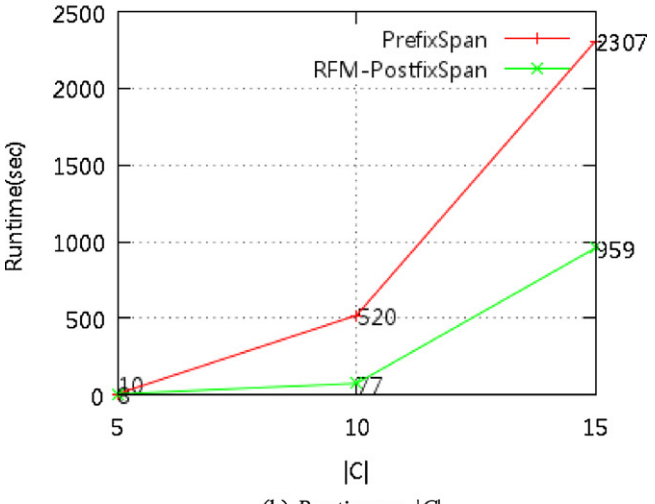


(b) *Runtime* vs. *Fminsup* (SC-POS)

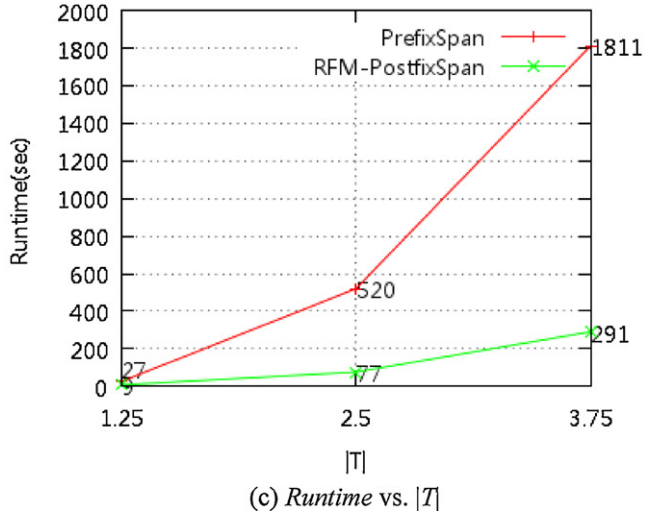
Fig. 7. The runtime analysis with different datasets. (a) Runtime vs. *Fminsup* (SYN-1) and (b) Runtime vs. *Fminsup* (SC-POS).



(a) *Runtime* vs.|*D*|



(b) *Runtime* vs. |*C*|



(c) *Runtime* vs. |*T*|

Fig. 8. The runtime analysis with different synthetic datasets. (a) Runtime vs. |*D*|, (b) Runtime vs. |*C*| and (c) Runtime vs. |*T*|.

Table 3
The results of runtime and # of RFM-SPs using SC-POS.

Rminsup	*Fminsup*	*Mminsup*	# of RFM-SPs
180	3000	200,000	1068
200	3000	200,000	1063
220	3000	200,000	859
200	3000	200,000	1063
200	3400	200,000	983
200	3800	200,000	888
200	3000	150,000	1329
200	3000	175,000	1194
200	3000	200,000	1063

Consequently, both algorithms need to construct more projected databases and then calculate pattern supports.

The results also indicate that, in all above tests, the RFM-PostfixSpan algorithm outperforms PrefixSpan. This is because while more constraints (i.e., recency and monetary constraints) are added, more uninteresting patterns can be removed. Thus, fewer patterns have to be processed and the search space (i.e., both the size and the number of projections) can be significantly reduced, resulting in the decrease of both runtime and memory consumption.

Next, we use several tests to evaluate the scale-up effect for three support thresholds unique to RFM-SPs based on SC-POS. Each time we only vary one support threshold and keep the other two constant. The results are shown in Table 3. We can see that as *Rminsup*, *Fminsup*, or *Mminsup* increase, the number of RFM-SPs decrease. The results also show that, among the three thresholds, monetary constraints can significantly reduce the number of patterns. This is quite useful if a business is focusing on discovering high-profit patterns. In summary, through setting adequate recency, frequency and monetary thresholds simultaneously, we retrieve more compact, representative and useful patterns in less time.

Next, we evaluated the value of patterns generated from the two algorithms based on the SC-POS dataset. For each pattern, we retrieve all transactions containing the pattern and calculate its revenue as the value of a pattern. Table 4 lists the number of patterns, the revenue of patterns, and the average revenue per pattern for the two algorithms by varying *Fminsup*. The results showed that, on average, RFM-PostfixSpan retains only about 81% of traditional sequential patterns (SP) but takes nearly 94% of the revenue of traditional SP. The average revenue per RFM-SP is also about 20 percent higher than that of traditional SP; that is, the set of RFM-SPs holds sequential patterns with relatively higher value.

Finally, we divided the SC-POS dataset into two sub-datasets, the training and the testing datasets, to investigate the effectiveness of the two algorithms. 70% of the original dataset is imposed on discovering patterns (Period-1: from 2001/12/28 to 2002/07/19), and 30% is used to test the predictive power of patterns (Period-2: from 2002/07/20 to 2002/10/15). Specifically, at Period-2, we evaluate the revenue gained from the patterns of Period-1. The decay speed is set as 0.001. The numbers of patterns found by the two algorithms at Period-1, the total revenue of the patterns at Period-2, and the average revenue per pattern at Period-2, are shown in Table 5.

The same as the findings in Table 4, the results all indicate that although the revenue of RFM-SP is not higher than that of SP, the average revenue of the former is higher than that of the latter. That is, each RFM-SP, in general, can hold more values than each conventional SP. Also, when we want to refer to the discovered results using sequential pattern mining, the former can provide a smaller number of patterns so that we can greatly remedy the problem of information overloading.

Table 4
The comparison of pattern values generated from two algorithms (SC-POS).

Parameter settings	*Rminsup*	200	200	200
	Fminsup	3000	3200	3400
	Mminsup	250,000	250,000	250,000
# of patterns	SP	1264	1059	957
	RFM-SP	932	896	856
Revenue (NTD)	SP	$4,263,928,263	$1,092,447,907	$891,541,577
	RFM-SP	$3,970,950,259	$995,916,410	$901,979,165
Average revenue per pattern (NTD)	SP	$3,373,361	$1,031,584	$931,600
	RFM-SP	$4,260,676	$1,111,514	$1,053,714

Table 5
The evaluation results using the training and testing datasets (SC-POS).

Parameter settings	*Rminsup*	600	600	600
	Fminsup	600	900	1200
	Mminsup	100,000	120,000	140,000
# of patterns at Period-1	SP	10,819	4448	2470
	RFM-SP	2991	2303	1406
Revenue at Period-2 (NTD)	SP	$1,793,468,478	$1,084,650,033	$765,139,355
	RFM-SP	$767,264,498	$672,672,666	$441,224,787
Average revenue per pattern at Period-2 (NTD)	SP	$166,370	$244,842	$310,275
	RFM-SP	$356,205	$435,669	$445,682

6. Conclusion

Sequential pattern mining is a useful method to discover customer purchasing behavior from large sequence databases. In this study, we propose a novel sequential pattern mining technique, which considers recency, frequency and monetary constraints, based on the concept of RFM analysis. A new type of sequential pattern, called RFM-SP, is defined, and an efficient projected-based algorithm, called RFM-PostfixSpan, is proposed to discover a complete set of RFM-SPs from a sequence database.

Seven synthetic datasets and a real-life dataset are used in our experiments. The results show that the proposed method is efficient and outperforms the traditional PrefixSpan algorithm in both runtime and the numbers of generated patterns. In practice, the RFM-PostfixSpan not only significantly reduces the runtime, but also retains more meaningful results for users.

The results of this study could be extended by further research. For example, future work could include fuzzy recency, frequency and monetary constraints, which would lead to a more flexible ways to uncover other meaningful patterns. The development of a desirable maintenance mechanism is also critical for users to properly tune the parameters in the mining process. Furthermore, the RFM-PostfixSpan algorithm could be useful in the market-basket analysis for various types of applications.

Acknowledgements

The authors would like to thank the Area Editor, Dr. K. Dutta, and anonymous reviewers for their helps and valuable comments to improve this paper. This research was supported by the National Science Council of the Republic of China under the grants NSC 100-2410-H-194-024-MY2 and NSC 100-2410-H-194-019-MY2.

References

Agrawal, R., Srikant, R., 1995. Mining sequential patterns. In: Proceedings of the Eleventh International Conference on Data Engineering, pp. 3–14.
Ahmed, C.F., Tanbeer, S.K., Jeong, B.-S., Lee, Y.-K., 2009. Efficient tree structures for high utility pattern mining in incremental databases. IEEE Transactions on Knowledge and Data Engineering 21 (12), 1708–1721.
Chen, S.S., Huang, T.C.K., Lin, Z.M., 2011. New and efficient knowledge discovery of partial periodic patterns with multiple minimum supports. Journal of Systems and Software 84 (10), 1638–1651.
Chen, M.S., Han, J., Yu, P.S., 1996. Data mining: an overview from a database perspective. IEEE Transaction on Knowledge and Data Engineering 8 (6), 866–886.
Chen, Y.L., Huang, T.C.K., 2005. Discovering fuzzy time-interval sequential patterns in sequence databases. IEEE Transactions on Systems, Man and Cybernetics, Part B 35 (5), 959–972.
Chen, Y.L., Chiang, M.C., Ko, M.T., 2003. Discovering time-interval sequential patterns in sequence databases. Expert Systems with Applications 25 (3), 343–354.
Chen, Y.L., Hu, Y.H., 2006. Constraint-based sequential pattern mining: the consideration of recency and compactness. Decision Support Systems 42 (2), 1023–1215.
Chu, C.-J., Tseng, V.S., Liang, T., 2008. An efficient algorithm for mining temporal high utility itemsets from data streams. Journal of Systems and Software 81 (7), 1105–1117.
Chen, Y.L., Kuo, M.H., Wu, S.Y., Tang, K., 2009. Discovering recency, frequency, and monetary (RFM) sequential patterns from customer's purchasing data. Electronic Commerce Research and Application 8 (5), 241–251.
Cheng, C.H., Chen, Y.S., 2009. Classifying the segmentation of customer value via RFM model and RS theory. Expert Systems with Applications 36 (3), 4176–4184.
Chiang, D.A., Wang, C.T., Chen, S.P., Chen, C.C., 2009. The cyclic model analysis on sequential patterns. IEEE Transactions on Knowledge and Data Engineering 21 (11), 1617–1628.
Eirinaki, M., Vazirgiannis, M., 2003. Web mining for web personalization. ACM Transactions on Internet Technology 3 (1), 1–27.
Erwin, A., Gopalan, R.P., Achuthan, N.R., 2007. CTU-Mine: an efficient high utility itemset mining algorithm using the pattern growth approach. In: 7th IEEE International Conference on Computer and Information Technology, Aizu-Wakamatsu, Fukushima, Japan, pp. 71–76.
Frawley, W.J., Shapiro, P.G., Matheus, C.J., 1992. Knowledge discovery in databases:an overview. AI Magazine 13 (3), 57–70.
Han, J.W., Kamber, M., 2006. Data Mining: Concepts and Techniques. Academic Press, New York.
Huang, C.L., Huang, W.L., 2010. Handling sequential pattern decay: developing a two-stage collaborative recommender system. Electronic Commerce Research and Applications 8 (3), 117–129.
Hughes, A.M., 1994. Strategic Database Marketing. Probus Publishing Company.
Hughes, A.M., 1996. Boosting response with RFM. Marketing Tools 5, 4–10.
Huang, S.C., Chang, E.C., Wu, H.H., 2009. A case study of applying data mining techniques in an outfitter's customer value analysis. Expert System Application 36 (3), 5909–5915.
Kim, C., Lim, J.H., Ng, R.T., Shim, K., 2007. SQUIRE: sequential pattern mining with quantities. Journal of Systems and Software 80 (10), 1726–1745.
Lin, F.R., Huang, K.J., Chen, N.S., 2006. Integrating information retrieval and data mining to discover project team coordination patterns. Decision Support Systems 42 (2), 745–758.
Liu, D.R., Shih, Y.Y., 2005a. Integrating AHP and data mining for product recommendation based on customer lifetime value. Information and Management 42 (3), 387–400.
Liu, D.R., Shih, Y.Y., 2005b. Hybrid approaches to product recommendation based on customer lifetime value and purchase preferences. Journal of Systems & Software 77 (2), 181–191.
Liu, D.R., Lai, C.H., Lee, W.J., 2009. A hybrid of sequential rules and collaborative filtering for product recommendation. Information Sciences 179 (20), 3505–3519.
Mannila, H., Toivonen, H., Verkamo, A.I., 1997. Discovery of frequent episodes in event sequences. Data Mining and Knowledge Discovering 1 (3), 259–289.
Miglautsch, J.R., 2002. Application of RFM principles: what to do with 1–1–1 customers? Journal of Database Marketing 9, 319–324.
Mannila, H., 1998. Database methods for data mining. In: ACM SIGKDD Conference on Knowledge Discovery & Data Mining, New York, USA.

Pei, J., Han, J., Mortazavi-Asl, B., Zhu, H., 2000. Mining access patterns efficiently from web logs. In: Proceeding of the 2000 Pacific-Asia conference on knowledge discovery and data mining (PAKDD'00), Kyoto, Japan, pp. 396–407.
Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., Hsu, M.-C., 2004. Mining sequential patterns by pattern-growth: the PrefixSpan approach. IEEE Transaction on Knowledge and Data Engineering 16 (11), 1424–1440.
Hu, Y.H., Kao, Y.H., 2011. Mining sequential patterns with consideration to recency, frequency, and monetary. In: In the 15th Pacific Asia Conference on Information Systems, Brisbane, Australia, Paper 78.
Hu, J., Mojsilovica, A., 2007. High-utility pattern mining: a method for discovery of high-utility item sets. Pattern Recognition 40 (11), 3317–3324.
Shaw, M.J., Subramaniam, C., Tan, G.W., Welge, M.E., 2001. Knowledge management and data mining for marketing. Decision Support Systems 31 (1), 127–137.
Srikant, R., Yang, Y., 2001. Mining web logs to improve website organization. In: Proceedings of the 10th International World Wide Web Conference, Hong Kong, pp. 430–437.
Tao, Y.H., Hong, T.P., Su, Y.M., 2008. Web usage mining with intentional browsing data. Expert Systems with Applications 34 (3), 1893–1904.
Yao, H., Hamilton, H.J., 2006. Mining itemset utilities from transaction databases. Data & Knowledge Engineering 59 (3), 603–626.
Yeh, I.C., Yang, K.J., Ting, T.M., 2009. Knowledge discovering on RFM model using Bernoulli sequence. Expert System with Applications 36 (3–2), 5866–5871.

Ya-Han Hu is currently an Assistant Professor of Department of Information Management at National Chung Cheng University, Taiwan. He received a PhD degree in Information Management from National Central University of Taiwan in 2007. His current research interests include data mining and knowledge discovery, decision support systems, and EC technologies. His research has appeared in Artificial Intelligence in Medicine, Data & Knowledge Engineering, Decision Support Systems, IEEE Transactions on Systems, Man, and Cybernetics – Part C, and Journal of Information Science.

Tony Cheng-Kui Huang received the PhD degree in Information Management from National Central University of Taiwan in 2006. He is an Associate Professor in the Department of Business Administration, National Chung Cheng University of Taiwan. His current research interests include data mining, decision support systems, soft computing, and IS/IT adoption. He has published papers in IEEE Transactions on Systems, Man and Cybernetics – Part B, Information Sciences, International Journal of Information Management, Data & Knowledge Engineering, Fuzzy Sets and Systems, Journal of Systems and Software, Applied Soft Computing, Computers & Education, International Journal of Innovative Computing, Information and Control, and Expert Systems with Applications.

Yu-Hua Kao is a system engineer in Quanta Storage Inc., Taiwan. She received her MS degree in Information Management from National Chung Cheng University of Taiwan. Her research interests include data mining, information systems, and EC technologies.