Fast RFM Model for Customer Segmentation

Shicheng Wan
Guangdong University of Technology
Guangzhou, China
scwan1998@gmail.com

Jiahui Chen
Guangdong University of Technology
Guangzhou, China
csjhchen@gmail.com

Zhenlian Qi
Guangdong University of Technology
Guangzhou, China
qzlhit@foxmail.com

Wensheng Gan*
Jinan University
Guangzhou, China
wsgan001@gmail.com

Lilin Tang
Harbin Institute of Technology
Shenzhen, China
hittang@126.com

ABSTRACT

With booming e-commerce and World Wide Web (WWW), a powerful tool in customer relationship management (CRM), called the RFM analysis model, has been used to ensure that major enterprises make more profit. Combined with data mining technologies, the CRM system can automatically predict the future behavior of customers to raise customer retention rate. However, a key issue is that the existing RFM analysis models are not efficient enough. Thus, in this study, a fast algorithm based on a compact list-based data structure is proposed along with several efficient pruning strategies to address this issue. The new algorithm considers recency (R), frequency (F), and monetary/utility (M) as three different thresholds to discover interesting patterns where the R, F, and M thresholds combined are no less than the user-specified minimum values. More significantly, the downward-closure property of frequency and monetary metrics are utilized to discover super-itemsets. Then, an extensive experimental study demonstrated that the algorithm outperforms state-of-the-art algorithms on various datasets. It is also demonstrated that the proposed algorithm performs well when considering the frequency metric alone.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; **Data analytics**.

KEYWORDS

RFM analysis, customer segmentation, RFM pattern.

ACM Reference Format:
Shicheng Wan, Jiahui Chen, Zhenlian Qi, Wensheng Gan, and Lilin Tang. 2022. Fast RFM Model for Customer Segmentation. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3487553.3524707

*Corresponding author, also with Pazhou Lab, Guangzhou, 510330, China

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524707

1 INTRODUCTION

The World Wide Web (WWW), commonly known as the Web, has led to worldwide information retrieval service and data intelligence. According to the Chinese e-Commerce report in 2019 [1], the total transaction volume of e-commerce was about 5.2 trillion dollars, including 1.7 trillion dollars of online retail sales, up by 16.5% from the preceding year. Encouraged by the boom in e-commerce, though many retail enterprises have already applied data mining technologies to marketing strategies [1, 11, 22], traditional high-utility pattern mining (HUPM) [4, 8, 10, 20, 28] cannot meet the requirements of the developing business needs of these enterprises. In the past, in most cases, database marketers focused on analyzing the commodity sale results directly. They often did not conduct an in-depth analysis as to what customers are willing to pay or why they hardly buy their products anymore. Thus, it is necessary to analyze customer behavior to offer better services. At present, the main idea of database marketing is to improve the effectiveness of service, increase the volume of sales, and bring more profits to companies. Behavioral segmentation offers more accurate behavior prediction than other methods because it provides sufficient information on customer shopping preferences [12]. Based on an analysis of customers' shopping behavior, product-centric enterprises can offer accurate suggestions to their customers, provide diversified products, and even personal services. For example, if some customers purchase only a few times, companies can improve the purchasing willingness by sending discount messages regularly; if consumers are shopping often, companies can recommend relatively more novel commodities to them; and for regular customers, companies can offer new products free in advance to enhance their loyalty. Overall, retailers adopt diverse marketing strategies for various customers to extend their life cycle. In other words, the business model of these companies is changing into a customer-centric model. Database marketing strategies based on customer behavior analysis have gradually become a powerful and competitive tool in major enterprises [27].

Customers are the main resources for profits, and customer relationship management (CRM) [19, 32] is a key technology that effectively increases profits. A satisfactory relationship between companies and their customers is important for the success of commercial enterprises in their marketing endeavor. How to offer the best-matched service to customers is a challenging task. Over the past decades, the concept of the RFM analysis model

[1] https://dzswgf.mofcom.gov.cn/news/5/2020/10/1602480531631.html

has been successfully applied in various data mining domains [5, 14, 15, 17, 23, 35, 35, 36]. The RFM analysis model has been widely adopted in real-world applications, such as protecting security of computers [18], the automobile industry [2, 23, 24], and optimizing electronics industry supply [6]. The ideas on how to utilize the RFM analysis technique can be divided into three methods. The first method combines clustering of the target customers with the K-means algorithm to identify the general trend in customers' true value and loyalty. Then, the original idea of the RFM analysis model was to divide customers into five groups and adopt distinct market strategies for target customers who are at different levels. Thus, it is natural that the second method of RFM analysis is called classification. The last method uses utility/frequent pattern-mining algorithms. It takes RFM scores as constraints to discover valuable customers, referred to as RFM customers. An RFM-customer has high recency (R), frequency (F), and monetary (M) scores. In other words, RFM customers have a strong willingness to interact with a company (high loyalty) and bring more profit to the company over a long period of time. However, in most cases, obtaining exact customer identification information is not accepted by the public. For example, when a customer orders takeaway food online, his/her IP address and device are not allowed to be published, because this information can reveal accurate user profiles. On the other hand, if customers check out by cash instead of using other payment options requiring ID information, or if different people share the same membership card, it becomes more difficult to determine the exact RFM customer. Therefore, combining the RFM analysis model with data mining technologies to discover interesting patterns in transactional databases without identifying customer information is a difficult. Fortunately, Hu and Yeh [16] first proposed the RFMP-Growth algorithm and addressed this issue. Because a database for customer identification information is lacking [16], they redefine customers as *RFM-patterns*, where RFM-customers are a subset of *RFM-patterns*. The mining task is revised to discover a set of interesting *RFM-patterns*. However, since 2014, there have been few studies in the literature dedicated to developing advanced algorithms. Motivated by this previous work, this paper focuses on proposing a novel RFM mining algorithm that is more efficient and scalable than RFMP-Growth.

In this study, the problem of fast data mining for RFM analysis is addressed by proposing the **RFM**-pattern **U**tility **L**ist-based mining algorithm (simplified as **RFMUL**) to efficiently discover a complete *RFM-pattern* set from a transaction database. The new algorithm extends the vertical structure of HUI-Miner [25] and FHM [7]), called RFM-List, to compress all vital information on potential *RFM-patterns*. It also adopts several efficient pruning strategies to reduce the search space. The main contributions of this study are as follows:

- This paper proposes an effective and efficient algorithm aims at mining a complete set of *RFM-patterns* from transactional databases without customer identification information.
- To avoid scanning databases multiple times and reduce the search space, a novel and compact list-based data structure called RFM-List is proposed to store key information during mining to avoid scanning the databases numerous times, thereby reducing memory consumption.
- Based on two upper-bounds, several pruning strategies are utilized to improve the mining process to save on execution time for efficiency.
- Extensive experimental evaluations have been conducted on both real and synthetic datasets to evaluate the proposed algorithm. The performance of RFMUL was also compared with the state-of-the-art algorithm, RFMP-Growth, both in terms of runtime and memory usage. The experiments show that RFMUL performs better than RFMP-Growth.

The remainder of this paper is organized as follows. In Section 2, related work is discussed. In Section 3, some basic preliminaries and the problem statement of RFM pattern mining are introduced. In Section 4, the details of our novel algorithm are provided. The experimental results are presented in Section 5. Conclusions and future work are presented in Section 6.

2 RELATED WORK

In this section, some studies on combining the RFM model and data mining techniques are reviewed. To the best of our knowledge, studies using the RFM analysis model in the data mining domain can be roughly divided into two types based on whether they 1) implement the RFM analysis model directly, and then segment target consumers into different groups; or 2) take R, F, and M scores as distinct constraints (i.e., thresholds) to measure the pattern value. However, in both approaches, the implicit assumption is that the databases contain customer identification information.

2.1 Web Mining Approaches Adopting RFM Analysis

Web service and data intelligence are commonly seen and powerful. For instance, Web data mining on rich type of data can achieve the segmentation of customer groups. Tavakoli *et al.* [36] pointed out that the parameters of the RFM analysis model were independent in previous studies. Thus, there is a lack of knowledge regarding the internal relation of user behavior records. They discussed the relationship between R, F, and M dimensions of the RFM analysis model and concluded that a connection exists between F and M such that the higher the frequency, the more the monetary profit is. In other words, the RFM analysis model offers useful knowledge to managers. It can be used to predict customers' future behavior online/offline, such as whether the client will visit the market soon, how frequently consumers shop, and how much they spend. Recently, integrating the classification method and RFM analysis was studied by Olson *et al.* [29], who analyzed the response possibilities of customers for the promotion of a specific product. They discussed the relative trade-off among data mining algorithms (e.g., logistic regression, decision tree, and neural network) in the context of customer segmentation. Cheng and Chen [5] also proposed an algorithm called LEM2, which combines RFM attributes and rough set theory. LEM2 aims to discover classification rules to help enterprises determine consumer features, which can strengthen customer relationship management (CRM). They adopted the RFM analysis model along with CRM to maximize profits. In addition, to prove the accuracy of the final classification rules, they experimented with the algorithm and then compared three different methods: decision tree, artificial neural networks, and naive Bayes.

The final results show that LEM2 outperforms the other algorithms in terms of accuracy. With respect to changes in customer requirements, Ha [12] adopted a decision tree to track changes in the RFM values. Then, they discovered classification rules to help predict RFM values in the future from current customer records.

2.2 Data Mining Utilizing RFM Variables

Data mining is a powerful tool [3, 9]. In early year, Pei *et al.* [30] developed the notion of convertible constraints. In 2007, they completed a case study in which constraints can be effectively and efficiently introduced into deeper levels to sequentially mine patterns under their proposed framework [31]. There have also been some studies [16, 17, 20, 21] on how to combine various constraints and high-utility pattern mining. However, as discussed in Section 1, there is plenty of information that should not be disclosed or collected. That is, the record database may not contain sensitive data (e.g., customer identification) in most cases. It seems impossible to integrate RFM variables into data mining techniques when key information is lost. RFM customers are hidden in transaction databases, and they can be identified exactly. Hu and Yeh [16] first retrieved *RFM-patterns* without customer identification from transaction databases. They proposed a tree-based mining algorithm to successfully discover a comprehensive set of *RFM-patterns* without using customer identification information. An *RFM-pattern* is denoted as a promising pattern, when the R, F, and M values are no less than the user-specified minimum R, F, and M thresholds. The recency value depends on the occurrence of a pattern in a specified period. Frequency is defined as the total number of times a pattern appears. Monetary value refers to the utility of the pattern. Subsequently, according to human behavior (recency effect), Kim *et al.* [17] found that the closer previous events are, the more important a role they play in the decision-making process of the user. Although the novel algorithm does not consider the M value, it still demonstrates the availability of the addition of other useful constraint variables in finding interesting patterns during the data mining process.

RFMP-Growth adopts an extended version of the FP-tree structure [13], and thus it has to scan the database multiple times during the mining process when a large number of candidates exist. The RFM-pattern-tree also requires massive memory resources when the database is dense. These problems motivated us to develop a novel algorithm that adopts a more efficient framework to mine *RFM-patterns* without private information.

3 PRELIMINARIES

3.1 Basic Concepts

In this subsection, most notations and definitions are given in Ref. [16], and some are listed in Table 1. In particular, user-specified minimal thresholds α is given as percentage, and β is a positive value. In the following, without loss of generality, it is assumed that the database discussed always lacks sensitive information such as customer ID and IP address. In addition, Table 2 presents a simple transaction database as our running example. It consists of ten transactions and six distinct items {A, B, C, D, E, F}. The corresponding $p(x_i)$ of each item x_i are \$3, \$15, \$1, \$5, \$10, and \$12, respectively.

Referring to studies [16, 33], the more recent the occurrence of a pattern, the higher the recency value is. Herein, our novel algorithm and RFMP-growth both adopt the same formula for calculating recency.

Table 1: A basic notion table.

Symbol	Description
x_i	An item (e.g., goods and products).
I	A finite set of distinct items, I = $\{x_1, x_2, \ldots, x_n\}$.
X	A finite subset of I.
T_j	A transaction w.r.t a set of distinct items with a unique ID (*TID*).
$\mathcal{D}$	A multiset of transactions, $\mathcal{D}$ = $\{T_1, T_2, \ldots, T_m\}$.
$q(x_i, T_j)$	A positive internal utility (e.g., quantity) of x_i in T_j.
$p(x_i)$	A positive external utility (e.g., unit profit) belongs to an item x_i.
$u(x_i, T_j)$	The monetary/utility of x_i in T_j, $p(x_i) \times q(x_i, T_j)$.
α	A user-specified minimum frequency threshold.
β	A user-specified minimum monetary threshold.
γ	A user-specified minimum recency threshold.

Definition 3.1. (**Recency pattern**). Given an itemset X in a transaction T_j, the recency value of X in T_j is defined as $R(X, T_j)$ = $(1-\delta)^{T_{last}-T_{current}}$[2]. Specifically, δ is a user-specified decay speed, where $0 < \delta < 1$ and $|\mathcal{D}|$ represents the total number of transactions containing in $|\mathcal{D}|$. In this study, it is assumed that *TID* is the recording timestamp of each transaction. Then, the last transaction is the closest. That is, $R(X, T_j)$ = $(1-\delta)^{|\mathcal{D}|-j}$. Furthermore, the summary of recency values of all transactions containing X is denoted by $R(X)$ = $\sum_{T_j \subseteq \mathcal{D}} R(X, T_j)$. It is assumed that X is an *R-pattern* if $R(X)$ is no less than γ.

Definition 3.2. (**Frequency pattern**). Frequency value represents the number of times an itemset occurs in a database $\mathcal{D}$, denoted as $F(X)$. An itemset X is called a *F-pattern* only if its $F(X)$ is higher than or equal to $|\mathcal{D}| \times \alpha$ (i.e., $F(X) \geq |\mathcal{D}| \times \alpha$).

Definition 3.3. (**Monetary pattern**). Given an itemset X in a transaction T_j, the monetary value of X in T_j is defined as $M(X, T_j)$ = $\sum_{x_i \in X} u(x_i, T_j)$. Furthermore, the monetary value of X in $\mathcal{D}$ is denoted by $M(X)$ = $\sum_{T_j \subseteq \mathcal{D}} M(X, T_j)$. If $M(X)$ of an itemset is no less than the minimal utility threshold, then X is a *M-pattern*, that is, $M(X) \geq \beta$. In addition, in this paper, the monetary value is also regarded as utility.

3.2 Problem Formulation

Definition 3.4. (**RFM-pattern**) If an itemset X satisfies the three following constraints: 1) $R(X) \geq \gamma$; 2) $F(X) \geq |\mathcal{D}| \times \alpha$; and 3) $M(X) \geq \beta$. Then, X is assumed to be an *RFM-pattern*.

Problem Statement. Based on the definitions so far introduced, given a transaction database $\mathcal{D}$ with three minimal thresholds for recency, frequency, and utility, the problem of the *RFM-pattern*

[2]There are also many other expressions that can be adopted, such as in [33].

Table 2: A sample transaction database.

TID	Transaction
T_1	$(A, 3)$ $(B, 2)$ $(D, 3)$
T_2	$(A, 2)$ $(D, 4)$ $(E, 2)$
T_3	$(A, 3)$ $(C, 5)$ $(F, 3)$
T_4	$(A, 1)$ $(C, 3)$ $(E, 1)$ $(F, 2)$
T_5	$(A, 1)$ $(D, 3)$ $(E, 2)$
T_6	$(A, 1)$ $(B, 2)$ $(D, 4)$
T_7	$(A, 2)$ $(B, 3)$ $(C, 2)$ $(E, 1)$ $(F, 1)$
T_8	$(F, 2)$
T_9	$(C, 3)$ $(D, 3)$
T_{10}	$(A, 3)$ $(D, 4)$

mining task is described as identifying a complete set of *RFM-patterns* in $\mathcal{D}$.

Example 1: Given an itemset {*A*, *E*}, assume α = 20%, β = \$43, γ = 3 and δ = 0.01. Then, $R(\{A, E\}) = R(\{A, E\}, T_2) + R(\{A, E\}, T_4) + R(\{A, E\}, T_5) + R(\{A, E\}, T_7) = 3.79$, Obviously, the total support of $\{A, E\}$ is 4, which is higher than 10 × 20%. Furthermore, because $M(\{A, E\})$ = \$78 > \$43, itemset {*A*, *E*} is an *RFM-pattern*. However, because of $F(\{A, B, C, E, F\}) = 1$, $\{A, B, C, E, F\}$ is not an *RFM-pattern*. Consider another itemset $\{A, C, E\}$, and its $M(\{A, C, E\})$ = \$34. Hence, it is not an *RFM-pattern*.

As introduced by Hu *et al.* [16], it is difficult to simultaneously consider three constraints during mining. Thus, the focus here is mainly to improve the efficiency and effectiveness of the utility metric. In this section, a new model and some basic operations for the novel algorithm in transaction databases are described.

3.3 Downward-Closure Property

The huge volumes of data, make the effective reduction of the search space a vital challenge in the data-mining domain. A widely accepted strategy is to utilize the downward-closure property, which avoids searching many useless patterns. The downward-closure property, was also adopted in this study for frequency and utility and is introduced as follows.

It is clear that any sub-itemset of a frequent itemset cannot be infrequent [1]. In other words, if an itemset is infrequent, it is not necessary to explore its super-itemsets.

As mentioned in *Example 1*, when an item has non-binary purchase quantities (namely internal utility) and unit profit (namely external utility) to indicate relative importance, the utility of the item is a numeric function that is neither monotonic nor anti-monotonic. This indicates that the utility of an itemset may be higher, lower, or equal to the utility values of its subsets. Therefore, the downward-closure property of the frequency metric cannot be used directly. Fortunately, a novel concept called transaction-weighted utilization [26] addresses this issue well.

Definition 3.5. (**Transaction utility**). Let there be a transaction T_j, its corresponding utility is the summation of the utilities of items T_j contains, which is denoted as $TU(T_j) = \sum_{x_i \in T_j} u(x_i, T_j)$.

Definition 3.6. (**Transaction-weighted utilization**). Given an itemset X, the transaction-weighted utilization (abbreviated as *TWU*) is defined as the sum of transaction utilities which contain X. Formally, $TWU(X) = \sum_{X \subseteq T_j} TU(T_j)$. Obviously, *TWU* is a loose upper-bound because the real utility of any itemset in T_j must be lower than or equal to the $TU(T_j)$ value[3]. Hence, X is a potential high-utility itemset (simplified as *pHUI*) if its *TWU* value is no less than β.

Table 3: Transaction-weighted utilization of items.

Item	*A*	*B*	*C*	*D*	*E*	*F*
TWU	\$385	\$182	\$183	\$238	\$199	\$189

Definition 3.7. (**RFT-pattern**). In a transaction database $\mathcal{D}$, if any itemset X is *R-pattern*, *F-pattern*, and *pHUI* at the same time, it is denoted as an *RFT-pattern*. This means that X is a potential *RFM-pattern*. Furthermore, *RFM-patterns* and any subset of an *RFT-pattern* are both *RFT-patterns* [16].

4 THE RFMUL ALGORITHM

4.1 RFM-List Structure

Definition 4.1. (**Global order**). Let $\prec$ be a global order on I (e.g., the lexicographical order). In this study, $\prec$ is defined as an ascending order based on the *TWU* value. For example, from Table 3, there is "$B \prec C \prec F \prec E \prec D \prec A$". Then, the proposed algorithm appends one item at a time to itemset to generate high-level itemsets by following the global order $\prec$.

Definition 4.2. (**Remaining utility** [25]). Based on the global order $\prec$, the set of items after X in T_j is defined as T_j/X. Then, the corresponding utility of the remaining itemset is denoted as $ru(X, T_j) = \sum_{x_i \in T_j/X} u(x_i, T_j)$. The remaining utility can be interpreted as the amount of profit that can be made when other items in T_j are appended to X.

Definition 4.3. (**RFM-List structure**). As shown in Figure 1, an RFM-List is a different highly compressed list structure of each itemset X and denoted as a set of tuples (<*tid*, *iutil*, *rutil*>). The term *tid* represents the identification of a transaction that contains X. The term *iutil* is the real utility of X, and *rutil* denotes the remaining utility of X in *tid*. Specifically, the symbol X_{List} is adopted to represent the RFM-List of X.

Definition 4.4. (**Join operation**). In order to construct RFM-List structure of an l-itemset ($l \geq 2$) without scanning the database multiple times, the lists of two distinct (l-1)-itemsets containing some of the same *tids* are directly joined. Then, with respect to the utility of the l-itemset, the term *ituil* is the summation of the corresponding utility of two constituent itemsets, where *rutil* is dependent on the smaller one.

Example 2: In Figure 1, the two RFM-lists A_{List} and E_{List} have the same four *tids* (i.e., T_2, T_4, T_5 and T_7). The mathematical formula is *iutil*({*A*, *E*}, T_2) = *iutil*({*A*}, T_2) + *iutil*({*E*}, T_2) = \$26. In addition, the *rutil* of {*A*, *E*} is \$0 in T_2 because *rutil*({*A*}, T_2) < *rutil*({*E*}, T_2).

[3] Due to the limited space of this paper, preclude us from providing the details of the proof. For details, please refer to [26].

{ C }		
tid	***iutil***	***rutil***
T_3	5	45
T_4	3	37
T_7	2	28
T_9	3	15

{ D }		
tid	***iutil***	***rutil***
T_1	15	9
T_2	20	6
T_5	15	3
T_6	20	3
T_9	15	0
T_{10}	20	9

{ A }		
tid	***iutil***	***rutil***
T_1	9	0
T_2	6	0
T_3	9	0
T_4	3	0
T_5	3	0
T_6	3	0
T_7	6	0
T_{10}	9	0

●●●

{ E }		
tid	***iutil***	***rutil***
T_2	20	26
T_4	10	3
T_5	20	18
T_7	10	6

{ F }		
tid	***iutil***	***rutil***
T_3	36	9
T_4	24	13
T_7	12	16
T_8	24	0

Figure 1: The RFM-List structures of 1-itemsets.

According to the RFM-List structure, all the key information in itemsets is saved after scanning the database for the first time. It is not necessary to check an *RFT-pattern* by scanning the database again, which is different from that of RFMP-Growth. Following the definition of HUI-Miner [25], this kind of mining framework is regarded as "one phase". Our study is the first to use the "one phase" algorithm to discover *RFM-patterns*.

4.2 Efficient Pruning Strategy

With the definitions of global order and join operation, the search space of the *RFM-pattern* mining process can be regarded as traversing a set-enumeration trees [34]. We assume the root node is null, and the n child nodes of the root have n 1-itemsets, respectively. Then, the remaining grandchild nodes represent other high-level itemsets (i.e., l-itemset where $l \geq 2$). Obviously, if a naive exhaustive search method is utilized, it has to check 2^n nodes, which is excessively time-consuming. Thus, several efficient pruning strategies are adopted to effectively reduce the search space.

In the high-utility itemset mining task, a general pruning method is based on the TWU concept. Because of the downward-closure property, TWU is a suitable natural upper bound eliminating the unpromising items in advance, which helps reduce the number of subsequent traversal nodes.

Property 1. If $TWU(X)$ is less than β, all supersets of X are low-utility itemsets. The mathematical inequality is $M(X') \leq TWU(X') \leq TWU(X) < \beta$, where $X \subseteq X'$. The details of the proof are provided in Ref. [26].

Strategy 1. Based on the Property 1, if there exist TWU of 1-itemsets less than the minimal threshold, there is no need to explore their supersets.

Because the RFM-List stores all the necessary information on the itemset in $\mathcal{D}$, the sum of the remaining utility indicates the future increases in utility value of the itemset. It also serves as a useful upper-bound in pruning unpromising itemsets in advance. This means that if the summation of all *iutils* and *rutils* of X is no less than the β, X and its supersets are *pHUIs*.

Property 2. Given an RFM-List X_{List}, if the sum of all *iutils* and *rutils* in X_{List} is less than β, its supersets are low-utility itemsets. The mathematical inequality is $M(X') \leq \sum_{T_j \in X_{List}} (iutil(X, T_j) + rutil(X, T_j)) < \beta$, where $X \subseteq X'$. The proof was first provided by Liu and Qu [25].

Strategy 2. Based on the Property 2, if the summation of whole *iutils* and *rutils* in some X_{List} is less than the minimal threshold, there is no need to construct RFM-Lists of supersets of X.

4.3 The Proposed Algorithm

The RFMUL algorithm adopts the depth-first search method to generate high-level itemsets by joining different RFM-Lists of low-level itemsets. In this section, the details are presented below. Algorithm 1 which takes six parameters as input: 1) the prefix itemset P, 2) the RFM-List of P, 3) a set of RFM-Lists of all P's with 1-extension, 4) the minimal recency threshold R_{min}, 5) the minimum frequency threshold, F_{min}, and 6) the minimal utility threshold M_{min}. The algorithm first traverses each RFM-List of itemset X in *RFM-Lists*. If X is an *R-pattern*, *F-pattern*, and *M-pattern* at the same time, X will be regarded as an *RFM-pattern* that should be output (Lines 2-4). Otherwise, it should be checked whether X is a *F-pattern* and *pHUI* (Line 5, using Strategy 2). If X is not both a *F-pattern* and *pHUI*, the procedure will select the next itemset after X and repeat the checking steps above. Otherwise, the current itemset represents the super-itemsets of X and maybe *RFM-patterns*, too. In line 6, the RFM-List of super-itemset X' is initialized as *NULL*. To explore the search space, the procedure joins X_{List} and each Y_{List} after X in *RFM-Lists*. Then, the *Construct* procedure is called, and a novel RFM-List of extension itemset XY is created (Line 8). If the real utility of XY is higher than \$0, XY becomes an element of X' (Lines 9-11), whereby, the intersection process between two distinct itemsets, X and Y, is completed. Finally, in Line 13, the procedure adds X as a new super-itemset. The *RFMUL* procedure is recursively processed until no new itemset is generated (Line 14).

Algorithm 2 shows how to construct the super-itemset according to two distinct RFM-Lists. It takes three parameters as input: 1) the RFM-List of the prefix itemset P, 2) the RFM-List of itemset Px, and 3) another distinct RFM-List of itemset Py. The *construct* procedure first initializes the RFM-List of the super-itemset as *NULL* (Line 1). The itemsets Px_e and Py_e are two different elements in Px_{List} and Py_{List}, respectively (Px precedes Py). If Px_e has the same *tid* term as Py_e, then *Construct* procedure starts to build the new RFM-List of Pxy (Line 3). Note that if the itemset P is *NULL*, then it directly constructs a new tuple (Line 8). Otherwise, it finds the element P_e in P_{List} that has the same *tid*, and then constructs a new tuple (Lines 5 and 6). In addition, the $iutil(P_e)$ is calculated twice because Px_e and Py_e have a common prefix itemset P. Thereafter, the obtained tuple Pxy is inserted into RFM-List Pxy_{List} (Line 10). Thus far, constructing and updating a new RFM-List of a super-itemset has been illustrated. The foregoing steps are repeated until the last element of X_{List} is processed. Finally, the *Construct* procedure outputs a complete RFM-List of the super-itemset Pxy (Line 13).

The RFMP-Growth algorithm utilizes tree-based technology, but a common trick used by the tree structure is to scan the original

Algorithm 1: The RFMUL algorithm

Input: P: the prefix itemset; P_{List}: the RFM-List of P;
RFM-Lists: the set of RFM-Lists of all P's 1-extensions;
R_{min}: the minimal recency threshold;
F_{min}: the minimal frequency threshold;
M_{min}: the minimal utility threshold.
Output: all the *RFM-patterns* with P as prefix.

for *each* X_{List} *of* X *in RFM-Lists* **do**
  if $R(X) \geq R_{min}$ *and* $F(X) \geq F_{min}$ *and* $M(X) \geq M_{min}$ **then**
    RFM-patterns $\leftarrow X$;
  end
  if $F(X) \geq F_{min}$ *and* $\sum(iutil(X) + rutil(X)) \geq M_{min}$ **then**
    initialize X'_{List} as *NULL*;
    for *each* Y_{List} *of* Y *after* X *in RFM-Lists* **do**
      XY_{List} = **Construct**(P_{List},X_{List},Y_{List});
      if $M(XY) > 0$ **then**
        $X'_{List} = X'_{List} + XY_{List}$;
      end
    end
    $P = P \cup X$;
    call **RFMUL**(P, X_{List},X'_{List},R_{min},F_{min},M_{min});
  end
end

Algorithm 2: The Construct procedure

Input: P_{List}: the RFM-List of prefix itemset P;
Px_{List}: the RFM-List of itemset Px;
Py_{List}: the RFM-List of itemset Py.
Output: Pxy_{List}: a RFM-List of super-itemset Pxy.

initialize Pxy_{List} as *NULL*;
for *each element* $Px_e \in Px_{List}$ **do**
  if $\exists Py_e \in Py_{List}$ *and* $Px_e.tid == Py_e.tid$ **then**
    if $P_{List} \neq NULL$ **then**
      find element $P_e \in P_{List}$ where $P_e.tid == Px_e.tid$;
      $Pxy_e = (Px_e.tid, iutil(pxy_e)\text{-}iutil(P_e), rutil(Py_e))$;
    else
      $Pxy_e = (Px_e.tid, iutil(pxy_e), rutil(Py_e))$;
    end
    insert Pxy_e into Pxy_{List};
  end
end
return Pxy_{List}

database multiple times. This is unacceptable and is a waste in terms of runtime and memory consumption. However, RFMUL compresses all key information on itemsets into lists by scanning the database once. On the other hand, RFMUL adopts a binary search method to lower the time complexity, while RFMP-Growth has to check the nodes from up to down. In the next section, in extensive experiments, it is demonstrated that the novel algorithm outperforms the RFMP-Growth algorithm.

5 THE EXPERIMENTAL ANALYSIS

In this section, the performance of the novel *RFM-pattern* mining algorithm is discussed on several famous benchmark datasets, in comparison with the state-of-the-art RFMP-Growth algorithm [16]. Because the precision and recall rate of mining results of RFMP-Growth has already been evaluated, we should make sure that the mining results of proposed algorithm are the same with that of RFMP-Growth, rather than repeating the same experiments. All experiments were carried out on a computer with a 64-bit Intel Core 3.0 GHz processor running the Windows 10 operating system with 16 GB of RAM. The experimental algorithms were implemented in the Java language.

5.1 Dataset Description and Parameter Settings

Dataset description. The three experimental datasets (one synthetic and two real-world datasets) were downloaded from the SPMF website[4]. All of these have been published and are available to researchers. Table 4 summarizes the characteristics of the datasets. First, some feature labels of the selected datasets are introduced: #Trans is the number of transaction datasets contained; #Items represent the number of distinct items in the dataset; #AvgLen is the average length of a transaction in the dataset; and #Type indicates that the dataset is sparse or dense. BMSPOS is a sparse dataset with short transactions. Foodmart has the same features as BMSPOS, but it is the smallest dataset in the experiments. T40I10D100K is a sparse and dense synthetic dataset, respectively. These datasets all do not contain privacy information, and the SPMF website provides additional details on these datasets.

Table 4: Dataset characteristics

Dataset	#Trans	#Items	#AvgLen	#Type
BMSPOS	515,366	1,656	6.51	Sparse
Foodmart	4,141	1,559	4.4	Sparse
T40I10D100K	100,000	942	39.6	Dense

Parameter settings. Unless the goal is to stress the differences, the decay speed δ of all datasets is always set to 0.001. Because the *RFM-pattern* mining task has three different constraints (i.e., recency, frequency, and monetary/ utility), the same parameter settings were kept when testing different datasets as much as possible. The details of the parameters of each experimental dataset are introduced in Table 5, and the "variable" represents the threshold used for testing. Moreover, as described in the formulation of the previous problem (Subsection 3.2), it is difficult to consider three constraints simultaneously. Thus, primarily, the excellent performance of the utility part of our novel algorithm is presented. Most experiments have the same minimal recency and minimum frequency thresholds of γ and α, respectively, but a distinct minimum utility threshold β. Because β was set to a too large value (most of them are more than \$100,000) "K" and "M" are used to represent a thousand and a million, respectively. The results of each method are displayed in the subsequent figures and tables. There are no specific results on runtime and memory consumption. If the runtime exceeds 100,000 s or if the algorithm is out of memory, then the result of related patterns are marked as "-".

[4]http://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php

Table 5: Thresholds setting of experimental datasets

Dataset	utility (β)	frequency (α)	recency (γ)
BMSPOS	variable	0.005	1
Foodmart	10,000	variable	0.005
T40I10D100K	variable	0.005	2

5.2 Runtime Consumption Analysis

First, in this subsection, the runtime analysis content is discussed. In all sub-figures of Figure 2, the runtime cost of the three algorithms keeps dropping as β increases, and the computation time of RFMUL (the red line) is the least. Especially in the T40I10D100K dataset, RFMUL has an obvious gap compared to RFMP-Growth. In BMSPOS dataset, RFMUL also performance better than RFMP-Growth in terms of runtime. For example, when β is \$3.9M, RFMUL and RFMP-Growth spend 20.9 s and 48.9 s on computations, respectively. On the other hand, following the β raises, the gap in the execution time usage between two algorithms is closing. It is easy to understand that the higher the β, the less satisfying the patterns have gotten. In conclusion, RFMUL performs better than RFMP-Growth in terms of runtime.

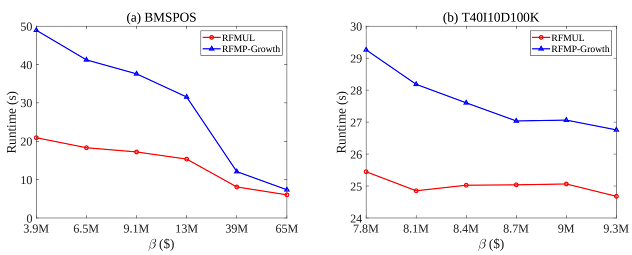

Figure 2: Runtime consumption under various β.

5.3 Memory Cost Analysis

As observed in Figure 3, the performance of the RFMP-Growth algorithm is not very well in dense dataset. For instance, in T40I10D100K, RFMP-Growth utilizes approximately 700 MB when β = \$9.3M, while RFMUL only requires 300 MB with the same β. In fact, when a database is sparse and large, the size of the corresponding prefix tree is relatively large, whereas the number of candidate itemsets is relatively small. However, RFMUL performs nor very well in BMSPOS. We suppose the reason is that BMSPOS is a sparse dataset and has a large number of distinct items, which makes RFMUL has to keep many useless RFM-Lists in memory. And we also can learn that there are fewer differences in memory cost between RFMUL and RFMP-Growth as β increases. The reason is the same as that described in the runtime analysis subsection. That is, β is too large to discover additional *RFM-patterns*.

5.4 Performance under Frequency Metric

In this subsection, the experiment is also performed to compare the performances of different frequency values between two algorithms. First, it should be noted that the number of transactions in Foodmart is small (= 4,141) and the number of distinct items is very large

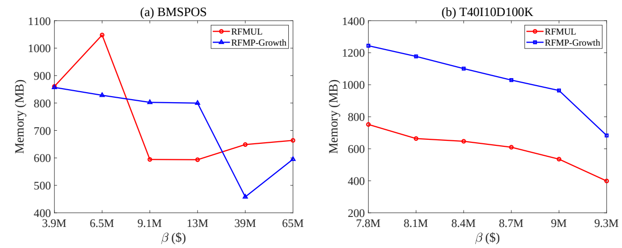

Figure 3: Memory cost under various β.

(= 1,559). This situation causes the minimal frequency to be very small, thereby prolonging the execution time. Hence, to ensure that each experimental algorithm obtains the correct results in time, γ was set to 0.005 and β to \$10,000, and α to a value in the range of 0.2% to 0.3%, as shown in Table 5. In Figure 4, RFMUL outperforms RFMP-Growth in terms of runtime and memory usage. For example, when α is 0.2%, RFMP-Growth consumes nearly 150 MB, which is approximately three times that of RFMUL. Although RFMUL generates more candidates than RFMP-Growth, it is still a viable approach. Because of considering runtime usage, the binary search method helps reduce the search time, while RFMP-Growth has to perform a downwards traversal of every branch from the root. In conclusion, RFMUL performs very well in terms of frequency without considering the utility metric.

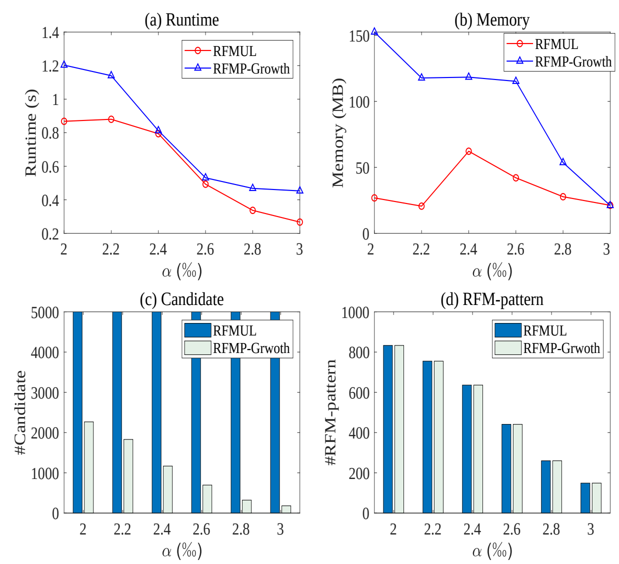

Figure 4: Influence of frequency metric on Foodmart.

5.5 Scalability Analysis

Finally, the scalability of the proposed algorithm is discussed. As shown in Figs. 2, 3 and 4, the variable-controlling approach is used to respectively evaluate the scalability performance of RFMUL under the frequency and utility metrics. For example, in T40I10D100K, the runtime and memory consumption have been decreasing steadily as β increases from \$7.8M to \$9.3M. This trend can also be seen in the Foodmart dataset. Hence, it is concluded that RMFUL performs well under both frequency and utility metrics.

6 CONCLUSIONS AND FUTURE WORK

Web data mining, such as RFM analysis, is powerful for direct marketing. In this paper, a "one phase" *RFM-pattern* mining framework called RFMUL is proposed to find interesting *RFM-patterns* in a transaction dataset without private information. Based on the remaining utility, the efficient pruning strategy is introduced to effectively reduce the search space. To evaluate the performance of RFMUL, extensive experiments are performed to demonstrate the effectiveness of the novel algorithm by comparing with the state-of-the-art algorithm on various datasets. The experimental results reveal that our new approach performs better than the other approach. In the future, several more efficient pruning strategies will be studied to improve the performance of different variants of RFMUL. The RFM-pattern mining task, without private information, can also be adopted in other subject domains, such as sequential pattern mining, episode mining, and association analysis.

ACKNOWLEDGMENTS

This research was partially supported by the National Key-Research and Development Program of China (Grant No. 2020YFB2104003), National Natural Science Foundation of China (Grant Nos. 61902079 and 62002136), Guangzhou Basic and Applied Basic Research Foundation (Grant Nos. 202102020928 and 202102020277).

REFERENCES

[1] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast algorithms for mining association rules. In *Proceedings of the 20th ACM International Conference on Very Large Data Bases*. Citeseer, 487–499.

[2] Chu-Chai Henry Chan. 2008. Intelligent value-based customer segmentation method for campaign management: A case study of automobile retailer. *Expert Systems with Applications* 34, 4 (2008), 2754–2762.

[3] Chien-Ming Chen, Lili Chen, Wensheng Gan, Lina Qiu, and Weiping Ding. 2021. Discovering high utility-occupancy patterns from uncertain data. *Information Sciences* 546 (2021), 1208–1229.

[4] Jiahui Chen, Shicheng Wan, Wensheng Gan, Guoting Chen, and Hamido Fujita. 2021. TOPIC: Top-k high-utility itemset discovering. *arXiv preprint, arXiv:2106.14811* (2021).

[5] Ching-Hsue Cheng and You-Shyang Chen. 2009. Classifying the segmentation of customer value via RFM model and RS theory. *Expert Systems with Applications* 36, 3 (2009), 4176–4184.

[6] Chui-Yu Chiu, I-Ting Kuo, and Po-Chia Chen. 2009. A market segmentation system for consumer electronics industry using particle swarm optimization and honey bee mating optimization. In *Global Perspective for Competitive Enterprise, Economy and Ecology*. Springer, 681–689.

[7] Philippe Fournier-Viger, Cheng-Wei Wu, Souleymane Zida, and Vincent S Tseng. 2014. FHM: Faster high-utility itemset mining using estimated utility co-occurrence pruning. In *Proceedings of the International Symposium on Methodologies for Intelligent Systems*. Springer, 83–92.

[8] Wensheng Gan, Chun-Wei Lin, Philippe Fournier-Viger, Han-Chieh Chao, Vincent Tseng, and Philip S Yu. 2021. A survey of utility-oriented pattern mining. *IEEE Transactions on Knowledge and Data Engineering* 33, 4 (2021), 1306–1327.

[9] Wensheng Gan, Jerry Chun-Wei Lin, Han-Chieh Chao, and Justin Zhan. 2017. Data mining in distributed environment: a survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7, 6 (2017), e1216.

[10] Wensheng Gan, Jerry Chun-Wei Lin, Philippe Fournier-Viger, Han-Chieh Chao, Tzung-Pei Hong, and Hamido Fujita. 2018. A survey of incremental high-utility itemset mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 2 (2018), e1242.

[11] Wensheng Gan, Jerry Chun-Wei Lin, Philippe Fournier-Viger, Han-Chieh Chao, and Justin Zhan. 2017. Mining of frequent patterns with multiple minimum supports. *Engineering Applications of Artificial Intelligence* 60 (2017), 83–96.

[12] Sung Ho Ha. 2007. Applying knowledge engineering techniques to customer analysis in the service industry. *Advanced Engineering Informatics* 21, 3 (2007), 293–301.

[13] Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining frequent patterns without candidate generation. *ACM SIGMOD Record* 29, 2 (2000), 1–12.

[14] Seyed Mohammad Seyed Hosseini, Anahita Maleki, and Mohammad Reza Gholamian. 2010. Cluster analysis using data mining approach to develop CRM methodology to assess the customer loyalty. *Expert Systems with Applications* 37, 7 (2010), 5259–5264.

[15] Nan-Chen Hsieh. 2004. An integrated data mining and behavioral scoring model for analyzing bank customers. *Expert Systems with Applications* 27, 4 (2004), 623–633.

[16] Ya-Han Hu and Tzu-Wei Yeh. 2014. Discovering valuable frequent patterns based on RFM analysis without customer identification information. *Knowledge-based Systems* 61 (2014), 76–88.

[17] Hakkyu Kim and DongWan Choi. 2020. Recency-based sequential pattern mining in multiple event sequences. *Data Mining and Knowledge Discovery* (2020), 1–31.

[18] Huy-Kang Kim, Kwang-Hyuk Im, and Sang-Chan Park. 2010. DSS for computer security incident response applying CBR and collaborative response. *Expert Systems with Applications* 37, 1 (2010), 852–870.

[19] Philip Kotler. 1974. Marketing during periods of shortage. *Journal of Marketing* 38, 3 (1974), 20–29.

[20] Jerry Chun-Wei Lin, Wensheng Gan, Philippe Fournier-Viger, Tzung-Pei Hong, and Vincent S Tseng. 2016. Efficient algorithms for mining high-utility itemsets in uncertain databases. *Knowledge-Based Systems* 96 (2016), 171–187.

[21] Ming-Yen Lin, Tzer-Fu Tu, and Sue-Chen Hsueh. 2012. High utility pattern mining using the maximal itemset property and lexicographic tree structures. *Information Sciences* 215 (2012), 1–14.

[22] Charles X Ling and Chenghui Li. 1998. Data mining for direct marketing: Problems and solutions.. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Vol. 98. 73–79.

[23] Duen-Ren Liu, Chin-Hui Lai, and Wang-Jung Lee. 2009. A hybrid of sequential rules and collaborative filtering for product recommendation. *Information Sciences* 179, 20 (2009), 3505–3519.

[24] Duen-Ren Liu and Ya-Yueh Shih. 2005. Integrating AHP and data mining for product recommendation based on customer lifetime value. *Information & Management* 42, 3 (2005), 387–400.

[25] Mengchi Liu and Junfeng Qu. 2012. Mining high utility itemsets without candidate generation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. 55–64.

[26] Ying Liu, Wei-Keng Liao, and Alok Choudhary. 2005. A fast high utility itemsets mining algorithm. In *Proceedings of the 1st International Workshop on Utility-based Data Mining*. 90–99.

[27] John R Miglautsch. 2000. Thoughts on RFM scoring. *Database Marketing & Customer Strategy Management* 8, 1 (2000), 67–72.

[28] Loan TT Nguyen, Phuc Nguyen, Trinh DD Nguyen, Bay Vo, Philippe Fournier-Viger, and Vincent S Tseng. 2019. Mining high-utility itemsets in dynamic profit databases. *Knowledge-Based Systems* 175 (2019), 130–144.

[29] David L Olson, Qing Cao, Ching Gu, and Donhee Lee. 2009. Comparison of customer response models. *Service Business* 3, 2 (2009), 117–130.

[30] Jian Pei, Jiawei Han, and Laks VS Lakshmanan. 2001. Mining frequent itemsets with convertible constraints. In *Proceedings of the 17th International Conference on Data Engineering*. IEEE, 433–442.

[31] Jian Pei, Jiawei Han, and Wei Wang. 2007. Constraint-based sequential pattern mining: the pattern-growth methods. *Journal of Intelligent Information Systems* 28, 2 (2007), 133–160.

[32] Joe Peppard. 2000. Customer relationship management (CRM) in financial services. *European Management Journal* 18, 3 (2000), 312–327.

[33] Georg Russ, Mirko Bottcher, and Rudolf Kruse. 2007. Relevance feedback for association rules using fuzzy score aggregation. In *Annual Meeting of the North American Fuzzy Information Processing Society*. IEEE, 54–59.

[34] Ron Rymon. 1992. Search through systematic set enumeration. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*. 539–550.

[35] Fariba Safari, Narges Safari, and Gholam Ali Montazer. 2016. Customer lifetime value determination based on RFM model. *Marketing Intelligence & Planning* (2016).

[36] Mohammadreza Tavakoli, Mohammadreza Molavi, Vahid Masoumi, Majid Mobini, Sadegh Etemad, and Rouhollah Rahmani. 2018. Customer segmentation and strategy development based on user behavior analysis, RFM model and data mining techniques: a case study. In *Proceedings of the 15th International Conference on E-Business Engineering*. IEEE, 119–126.