

Article

A New Method Combining Pattern Prediction and Preference Prediction for Next Basket Recommendation

Guisheng Chen ^{1,2}  and Zhanshan Li ^{1,2,*} 

¹ College of Computer Science and Technology, Jilin University, Changchun 130012, China; chengs16@mail.jlu.edu.cn

² Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun 130012, China

* Correspondence: lizs@jlu.edu.cn

Abstract: Market basket prediction, which is the basis of product recommendation systems, is the concept of predicting what customers will buy in the next shopping basket based on analysis of their historical shopping records. Although product recommendation systems develop rapidly and have good performance in practice, state-of-the-art algorithms still have plenty of room for improvement. In this paper, we propose a new algorithm combining pattern prediction and preference prediction. In pattern prediction, sequential rules, periodic patterns and association rules are mined and probability models are established based on their statistical characteristics, e.g., the distribution of periods of a periodic pattern, to make a more precise prediction. Products that have a higher probability will have priority to be recommended. If the quantity of recommended products is insufficient, then we make a preference prediction to select more products. Preference prediction is based on the frequency and tendency of products that appear in customers' individual shopping records, where tendency is a new concept to reflect the evolution of customers' shopping preferences. Experiments show that our algorithm outperforms those of the baseline methods and state-of-the-art methods on three of four real-world transaction sequence datasets.

Keywords: recommendation systems; market basket recommendation; data mining; periodic pattern; sequential rule; association rule



Citation: Chen, G.; Li, Z. A New Method Combining Pattern Prediction and Preference Prediction for Next Basket Recommendation. *Entropy* **2021**, *23*, 1430. <https://doi.org/10.3390/e23111430>

Academic Editor: Friedhelm Schwenker

Received: 16 September 2021

Accepted: 25 October 2021

Published: 29 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Data mining technology is an efficient tool for business. Early in 1993, Agrawal et al. [1] proposed association rule mining for transaction databases to discover the intrinsic connection between different products and the shopping habits of customers. This technology makes sales prediction easier. By predicting what customers will buy in the next shopping basket and then recommending the products to them, retailers can improve their services and promote sales. We call such a technique market basket prediction, which is the basis of product recommendation systems.

Since Agrawal et al. proposed association rule mining, both data mining and recommendation systems have been developing rapidly. On the one hand, sequential patterns [2], sequential rules [3], coverage patterns [4], temporal patterns [5], subgraph patterns [6] and periodic patterns [7] have been proposed. Data mining, as an increasingly sophisticated technology, has been used for many domains, such as time series analysis [8], medicine [9] and image processing [10]. On the other hand, recommendation systems include other kinds of implementation methods including pattern-based models, collaborative filtering [11] and Markov chains [12]. Advances in data mining technology make pattern-based models promising. There are many efficient and ready-made algorithms for pattern mining [13], and they can be easily used to implement pattern-based recommendation systems.

A pattern reveals the relation between different products, which makes pattern-based models comprehensive. Among them, a sequential rule reveals the relation between the products in two consecutive transactions. This means that a customer bought a product at some time and will buy another product at a future time. For example, if a customer buys a computer, he or she will need a U-Disk or printer possibly when working on the computer. The higher the confidence of a sequential rule, the higher the possibility. A number of sophisticated and efficient sequential rule mining algorithms have been proposed, including RuleGen [13], ERMiner [14] and RuleGrowth [3,15]. Because all products have a limited service life, when a product is used out, we will buy another product again. Therefore, some products periodically appear in our market baskets. If we know the period of a periodic pattern, then we can predict when it will appear again. Some periodic pattern mining algorithms have been proposed, including SPP [16], MPFPS [17] and LPPM [18]. The association rule reveals the relation between the products in a basket. This means that if a customer buys a product, he or she will buy another product at the same time. Some efficient algorithms such as TopKRules [19] and TNR [20] focus on association rule mining.

As described above, pattern-based models have the advantages of popularity and comprehensibility. However, existing pattern-based recommendation algorithms are insufficient to capture customers' shopping habits. For example, Ref. [21] focused on the association rule only, and the periodicity was neglected. Furthermore, the statistical characteristics inside a pattern, e.g., the distribution of periods of a periodic pattern, are also neglected. For example, by the current definition of periodic patterns [22], when a periodic pattern is used to make a prediction, we can only predict that the pattern will reoccur between a time interval but the probability at an exact time.

In terms of the disadvantages of existing methods, our method leverages not only the association rule but also the sequential rule and periodic pattern for prediction at the same time. We call such a strategy pattern prediction. Furthermore, the frequency and tendency of a product will be considered the preference that customers have for this product and the evolution of preference, respectively, to make predictions, which we call preference prediction. Combining pattern prediction and preference prediction, we propose a new algorithm for market basket prediction, which we call *SPAP* (Sequential rule, Periodic pattern, Association rule, and Preference).

In this paper, first, we present a new definition of periodic patterns and tendencies. Generally, if a product is bought periodically by a customer, then the period will be nearly equal to the service life of the product. However, service lives of a kind of product may differ from one another, leading to a fluctuation in the period. What type of pattern is a periodic pattern, which has a virtue to reveal the periodicity of product purchases that have not a fixed period? Obviously, if a pattern has periods that the fluctuation is too large compared to the average period, it will not be periodic. Taking the average period and standard deviation into account, the coefficient of variation, which is the specific value of the standard deviation and mean value, is used to measure the periodicity of patterns in our definition. The concept of tendency is based on the following considerations: if a product is more frequently bought in recent baskets than in early baskets of a customer, then the customer tends to be increasingly inclined to the product. Otherwise, if a product is more frequently bought in early baskets than recent ones, the customer tends to be increasingly estranged from the product. We use a new concept of tendency to reflect this fact.

Second, we propose probability models for pattern prediction. The sequential rule reveals the relation of two patterns belonging to two consecutive transactions. The former pattern is called the antecedent, and the latter the consequent. When a sequential rule is used for predicting the next basket, the time interval between antecedent and consequent is usually neglected. That is, the consequent will follow the antecedent with a given confidence, however, we do not know for sure at what time it occurs. In this paper, we learn the statistical model for the time interval of all sequential rules in the training data. In prediction, we use the statistical model to compute the probability of the occurrence of con-

sequents at an exact time stamp. For periodic patterns, the statistical model is determined by the average period and standard deviation. After training, we obtain all sequential rules, periodic patterns and association rules, along with their statistical characteristics. Consequently, we can calculate the probability of all products in a customer's next basket. Products that have a higher probability will have priority to be recommended. If the quantity of recommended products is insufficient, then we will make a preference prediction to select more products.

Preference prediction is based on this observation: if a product is more frequently bought by a customer, then we draw the conclusion that the customer has a preference for this product, and this product will have priority to be recommended. If some products have the same frequency, then the product with a higher tendency will be selected first in such a case.

Our contributions in this paper are summarized as follows:

- We present a new definition for periodic patterns and the tendency of patterns.
- We propose probability models for pattern prediction to predict the next basket.
- We design a new algorithm combining pattern prediction and preference prediction for next basket recommendation.
- Empirically, we show that our algorithm outperforms the baseline methods and state-of-the-art methods on three of four real-world transaction sequence datasets under the evaluation metrics of *F1-Score* and *Hit-Ratio*.

The remainder of this paper is organized as follows: Section 2 reviews the existing approaches. Section 3 includes the preliminary. We introduce our prediction method in Section 4. The implementation of our algorithm is described in Section 5, and the experimental analysis is reported in Section 6. Finally, we draw a conclusion in Section 7.

2. Related Work

Implementation methods of recommendation systems can be categorized into sequential, general, pattern-based, and hybrid models. Sequential models [23,24], mostly relying on Markov chains, explore sequential transaction data by predicting the next purchase based on the last actions to capture sequential behavior. A major advantage of this model is its ability to capture sequential behavior to provide good recommendations. The general model [25], in contrast, does not consider sequential behavior but makes recommendations based on customers' whole purchase history. The key idea is collaborative filtering. The pattern-based model bases predictions on the frequent patterns that are extracted from the shopping records of all customers [26]. Among them, the hybrid model combines the models mentioned above or other ideas, such as graph-based models [27] and recurrent neural network models [28,29]. Since there are so many works devoted to recommendation systems, it is impossible to list all here. So, we only briefly review pattern-based approaches in the next paragraph.

Fu et al. [30] first used an association rule for recommendation systems. Candidate items are listed for her in their order of support. Wang et al. [31] proposed an association rule mining algorithm with maximal nonblank for recommendation. The weighted association rule mining algorithm based on *FP-tree* and its application procedure in personalization recommendation was given by Wang et al. [32]. Ding et al. [33,34] proposed a method for personalized recommendation, which could decrease the number of association rules by merging different rules. Li et al. [21,35] proposed the notion of strongest association rules (*SARs*), and developed a matrix-based algorithm for mining *SAR* sets. As the subset of the entire association rule set, the *SAR* set includes many fewer rules with the special suitable form for personalized recommendation without information loss. Lazcorreta et al. [26] applied a modified version of the well-known *apriori* data mining algorithm towards personalized recommendation. Najafabadi et al. [36] applied the users' implicit interaction records with items to efficiently process massive data by employing association rules mining. It captures the multiple purchases per transaction in association rules, rather than just counting total purchases made. Chen et al. [37] mined

simple association rules with a single item in consequent to avoid exponential pattern growth. The method proposed by Zhou et al. [38] involved implementation of genetic network programming and ant colony optimization to solve the sequential rule mining problem for commercial recommendations in time-related transaction sequence databases. Maske et al. [39] proposed a method describing how customer behavior predicted based on the customer purchase items by association rule mining algorithm Apriori.

In the other methods, Cumby et al. [40] proposed a predictor that embraces a user-centric vision by reformulating basket prediction as a classification problem. They build a distinct classifier for every customer and perform predictions by relying just on their personal data. Unfortunately, this approach assumes the independence of items purchased together. Wang et al. [41] employed a two-layer structure to construct a hybrid representation over customers and items purchase history from last transactions: the first layer represents the transactions by aggregating item vectors from the last transactions, while the second layer realizes the hybrid representation by aggregating the customer's vectors and the transactions representations. Guidotti et al. [42–44] defined a new pattern named the Temporal Annotated Recurring Sequence (TARS), which seeks to simultaneously and adaptively capture the co-occurrence, sequentiality, periodicity and recurrence of the items in the transaction sequence. Jain et al. [45] designed a business strategy prediction system for market basket analysis. Kraus et al. [46] proposed similarity matching based on sequential dynamic time warping as a novel predictor of market baskets, and leverage the Wasserstein distance for measuring the similarity among embedded purchase histories. Hu et al. [47] presented a k-nearest neighbors (kNN) based method to directly capture two useful patterns: repeated purchase pattern and collaborative purchase pattern that associate with personalized item frequency. Faggioli et al. [48] proposed an efficient solution to achieve the next basket recommendation, under a more general top-n recommendation framework by exploiting a set of collaborative filtering based techniques to capture customers' shopping patterns and intentions.

3. Preliminary

Retailers usually preserve their customers' shopping histories in a database that we call the transaction sequence database. A customer's shopping history contains many transactions. Transaction, also called basket, usually contains ID, date, product list and quantity. All transactions of a customer are sorted according to date. This is called the transaction sequence, as Table 1 shows. A transaction sequence database contains all customers' transaction sequence, as Table 2 shows.

Let $C = \{c_1, \dots, c_n\}$ be a set of n customers and $\mathcal{I} = \{i_1, \dots, i_m\}$ be a set of m items or products in the market. The transaction sequence of customer c is denoted as B_c , and $B_c = \langle b_1, \dots, b_{r_c} \rangle$, where $b_i \subseteq \mathcal{I}$, $i \in \{1, \dots, r_c\}$, denotes a transaction or basket. The terms transaction, basket and itemset will be used interchangeably, due to the fact that we are referring to an unordered set of items (or products). The size of sequence B_c is denoted as $|B_c|$, and $|B_c| = r_c$. b_{r_c+1} denotes the next basket that will be purchased by customer c at the next time. We use indexes set $\{1, \dots, r_c\}$ of baskets in transaction sequence rather than formal dates as timestamps to simplify the problem. The interval of two transactions is denoted as $Gap(b_i, b_j)$ and defined as $Gap(b_i, b_j) = j - i$, where $1 \leq i < j \leq r_c$. The transaction sequence dataset $D = \{B_{c_1}, \dots, B_{c_n}\}$ consists of transaction sequences of n customers.

Problem 1. Assume a transaction sequence dataset, our aim is to predict the next basket for each customer according to their transaction sequence. Then, we will select k products to recommend to him or her. Formally, given dataset D , for all transaction sequence $B_c \in D$ and $B_c = \langle b_1, \dots, b_{r_c} \rangle$ to predict b_{r_c+1} , which contains a set of candidate items for recommendation. Let b_c^* denote the selected item set; then, b_c^* contains k items selected from b_{r_c+1} to recommend to customer c .

Table 1. Transaction sequence of a customer.

Transa. ID	Timestamp	Basket	Transa. ID	Timestamp	Basket
b_1	2010-01-01	$\{a, b, c, f\}$	b_9	2010-02-06	$\{a, b, g, h\}$
b_2	2010-01-05	$\{b, c, d\}$	b_{10}	2010-02-14	$\{b, c, d\}$
b_3	2010-01-07	$\{a, b, e, f, h\}$	b_{11}	2010-02-19	$\{a, b, f, h\}$
b_4	2010-01-11	$\{c, d, f, h\}$	b_{12}	2010-02-25	$\{c, d, f, g, h\}$
b_5	2010-01-16	$\{a, d, e, f, g\}$	b_{13}	2010-03-02	$\{a, b, c\}$
b_6	2010-01-19	$\{c, e, g, h\}$	b_{14}	2010-03-09	$\{b, e, g\}$
b_7	2010-01-28	$\{a, b, c, g\}$	b_{15}	2010-03-17	$\{a, c, d, e\}$
b_8	2010-01-31	$\{a, c, f, g\}$	b_{16}	2010-03-28	$\{c, f\}$

Table 2. Illustrative transaction sequence database containing seven customers.

Customer ID	Transaction Sequences
c_1	$\langle \{a, c\}, \{b, e\}, \{f, g\} \rangle$
c_2	$\langle \{b, c, d\}, \{a, c\}, \{f, g\}, \{d, e\} \rangle$
c_3	$\langle \{d, e, g\}, \{a, c\}, \{f, g\} \rangle$
c_4	$\langle \{a, c\}, \{d, e\}, \{c\}, \{f, g\} \rangle$
c_5	$\langle \{a, c\}, \{e\}, \{f, g\} \rangle$
c_6	$\langle \{a, c, e\}, \{a, c\}, \{f, g\}, \{b, f, g\} \rangle$
c_7	$\langle \{d\}, \{a, b, c\} \rangle$

Definition 1. (Frequent Itemset Pattern) Given an itemset p , $p \subseteq \mathcal{I}$, frequency threshold θ , and transaction sequence $B = \langle b_1, \dots, b_r \rangle$. If $\exists b_i \in B$, we have $p \subseteq b_i$, then we call b_i a support for p . Let $Sup(p)$ denotes all supports of p , then $Sup(p) = \{b_i | \forall b_i \in B, p \subseteq b_i\}$. The absolute frequency of p is defined as $Freq(p) = |Sup(p)|$, and the relative frequency is $Freq(p) = |Sup(p)| / |B|$. If $Freq(p) \geq \theta$, then we call p a frequent itemset pattern and itemset pattern for short. If p contains only a single item, that is, $|p| = 1$, we call it a single item pattern.

Definition 2. (Association Rule) Given two itemset patterns p_1 and p_2 of a transaction sequence, confidence threshold η , if

- (1) $p_1 \neq \emptyset, p_2 \neq \emptyset$ and $p_1 \cap p_2 = \emptyset$,
- (2) $p = p_1 \cup p_2$ is an itemset pattern, and
- (3) $Freq(p) / Freq(p_1) \geq \eta$,

then $p_1 \rightarrow p_2$ is an association rule. Its frequency is denoted as $Freq(p_1 \rightarrow p_2)$ and defined as $Freq(p_1 \rightarrow p_2) = Freq(p)$. Its confidence is denoted as $Conf(p_1 \rightarrow p_2)$ and defined as $Conf(p_1 \rightarrow p_2) = Freq(p) / Freq(p_1)$. We call p_1 the antecedent, and p_2 the consequent.

Definition 3. (Frequent Sequential Pattern) A sequence $s = \langle p_1, \dots, p_k \rangle$ is a subsequence of transaction sequence $B = \langle b_1, \dots, b_r \rangle$, denoted as $s \prec B$, if and only if there exist k integers $\{e_1, \dots, e_k\}$ such that $k \leq r, e_1 < \dots < e_k$ and $\forall i \in \{1, \dots, k\}$ it holds that $p_i \subseteq b_{e_i}$. We call this integer set an embedding of s in B , denoted as $Emb(s)$, $Emb(s) = \{e_1, \dots, e_k\}$. $Embs(s)$ denotes the set of all embeddings of s in B . In transaction sequence dataset D , if $s \prec B_c$, we call B_c a support of s . The set of all supports of s is denoted as $Sup(s)$ and $Sup(s) = \{B_c | \forall B_c \in D, s \prec B_c\}$. The absolute frequency of s is defined as $Freq(s) = |Sup(s)|$ and the relative frequency $Freq(s) = |Sup(s)| / |D|$. Given a threshold θ for frequency, if $Freq(s) \geq \theta$, then s is a frequent sequential pattern, sequential pattern for short.

In this paper, the length of s is denoted as $Len(s)$, and defined as $Len(s) = \sum_{v \in \{1, \dots, k\}} |p_v|$. If a sequence s contains only an itemset p , that is, $s = \langle p \rangle, |s| = 1$, then it can be mapped into itemset p , and we have $s = p$. If $Len(s) = 1$, viz. s has only a single itemset, and this itemset contains only a single item. We call it a single item pattern.

Definition 4. (Sequential Rule) Given two sequential patterns s_1 and s_2 of a transaction sequence dataset and confidence threshold η , if

- (1) $s_1 \neq \emptyset, s_2 \neq \emptyset$,
- (2) $s = \langle s_1, s_2 \rangle$ (s_1 concatenates with s_2) is a sequential pattern, and
- (3) $Freq(s)/Freq(s_1) \geq \eta$,

then we call $s_1 \rightarrow s_2$ a sequential rule. Its frequency $Freq(s_1 \rightarrow s_2) = Freq(s)$, and confidence $Conf(s_1 \rightarrow s_2) = Freq(s)/Freq(s_1)$. We call s_1 the antecedent, and s_2 the consequent.

Property 1. Assume a sequential rule $s_1 \rightarrow s_2 : Conf$, where $s_2 = \langle p_1, \dots, p_k \rangle$, then we have $s_1 \rightarrow \langle p_1 \rangle : Conf_1, \dots, s_1 \rightarrow \langle p_k \rangle : Conf_k$, and $Conf \leq Conf_1, \dots, Conf \leq Conf_k$.

Proof. The proof comes from the anti-monotonicity of sequential patterns' frequency. \square

Definition 5. (Periodic Pattern) Let p be an itemset pattern of transaction sequence $B = \langle b_1, \dots, b_r \rangle$, $\lambda = Freq(p)$. The occurrence list of p is denoted as $Occur(p)$, and defined as $Occur(p) = \{i | \forall i \in \{1, \dots, r\} \text{ s.t. } b_i \in B \text{ and } p \subseteq b_i\}$. Then, the period list of p in B is denoted as $Per(p)$, and defined as $Per(p) = \{per_a = w_{a+1} - w_a | \forall a \in \{1, \dots, \lambda - 1\}, w_a \in Occur(p)\}$. The coefficient of variation of p is denoted as $Coefva(p)$, and $Coefva(p) = std(Per(p))/mean(Per(p))$, where $std(*)$ and $mean(*)$ are the standard deviation and mean, respectively. Given a threshold δ for the coefficient of variation, if $Coefva(p) \leq \delta$, then p is a periodic itemset pattern or periodic pattern for short.

Definition 6. (Gap of Two Itemsets) Given two itemsets p_1 and p_2 of transaction sequence B . If $\exists o_1 \in Occur(p_1), \exists o_2 \in Occur(p_2)$ and $o_1 < o_2$, then the gap of p_1 and p_2 in B is defined as $Gap(p_1, p_2) = \min\{g | \forall o_1 \in Occur(p_1), \forall o_2 \in Occur(p_2), o_1 < o_2, g = o_2 - o_1\}$; otherwise, $Gap(p_1, p_2) = \infty$.

Definition 7. (Gap of Two Subsequences) Given two subsequences s_1 and s_2 of transaction sequence B . If $\exists E_1 \in Embs(s_1), \exists E_2 \in Embs(s_2)$ and $\max(E_1) < \min(E_2)$, then the gap of s_1 and s_2 in B is defined as $Gap(s_1, s_2) = \min\{g | \forall E_1 \in Embs(s_1), \forall E_2 \in Embs(s_2), \max(E_1) < \min(E_2), g = \min(E_2) - \max(E_1)\}$; otherwise, $Gap(p_1, p_2) = \infty$.

Definition 8. (Tendency) Given an itemset p of transaction sequence B , we call $Ten(p) = mean(Occur(p))$ the tendency of itemset p in transaction sequence B .

Example 1. For the transaction sequence given in Table 1, the support of itemset $p = \{a, h\}$ is $Sup(p) = \{b_3, b_9, b_{11}\}$, frequency $Freq(p) = 3$, $Occur(p) = \{3, 9, 11\}$, $per_1 = 9 - 3 = 6$, $per_2 = 11 - 9 = 2$, periods list $Per(p) = \{6, 2\}$, $Coefva(p) = 0.71$. In Table 2, the frequency of sequence $s_1 = \langle \{a, c\} \rangle$ is $Freq(s_1) = 7$, the frequency of sequence $s_2 = \langle \{f, g\} \rangle$ is $Freq(s_2) = 6$, and the frequency of sequence $s = \langle s_1, s_2 \rangle$ is $Freq(s) = 6$. We have $Conf(s_1 \rightarrow s_2) = 6/7$. $Embs(s)_{c_1} = \{\{1, 3\}\}$. $Embs(s)_{c_6} = \{\{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}\}$. The frequencies of itemsets $p_1 = \{a, c\}$ and $p_2 = \{f, g\}$ in transaction sequence B_{c_6} are $Freq(p_1)_{c_6} = 2$ and $Freq(p_2)_{c_6} = 2$, respectively. The occurrence list of itemset $\{d\}$ in transaction sequence B_{c_2} is $Occur(\{d\})_{c_2} = \{1, 4\}$, tendency $Ten(\{d\})_{c_2} = 2.5$.

4. Framework

Our method includes pattern prediction and preference prediction. In pattern prediction, first, all sequential rules, periodic patterns and association rules are found together with their statistical characteristics. Then, probability models are built based on their statistical characteristics. Afterward, we use the probability models to calculate the probability of all products in the next basket for a customer. The products that have a higher probability will be selected preferentially to recommend to him or her. If k products have been selected, then continue to the prediction of the next customer; otherwise, make preference predictions. In preference prediction, the product that is more frequent in the individual shopping records will be selected first. If some products have the same frequency, then the product

that has a higher tendency will be selected. Until all k products are selected, we continue to predict the next customer. See Algorithm 1. We introduce the probability list pl to preserve the probability of all items in b_{r+1} , viz. $pl := \{(item1 : value1), (item2 : value2), \dots\}$.

Algorithm 1: SPAP($D, \theta_r, \eta, \theta_p, \delta, k$).

```

1 SeqRuleSet ← SequentialRuleMining( $D, \theta_r, \eta$ );
2 for all  $B_c \in D$  do
3   PPSet ← PeriodicPatternMining( $B_c, \theta_p, \delta$ );
4   AssRuleSet ← AssociationRuleMining( $B_c, \theta_r, \eta$ );
5    $b_{r+1} = Null, pl = Null, b_c^* = Null$ ;
6    $b_{r+1}, pl =$  SequentialRulePrediction( $SeqRuleSet, B_c, b_{r+1}, pl$ );
7    $b_{r+1}, pl =$  PeriodicPatternPrediction( $PPSet, B_c, b_{r+1}, pl$ );
8    $b_{r+1}, pl =$  AssociationRulePrediction( $AssRuleSet, b_{r+1}, pl$ );
9   sort  $b_{r+1}$  by probability in descending order;
10   $k^* = \min(|b_{r+1}|, k)$ ;
11   $b_c^* = b_c^* \cup \{\text{select the first } k^* \text{ items from } b_{r+1}\}$ ;
12  if  $(k^* = k - |b_c^*|) > 0$  then
13    |  $b_c^* = b_c^* \cup \{\text{select } k^* \text{ items by preference prediction}\}$ ;
14  end
15 end
16 return  $b_c^*$  for all  $B_c \in D$ ;

```

4.1. Sequential Rule Prediction

4.1.1. Probability Model of Sequential Rule

Sequential rules reveal the relation between products in two consecutive transactions. This means that a customer bought a product at some time, and he or she will buy another product at a future time. However, the sequential rule defined by the previous section has a limitation to use for the next basket prediction. For example, given sequential rule $E_1 \rightarrow E_2 : Conf = 0.8$. If event E_1 occurs, then it will lead to event E_2 occurring at a confidence of 0.8, and the confidence is considered as the probability here, that is, $P(E_2|E_1) = 0.8$. However, we did not know for sure at what time event E_2 occurs, and did not know the probability of event E_2 occurring at an exact time after event E_1 occurred. To address this limitation, we build a probability model for time intervals of the sequential rule. The time interval of a sequential rule, i.e., the time interval between event E_1 and event E_2 , is a random variable and represented by X here, $X = Gap(E_1, E_2)$. Generally, the larger the time interval, the lower the relevance of E_1 and E_2 , and vice versa. We suppose that the probability model nearly follows an exponential distribution with a parameter of $mean(Gap(s_1, s_2))$.

Example 2. For a transaction sequence dataset showed in Table 2, let $s_1 = \langle \{a, c\} \rangle$, $s_2 = \langle \{f, g\} \rangle$ and $s = \langle s_1, s_2 \rangle = \langle \{a, c\}, \{f, g\} \rangle$, we have $Sup(s_1) = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$, $Sup(s_2) = \{c_1, c_2, c_3, c_4, c_5, c_6\}$. If we set $\theta = 5$, $\eta = 0.8$, and we have $Conf(s_1 \rightarrow s_2) = 6/7 > \eta$, then $s_1 \rightarrow s_2$ is a sequential rule. The time interval between s_1 and s_2 is $Gap(s_1, s_2)_{c_1} = 3 - 1 = 2$ in c_1 , where $Occur(s_1)_{c_1} = \{1\}$ and $Occur(s_2)_{c_1} = \{3\}$. In a similar way, $Gap(s_1, s_2)_{c_2} = 1$, $Gap(s_1, s_2)_{c_3} = 1$, $Gap(s_1, s_2)_{c_4} = 3$, $Gap(s_1, s_2)_{c_5} = 2$, $Gap(s_1, s_2)_{c_6} = 1$. Note that in sequence c_6 , we have $Occur(s_1)_{c_6} = \{1, 2\}$ and $Occur(s_2)_{c_6} = \{3, 4\}$, leading $Gap(s_1, s_2)_{c_6}$ to be multiple-valued. According to Definition 6, we have $Gap(s_1, s_2)_{c_6} = \min\{(3 - 1), (3 - 2), (4 - 1), (4 - 2)\} = 1$. Consequently, we obtain a probability model for $Gap(s_1, s_2)$ as $P\{Gap(s_1, s_2) = 1\} = 1/2$, $P\{Gap(s_1, s_2) = 2\} = 1/3$ and $P\{Gap(s_1, s_2) = 3\} = 1/6$, respectively.

4.1.2. Principle of Sequential Rule Prediction

Given transaction sequence $B = \langle b_1, \dots, b_r \rangle$, sequential rule $s_1 \rightarrow s_2 : Conf$, and its probability distribution $P\{X = Gap(s_1, s_2)\}$ of time intervals between s_1 and s_2 . Suppose the consequent contains only a single itemset, that is, $|s_2| = 1$ (Since if $|s_2| > 1$, then we

can break it down into several sequential rules, which have a consequent containing only a single itemset, according to Property 1). If $s_1 \not\prec B$, then $P(s_1 \prec B) = 0$. Otherwise, $P(s_1 \prec \langle b_1, \dots, b_\lambda \rangle) = 1$, where $\lambda = \max\{e | \forall E \in \text{Embs}(s_1), e \in E\}$. Since sequential rule $s_1 \rightarrow s_2 : \text{Conf}$ means that $P(s_2 \prec \langle b_{\lambda+1}, b_{\lambda+2}, \dots \rangle | s_1 \prec \langle b_1, \dots, b_\lambda \rangle) = \text{Conf}$, then $P(s_2 \prec \langle b_{\lambda+1}, b_{\lambda+2}, \dots \rangle) = \text{Conf} \times P(s_1 \prec \langle b_1, \dots, b_\lambda \rangle) = \text{Conf}$. Suppose A_1 denote the event $s_2 \subseteq b_{r+1}$. In a similar way, A_u denote the event $s_2 \subseteq b_{r+u}$. If event A_1 occurs, then $\text{Gap}(s_1, s_2) = r + 1 - \lambda$, $P(A_1 | s_2 \prec \langle b_{\lambda+1}, b_{\lambda+2}, \dots \rangle) = P\{X = (r + 1 - \lambda)\}$. In a similar way, if event A_u occurs, then $\text{Gap}(s_1, s_2) = r + u - \lambda$, $P(A_u | s_2 \prec \langle b_{\lambda+1}, b_{\lambda+2}, \dots \rangle) = P\{X = (r + u - \lambda)\}$. Since $P(s_2 \prec \langle b_{\lambda+1}, b_{\lambda+2}, \dots \rangle) = \text{Conf}$, we obtain the probability of $s_2 \subseteq b_{r+u}$, $P(A_u) = P\{X = (r + u - \lambda)\} \times \text{Conf}$.

First, the time variable is continuous in general. However, in our case, we use the index of baskets as the timestamp, so the time variable is discretized. The probability of an exact value of the time variable is the probability within a unit interval over this value. Second, if an item is contained in the consequents of several sequential rules, then it will be predicted several times. In such a case, we update its probability as the maximal value. See Algorithm 2.

Algorithm 2: SequentialRulePrediction(*RuleSet*, *B*, *b_{r+1}*, *pl*).

```

1 for all rule( $s_1 \rightarrow s_2 : \text{Conf}$ )  $\in$  RuleSet do
2   if  $B \in \text{Sup}(s_1)$  then
3      $\lambda = \max\{e | \forall E \in \text{Embs}(s_1), e \in E\}$ ;
4      $P(A_1 | s_2 \prec \langle b_{\lambda+1}, b_{\lambda+2}, \dots \rangle) = P\{X = (r + 1 - \lambda)\}$ ;
5      $P(A_1) = P\{X = (r + 1 - \lambda)\} \times \text{Conf}$ ;
6     for all  $i \in s_2$  do
7       if  $i \in b_{r+1}$  and  $\text{val}_i \geq P(A_1)$ , where  $(i : \text{val}_i) \in \text{pl}$  then
8         continue;
9       else if  $i \in b_{r+1}$  then
10        update  $\text{val}_i$  as  $P(A_1)$ , where  $(i : \text{val}_i) \in \text{pl}$ ;
11      else
12         $b_{r+1} = b_{r+1} \cup \{i\}$ ;
13         $\text{pl} = \text{pl} \cup \{(i : P(A_1))\}$ ;
14      end
15    end
16  end
17 end
18 return  $b_{r+1}, \text{pl}$ ;

```

Continue with the Example 2 Let $B = \langle \{a, b, c\}, \{b, e, g\}, \{a, c, d, e\}, \{c, f\} \rangle$; we predict the probability of all products in b_5 . In Example 2, we obtain a sequential rule $s_1 \rightarrow s_2 : \text{Conf}$ and its probability model. $\text{Embs}(s_1) = \{\{1\}, \{3\}\}$, $\lambda = 3$. If $s_2 \subseteq b_5$, then $\text{Gap}(s_1, s_2) = 2$. We have $P\{\text{Gap}(s_1, s_2) = 2\} = 1/3$ from the probability distribution. Finally, we have $P(\{f, g\} \subseteq b_5) = 6/7 \times 1/3 = 6/21$, as Figure 1 shows.

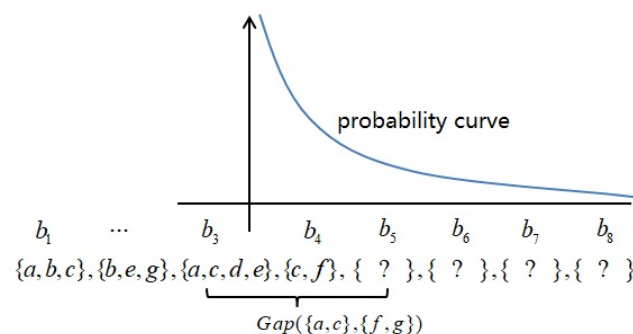


Figure 1. Probability model of sequential rule.

4.2. Periodic Pattern Prediction

4.2.1. Probability Model of Periodic Pattern

Periodic events intrinsically reoccur with a fixed period. However, in the real world, the period is influenced by different factors, leading to a fluctuation in the period. In this case, we only have an average value for periods. As we all know, the smaller the fluctuation, the better. So, we compare the fluctuation with the average period. If the fluctuation is too large compared to the average period, we do not define it as a periodic event. The coefficient of variation, which is the specific value of the standard deviation and mean value, is suitable for periodicity measure, since standard deviation is a good measure for fluctuation. The service life of a product is no exception. If a product is bought periodically by a customer, then the period will be nearly equal to the service life of the product. However, service lives of a kind of products may differ from one another, leading to a fluctuation in the period.

According to Definition 5, we define periodic patterns based on the coefficient of variation. If a pattern has a higher coefficient of variation, which means that the standard deviation is too large compared to the mean value, the pattern is not periodic. Otherwise, we classify it as a periodic pattern. Given a periodic pattern, if it occurs at some time, then it is more likely to reoccur after an average period of time. The probability at a time closer to the time after an average period later has a higher value; otherwise, it has a lower value. We suppose that the probability model of periods follows a normal distribution and has two parameters: $mean(Per(p))$ and $std(Per(p))$, as Figure 2 shows.

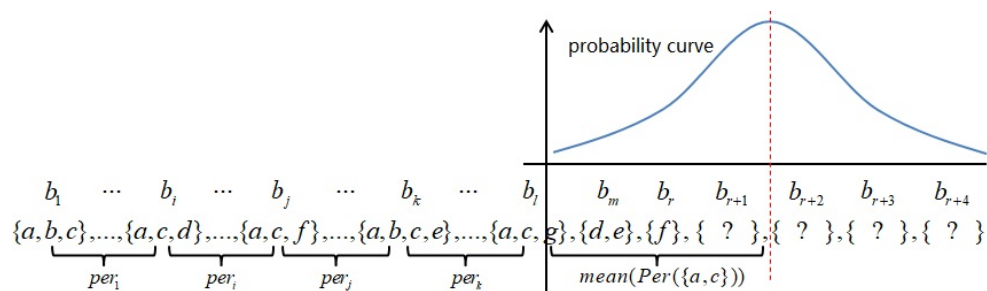


Figure 2. Probability model of periodic pattern $\{a, c\}$.

4.2.2. Principle of Periodic Pattern Prediction

Given transaction sequence $B = \langle b_1, \dots, b_r \rangle$, and a periodic pattern p of B , the prediction of the probability of $P(p \subseteq b_{r+1})$ is analogous to the sequential rule prediction.

4.3. Association Rule Prediction

Given an association rule $p_1 \rightarrow p_2 : Conf$, if itemset pattern $p_1 \subseteq b$, then itemset pattern $p_2 \subseteq b$ with a probability of $P(p_2 \subseteq b | p_1 \subseteq b) = Conf$. If $p_1 \not\subseteq b$, then $P(p_1 \subseteq b) = 0$, and $P(p_2 \subseteq b | p_1 \subseteq b) = 0$.

After sequential rule prediction and periodic pattern prediction, we obtain a set of candidate products with their probability in b_{r+1} . At the same time, we define the probability of $p_1 \subseteq b_{r+1}$ as $P(p_1 \subseteq b_{r+1}) = \min\{P(i) | \forall i \in p_1, (i : P(i)) \in pl\}$. Therefore, we get the probability of $p_2 \subseteq b_{r+1}$ as $P(p_2 \subseteq b_{r+1}) = P(p_1 \subseteq b_{r+1}) \times Conf$. See Algorithm 3.

After pattern prediction, if a product has not been predicted, then it will have a default probability value of zero and is not included in b_{r+1} to be recommended in the pattern prediction stage. Therefore, we obtain a set of candidate products along with their probabilities in the next basket b_{r+1} . In b_{r+1} , a product with a higher probability will have priority to be recommended to customers. If $|b_{r+1}| < k$, that is, the number of candidate items is less than k in b_{r+1} , then we will make a preference prediction to continue to select products to recommend until we get all k products.

Algorithm 3: AssociationRulePrediction(*RuleSet*, b_{r+1} , *pl*).

```

1 for all rule( $p_1 \rightarrow p_2 : Conf$ )  $\in$  RuleSet do
2   if  $p_1 \subseteq b_{r+1}$  then
3      $P(p_1 \subseteq b_{r+1}) = \min\{P(i) | \forall i \in p_1, (i : P(i)) \in pl\}$ ;
4      $P(p_2 \subseteq b_{r+1}) = P(p_1 \subseteq b_{r+1}) \times Conf$ ;
5     for all  $i \in p_2$  do
6       if  $i \in b_{r+1}$  and  $val_i \geq P(p_2 \subseteq b_{r+1})$ , where  $(i : val_i) \in pl$  then
7         continue;
8       else if  $i \in b_{r+1}$  then
9         update  $val_i$  as  $P(p_2 \subseteq b_{r+1})$ , where  $(i : val_i) \in pl$ ;
10      else
11         $b_{r+1} = b_{r+1} \cup \{i\}$ ;
12         $pl = pl \cup \{(i : P(p_2 \subseteq b_{r+1}))\}$ ;
13      end
14    end
15  end
16 end
17 return  $b_{r+1}, pl$ ;

```

4.4. Preference Prediction

In pattern prediction, we selected a set of products to recommend to a customer. If the number of selected products is less than k , then we continue to select based on preference prediction.

In preference prediction, if a product is more frequently bought by a customer, then we conclude that the customer has a preference for this product. However, the preference will evolve over time. The purchase distribution of a product over a shopping history will indicate a change in preference. If a product is more frequently bought in the recent baskets than the earlier ones of a customer, then the customer is more and more inclined toward the product; otherwise, if a product is more frequently bought in earlier baskets than recent ones, then the customer tends to be increasingly estranged from the product. According to Definition 8, tendency reflects such a fact. The preference prediction is based on the frequency and tendency of a product in customers' shopping histories. We first select the products that are more frequent to recommend. If some products have the same frequency, then the product that has a higher tendency is prioritized.

4.5. A Comprehensive Example

For the transaction sequence given in Table 1, itemset $p = \{c, f\}$ has an occurrence list of $Occur(p) = \{1, 4, 8, 12, 16\}$ and a period list of $Per(p) = \{3, 4, 4, 4\}$. By calculating, we get $mean(Per(p)) = 3.75$ and $std(Per(p)) = 0.866$, $Coefva(p) = std(Per(p)) / mean(Per(p)) = 0.23$. If we set $\theta = 5$ and $\delta = 0.5$, then p is a periodic pattern and its period follows $per \sim N(3.75, 0.866)$. Now let us predict b_{17} and select 5 products to recommend, viz. $k = 5$. Suppose we have a sequential rule of Example 2 and an association rule of $\{c, f, g\} \rightarrow \{d\} : 0.9$ at the same time. First, in sequential rule prediction, we have $P(\{f, g\} \subseteq b_{17}) = 6/21 = 0.2857$, $pl = \{(f : 0.2857), (g : 0.2857)\}$. Second, we use periodic pattern $p = \{c, f\}$ to predict. As $Occur(p) = \{1, 4, 8, 12, 16\}$, $\lambda = 16$. If $p \subseteq b_{17}$, then $per_5 = 17 - 16 = 1$. We have $P\{per = 1\} = 0.0046$ from the probability model. Merging $\{(c : 0.0046), (f : 0.0046)\}$ into pl , we get $pl = \{(c : 0.0046), (f : 0.2857), (g : 0.2857)\}$. Third, the association rule is used to predict. By definition, we have $P(\{c, f, g\} \subseteq b_{17}) = \min\{0.0046, 0.2857, 0.2857\} = 0.0046$, then $P(\{d\} \subseteq b_{17}) = 0.0046 \times 0.9 = 0.0041$. Finally, the products are selected in the order of $\langle f, g, c, d \rangle$. However, there are not enough products to recommend. Thus, we will have a preference prediction.

In preference prediction, all products that are in the individual shopping history, excluding selected products, are sorted by frequency. We have $Freq(\{a\}) = 9$, $Freq(\{b\}) = 9$, $Freq(\{h\}) = 6$ and $Freq(\{e\}) = 5$. Because the frequencies of items a and b are the

same, their tendencies are calculated. Item a has an occurrence list of $Occur(\{a\}) = \{1, 3, 5, 7, 8, 9, 11, 13, 15\}$, and item b has $Occur(\{b\}) = \{1, 2, 3, 7, 9, 10, 11, 13, 14\}$. We get $Ten(\{a\}) = 8$ and $Ten(\{b\}) = 7.78$, and select a . Finally, all 5 products are selected.

4.6. Relation of Two Prediction Strategies

In pattern prediction, we must set a value to the frequency threshold, confidence threshold and threshold for the coefficient of variation in pattern mining. These parameters determine the weight of pattern prediction and preference prediction.

The confidence threshold parameter is set for sequential rule mining and association rule mining. A higher value of the confidence threshold is set, and fewer sequential rules and association rules are found. The threshold parameter for the coefficient of variation is set for periodic pattern mining. If it has a lower value, then fewer periodic patterns are found. The frequency threshold parameter is set for all three types of patterns. If a higher value is set, then fewer patterns are found. There is no parameter for preference prediction.

In general, if we have a higher value on the frequency threshold and confidence threshold and a lower value on the threshold for the coefficient of variation, then we will select fewer products in pattern prediction. Preference prediction will have a higher weight on the selected result, and pattern prediction will have the opposite effect. Conversely, preference prediction will have a lower weight on the selected result.

5. Implementation

5.1. Optimizations

In pattern mining, the result grows exponentially as the pattern expands in size. It is a severe problem that traditional pattern mining approaches face to efficiently mine large patterns in dense datasets. The growth of the number of patterns is not proportional to the improvement of prediction performance in our method but leads to heavier workload. In the pattern prediction of our method, if an item is contained in several patterns, it will probably be repeatedly predicted several times, leading to redundancies in workload. The larger the number of patterns or the larger the size of the pattern, the more redundancies in the workload.

For the two reasons mentioned above, we use simple association rules, simple sequential rules and simple periodic patterns to implement pattern prediction.

Definition 9. (Simple Association Rule) Given an association rule $p_1 \rightarrow p_2 : Conf$, if both p_1 and p_2 are single item patterns, then we call it a simple association rule.

Definition 10. (Simple Sequential Rule) Given a sequential rule $s_1 \rightarrow s_2 : Conf$, if both s_1 and s_2 are single item patterns, then we call it a simple sequence rule.

Definition 11. (Simple Periodic Pattern) Given a periodic pattern p , if p is a single item pattern, then we call it a simple periodic pattern.

This strategy dramatically reduces the number of patterns and is easier to implement. As a result, our algorithm is incomplete. We use two ready-made algorithms introduced by Philippe Fournier-Viger et al. [13], namely, *FPGrowth* and *RuleGen*, to mine simple association rules and simple sequential rules, respectively. The implementation of simple periodic pattern mining will be discussed in the next subsection.

5.2. Data Structure

Inspired by Fabio Fumarola et al. [49], we propose a new data structure named *vertical bit-list* to represent the dataset. Each item $i \in \mathcal{I}$ is assigned a *vertical bit-list*. A *vertical bit-list* is made up of a bit vector and several integer arrays, as Figure 3 shows. Bit vectors have a size of the dataset size, i.e. the number of sequences in the dataset. The j th bit, $j \in \{1, \dots, |D|\}$, in each bit vector correspond to the j th sequence in the dataset D . If the

j th bit in the bit vector of a *vertical bit-list* is flipped, then means there exist an itemset in the j th sequence contains the item to which the *vertical bit-list* is assigned, and each flipped bit is assigned an integer array to preserve the occurrence list of the item in the sequence to which the flipped bit correspond. If the j th sequence contains item i , that is, there exist itemsets in the j th sequence containing item i , then the frequency and tendency of i in the j th sequence can be calculated by its corresponding integer array, and we can determine whether the item i is a simple periodic pattern in the j th sequence based on the integer array according to Definition 5, since simple periodic pattern mining is free from pattern growth.

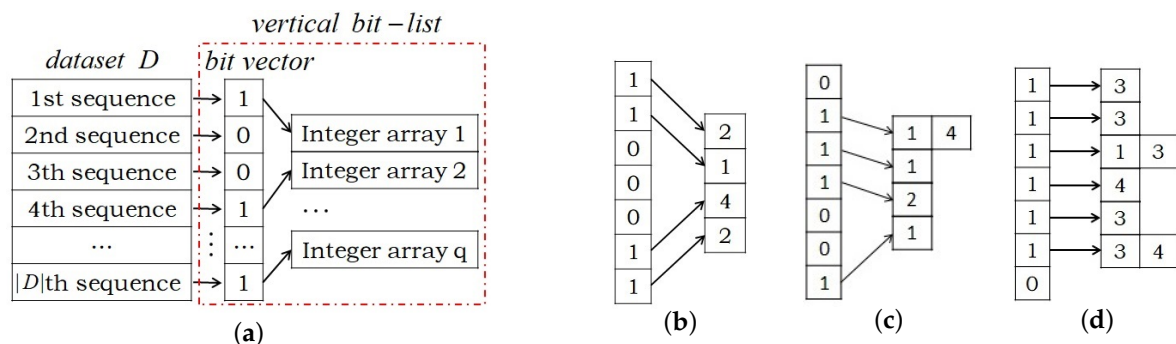


Figure 3. (a) shows the *vertical bit-list* of item $i \in \mathcal{I}$, where $q = |\{B_c | \forall B_c \in D, \exists b \in B_c, i \in b\}|$. (b–d) show *vertical bit-list* of items b, d and g , respectively, in dataset of Table 2.

5.3. Complexity

In the stage of pattern mining, we dramatically reduce the complexity of our algorithm by mining only simple sequential rules, simple association rules and simple periodic patterns. We mine simple sequential rules and simple association rules in $O(m^2)$, where m is the number of distinct items in \mathcal{I} since the rule of two types only contains two items. Simple periodic pattern mining and tendency calculation have a complexity of $O(n \times v)$, n is the number of all customers and v is the average count of products the customers have bought, that is, one scan over the dataset, and the complexity of the prediction stage is the same.

6. Experiment

To evaluate the performance of our *SPAP* algorithm, experiments were conducted on four real-world datasets. First, we assessed the influences of parameters on the weight of pattern prediction and preference prediction. Second, we compared our algorithm with those of the baseline methods and state-of-the-art methods in the evaluation metrics of *F1-Score* and *Hit-Ratio*. The experiments were conducted on a computer with an Intel Core I7-8550U 1.8 GHz processor and 8 GB of RAM, running Windows 10 (64-bit version). The *SPAP* is implemented in Java.

6.1. Datasets

We performed our experiments on four real-world transaction sequence datasets. Three of the datasets, Ta-Feng (<https://www.kaggle.com/chiranjivdas09/ta-feng-grocery-dataset> (Accessed data: 19 October 2021)), Dunnhumby (<https://www.kaggle.com/frtgnn/dunnhumby-the-complete-journey> (Accessed data: 19 October 2021)) and X5-Retail-Hero (<https://www.kaggle.com/mvyurchenko/x5-retail-hero> (Accessed data: 19 October 2021)), are based on physical markets; and one dataset, T-Mall (Provided by Guidotti et al. [50] at <https://github.com/riccotti/Customertemporalregularities/tree/master/datasets> (Accessed data: 19 October 2021)), is based on an online market.

- Ta-Feng is a dataset of a physical market, covering food, stationery and furniture, with a total of 23,812 different items in China. It contains 817,741 transactions made by 32,266 customers over 4 months.
- The Dunnhumby (Dunnh for short) dataset contains household level transactions over two years from a group of 2500 households who are frequent shoppers at a retailer.
- The X5-Retail-Hero (X5RH for short) dataset contains 183,258 transactions made by 9305 customers over 4 months and a total of 27,766 distinct items.
- The T-Mall dataset records four months of online transactions of an online e-commerce website. It contains 4298 transactions belonging to 884 users and 9531 distinct brands considered as items.

In preprocessing these datasets, we remove customers who have fewer than 10 baskets for Ta-Feng, Dunnh and T-Mall, and remove customers who have fewer than 21 baskets for X5RH. For simplicity, we adopt the index of the basket in sequence as the time unit rather than the real date. Table 3 shows the details of these datasets used in our experiment.

Table 3. Characteristics of running datasets.

Dataset	# Customers	# Items	# Baskets	Average Basket Size
Ta-Feng	2374	18,138	39,533	5.58
Dunnh	2402	91,779	273,474	9.36
X5RH	2924	22,921	99,357	5.40
T-Mall	748	9436	31,035	2.52

6.2. Evaluate Metrics F1-Score and Hit-Ratio

Following Guidotti et al. [44], first, we sort the transactions by the timestamps for each customer. Then, we split the dataset into training set and testing set. Testing set contains the latest transaction of all customers for model evaluation. Training set contains the remainder of the transactions of all customers for model training. This is known as *leave-one-out* strategy. The product set that customer c actually buys is denoted as b'_c . The product set recommended to customer c is denoted as b_c^* . The metrics we use for evaluation, *F1-Score* and *Hit-Ratio*, are defined as

$$precision(b'_c, b_c^*) = \frac{|b'_c \cap b_c^*|}{|b_c^*|} \quad (1)$$

$$recall(b'_c, b_c^*) = \frac{|b'_c \cap b_c^*|}{|b'_c|} \quad (2)$$

$$F1 - Score = \frac{2 \times precision(b'_c, b_c^*) \times recall(b'_c, b_c^*)}{precision(b'_c, b_c^*) + recall(b'_c, b_c^*)} \quad (3)$$

$$Hit - Ratio = \frac{\sum_{c \in C} \mathbb{I}(b'_c \cap b_c^* \neq \emptyset)}{|C|} \quad (4)$$

where $\mathbb{I}(\ast)$ is an indicator function. The *F1-Score* is reported by the average value of all customers.

Furthermore, to evaluate the contribution of two prediction strategies, we introduce a new measure: *Weight*. Let b_{pa} denote all the items selected in pattern prediction for all customers, and b_{pr} denote all the items selected in preference prediction for all customers. The number of all items selected to recommend to all customers is $\sum_{c \in C} |b_c^*|$, and we have $|b_{pa}| + |b_{pr}| = \sum_{c \in C} |b_c^*|$. The weights of pattern prediction and preference prediction are denoted as $Weight_{pa}$ and $Weight_{pr}$, respectively, and defined as

$$Weight_{pa} = \frac{|b_{pa}|}{\sum_{c \in C} |b_c^*|} \quad (5)$$

$$Weight_{pr} = \frac{|b_{pr}|}{\sum_{c \in C} |b_c^*|} = 1 - Weight_{pa} \quad (6)$$

due to the complementation of $Weight_{pa}$ and $Weight_{pr}$, we report $Weight_{pa}$ only. A higher value of $Weight_{pa}$ means that more items are selected in pattern prediction, and vice versa. In the remainder of this paper, we use $Weight$ to replace $Weight_{pa}$. Note that if the number of all distinct products customer c has bought is less than k , then we cannot select a set of products including more than or equal to k items to recommend to him or her, that is, $|b_c^*| \leq k$, leading to $\sum_{c \in C} |b_c^*| \leq k \times n$, where $|C| = n$.

6.3. Influence of Parameters

Our algorithm is composed of two prediction strategies. We obtain the best performance only on the right proportion of weight on pattern prediction and preference prediction. There are four parameters in pattern prediction: relative frequency threshold θ_r and confidence threshold η for sequential rule mining and association rule mining, threshold for coefficient of variation δ and absolute frequency threshold θ_p for periodic pattern mining. All of these parameters have an influence on the weight of these two prediction strategies. In this subsection, we will evaluate the influences of these parameters. k is set to the average basket size of each dataset. The values for the remainder of the parameters are preset to be $\theta_r = 0.16$, $\eta = 0.2$, $\delta = 0.2$ and $\theta_p = 5$ for Ta-Feng, $\theta_r = 0.35$, $\eta = 0.7$, $\delta = 0.4$ and $\theta_p = 12$ for Dunnh, $\theta_r = 0.15$, $\eta = 0.4$, $\delta = 0.8$ and $\theta_p = 10$ for X5RH, and $\theta_r = 0.1$, $\eta = 0.4$, $\delta = 0.8$ and $\theta_p = 5$ for T-Mall.

First, we test the parameter of relative frequency threshold θ_r , and the results are shown in Figure 4. We can see that when θ_r rises, the total number of distinct rules, including sequential rules and association rules, and $Weight$ decrease in all cases of the four datasets. $Weight$ is always lower than 0.5, which means that preference prediction plays a dominant role. For the Ta-Feng dataset, when θ_r has a value less than 0.16, both $F1-Score$ and $Hit-Ratio$ remain unchanged and achieve the optimal values, and the situation is the same on the Dunnh dataset when θ_r is greater than 0.75. Both $F1-Score$ and $Hit-Ratio$ are constant on the X5RH dataset when θ_r is greater than 0.63. For the T-Mall dataset, when θ_r is near to 0.11, both $F1-Score$ and $Hit-Ratio$ achieve the optimal values. Figure 5 shows the influences of different values for threshold η . As η rises, the number of rules and $Weight$ decrease on all four datasets. Both $F1-Score$ and $Hit-Ratio$ achieve the optimal values on the Ta-Feng dataset when η is less than 0.38, achieve the optimal values on the Dunnh when η is greater than 0.9 and achieve the optimal values on the T-Mall dataset when η is equal to 0.5, respectively. When η is greater than 0.8, $F1-Score$ or $Hit-Ratio$ achieves the optimal value on the X5RH dataset.

The threshold for the coefficient of variation δ is set for periodic pattern mining. A higher value for δ will lead to a larger number of periodic patterns, as Figure 6 shows. At the same time, a larger number of periodic patterns results in a higher $Weight$ on all four datasets. When δ is less than 0.4, $F1-Score$ or $Hit-Ratio$ achieves the optimal value on the Ta-Feng dataset. Both $F1-Score$ and $Hit-Ratio$ remain unchanged and achieve the optimal values on the Dunnh dataset when δ is greater than 2.8, and the situation is the same on the X5RH and T-Mall datasets when δ is greater than 2.0.

Finally, we test the absolute frequency threshold θ_p for periodic pattern mining. As shown in Figure 7, when θ_p rises, the number of periodic patterns and $Weight$ decrease on all four datasets. Both $F1-Score$ and $Hit-Ratio$ remain unchanged and achieve the optimal values on the Ta-Feng dataset when θ_p is greater than 11, and the situation is the same on the Dunnh dataset when θ_p is greater than 27. In the case of X5RH, when θ_p is equal to 7, both $F1-Score$ and $Hit-Ratio$ achieve the optimal values. For the T-Mall dataset, $F1-Score$ and $Hit-Ratio$ achieve their optimal values when θ_p is equal to 10 and 5, respectively.

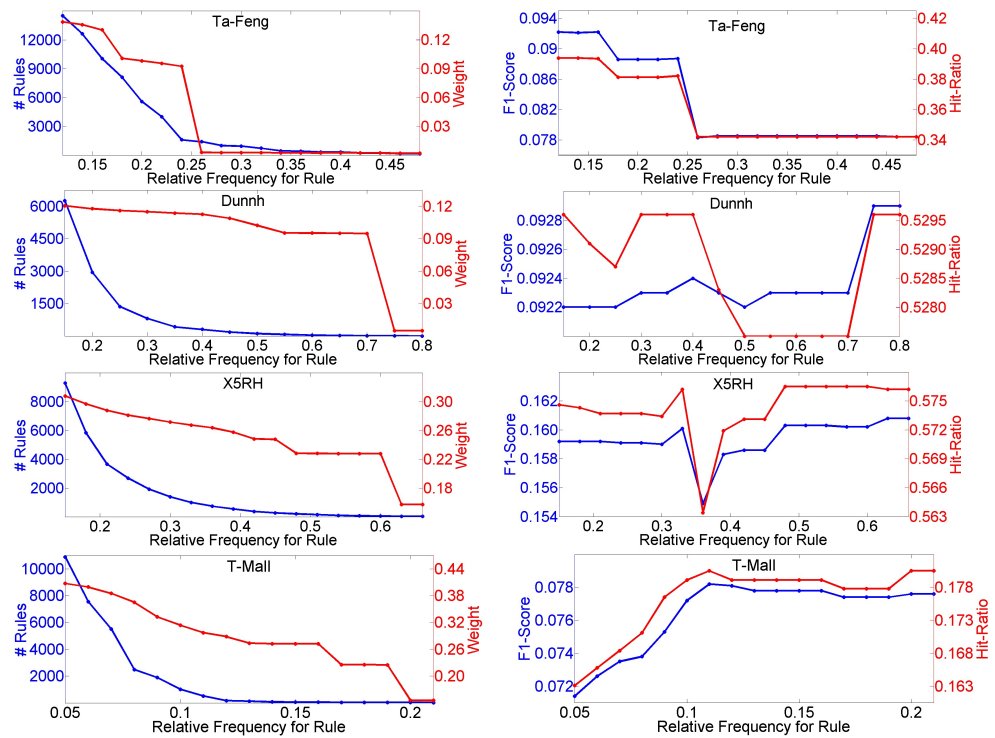


Figure 4. Influences of different values of θ_r on number of rules, *Weight*, *F1-Score* and *Hit-Ratio*.

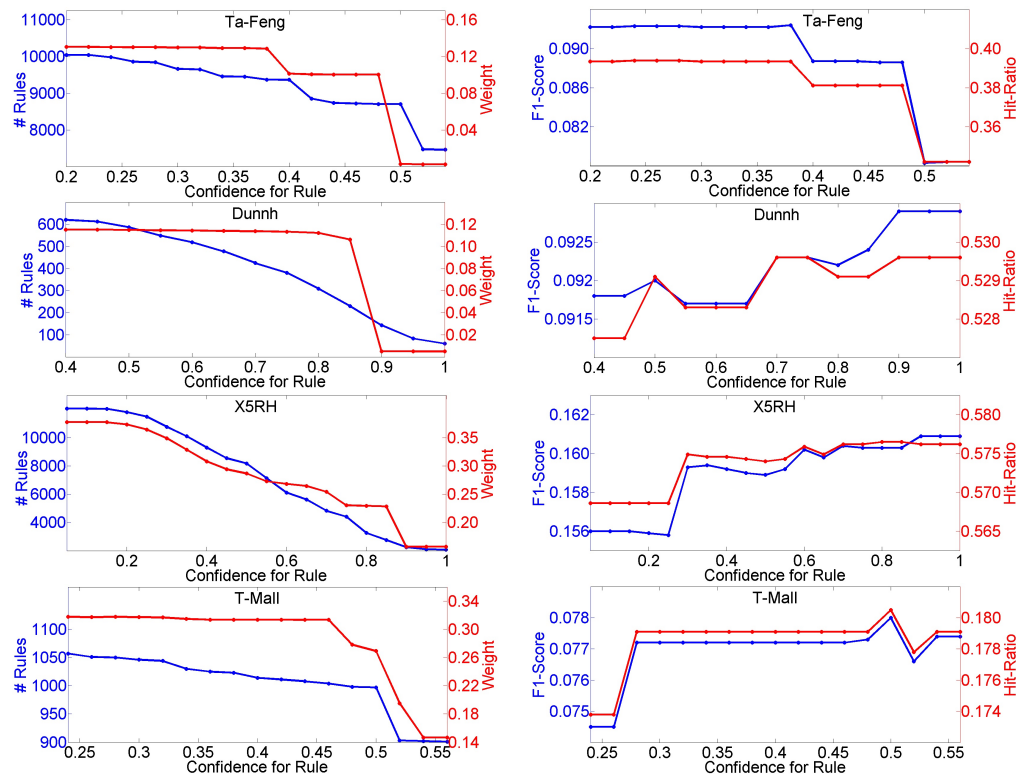


Figure 5. Influences of different values of η on number of rules, *Weight*, *F1-Score* and *Hit-Ratio*.

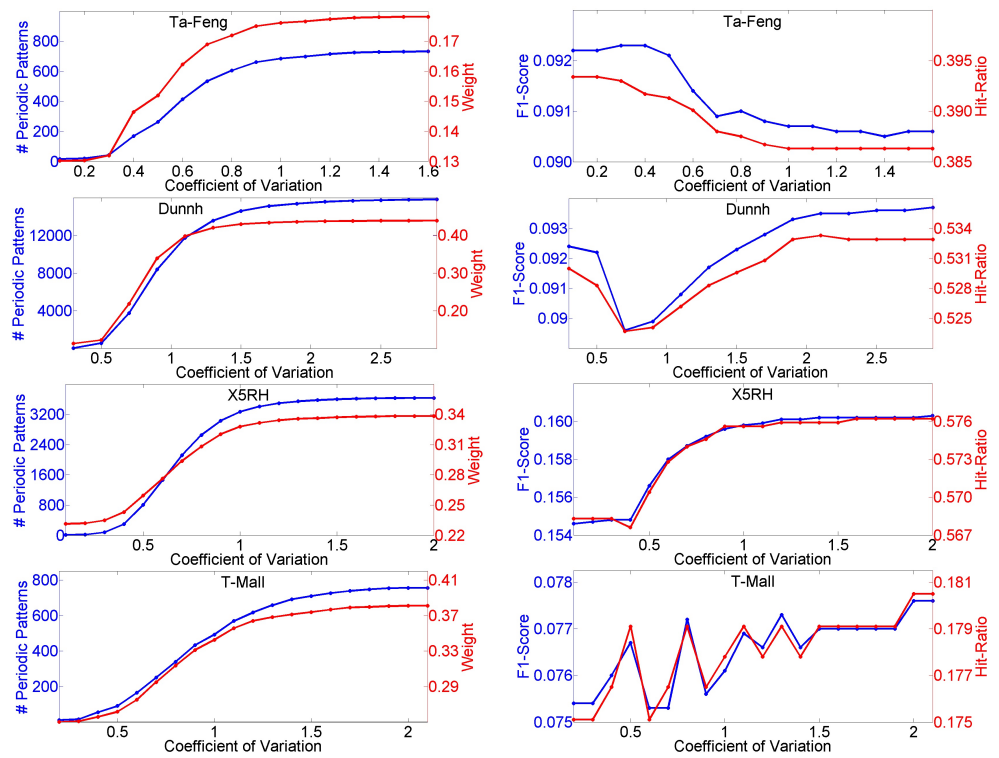


Figure 6. Influences of different values of δ on number of periodic patterns, *Weight*, *F1-Score* and *Hit-Ratio*.

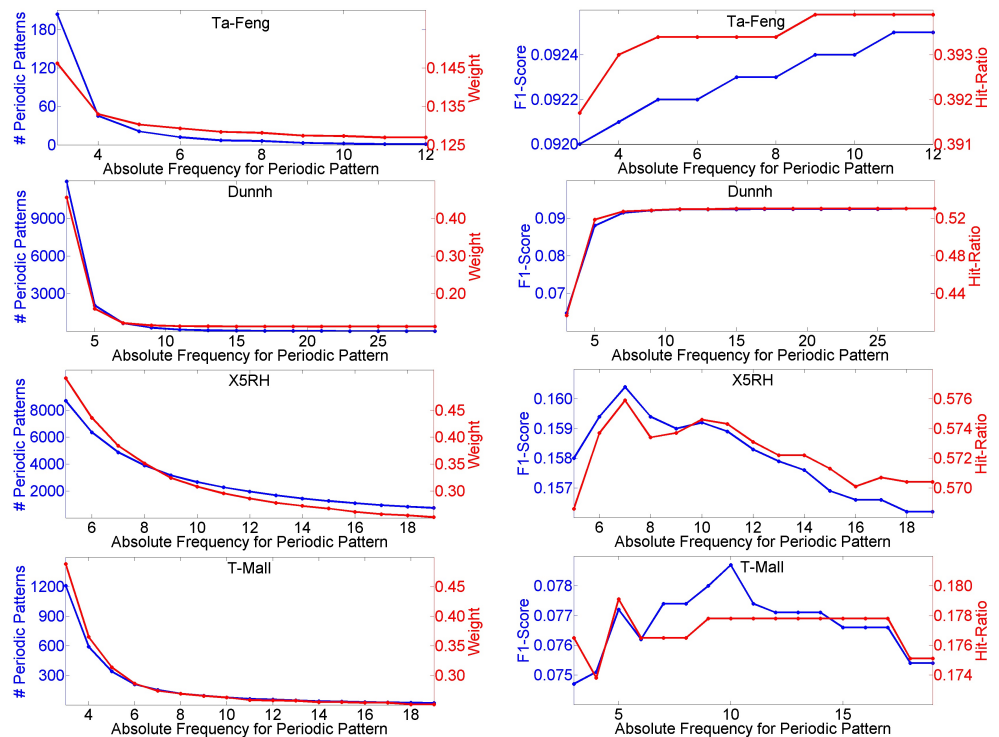


Figure 7. Influences of different values of θ_p on number of periodic patterns, *Weight*, *F1-Score* and *Hit-Ratio*.

6.4. Comparison with Baseline Methods and State-of-the-Art Methods

In this subsection, we report the comparisons of our method with baseline methods, including *TOP*, *MC*, *CLF* and *NMF*; and state-of-the-art methods, including *HRM*, *TBP*, *TIFU* and *UPCF*.

- *TOP* predicts the top- k most frequent items with respect to their appearance, i.e., the number of times that they are purchased, in a customer's purchasing history B_c .
- *MC* [40] makes the prediction based on the last purchase b_{r_c} and on a Markov chain calculated on B_c .
- *CLF* [40]: Due to space limitations, we do not discuss here. See [40] for more details.
- *NMF* (Non-negative Matrix Factorization) [51] is a collaborative filtering method that applies a non-negative matrix factorization to the customers-items matrix. The matrix is constructed from the purchase history of all customers.
- *HRM* (Hierarchical Representation Model) [41] employs a two-layer structure to construct a hybrid representation over customers and items purchase history B from last transactions: the first layer represents the transactions by aggregating item vectors from the last transactions, while the second layer realizes the hybrid representation by aggregating the user's vectors and the transactions representations.
- *TBP* [44] is a new pattern-based method proposed by Guidotti et al. [44] that seeks to simultaneously capture the co-occurrence, sequentiality, periodicity and recurrence of the items in basket sequences.
- *TIFU* [47] (<https://github.com/HaojiHu/TIFUKNN> (Accessed data: 19 October 2021)) is a k -nearest neighbors (kNN) based method.
- *UPCF* [48] (<https://github.com/MayloIFERR/RACF> (Accessed data: 19 October 2021)) denotes User Popularity-based Collaborative Filtering. The model considers a user-based collaborative approach that relies on similar users to find new items that can be of interest to the target user.

Source code for *MC*, *CLF*, *NMF*, *HRM* and *TBP* are provided by [44] (<https://github.com/GiulioRossetti/tbp-next-basket> (Accessed data: 19 October 2021)); and all the algorithms run under the recommended parameter values. Results are shown in Figures 8 and 9. *TBP* runs overtime on the Dunnh dataset.

Figure 8 shows the results of *F1-Score*, we can see that our *SPAP* algorithm has the best *F1-Score* on the Ta-Feng dataset when k is less than 14, and closes to *TOP* on the Dunnh and X5RH datasets. Figure 9 shows the results of *Hit-Ratio*, *SPAP* outperforms the other algorithms on the Ta-Feng dataset, and closes to *TOP* on the Dunnh and X5RH datasets.

We noticed that all algorithms, except *TBP*, exhibit poor performance for *F1-Score* when k is too higher or lower than the average basket size. The most obvious finding is on T-Mall dataset, which has the smallest average basket size. Because *F1-Score* is relevant to the size of recommended basket b_c^* . A higher value for k means we will select more products to recommend, leading to a large size for b_c^* and a lower value for *precision*(b_c', b_c^*); otherwise, we have a lower value for *recall*(b_c', b_c^*). However, when the value of k increases, it results in a higher possibility of being hit, and *Hit-Ratio* rises naturally. Only k has a value nearer to the average basket size, and we have a fairer comparison.

As described above, almost all algorithms exhibited their best performance when k was set as a value of the average basket size. Table 4 lists the performance results in which k has a value of the average basket size, $k = 6$, $k = 9$, $k = 5$ and $k = 3$ for Ta-Feng, Dunnh, X5RH and T-Mall, respectively. *TIFU* has the best performances on the T-Mall dataset. However, our *SPAP* algorithm outperforms other algorithms on the Ta-Feng, Dunnh and X5RH datasets.

Table 4. Comparison with those of the baseline methods and state-of-the-art methods when k is set as average basket size value. The bold is the maximum of the all, and the underline is the maximum of the competing algorithms.

	Dataset	SPAP	TOP	MC	CLF	NMF	HRM	TBP	TIFU	UPCF
<i>F1-Score</i>	Ta-Feng ($k = 6$)	0.0922	<u>0.0796</u>	0.0415	0.0765	0.0617	0.0570	0.0742	0.0731	0.0672
	Dunnh ($k = 9$)	0.0935	<u>0.0926</u>	0.0514	0.0675	0.0685	0.0545	-	0.0864	0.0839
	X5RH ($k = 5$)	0.1604	<u>0.1596</u>	0.1100	0.1316	0.1057	0.0879	0.1502	0.1574	0.1417
	T-Mall ($k = 3$)	0.0783	0.0691	0.0379	0.0735	0.0626	0.0440	0.0757	0.0944	0.0847
<i>Hit-Ratio</i>	Ta-Feng ($k = 6$)	0.3934	<u>0.3517</u>	0.1963	0.3096	0.2894	0.2768	0.2850	0.3158	0.2991
	Dunnh ($k = 9$)	0.5329	<u>0.5287</u>	0.3370	0.4165	0.4554	0.4049	-	0.4917	0.4833
	X5RH ($k = 5$)	0.5771	<u>0.5728</u>	0.4454	0.5056	0.4457	0.4008	0.5075	0.5486	0.5266
	T-Mall ($k = 3$)	0.1778	0.1631	0.0869	0.1725	0.1457	0.1072	0.1602	0.2200	0.1979

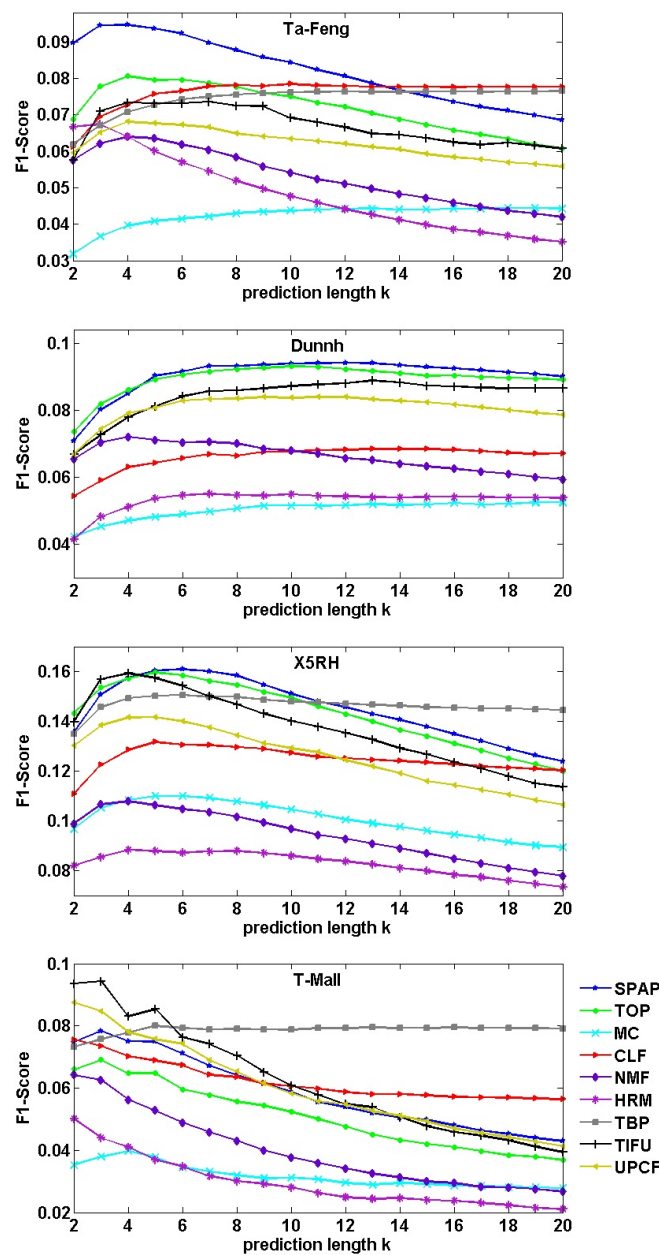


Figure 8. Comparison of *F1-Score* with those of the baseline methods and state-of-the-art methods under different values of k .

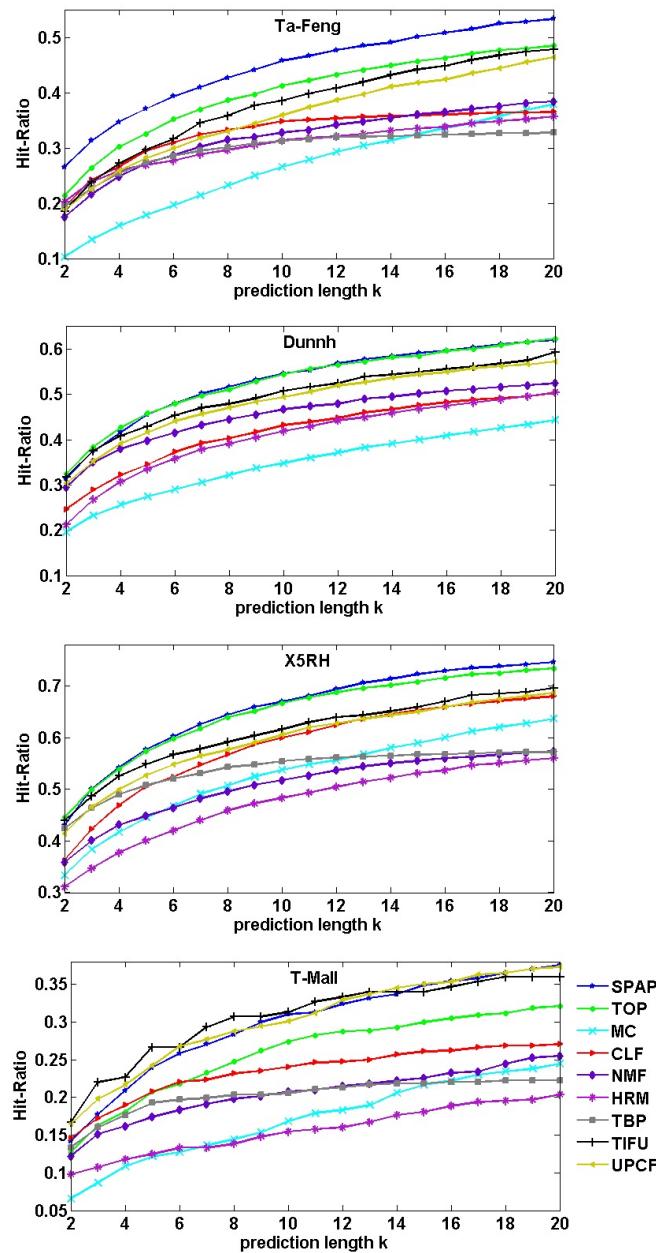


Figure 9. Comparison of *Hit-Ratio* with those of the baseline methods and state-of-the-art methods under different values of *k*.

Patterns have a virtue of reflecting customers’ shopping habits. Due to the fact that people visit physical stores more regularly than online stores, our pattern-based model *SPAP* achieves the best performances on physical store datasets Ta-Feng, Dunnh and X5RH. Table 5 lists the improvements of *SPAP* compared with *TOP* on the Ta-Feng, Dunnh and X5RH datasets; and compared with *TIFU* on the T-Mall dataset.

Table 5. Improvements vs. the maximum of the competing algorithms.

	Improvements vs. <i>TOP</i>			Improvements vs. <i>TIFU</i>
	Ta-Feng	Dunnh	X5RH	T-Mall
<i>F1-Score</i>	15.8%	1%	0.5%	−17.1%
<i>Hit-Ratio</i>	11.9%	0.8%	0.8%	−19.2%

6.5. Running Time

All of these algorithms have a running time ranging from several seconds to several hours. *TOP* is always the fast one [44]. In this subsection, we compare the running time (training time and prediction time) of our algorithm with *TOP*. Both of them are implemented in Java. The results are shown in Table 6. Source code for *MC*, *CLF*, *NMF*, *HRM*, *TBP*, *TIFU* and *UPCF* are implemented in Python, so we do not report their running times here. We terminate the process of *TBP* on the Dunnh dataset when it runs over two days.

Table 6. Running times.

	Ta-Feng	Dunnh	X5RH	T-Mall
<i>TOP</i>	0.0476 s	0.444 s	0.192 s	0.078 s
<i>SPAP</i>	0.749 s	12.838 s	2.824 s	0.234 s

7. Conclusions

In this paper, we propose a pattern-based model for next basket prediction. The method includes pattern prediction and preference prediction. In pattern prediction, first, all sequential rules, periodic patterns and association rules are found together with their statistical characteristics. Then, probability models are built based on their statistical characteristics. Afterward, we use the probability models to calculate the probability of all products in the next basket for a customer. The products that have a higher probability will be selected to recommend to him or her. If k products have been selected, then continue to the prediction of the next customer; otherwise, make preference predictions. In preference prediction, the product that is more frequent in the individual shopping records will be selected first. If some products have the same frequency, then the product that has a higher tendency will be selected. Until all k products are selected. Experiments show that our algorithm outperforms those of the baseline methods and state-of-the-art methods on three of four real-world transaction sequence datasets.

Author Contributions: G.C. implemented the experiment and wrote the first draft of the paper, Z.L. provided funding for the paper and revised it. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61672261 and Grant 61802056, in part by the Natural Science Foundation of Jilin Province under Grant 20180101043JC, in part by the Industrial Technology Research and Development Project of Jilin Development and Reform Commission under Grant 2019C053-9, and in part by the Open Research Fund of Key Laboratory of Space Utilization, Chinese Academy of Sciences, under Grant LSU-KFJJ-2019-08.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets are available at the following links: <https://www.kaggle.com/chiranjivdas09/ta-feng-grocery-dataset> (Accessed data: 19 October 2021), <https://www.kaggle.com/frtgnn/dunnhumby-the-complete-journey> (Accessed data: 19 October 2021), <https://www.kaggle.com/mvyurchenko/x5-retail-hero> (Accessed data: 19 October 2021), <https://github.com/riccotti/CustomertemporalRegularities/tree/master/datasets> (Accessed data: 19 October 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Agrawal, R.; Imielinski, T.; Swami, A. Mining association rules between sets of items in large databases. *ACM SIGMOD Rec.* **1993**, *22*, 207–216. [CrossRef]
2. Jamshed, A.; Mallick, B.; Kumar, P. Deep learning-based sequential pattern mining for progressive database. *Soft Comput.* **2020**, *24*, 17233–17246. [CrossRef]

3. Fournier-Viger, P.; Wu, C.; Tseng, V.S.; Cao, L.; Nkambou, R. Mining partially-ordered sequential rules common to multiple sequences. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 2203–2216. [[CrossRef](#)]
4. Srinivas, P.G.; Reddy, P.K.; Trinath, A.V.; Bhargav, S.; Kiran, R.U. Mining coverage patterns from transactional databases. *J. Intell. Inf. Syst.* **2015**, *45*, 423–439. [[CrossRef](#)]
5. Chen, Y.; Peng, W.; Lee, S. Mining temporal patterns in time interval-based data. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 3318–3331. [[CrossRef](#)]
6. Ray, A.; Holder, L.B.; Bifet, A. Efficient frequent subgraph mining on large streaming graphs. *Intell. Data Anal.* **2019**, *23*, 103–132. [[CrossRef](#)]
7. Fournier-Viger, P.; Wang, Y.; Yang, P.; Lin, C.W.; Kiran, R.U. Tspin: Mining top-k stable periodic patterns. *Appl. Intell.* **2021**. [[CrossRef](#)]
8. Esling, P.; Agon, C. Time-series data mining. *ACM Comput. Surv.* **2012**, *45*, 12:1–12:34. [[CrossRef](#)]
9. Khaleel, M.A.; Dash, G.N.; Choudhury, K.S.; Khan, M.A. Medical data mining for discovering periodically frequent diseases from transactional databases. In *Computational Intelligence in Data Mining*; Springer: New Delhi, India, 2015; Volume 1, pp. 87–96.
10. Chakraborty, S.; Karâa, W.B.A.; Dey, N.; Banerjee, S.; Azar, A.T. Image mining framework and techniques: A review. *Annales De Dermatologie Et De Vénéréologie* **2015**, 237–244. in press.
11. Guo, L.; Liang, J.; Zhu, Y.; Luo, Y.; Sun, L.; Zheng, X. Collaborative filtering recommendation based on trust and emotion. *J. Intell. Inf. Syst.* **2019**, *53*, 113–135. [[CrossRef](#)]
12. Yza, B.; Zsab, D.; Wza, B.; Lin, Y.; Sla, B.; Xue, L. Joint personalized markov chains with social network embedding for cold-start recommendation. *Neurocomputing* **2020**, *386*, 208–220.
13. Fournier-Viger, P.; Gomariz, A.; Gueniche, T.; Soltani, A.; Wu, C.; Tseng, V.S. SPMF: A java open-source pattern mining library. *J. Mach. Learn. Res.* **2014**, *15*, 3389–3393.
14. Fournier-Viger, P.; Gueniche, T.; Zida, S.; Tseng, V.S. ERMiner: Sequential rule mining using equivalence classes. In *Advances in Intelligent Data Analysis XIII*; Springer: Cham, Switzerland, 2014; pp. 108–119.
15. Fournier-Viger, P.; Faghihi, U.; Nkambou, R.; Nguifo, E.M. CMRules: Mining sequential rules common to several sequences. *Knowl. Based Syst.* **2012**, *25*, 63–76. [[CrossRef](#)]
16. Fournier-Viger, P.; Yang, P.; Lin, J.C.; Kiran, R.U. Discovering stable periodic-frequent patterns in transactional data. In *Advances and Trends in Artificial Intelligence. From Theory to Practice*; Springer: Cham, Switzerland, 2019; pp. 230–244.
17. Fournier-Viger, P.; Li, Z.; Lin, J.C.; Kiran, R.U.; Fujita, H. Efficient algorithms to identify periodic patterns in multiple sequences. *Inf. Sci.* **2019**, *489*, 205–226. doi: 10.1016/j.ins.2019.03.050. [[CrossRef](#)]
18. Fournier-Viger, P.; Yang, P.; Kiran, R.U.; Ventura, S.; Luna, J.M. Mining local periodic patterns in a discrete sequence. *Inf. Sci.* **2021**, *544*, 519–548. [[CrossRef](#)]
19. Fournier-Viger, P.; Wu, C.; Tseng, V.S. Mining top-k association rules. In *Advances in Artificial Intelligence—25th Canadian Conference on Artificial Intelligence, Canadian AI 2012*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 61–73.
20. Fournier-Viger, P.; Tseng, V.S. Mining top-k non-redundant association rules. In *Foundations of Intelligent Systems—20th International Symposium, ISMIS 2012*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 31–40.
21. Li, J.; Xu, Y.; Wang, Y.; Chu, C. Strongest association rules mining for personalized recommendation. *Syst. Eng. Theory Pract.* **2009**, *29*, 144–152. [[CrossRef](#)]
22. Fournier-Viger, P.; Lin, C.W.; Duong, Q.H.; Dam, T.L.; Voznak, M. PFP: Discovering periodic frequent patterns with novel periodicity measures. In Proceedings of the 2nd Czech-China Scientific Conference, Ostrava, Czech Republic, 7 June 2016; IntechOpen: London, UK, 2017.
23. He, R.; Kang, W.; McAuley, J.J. Translation-based recommendation: A scalable method for modeling sequential behavior. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, 13–19 July 2018; pp. 5264–5268. [[CrossRef](#)]
24. Zhao, C.; You, J.; Wen, X.; Li, X. Deep bi-lstm networks for sequential recommendation. *Entropy* **2020**, *22*, 870. [[CrossRef](#)] [[PubMed](#)]
25. Huang, T.; Zhang, D.; Bi, L. Neural embedding collaborative filtering for recommender systems. *Neural Comput. Appl.* **2020**, *32*, 17043–17057. [[CrossRef](#)]
26. Lazcorreta, E.; Botella, F.; Fernandez-Caballero, A. Towards personalized recommendation by two-step modified apriori data mining algorithm. *Expert Syst. Appl.* **2008**, *35*, 1422–1429. [[CrossRef](#)]
27. Najafabadi, M.K.; Mohamed, A.H.; Onn, C.W. An impact of time and item in fluencer in collaborative filtering recommendations using graph-based model. *Inf. Process. Manag.* **2019**, *56*, 526–540. [[CrossRef](#)]
28. Cui, Q.; Wu, S.; Liu, Q.; Zhong, W.; Wang, L. MV-RNN: A multi-view recurrent neural network for sequential recommendation. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 317–331. [[CrossRef](#)]
29. Zhang, L.; Wang, P.; Li, J.; Xiao, Z.; Shi, H. Attentive hybrid recurrent neural networks for sequential recommendation. *Neural Comput. Appl.* **2021**, *33*, 11091–11105. [[CrossRef](#)]
30. Fu, X.; Budzik, J.; Hammond, K.J. Mining navigation history for recommendation. In Proceedings of the IUI 2000—International Conference on Intelligent User Interfaces, New Orleans, LA, USA, 9–12 January 2000; pp. 106–112. [[CrossRef](#)]
31. Wang, D.; Yu, G.; Bao, Y. An approach of association rules mining with maximal nonblank for recommendation. *J. Softw.* **2004**, *15*, 1182–1188.

32. Wang, T.; Yang, A. Research of weighted association rule and its application in personalization recommendation system. *J. Zhengzhou Univ. Nat. Sci.* **2007**, *39*, 65–69.
33. Ding, Z.; Chen, J. Individuation recommendation system based on association rule. *Comput. Integr. Manuf. Syst.* **2003**, *9*, 891–893.
34. Ding, Z.; Wang, J.; Wang, D.; Bao, Y.; Yu, G. A web personalized recommendation method based on uncertain consequent association rules. *Comput. Sci.* **2003**, *30*, 69–72. 88.
35. Li, J.; Wang, Y.; Xu, Y. Personalized recommendation based on strong association rule mining for mass customization. In Proceedings of the 3rd Interdisciplinary of World Congress of Mass Customization and Personalization MCPC2005, Hong Kong, China, 18–21 June 2005.
36. Najafabadi, M.K.; Mahrin, M.N.; Chuprat, S.; Sarkan, H.M. Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data. *Comput. Hum. Behav.* **2017**, *67*, 113–128. [\[CrossRef\]](#)
37. Chen, G.; Wei, Q.; Liu, D.; Wets, G. Simple association rules (SAR) and the SAR-based rule discovery. *Comput. Ind. Eng.* **2002**, *43*, 721–733. [\[CrossRef\]](#)
38. Zhou, H.; Hirasawa, K. Evolving temporal association rules in recommender system. *Neural Comput. Appl.* **2019**, *31*, 2605–2619. [\[CrossRef\]](#)
39. Maske, A.R.; Joglekar, B. An Algorithmic Approach for Mining Customer Behavior Prediction in Market Basket Analysis. In *Innovations in Computer Science and Engineering; Lecture Notes in Networks and Systems*; Saini H., Sayal R., Govardhan A., Buyya R., Eds.; Springer: Singapore, 2019; Volume 74. [\[CrossRef\]](#)
40. Cumby, C.M.; Fano, A.E.; Ghani, R.; Krema, M. Predicting customer shopping lists from point-of-sale purchase data. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 402–409. [\[CrossRef\]](#)
41. Wang, P.; Guo, J.; Lan, Y.; Xu, J.; Wan, S.; Cheng, X. Learning hierarchical representation model for next basket recommendation. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 403–412. [\[CrossRef\]](#)
42. Guidotti, R.; Rossetti, G.; Pappalardo, L.; Giannotti, F.; Pedreschi, D. Next Basket Prediction using Recurring Sequential Patterns. *arXiv* **2017**, arXiv:1702.07158. .
43. Guidotti, R.; Rossetti, G.; Pappalardo, L.; Giannotti, F.; Pedreschi, D. Market basket prediction using user-centric temporal annotated recurring sequences. In Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM), ICDM 2017, New Orleans, LA, USA, 18–21 November 2017; IEEE Computer Society: 2017; pp. 895–900. [\[CrossRef\]](#)
44. Guidotti, R.; Rossetti, G.; Pappalardo, L.; Giannotti, F.; Pedreschi, D. Personalized market basket prediction with temporal annotated recurring sequences. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 2151–2163. [\[CrossRef\]](#)
45. Jain, S.; Sharma, N.K.; Gupta, S.; Doohan, N. Business Strategy Prediction System for Market Basket Analysis. In *Quality, IT and Business Operations*; Springer Proceedings in Business and Economics; Kapur P., Kumar U., Verma A., Eds.; Springer: Singapore, 2018. [\[CrossRef\]](#)
46. Kraus, M.; Feuerriegel, S. Personalized Purchase Prediction of Market Baskets with Wasserstein-Based Sequence Matching. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2643–2652.
47. Hu, H.J.; He, X.N.; Gao, J.Y.; Zhang, Z.L. Modeling Personalized Item Frequency Information for Next-basket Recommendation. In Proceedings of the SIGIR 2020: The 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event China, 25–30 July 2020; pp. 1071–1080.
48. Faggioli, G.; Polato, M.; Aiolli, F. Recency aware collaborative filtering for next basket recommendation. In Proceedings of the UMAP'20: 28th ACM Conference on User Modeling, Adaptation and Personalization, Genoa, Italy, 14–17 July 2020; pp. 80–87. [\[CrossRef\]](#)
49. Fumarola, F.; Lanotte, P.F.; Ceci, M.; Malerba, D. CloFAST: Closed sequential pattern mining using sparse and vertical id-lists. *Knowl. Inf. Syst.* **2016**, *48*, 429–463. [\[CrossRef\]](#)
50. Guidotti, R.; Gabrielli, L.; Monreale, A.; Pedreschi, D.; Giannotti, F. Discovering temporal regularities in retail customers' shopping behavior. *Epj Data Sci.* **2018**, *7*, 6. [\[CrossRef\]](#)
51. Lee, D.D.; Seung, H.S. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13; Papers from Neural Information Processing Systems (NIPS) 2000*; MIT Press: Cambridge, MA, USA, 2000; pp. 556–562.