Journal of Retailing and Consumer Services 60 (2021) 102431

Contents lists available at ScienceDirect

Journal of Retailing and Consumer Services

journal homepage: http://www.elsevier.com/locate/jretconser

Temporary rules of retail product sales time series based on the matrix profile

Hailin Li[a,b], Yenchun Jim Wu[c,d,*], Shijie Zhang[a], Jinchuan Zou[a]

[a] *College of Business Administration, Huaqiao University, Quanzhou, 362021, China*
[b] *Research Center for Applied Statistics and Big Data, Huaqiao University, Xiamen, 361021, China*
[c] *Graduate Institute of Global Business and Strategy, National Taiwan Normal University, Taipei, 10645, Taiwan*
[d] *National Taipei University of Education, Taipei, 10671, Taiwan*

ARTICLE INFO

Keywords:
Product sales correlation analysis
Market basket analysis
Temporary association rules
Time series data mining
Matrix profile
Motif discovery

ABSTRACT

Correlation analysis in the retail industry mainly involves market basket analysis. This kind of correlation analysis of retail product sales does not reflect the information regarding the time or quantity of the sales. Product sales datasets contain rich information about the correlations between different products at different times. The co-occurrence of similar sales subsequences reveal that product sales are correlated in a specific time period. Therefore, searching for similar co-occurrence patterns can help analyze the temporary correlations between products. The search for similar subsequences can be viewed as motif discovery in time-series datasets. In the field of motif discovery, the matrix profile (MP) provides an overwhelming advantage in detecting motifs. In this study, our aim is to discover motifs using MP, and hence, analyze the temporary sales correlations between products. The results of our numerical experiments indicate what products customers will purchase at what time. As opposed to strong association rules, we name the correlation rules in this work as temporary rules (TRs). Our results also show that customers' preferences are not stable and change with time. In the retail industry, TRs can help business owners make suitable product promotions at appropriate times. Moreover, our analysis demonstrates that TRs can extract more interesting information and patterns than mining with association rules.

1. Introduction

Several enterprises store massive data in their databases or data warehouses, especially in the retail industry (Fayyad et al., 1996; Holzer, 2020). These databases are so extensive that it is not easy to extract useful information from them (Halkidi, 2000; Moodley et al., 2020). In the field of retail studies, the correlation analysis of product sales is mainly based on market basket analysis (MBA), which in turn uses the Apriori and FP growth algorithms to mine transaction records (Halkidi, 2000; Agrawal et al., 1994). Regarding the efficiencies of these algorithms, the Apriori algorithm involves a high I/O cost because of the continuous scanning of the transactions; whereas, the FP growth algorithm scans the database only twice and stores the frequency patterns in the FP-tree instead of performing a continuous scan (Zeng et al., 2015). Parallel implementations have become increasingly popular owing to the increase in efficiency of the Apriori algorithm. Agrawal and Shafer (1996) introduced three Apriori approaches supporting parallel calculations, namely, the count distribution, data distribution, and candidate distribution algorithms. However, the traditional association rules mined by the Apriori or FP growth algorithms provide business enterprises with limited information and knowledge, as they only yield the probability of products that were purchased together. Many business enterprises accumulate large quantities of data from their day-to-day operations, sequential rules can increase the timeliness or temporal information content of these data based on standard association rules. However, there are several limitations of sequential rule mining, considering which Fournier et al. (2015) proposed the RuleGrowth algorithm and its extension, the TRuleGrowth algorithm (which adds a separate time window for mining rules). Fournier et al. (2015) proved that the sequential rules generated by these algorithms have a higher prediction accuracy compared to other algorithms based on the gathered data. Sequential rule mining is also used for making predictions and to facilitate fast decision-making. To this end, the TRuleGrowth and CMRules algorithms have been used to generate sequential rules to

* Corresponding author. Graduate Institute of Global Business and Strategy, National Taiwan Normal University, Taipei, 10645, Taiwan.
E-mail address: wuyenchun@gmail.com (Y.J. Wu).

https://doi.org/10.1016/j.jretconser.2020.102431
Received 18 August 2020; Received in revised form 11 December 2020; Accepted 12 December 2020
Available online 7 January 2021

predict the future interest of users in web pages (Singh et al., 2017). It has been shown that the sequential rule prediction accuracy of the TRuleGrowth algorithm can be increased by adding constraints such as length, attribute, regular expression, gap, and duration (Pujari et al., 2017). Such an approach also reduces the time and memory consumed to generate sequential rules. The TRuleGrowth algorithm has been further extended and improved to obtain the M_TRuleGrowth algorithm (Sagare et al., 2020), which has been applied in predictive data analytics to provide support for corporate decision-making. Moreover, it is useful for making predictions in various fields. Numerical experiments using real-world datasets have shown that M_TRuleGrowth has better execution times even if there are new developments in the association rules. Recently, there has been much focus on the multi-dimensional scaling technique, which is considered to be the second stage of research in the sales correlation of retail products.

In the second stage of sales research, it is usual to carry out a correlation analysis of the consumption of products in terms of factors such as time and quantity. Using the techniques of time-series data mining, similar trends in the sales of different products starting at a particular time and lasting for a specific period can be detected. These sales trends, regarded as co-occurrence patterns, represent the correlation between the products purchased within a specific period. Two kinds of information can be obtained from these trends, namely, the time when similar sales trends tend to occur, and the duration for which these trends last. We refer to these patterns as temporary rules (TRs) in this work.

Fig. 1 shows the sales patterns of two products, A and B. Their sales trends are similar in the subsequences denoted in red. This co-occurrence of similar patterns indicates that there exists a temporary sales correlation between products A and B in a specific time window. We assume that products A and B were purchased 30 times during the similar subsequence. If the traditional Apriori algorithm is used to scan the transactions, then the support of A and B (i.e., sup = 30) may not meet the minimum support threshold in this case (i.e., sup = 40). Thus, Apriori regards these product sales to be uncorrelated. Note that A and B exhibit a co-occurrence of similar sales trends only in the specific periods in which they are correlated. By contrast, TRs can extract these similar patterns and point out potentially useful information, such as the start time and duration of the correlated sales windows, that traditional methods tend to ignore.

In the retail industry, cross-marketing is an extremely popular way to increase sales growth. Several scholars have discussed the potential benefits of cross-marketing, which we discuss here briefly (Neslin and Shankar, 2009; Zhang et al., 2010; Berry et al., 2010; Behera et al., 2020). More recently, Cao and Li (2015) confirmed that cross-marketing has a positive influence on sales growth in a quantitative way. However,

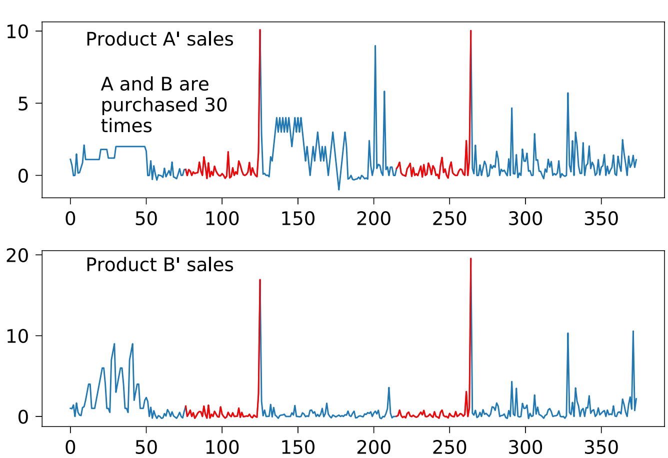

Fig. 1. Sales patterns of products A and B. The similar sales subsequences are denoted in red.

to the best of our knowledge, few studies have considered the time factor in their analyses. Time is critical for product promotions in cross-marketing. Owing to a change in the customers' purchase behavior with time, the sales correlation between two products may not always remain strong. For example, peanut and beer may be frequently purchased together by some people during summer, whereas other people may prefer to purchase these products together during winter. Therefore, for cross-promotion to work, business managers and leaders should choose "similar" products to promote to the "right" customers at the "right" time. Our approach can solve this problem by discovering appropriate TRs to facilitate cross-promotion to attract customers' attention, and hence, maximize profit.

Nearly 70% of e-commerce leaders use predictive analysis to assess the sales of their products (See-To and Ngai, 2018). This indicates that sales prediction is extremely meaningful for business owners. This is where TRs can play an important role by forecasting product sales. Generally, sales correlation between products is not occasional, and is often periodic. According to the frequency of TRs, retailers can forecast the next set of similar trends that may occur in their sales. In addition, the discovery of TRs in turn depends on the occurrence of similar sales patterns, which keep repeating the same as TRs. These patterns show that retail product sales change within a specific time period. Moreover, previous sales records can help retailers predict future sales.

The motivation for the present study is threefold. First, in MBA, it is important to know which products have sales correlation to facilitate bundle sales. This issue can be addressed by a similarity search in sales subsequences that can be obtained from the time-series data of different product sales. Second, it is essential to know when is it suitable to recommend the products to the consumers, and how long this recommendation can last. This can be achieved by using multi-motifs derived from similar subsequences that indicate which products have sales correlation during which time. Moreover, the length of the multi-motifs indicates how long the sales correlation lasts. Third, it is important to be able to effectively extract the sales correlation between products from retail datasets. In this work, we use the fast matrix profile (MP) to discover motifs in time-series datasets and also design the procedure to discover TRs.

To quickly discover what products can be sold to the right customer at the right time, we use the techniques of time-series data mining based on the MP to design our algorithm. We first measure the similarity between a series of product sales. These relationships between products can be used to construct a similarity network of product sales. According to this network, different products can be selected to form groups. We consider the time series of product sales in a group to be a multivariate time-series dataset. Note that multi-motifs can be found in multivariate time-series data. If these multi-motifs are found to be repeating, then that indicates that the temporary relationships between product sales are frequent. Thus, TRs can be generated from a set of multi-motifs.

2. Preliminary studies

2.1. Related work

Machine learning and artificial intelligence have also promoted the development of pattern mining technology, which plays an important role in fraud detection and trend learning. Owing to the different attributes of datasets, the patterns mined from them can have a wide range of variety, including frequent patterns, sequential patterns (Kim and Choi, 2020; Cheng et al., 2015), high utility patterns (Ahmed et al., 2012b, a), subgraph patterns (Braun et al., 2014; Cuzzocrea et al., 2015), episodes (Amphawan et al., 2015), infrequent patterns (Cagliero et al., 2018), and probabilistic frequent patterns (Li et al., 2018). Frequent patterns are itemsets, subsequences, or substructures that appear in a dataset frequently, that is, with a frequency no less than the user-specified threshold. Sequential pattern mining is a data mining task specialized for analyzing sequential data. It has numerous real-life applications because data is often naturally encoded as a sequence of symbols in many fields, such as bioinformatics, e-learning, MBA, text analysis, and webpage clickstream analysis. Sequential patterns are used to find complex itemsets in sequential order. Similarly, high utility patterns are used to find items with a high value or utility, whereas subgraph patterns are used to find items that meet user-specified thresholds. Episodes are used to discover continuous events, infrequent patterns are used to find itemsets with low frequency, whereas probabilistic frequent modes are used to find frequent itemsets in uncertain data streams that meet the minimum support as well as the minimum probability parameters. We can distinguish between all these patterns from three aspects: whether the items or itemsets are in order, their weights are equal, and whether they are continuous. Different types of patterns can be mined from different data attributes; moreover, different patterns have specific applications of their own. The key technologies of pattern mining include the windows method (Manku and Motwani, 2002; Li et al., 2008) and attenuation method (Chang and Lee, 2006; Shin et al., 2014). Data mining consists of extracting information from databases to understand the underlying data or to make decisions. Pattern mining is usually used in combination with existing methods, such as precise methods or approximate methods, which depend on the pattern in question. Pattern correlation is an important research direction in the field of pattern mining, especially in data that have time attributes. Correlation patterns can be either positive or negative and may contain time gaps as well (Liu et al., 2018). By analyzing correlation patterns, a broad insight can be gained into the nature of each pattern. For example, the correlation analysis of sales data, gene expression data (Gong et al., 2010), and production data (Fu et al., 2019) can find widespread positive or negative correlation patterns, which can provide useful decision-making directions for real-world applications. Commonly used measures of pattern correlation include bond, all-confidence, and frequency affinity, as well as various statistical tests (Shen, 2019) and newly developed algorithms (Ren et al., 2020; Nai et al., 2019).

Several techniques have been applied to the field of time-series data mining (Esling and Agon, 2012; Li, 2019; Li and Wei, 2020), of which motif discovery has garnered a lot of interest, as it is an efficient and effective way to discover TRs in retail sales datasets. In a previous work (Li et al., 2020), we conducted a preliminary study of the sales correlation between products; however, we used methods such as time-series data clustering and similarity search to find the relevance of the commodity sales, which although valid were not very efficient. Moreover, the methods used in our previous work were effective only for univariate time-series data but failed to discover motifs in multivariate time-series data. Repeated patterns are generally regarded as motifs in a time-series dataset, and motif discovery is the process of searching for similar patterns that form a superset of repeating patterns. This is because repeated patterns are also detected to be the most similar. In this section, we introduce the theory of motif discovery.

Yeh et al. (2016) introduced the MP, which is a subversive method that has important implications for time-series data mining because of its generality, versatility, simplicity, and scalability. Compared with the algorithms presented by Mueen et al. (2009a), Mueen et al. (2009b), and Mueen and Chavoshi (2015), MP and its extension STAMP have an overwhelming advantage in terms of time complexity and accuracy (Yeh et al., 2016). Keogh and collaborators extended the MP algorithm and conducted several inspiring studies in various fields, such as seismology (Shakibay Senobari et al., 2018), biology (Samee et al., 2019), and electric power research (Funde et al., 2019). Interestingly, the STOMP algorithm broke the one hundred million barrier to search for motifs (Zhu et al., 2016). Several researchers have also attempted to explore scalable algorithms and the scalable discovery of variable-length motifs (Zhu, 2018; Linardi et al., 2018). In the context of multivariate time series, Yeh et al. (2017) introduced the multi-matrix profile to search for multi-motifs.

In this work, we intend to introduce the MP and STAMP algorithms for univariate time-series mining and the multi-matrix profile and mSTAMP algorithms for multivariate time-series mining.

2.2. Definitions and notation

In this section, we provide all the required definitions to help readers better understand the background of our work.

Definition 1. A time series T is a vector of real numbers t_i, such that $T = t_1, t_2, \cdots, t_n$, where n is the length of T.

Definition 2. A subsequence $T_{i,w}$ is the continuous part of $T_{i,w}$, where i is the starting position and w is the length of the subsequence $T_{i,w}$. Therefore, there are $n - w + 1$ subsequences in T obtained by applying a sliding window of length w to the entire time series.

Definition 3. A distance profile D is a vector consisting of the Euclidean distances between a query sequence and every subsequence in time series T.

Definition 4. A distance profile set D_{AB} is a matrix of the Euclidean distances between every subsequence of time series T_A and T_B.

Definition 5. An MP P_{AB} is a vector of Euclidean distances between each subsequence in T_A and its most similar subsequence in T_B. We can create this vector by scanning every column in the distance profile set D_{AB} to select the minimum value therein as a value in vector P_{AB}.

Definition 6. An MP index I_{AB} is a vector of the starting positions of the subsequences in T_B which are most similar to the subsequences in T_A. Note that before introducing the algorithm to compute the MP, we introduce the MASS algorithm below.

Definition 7. MASS algorithm is a method designed to compute the distance between full subsequence joins. It requires query Q and a time series T as input and returns a distance profile D as output.

Note that MASS stands for Mueen's ultra-fast algorithm for similarity search. Overall, it is the fastest similarity search algorithm for time-series subsequences under Euclidean distances. This algorithm is exceptionally robust and is independent of query and data. It utilizes the fast Fourier transform to deal with query and data, and then performs a classical convolution to obtain the dot product QT between query and data. The remaining calculation can be summarized as follows:

$$D = \sqrt{2w\left(1 - \frac{QT - w\mu_Q M_T}{w\sigma_Q \sum_T}\right)} \tag{1}$$

where D represents the distance profile between the query and all subsequences in the time series; QT denotes the dot product between the query and all subsequences in the time series; μ_Q and σ_Q are the mean and standard deviation of the query, respectively; M_T and $\sum_T$ represent the vectors that store the mean and standard deviation of each subsequence of the time series, respectively; and finally, w is the length of the query.

Definition 8. The STAMP algorithm presented in Algorithm 1 is a method designed to compute the MP. It requires two time series T_A and T_B as input and returns the MP P_{AB} and position index I_{AB} as output.

Algorithm 1. STAMP(T_A,T_B,w)

Note that STAMP based on the MASS algorithm. In line 1 of Algorithm 1, the initialization is carried out. In the main loop, the algorithm attempts to obtain the MP through iterations (lines 2 to 7). In each iteration, the algorithm calculates the distance profile between the current pattern in T_B and all patterns in T_A (line 3). Then, the SelectValuesMin function selects the values in the distance profile that are less than the corresponding values in the MP, and updates the MP (line 5). Finally, the algorithm returns the MP and the corresponding index.

Definition 9. The motif $motif_{AB}$ represents the most similar pair of

Algorithm 1 STAMP(T_A,T_B,w)

Input: Two time series T_A and T_B, and length of subsequence w.

Output: MP P_{AB} and MP index I_{AB}.

```
P_AB ⟵ Initialization(inf), I_AB ⟵ Initialization(0), L_B ⟵ Length(T_B), nums ⟵ L_B − w + 1
for item in nums do
    D = MASS(T_B(item), T_A)
    if any(D¡P_AB) then
        P_AB, I_AB ⟵ SelectValuesMin(P_AB, I_AB, D, item)
    end if
end for
Return P_AB, I_AB
```

subsequences in the MP P_{AB}. In practice, we can retrieve several similar pairs to form a motif set or chain.

In fact, using the MP P_{AB}, we can obtain the corresponding motif by selecting the minimum value pairs, as shown in Fig. 2.

Fig. 2 (a) shows a segment of time series Q having 375 data points and a pair of motifs highlighted in red. The selection of the motif depends on the MP. Fig. 2(b) shows the MP of time series Q. The position of the minimum value in the MP is 110. Therefore, the subsequence starting at this position and pointing to another subsequence (i.e., the index stored in MP Index) is extracted from the time series.

We have already introduced motif discovery for a single time series. Similarly, a multi-matrix profile can be designed for a multi-motif discovery. We describe some terms commonly used in multivariate time series in the following definitions (Yeh et al., 2017).

Definition 10. A multivariate time series mT is a set of co-evolving time series mT^i, such that $mT = [mT^1, mT^2, \cdots, mT^d]$, where d is the number of dimensions.

Definition 11. A subsequence of multivariate time series $mT_{i,w}$ is a subset of mT, where i is the starting position and w is the length of the subsequence. Formally, $mT_{i,w} = [mT^1_{i,w}, mT^2_{i,w}, \cdots, mT^d_{i,w}]$.

Definition 12. A sub-dimensional subsequence $mT_{i,w}(X)$ is a subset of $mT_{i,w}$, where X implies that certain specific dimensions are selected.

Definition 13. A distance measure function for a time series consisting of multi-dimensional subsequences computes the distance between $mT_{i,w}$ and $mT_{j,w}$, which is the minimum value of the distance between $mT^X_{i,w}$ and $mT^X_{j,w}$.

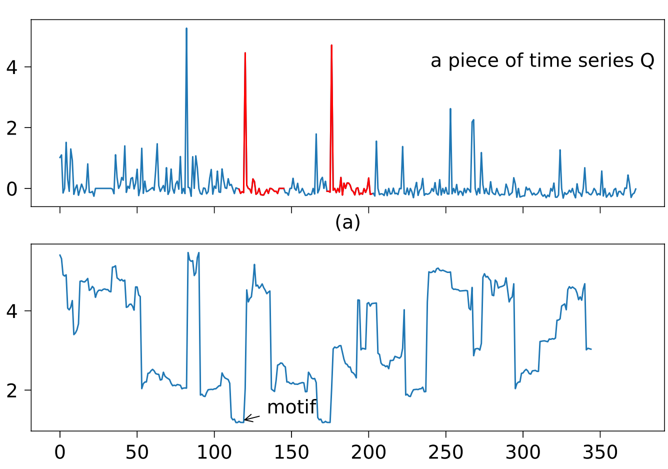

Fig. 2. Motif discovery based on MP. (a) A segment of time series Q with the motif highlighted in red. (b) MP of time series Q.

Note that because suitable dimensions need to be selected to determine the minimum distance, we need to store information about the selection of dimensions.

Definition 14. A distance profile mD is a vector measuring the Euclidean distance between a query and every subsequence in time series mT.

Definition 15. A multi-dimensional distance profile set mD_{AB} is a matrix consisting of the Euclidean distance between every subsequence of time series mT_A and mT_B.

Definition 16. A multi-dimensional MP mP_{AB} is a vector consisting of the Euclidean distance between every subsequence in mT_A and its most similar subsequence in mT_B. This vector can be updated by selecting the minimum value of each column of the distance profile set mD_{AB}.

Definition 17. The mSTAMP algorithm presented in Algorithm 2 is a method that computes the multi-matrix profile. It requires a multivariate time series as input and returns a multi-matrix profile and its relative index as output.

Algorithm 2. mSTAMP(mT_A,mT_B,w)

Although mSTAMP is similar to STAMP, the distance measure between the corresponding child elements is different. From lines 4 to 7 of Algorithm 2, the algorithm calculates the distance between the current query and other subsequences in the same dimension. Next, the function distanceProfile calculates the distance matrix based on the MASS algorithm, which is the fastest similarity search algorithm for time-series subsequences under Euclidean distances. In line 8, the algorithm sorts the distances obtained from all dimensions using a greedy fashion strategy. Then, the minimum distance of the subsequences is selected from some specific dimensions, which represents the distance between the multi-subsequences (lines 9 to 14). Finally, the algorithm returns the multi-matrix profile and its relative indices.

Definition 18. The multi-motif $mmotif_{AB}$ is the most similar pair of subsequences in mP_{AB}.

Fig. 3 shows a multivariate time series and its multi-motif. It shows that the multi-motif appears only in the first two time series of mT.

3. TR discovery method

Our research aims to mine the TRs, and thus analyze the sales correlations between products. Our work can be divided into two stages, as shown in Fig. 4, namely, motif discovery and TR discovery. Motif discovery is an antecedent of TR discovery. In stage 1, that is the process of motif discovery, we first measure the similarity between the different product sales time series. These relationships between products can help

Algorithm 2 $mSTAMP(mT_A, mT_B, w)$

Input: Two multivariate time series mT_A and mT_B, and the length of subsequence w.

Output: Multi-matrix profile mP_{AB}, MP index mI_{AB}, and MP stock mC_{AB}.

$mP_{AB} \longleftarrow Initialization(inf)$, $[mI_{AB}, mC_{AB}, D, D'] \longleftarrow Initialization(0)$
$L \longleftarrow Length(T_A), nums \longleftarrow L - w + 1$
for item in nums **do**
  for i in 1 to d **do**
    $Q \longleftarrow mT_B(i, item : item + w - 1)$
    $D(i,:) \longleftarrow distanceProfile(Q, mT_A(i,:))$
  end for
  $D \longleftarrow columnWiseAscendingSort(D)$
  for i in 1 to d **do**
    $D' \longleftarrow D' + D(i,:)$
    $D'' \longleftarrow D'/i$
    $[mP_{AB}(i,:), mI_{AB}(i,:), mC_{AB}(i,:)] \longleftarrow elementWiseMin(mP_{AB}(i,:), D'')$
  end for
end for
Return $mP_{AB}, mI_{AB}, mC_{AB}$

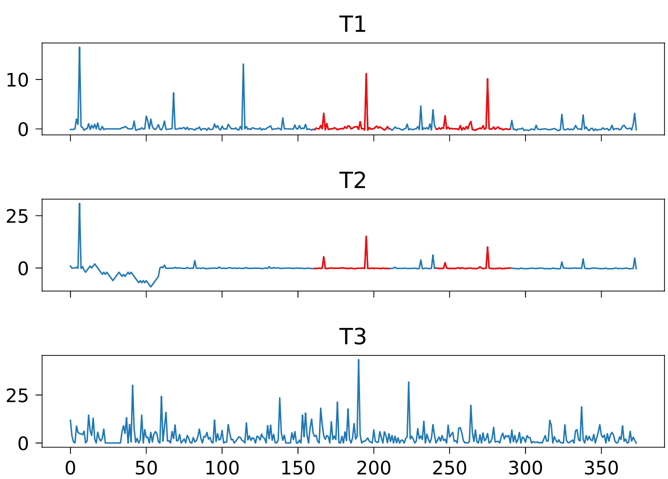

Fig. 3. Segment of a multivariate time series mT consisting of three co-evolving single time series $T1$, $T2$ and $T3$. The multi-motifs are denoted in red.

construct a similarity network of products. According to this network, different groups of products can be selected. We consider the product sales time series in such a group to be a multivariate time series. Multi-motifs can be found in such multivariate time series. If these multi-motifs are repeating, then it implies that the temporary relationships that they represent are frequent. Consequently, in stage 2, TRs can be generated from a multi-motif set.

3.1. *Motif discovery*

In general, motif discovery involves two steps: measuring the similarity between product sales time series, and multi-motif discovery. Similarity measurement is a basic function of multi-motif discovery. We can distinguish between different groups of product sales time series according to their similarity. In a product sales group, a multi-matrix profile can be computed effectively to obtain multi-motifs.

3.1.1. *Similarity measure*

Products that have more co-occurrences of similar sales subsequences are clearly more related to each other. Therefore, we intend to measure the similarity of product sales by counting the first several pairs of similar sales subsequences.

The STAMP algorithm computes the MP between the sales time series of two products based on the Euclidean distance instead of dynamic time warping (Li, 2021). In this case, we need to search for the most similar subsequences among the sales time series of all the products. This implies that the MP of the product sales time series needs to update with the iterations to remain the minimum distance. Therefore, we employ our algorithm named calculateProductPatterns (CPP), as shown in Algorithm 3, which is based on STAMP.

The CPP algorithm requires the set of product sales time series S_T and length of subsequence w as input and returns P, I, and C that represent the sets of MP, MP index, and MP code, respectively, as output. The MP index I stores the index of the most similar subsequence, whereas MP code C records the stock code of the corresponding subsequence. The structure of these sets is shown in Fig. 5.

Algorithm 3. calculateProductPatterns(S_T,w)

In line 1 of Algorithm 3, the memory for the MP and some related information is initialized. For each iteration in the main loop (lines 2 to 8), the distance between the sales subsequences of T_A and other products (expect T_A) is computed to maintain the minimum value of P_A. P_A is the MP between product A and other products, which is stored in the MP set P of all products. When P_A is updated, the corresponding index I_A and code C_A of the product are also recorded, as shown in line 6.

The sets P, I, and C can be used to measure the similarity between the sales time series of different products. According to these sets, several similar pairs can occur in different products. The number of most similar subsequences determines the degree of similarity between the product sales. This process is represented by our algorithm countingSimilarPatterns (CSP), as shown in Algorithm 4.

In this process, the similar patterns must co-occur. If there is no co-occurrence, it is highly likely that these patterns are similar at

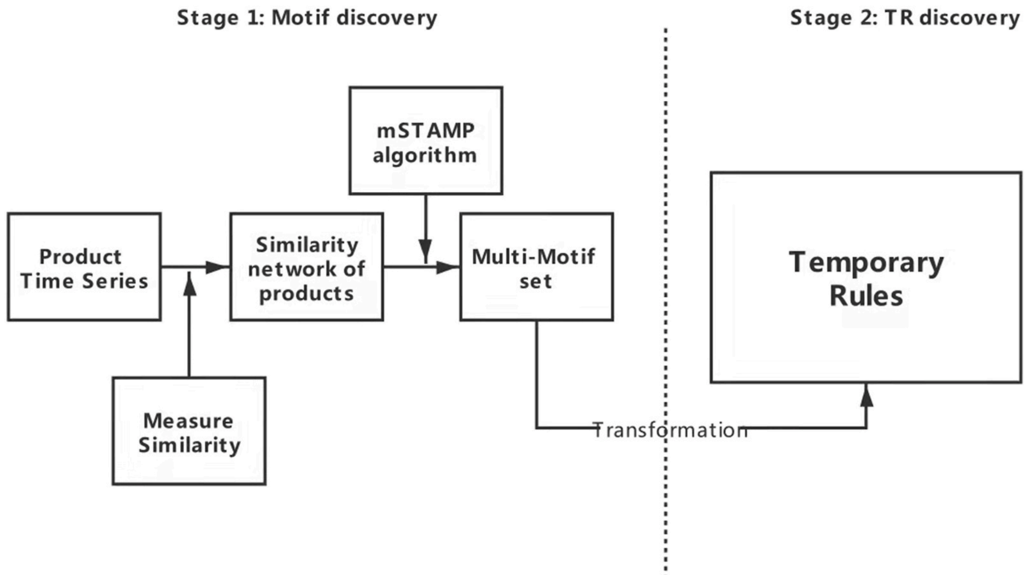

Fig. 4. The flow of TR discovery.

Algorithm 3 calculateProductPatterns(S_T, w)

Input: a set of product sales time series S_T and length of subsequence w.

Output: MP P, MP index I, and MP code C.

```
P ⟵ Initialization(inf), I, C ⟵ Initialization(0)
for each T_A in S_T do
    for each T_B in S_T but T_B is not T_A do
        P_AB, I_AB ⟵ STAMP(T_A, T_B, w)
        if any(P_AB < P_A) then
            P_A, I_A, C_A ⟵ selectMinV(P_AB, I_AB, name(T_B))
        end if
    end for
    return T_A to S_T
end for
Return P, I, C
```

different times. In reality, one product's sale in summer would not affect another product's sale in winter. Therefore, when counting the number of similar patterns, we consider only those occurring at the same time. In line 1 of Algorithm 4, the similarity table *N* is initialized and organized to represent the sales correlation between products. The main loop (lines 2 to 8) counts the codes in MP code *C*. In line 6, the degree of similarity

Fig. 5. Structures of MP, MP index, and MP code.

between the two product sales is calculated, which is the number of subsequences in C_A pointing to another product. Finally, we can obtain the similarity table N between all product sales. The general form of table N is given by Table 1.

Algorithm 4. countingSimilarPatterns(I,C)

A similarity measure between product sales is performed by the algorithm measuringProductSimilarity (MPS), as shown in Algorithm 5, which relies on both the CPP and CSP algorithms.

Algorithm 5. measuringSimilarSimilarity(S_T,w)

3.1.2. *Motif extraction*

To search for motifs quickly according to the similarity table N, we need to divide the products into two groups, namely, strongly correlated sales products and weakly correlated sales products. Within these different product groups, the product sales can be treated as a multivariate time series. The mSTAMP algorithm can retrieve multi-motifs from such a multivariate time series. Note that it is necessary to mine in two groups to find motifs. This is because some products may not be correlated in general but may be similar within a specific time period. The search for motif sets can be divided into three steps.

a) Motif pre-search calculations: Before mining the product data in groups, we need to fit multi-matrix profiles to the multi-time series of the products. This is because the multi-time series for a group of products is not a real multivariate time series. In other words, a single time series may not be strongly related to another time series within such a multi-time series. For example, some product sales need not always be synchronized despite belonging to the strongly correlated sales group. Thus, although the mSTAMP algorithm is efficient and has the ability to select appropriate dimensions, it performs certain unnecessary calculations. Therefore, certain pre-calculations are required to determine the dimensions or modes in which the motifs occur. Appropriate dimension selection helps improve the efficiency of extracting motifs.

b) Motifs in strongly correlated sales products: Multi-motifs occur frequently in strongly correlated products. The MotifsInCorrelated-Products (MCP) algorithm presented in Algorithm 6 is designed to search for such multi-motifs.

In line 1 of Algorithm 6, the set of multi-motifs is initialized. The selectingSimilarGroup function selects the top k similar sales from the product groups. The degree of similarity between product groups is the mean of all product correlations. The mSTAMP algorithm is applied to these groups, which returns the multi-matrix profile (lines 3 to 6). Finally, we retrieve k multi-motifs from the k product groups.

Table 1
Product similarity table.

Product 1 code	Product 2 code	Similarity
22962	22963	198
22963	22962	187
22629	22630	181
22726	22727	169
21122	21124	167
…	…	…

Algorithm 6. MotifsInCorrelatedProducts(ST,w)

c) Motifs in weakly correlated sales products: Compared with strongly correlated products, mining multi-motifs in weakly correlated sales products is more complicated and difficult. In general, the correlations between dissimilar product sales are extremely weak. Although these relationships are usually ignored in traditional association analysis, they are meaningful and valuable to business owners.

We describe the procedure of mining motifs in weakly correlated sales products as MotifsInUncorrelatedProducts (MUP) in Algorithm 7. In general, this algorithm requires the similarity table N and number of multi-motifs k as input, and returns the set of multi-motifs S_m as output.

Algorithm 7. MotifsInUnCorrelatedProducts(ST,w)

In line 1 of Algorithm 7, the set of multi-motifs is initialized. Next, the top k weakly correlated sales product groups are recognized according to the similarity table N in line 2. Note that we have to select a large number of product groups because motifs are rare to find in weakly correlated sales products. Then, the mSTAMP algorithm searches for multi-motifs in each group of product sales (lines 3 to 6). Finally, the algorithm returns the top k similar patterns or multi-motifs (line 7).

We can now conclude the entire process of motif discovery in product sales time series using the algorithm motifDiscovery (MD), as shown in Algorithm 8. This algorithm mines out the set of multi-motifs S_m from the time series set S_T.

Algorithm 8. MotifsDiscovery(S_T,w,k_1,k_2)

3.2. *Temporary rules*

Using the MD algorithm, we can obtain the set of multi-motifs S_m. However, it still needs to be determined whether these motifs are meaningful. More specifically, these patterns may only be similar oc-

---

Algorithm 4 countingSimilarPatterns(I, C)

Input: MP index set I and MP code set C.

Output: Similarity table N of product sales.

```
N ⟵ Initialization()
for each C_A in C do
    Indexs ⟵ length(C_A)
    for each idx in indexs do
        if idx = I_A[idx]
            N_{A,C_A[idx]} ⟵ N_{A,C_A[idx]} + 1 then
        end if
    end for
end for
Return N
```

Algorithm 5 measuringSimilarSimilarity(S_T, w)

Input: a set of product sales time series S_T and length of subsequence w.

Output: Similarity table of product sales.

$[P, I, C, N] \longleftarrow Initialization()$

$P, I, C \longleftarrow calculateProductPatterns(S_T, w)$

$N \longleftarrow countingSimilarPatterns(I, C)$

Return N

Algorithm 6 MotifsInCorrelatedProducts(S_T, w)

Input: Similarity table N of product sales and number k of retrieved multi-motifs.

Output: Set of multi-motifs S_m

$S_m \longleftarrow Initialization()$

$strongGroups \longleftarrow selectingSimilarGroup(N, k)$

for each group in strongGroups **do**

  $item_{motif} \longleftarrow mSTAMP(group, w)$

  $S_m.add(item_{motif})$

end for

Return S_m

Algorithm 7 MotifsInUnCorrelatedProducts(S_T, w)

Input: Similarity table N of product sales and number k of retrieved multi-motifs.

Output: Set of multi-motifs S_m

$S_m \longleftarrow Initialization()$

$weakGroups \longleftarrow selectingDisimilarGroup(N, k)$

for each group in weakGroups **do**

  $item_{motif} \longleftarrow mSTAMP(group, w)$

  $S_m.add(item_{motif})$

end for

Return S_m

casionally and hence are of little help to the corresponding retail enterprises. Therefore, the set of multi-motifs needs to be filtered to obtain meaningful multi-motifs.

At present, the MP algorithm is widely applied in many fields mainly because of its domain agnostic nature. When performing data analysis in a certain field, it does not require professional knowledge of the field to extract the motif. According to the obtained motif, domain experts perform motif interpretation. Similarly, to determine the value of the motifs, professional sales personnel can be contacted directly for interpretation. Another method to determined the value of the motifs is that product sales data can be tracked, the relevant data can be obtained for a longer period of time, and it can also be observed whether the acquired motifs reappear later in time. Both methods can judge if the motifs are valuable and whether they need to be filtered.

Fig. 6(a) shows three product sales time series named T_1, T_2, and T_3, respectively. The multi-motifs that are discovered are highlighted in red. If this relationship is helpful and repeated, we regard these multi-motifs as a TR. In Fig. 6(b), the first two patterns highlighted in red in both T_1 and T_2 are the same as the corresponding multi-motifs in Fig. 6(a), whereas the last two similar patterns appear once the time series become longer than in Fig. 6(a). This means that the multi-motifs repeat twice, and hence, can be considered to be a TR.

After the required filtering, these multi-motifs can be transformed into TRs. In the following, we explain three aspects of TRs with respect to multi-motifs.

3.2.1. *Timeliness*

The timeliness aspect of TRs refers to the fact that TRs function in specific time periods. From Fig. 6(a), we can easily recognize that T_1 and T_2 are not always synchronized, and a multi-motif only appears within a specific time period. In these periods, T_1 and T_2 are correlated, which implies that the customers prefer to purchase T_1 and T_2 together during this time. Thus, it is useful to promote T_1 and T_2 in these time windows than in others.

Algorithm 8 MotifsDiscovery(S_T, w, k_1, k_2)

Input: a set of product sales time series S_T, length of subsequence w, and number of retrieved multi-motifs k_1, k_2.

Output: Set of multi-motifs S_m

$S_m \longleftarrow Initialization()$
$N \longleftarrow measuringProductSimilarity(S_T, w)$
$S'_m \longleftarrow MotifsInCorrelatedProducts(N, k_1)$
$S_m.add(S'_m)$
$S''_m \longleftarrow MotifsInUncorrelatedProducts(N, k_2)$
$S_m.add(S''_m)$
Return S_m

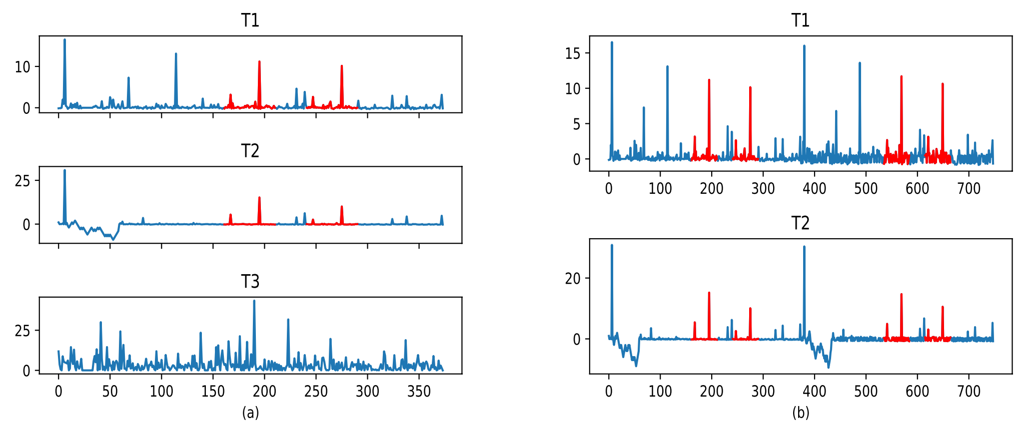

Fig. 6. Mining for TRs. (a) Segment of a multivariate time series composed of single time series T_1, T_2, and T_3. The multi-motifs are highlighted in red. (b) Repeating multi-motifs in longer time series.

3.2.2. Value

TRs can help exhibit the value of products with regard to their consumption. This is because TRs are based on motifs, and motifs are the similar subsequences of a sales time series. Different motifs reflect different features of product sales. Note that products containing high-sales motifs are worthy of promotions. For example, the motif patterns in Fig. 6 show that the sales would experience at least two dramatic jumps, which is more desirable than some of the remaining low-sale motifs.

3.2.3. Prediction

TRs can predict certain aspects of product sales based on the timeliness and quantity of products sold. In Fig. 6(b), the red-highlighted regions represent the motifs as well as the patterns similar to the motifs. These patterns result from motifs and TRs repeating continuously. Therefore, from the timeliness of TRs, if we can determine the frequency with which the motifs appear, then we can predict when the next motif will appear. In addition, motifs show how the sales change with time. Thus, according to the frequency of motifs and the trend of patterns, we can predict the sales at the time when the next motif appears.

4. Experimental evaluation

In this section, we present the settings of the experiments that we carried out to mine motifs. The details of the STAMP and mSTAMP algorithms are already available in the literature (Li et al., 2020; Linardi et al., 2018). The dataset used in this work was provided by UCR, which contains all the transactions from December 01, 2010 to December 09, 2011 for a registered, UK-based, online retail store. The total number of the instances is 541,909.

4.1. Data preprocessing

The online retail dataset has five attributes (i.e., "InvoiceNo", "StockCode", "Description", "Quantity", and "InvoiceDate"), all of which were used in this work. There were three steps involved in the data preprocessing, namely, transaction preprocessing, transaction transformation, and product sales time series preprocessing. The transaction preprocessing was performed to omit errors in the original dataset, whereas the remaining two steps transformed the transactions into a time series.

1) Transaction preprocessing: In this step, we removed the errors from the original transactions. Certain observations contained negative values of the quantity attribute. These data are not meaningful for our research, and hence, we removed these observations as well.
2) Transaction transformation: Transactions need to be transformed into a product sales time series according to the sales Quantity and InvoiceDate for a given Stockcode. First, we determined the span of the time series from the date range given by InvoiceDate. Next, we collected the set of products by scanning the column StockCode of the dataset. Finally, we counted the total product sales during the

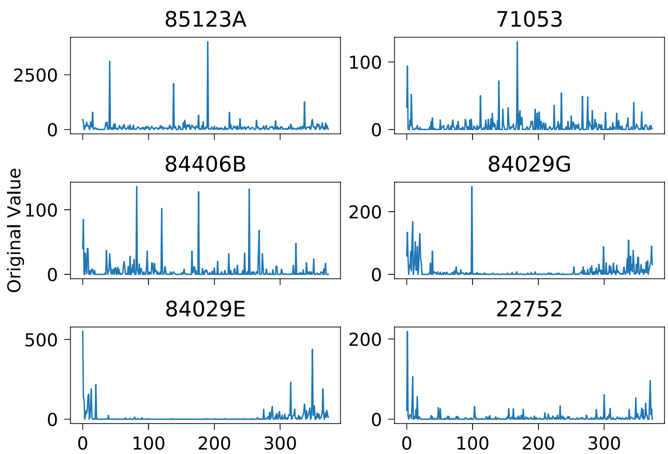

Fig. 7. Sales time series of the first six products in the UCR dataset.

span of the time series by scanning the entire dataset. Fig. 7 shows the sales time series of the first six products in the dataset.

3) Time series preprocessing: Although the product time series provides the sales information of the products, they still need to be transformed and cleaned as per the following steps.
   a) Sales normalization problem: Product sales fluctuate dramatically. Hence, the sales peak values may affect the similarity measure between the subsequences. Therefore, we need a z-score function to normalize each sales time series.
   b) The trivial match problem: Owing to the characteristic nature of the numbers occurring in sales, often a lot of zeros appear causing the trivial match problem. This is because subsequences containing too many zero values can be erroneously recognized as extremely similar pairs. However, this problem can be fixed by following certain steps. First, some product sales have near zero values, which are not worthwhile for mining motifs, and hence, should be deleted. In addition, some subsequences containing too many zero values could easily be regarded as similar patterns, which in turn can lead to the generation of meaningless motifs. To prevent these subsequences from being detected as motifs, we used the random walk theory and replaced those subsequences with random walk sales data. According to the random walk theory, because random walk sales data are independent of each other, the corresponding subsequences would not be detected as being similar to each other. Thus, the trivial match problem can be fixed.

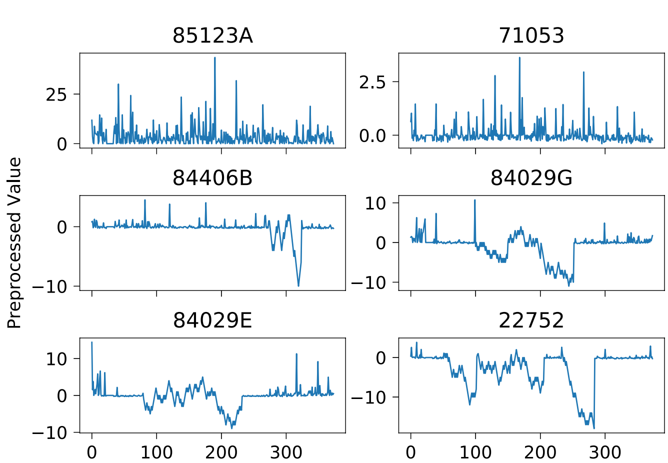

Fig. 8. Sales time series of the first six products in the UCR dataset after data preprocessing.

Following all these steps, the transactions were cleaned and transformed into the correct product sales time series. The sales time series of the first six products were transformed into the ones shown in Fig. 8 after data preprocessing. Note that the length of each time series is 375 days.

4.2. Motif discovery in product sales time series

Motif discovery involves two steps: (i) measuring the similarity between products, and (ii) searching for motif sets in the similarity table. Note that because of the limitations of modern computational technology, we search for motifs in the sales time series of the first 2000 products only.

The MPS algorithm is designed to measure the similarity between the product sales time series. Within the MPS algorithm, the CPS algorithm computes the MP set of the first 2000 products. Then, the CSP algorithm returns the MP set and measures the similarity between the product sale time series by counting the similar subsequences. Finally, we can obtain the similarity table N (see e.g., Table 1).

In Table 1, there are 4553 rows that show the correlations between product sales. The similarity attribute in the table is the number of pairs having the most similar sales subsequences. For example, there are 198 sales subsequences of product 22963 that are most similar to that of product 22962. We also observe that these correlations have direction, which is evident from the different similarity values of the first two rows in Table 1.

The similarity table can be visualized as a similarity network. We can then choose just a part of this network to visualize and study the correlations between different product sales, as shown in Fig. 9.

In Fig. 9, each product represents a node of the similarity network. The size of the node represents the degree, whereas the edge represents the sales correlation between the source and target products. In addition, the width of the edge reflects the degree of sales correlation. This figure truly helps to understand the correlation between products and has several benefits for future work. Although groups of product sales can be treated as multivariate time series, they are not actual multivariate time series. This essentially means that not all the underlying single time series are related. To properly extract multi-motifs from the dataset, we need to perform certain tests and discover the ideal model for using multi-matrix profiles in product sales.

We select some objective products (such as 79051A, 22613, 22679, and 22680) that have the largest degrees in the similarity table N. From the similarity network, we can extract a product sales group related to the objective product. This group can be considered to be a multivariate time series, which can be inputted to the mSTAMP algorithm to generate a set of multi-matrix profiles. For example, we visualize the multi-matrix profiles of product 79051A in Fig. 10. Each curve shows the multi-matrix profile corresponding to a multivariate sales time series having different dimensions. For example, 2-d indicates that the multivariate time series is composed of two product sales time series.

Table 2 presents the minimum value of the multi-matrix profiles of the four products in each subsequence. We see that there is an overwhelming advantage of the dimension 2-d. The minimum distances of all the subsequences belong to 2-d. Therefore, in the sales correlation analysis, the most useful sales relationship in the online retail dataset usually occurs between two products. Thus, in the remaining work, we only searched for multi-motifs in 2-d.

In the real world, multi-motifs are more likely to exist in strongly correlated sales product groups. The method described above is particularly relevant for mining multi-motifs in strongly correlated groups, which can be extracted from the similarity network shown in Fig. 9.

Now that we have established that dimension 2-d is most suitable for multi-motif discovery in product sales, we select products that have the top 100 strongly correlated sales, and combine them to create a multivariate time series. The mSTAMP algorithm then returns the multi-motifs in these correlated products. We present these motifs in Table 3.

In Table 3, the product codes and names are provided, as well as the

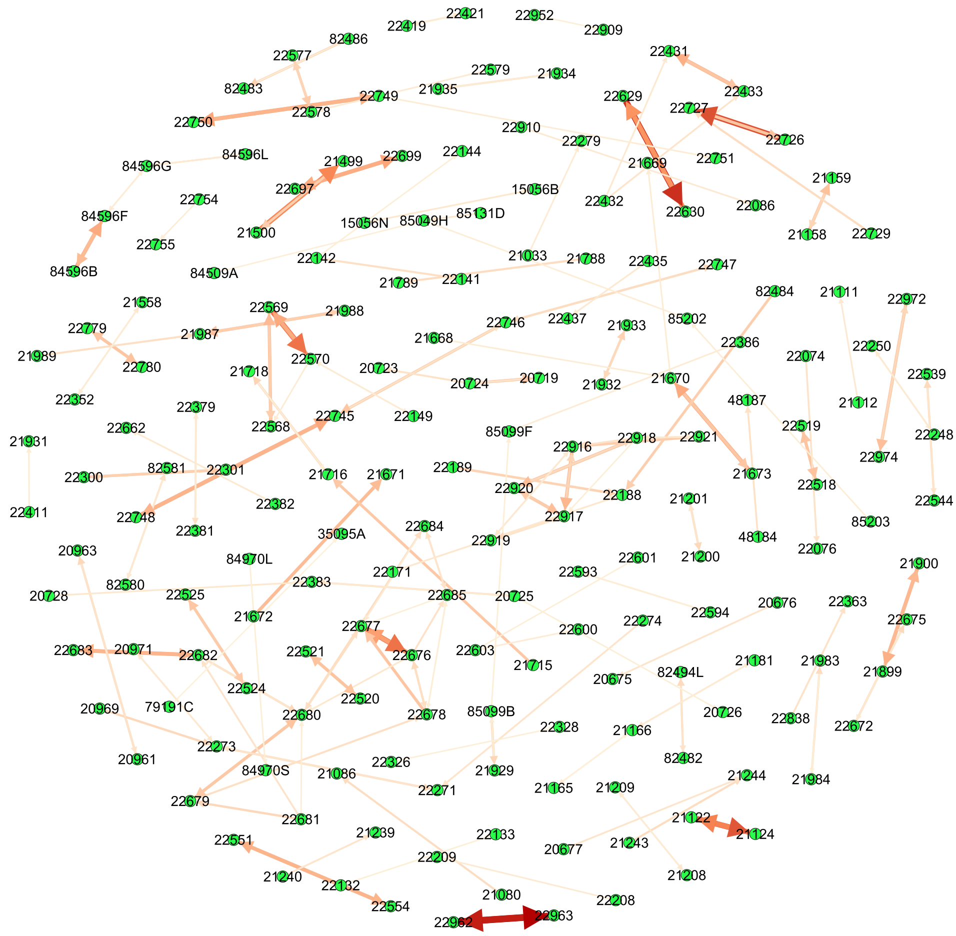

Fig. 9. A part of the product sales similarity network.

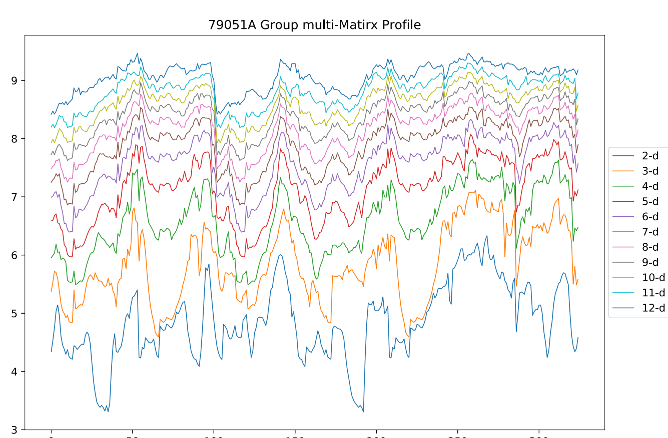

Fig. 10. Multi-matrix profiles of product 79051A.

Table 3
Motifs in strongly correlated sales products.

Product 1 code	Product 1 name	Product 2 code	Product 2 name	Motif Period
22962	JAM JAR WITH PINK LID SPACEBOY	22963	JAM JAR WITH PINK LID GIRL	$70^{th} to 100^{th}$ $227^{th} to 267^{th}$ $176^{th} to 206^{th}$
22629	LUNCH BOX ALARM CLOCK	22630	LUNCH BOX ALARM CLOCK	$262^{th} to 292^{th}$ $96^{th} to 126^{th}$
22726	BAKELIKE GREEN SET/10 PINK	22727	BAKELIKE RED SET/10 BLUE	$235^{th} to 265^{th}$ $56^{th} to 86^{th}$
21122	CANDLES	21124	CANDLES	$174^{th} to 204^{th}$
…	…	…	…	…

Table 2
Minimum value of multi-matrix profile.

Dimension	2-d	3-d	…	x-d
Count	1300	0	0	0

period of motif occurrence. To better understand these motifs, we consider products 22726 and 22727 as an example and visualize their motif in Fig. 11.

Multi-motifs in weakly correlated sales products are rare. This type of relationships tends to be hidden under a massive amount of retail data. Therefore, the MUP algorithm intends to mine multi-motifs in massive groups of weakly correlated sales products. We consider the products that have the 50 lowest correlations to be the weakly correlated sales products. Each weakly corrected sales product can form a group according to the network in Fig. 9. Some multi-motifs in weakly correlated sales products are not meaningful, and hence, we have to filter them out. Table 4 shows some of the meaningful motifs in weakly correlated sales products.

For an intuitive visualization of the weakly correlated multi-motifs,

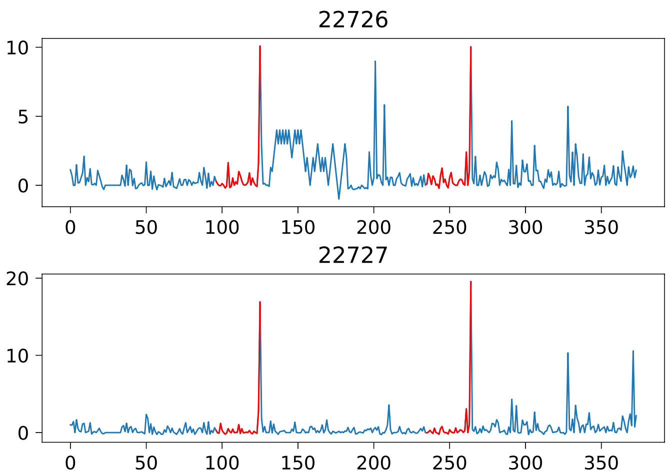

Fig. 11. Multi-motifs (highlighted in red) in two strongly correlated sales products 22726 and 22727.

Table 4
Motifs in weakly correlated sales products.

Product 1 code	Product 1 name	Product 2 code	Product 2 name	Motif Period
22870	NUMBER TILE COTTAGE GARDEN	21253	SET OF PICTURE STICKERS	190^{th}to210^{th} 241^{th}to271^{th}
22797	CHEST OF DRAWERS GINGHAM HEART	90087	CRYSTAL SEA HORSE PHONE CHARM	89^{th}to119^{th} 344^{th}to374^{th}
84510C	SET OF 4 POLKADOT COASTERS	22761	CHEST 7 DRAWER MA CAMPAGNE	37^{th}to67^{th} 292^{th}to322^{th}
22428	ENAMEL FIRE BUCKET CREAM	21913	SEASIDE JIGSAW PUZZLES	191^{th}to221^{th} 344^{th}to374^{th}
…	…	…	…	…

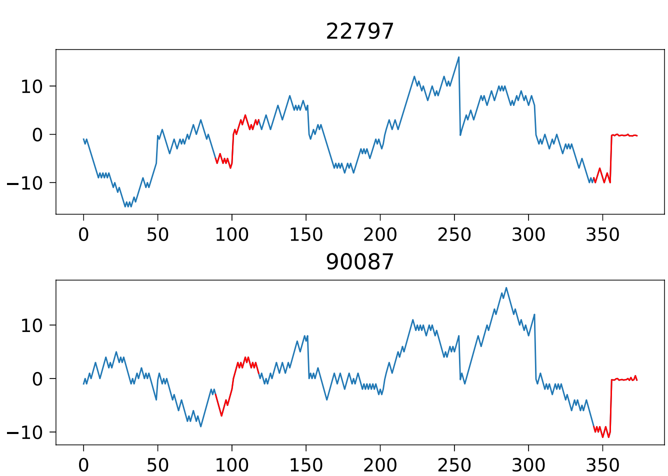

Fig. 12. Multi-motifs (highlighted in red) in two weakly corrected sales products 22797 and 90087.

we present the sales data of products 22797 (i.e., “chest of drawers gingham heart”) and 90087 (i.e., “crystal sea horse phone charm”) in Fig. 12.

Note that motif discovery depends on the number of motifs that retailers want to retrieve. Therefore, we selected the top 100 strongly correlated sales motifs and 5 weakly correlated sales motifs, as shown in Table 5. The next step is to transform this motif set into TRs.

4.3. *TR discovery results*

Now that we have obtained several multi-motifs, as shown in Table 5, we consider the products 22726 and 22727 from Fig. 11 as an example to discover TRs.

4.3.1. *Verification of multi-motifs*

We need to verify whether there are repeated patterns that are similar to the multi-motifs. We searched for such similar patterns by looking at long time-series datasets. The repeated patterns are recognized by comparing them with existing similar patterns. In our simulations, we selected the multi-motifs from products 22726 and 22727 and verified whether they have repeated patterns. In addition, we simulated the sales data for one year and added it to the original dataset. Next, we calculated the distance profile and selected the most similar patterns therein. We visualize the results obtained in Fig. 13. Four similar subsequences were found to occur in two years. If these were real time-series datasets, we could presume that the motifs would repeat because of the existence of several similar patterns.

4.3.2. *Transforming motifs into TRs*

We chose the sales patterns of products with StockCode 22726 and 22727, which represent “alarm clock bakelike green” and “alarm clock bakelike red”, respectively. Their multi-motifs are presented in Fig. 13. Keeping the characteristics of multi-motifs in mind, we generate TRs having the following aspects.

a) Timeliness aspect: A pair of multi-motifs in the time series for the green and red clocks appeared between the 96th and 126th day, and between the 235th and 265th day. In these two periods, the sales of the two clocks were correlated. Thus, it would be advisable for retailers to promote these clocks together during these periods.
b) Quantity aspect: In essence, multi-motifs are the similar sales subsequences between two products. The pattern shows that the sales of the two clocks jumps to a high level at the same time. This implies that in this specific period, customers begin to purchase a large number of both the clocks. Thus, research-based promotions can improve the profits dramatically.
c) Prediction aspect: If we can understand the frequency of occurrence of the multi-motifs in terms of the sales quantity of the motif, it is easy to predict the sales trend at the time the next multi-motif appears. For example, from Fig. 13, we can predict from the motif in the second year that the sales of both the clocks will jump to a high level on the 440th day.

4.3.3. *Influence of motif length*

TRs can be obtained from multi-motifs and are extremely helpful for businesses. However, in practice, when mining for multi-motifs or TRs, we need to determine the length of a motif, as it affects the accuracy of TRs. Therefore, we conducted experiments to determine the effect of the length of a motif on the acquisition of multi-motifs or TRs.

In our experiments, we first determined the size of the time window needed to discover the TRs. Generally, product sales are periodic, and hence, it is reasonable to use a week, two weeks, a month, or 2 months as the size of the time window. Next, we selected the first 200 products, and carried out our analysis by implementing different window sizes.

Table 5
Selected motifs from strongly and weakly correlated sales products.

	Product 1 code	Product 1 name	Product 2 code	Product 2 name	Motif Period
Strongly correlated sales Products	22870	NUMBER TILE COTTAGE GARDEN	21253	SET OF PICTURE STICKERS	$190^{th}to210^{th}$ $241^{th}to271^{th}$
	22797	CHEST OF DRAWERS GINGHAM HEART	90087	CRYSTAL SEA HORSE PHONE CHARM	$89^{th}to119^{th}$ $344^{th}to374^{th}$
Weakly correlated sales Products	84510C	SET OF 4 POLKADOT COASTERS	22761	CHEST 7 DRAWER MA CAMPAGNE	$37^{th}to67^{th}$ $292^{th}to322^{th}$
	22428	ENAMEL FIRE BUCKET CREAM	21913	SEASIDE JIGSAW PUZZLES	$191^{th}to221^{th}$ $344^{th}to374^{th}$
	…	…	…	…	…

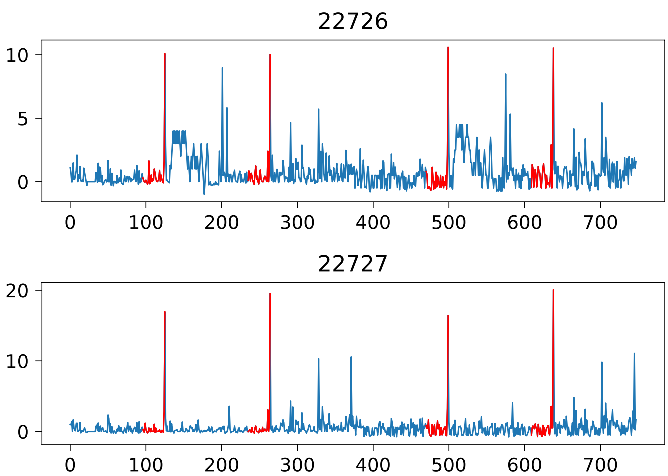

Fig. 13. Verification of multi-motif generation using simulations.

1) Similarity of products for different time windows: Here, we show the influence of different motif lengths on the similarity network. We can see from Fig. 14 that the motifs in the different panels have different lengths of 7, 14, 30, and 60 days. Each panel shows the product similarity distribution, where the x- and y-axes represent the similarity and the distribution of products, respectively. More specifically, we used the auto mode to cut out the bins of the histogram and set the overall density to 1. We conclude two key points about the influence of the motif length on the similarity of networks, as discussed below.
   a) *Longer the motif length, more similar are the products.* As we can see from Fig. 14, as the length of the motif increases, highly similar products start emerging. For the "products in similarity distribution_200_7", the maximum similarity is approximately 16 for a window size of 7, whereas the maximum similarity is approximately 40, 150 and 200 for window sizes 14, 30 and 60, respectively. It is actually reasonable that a short motif length corresponds to a low degree of similarity between the products in the network. This is because shorter patterns are more easily matched with other patterns compared to longer patterns. Moreover, it is more likely for shorter patterns to have highly similar counterparts in a large number of products. Therefore, it is possible for all the subsequences of a product to point to several other products such that the overall degree of similarity is reduced.
   b) *Most of the products are dissimilar.* From all the cases in Fig. 14, we observe that the product distribution peaks in very low similarity. Thus, we can easily recognize the highly similar products that are convenient for detecting TRs using multi-motifs.
2) Multi-motifs for different time windows: Eventually, we shall set the hyperparameter k to retrieve k motifs only. However, for the present discussion, we set k according to the similarity distribution of the products, so as to mine as many useful motifs as possible. Therefore, we mainly focus on the highly correlated products.

According to the similarity distribution of products in Fig. 14, we observe that highly correlated products are rare. Therefore, we chose k to be a small number, say 4. We then retrieved the first 4 multi-motifs for different time windows. We also filtered out the cases with window sizes 7 and 14 days.

We present the multi-motifs of the top 4 pairs of similar products for a window size of 30 days in Fig. 15. From this figure we can easily determine the accuracy of the patterns as well as the most similar trends. For example, products 22726 and 22727 have a boost in sales at the same time. The same is also true for products 22962 and 22963. This co-increase in the sales is repeated again at a later time. When the window size is set to 60 days, we again observe a similar trend in the products, as shown in Fig. 16.

Thus, from our results we conclude that window sizes of 30 and 60 days are most appropriate for extracting motifs when the total time period is 375 days. This choice allows us to find more helpful patterns or motifs.

4.3.4. Comparing TRs with traditional association rules

In this section, we compare the findings of the present study with that of Fournier-Viger et al. (2019) for a more comprehensive evaluation of our work. Fournier-Viger et al. (2019) analyzed three kinds of datasets: local high utility itemsets (LHUI), peak high utility itemsets (PHUI) and non-redundant peak high utility itemsets (NPHUIs). The aim of the present work, however, was not to test datasets but to perform a thorough comparison between different data mining algorithms, and present the results of our analysis.

Although our ideas are similar to that of Fournier-Viger et al. (2019), we employed very different methods. We considered all the core ideas that represent the relations between commodities over time. One of the major differences between the two works is that while we aimed to discover repeated patterns or TRs, Fournier-Viger et al. (2019) focused more on recognizing highly correlated products during a specific time period in itemsets. In our work, we not only recognized repeated patterns but also transformed them into TRs, which are extremely valuable

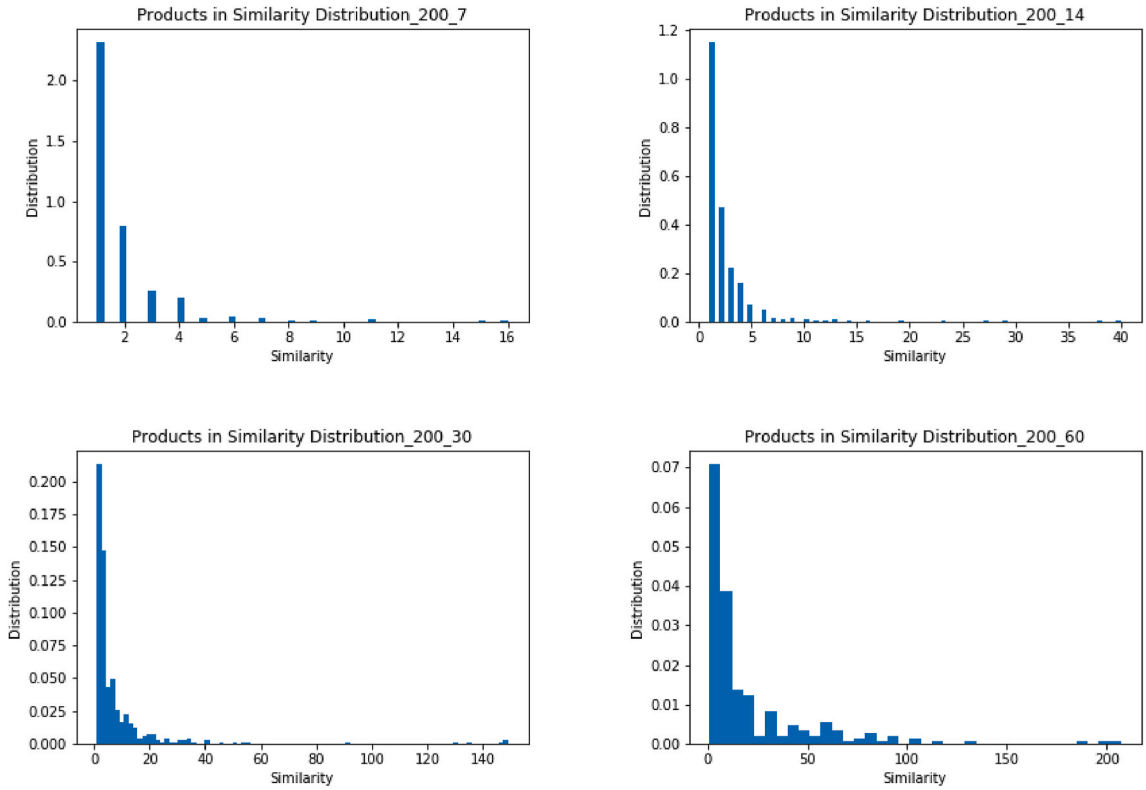

Fig. 14. The similarity network of products for different time windows.

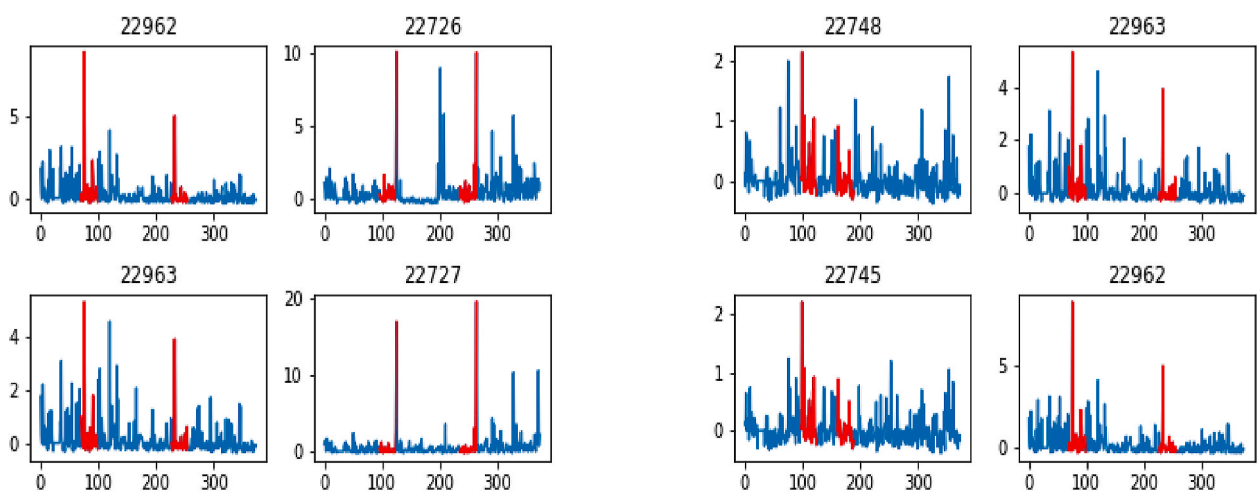

Fig. 15. Multi-motifs of the top 4 pairs of similar products for a window size of 30 days.

to retail businesses. More importantly, we have presented a valuable representation of our algorithm, and clearly outlined the advantages of our findings.

1) Hyperparameters: Compared to Fournier-Viger et al. (2019), we have fewer effective hyperparameters. Although both works need to set a "window size", Fournier-Viger et al. (2019) had to set an extra parameter, called "minutil", as the threshold, which dramatically affects their results. Minutil needs to be tuned several times manually, which can be avoided in our work. Hence, we selected the top *k* products and motifs to avoid specifying a threshold.
2) Weak correlations: The main goal of Fournier-Viger et al. (2019) is to mine the highly correlated products using parameters "minutils" and "minlength" in their algorithm, without invoking the concept of similarity. On the contrary, our work introduced the concept of similarity to determine which products to group together to explore the correlations between them more efficiently. The dissimilar products discovered by our algorithm are equally valuable, as they are essential for mining local high utility itemsets. We note that some low utility products that are valuable to retail businesses can be easily ignored by the LHUI algorithm in Fournier-Viger et al. (2019) in absence of the similarity measurement performed in our work.

In the field of correlation analysis, traditional association rules such as the Apriori algorithm are classical and powerful. Thus, to better understand association rules in general, we compared traditional association rules and the TRs proposed in this work.

For a fair comparison, we applied the Apriori algorithm to the same

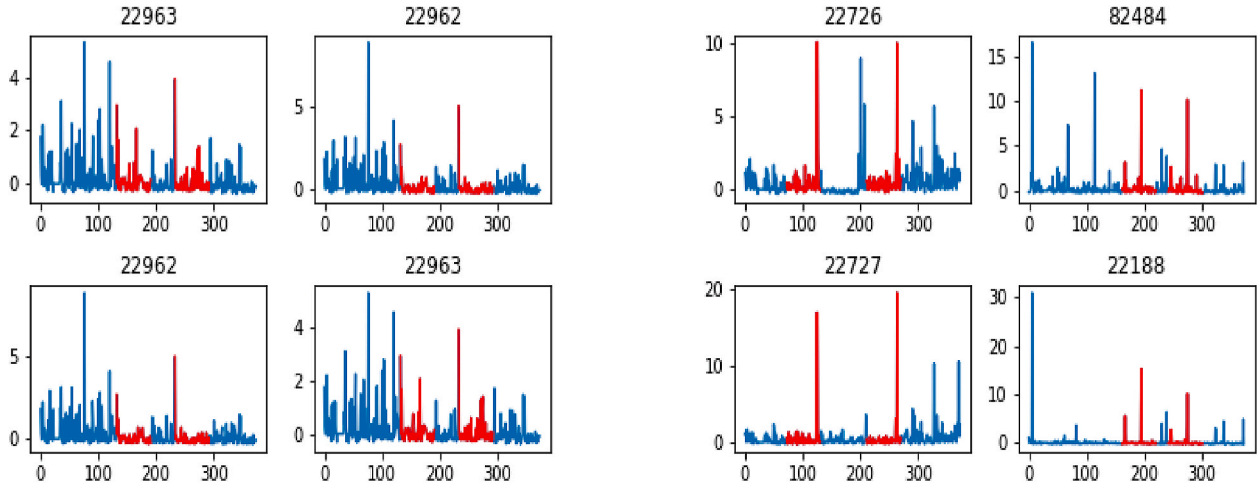

Fig. 16. Multi-motifs of the top 4 pairs of similar products for a window size of 60 days.

Table 6
Strong association rules.

Rules	Support	Confidence
[21136]←[84879],[84879]← [21136] [47590*B*]←[47590*A*],[22727]← [22729]	0.01	0.01
[23202]←[23203],[23203]← [23202] [21928]←[85099*B*],[22727]← [22726]	0.02	0.01
[23202]←[23203],[21928]← [85099*B*] [85099*F*]←[85099*B*],[22727]← [22726]	0.02	0.5

online retail dataset used in this work to mine the association rules. In Table 6, we present the strong association rules for different support and confidence.

TRs are based on motifs. Compared with the strong association rules in Table 6, TRs can reflect more information about time. In addition, motifs are the similar subsequences of product sales, which provide information regarding the change in quantity of a product during a specific period. We list the differences between the two kind of rules in Table 7.

5. Conclusion

We applied the MP to product sales time series to find the local correlation between products and TRs. TRs are different from traditional association rules. They essentially reflect the sales relationship between products within a specific time period. Based on the sales, TRs show the sales correlation between products in both time and quantity. These temporary association rules are not a sliver bullet but are nevertheless valuable to the retail industry. In this study, we divided the products in the dataset into different groups to mine the useful motifs. We discovered that some products that are essentially similar but have different styles or colors are highly likely to be correlated during their entire sales period. We also mined several meaningful patterns in weakly corrected (or uncorrelated) product groups.

TRs can help promote retail industries by improving cross-marketing, supply, and evaluation of products. TRs provide the retailers with information about which products are more correlated in a specific period. As the purchasing habits of customers keep changing, TRs also change accordingly. Using TRs, business leaders can effectively recommend suitable products to customers at a suitable time, which can contribute to significant profit. Moreover, compared with traditional

Table 7
Comparison between strong association rules and TRs.

Aspect	Strong Association Rules	Temporary Rules
Probability	Confidence and support	No
Prediction	Probability of products purchased	Time and Quantity
Timeliness	No	Yes
Quantity	No	Yes

association rules, TRs are more flexible and effective.

The sales trend of a product that has a high degree in the similarity table N can be captured by TRs. Retailers can predict sales quantity by researching the frequency of TRs or similar patterns. By observing the regular appearance of TRs, supply chain enterprises can prepare in advance by storing more products even before the sales reach a peak.

In addition, TRs can help businesses with product promotions. Usually only popular products are selected for promotion to increase the profit. On the contrary, TRs can provide additional information about the sales trends of the products so that the products can be better evaluated, and the most suitable product can be selected for promotion. We also note that TRs are not meant to replace traditional association rules but are supposed to supplement them. TRs still have some shortcomings, such as the lack of a proper numerical standard. In the future, we plan to update and complete our research on TRs.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jretconser.2020.102431.

Funding

This work has been supported by the National Natural Science Foundation of China (71771094,61300139) and the Natural Science Foundation of Fujian (2019J01067), and Ministry of Science & Technology, Taiwan (108-2511-H-003-034-MY2 & 109-2511-H-003-049-MY3).

References

Agrawal, R., Shafer, J.C., 1996. Parallel mining of association rules. IEEE Trans. Knowl. Data Eng. 8, 962–969.

Agrawal, R., Srikant, R., et al., 1994. Fast algorithms for mining association rules. In: Proc. 20th Int. Conf. Very Large Data Bases, vol. 1215. VLDB, pp. 487–499.
Ahmed, C.F., Tanbeer, S.K., Jeong, B.-S., Choi, H.-J., 2012a. Interactive mining of high utility patterns over data streams. Expert Syst. Appl. 39, 11979–11991.
Ahmed, C.F., Tanbeer, S.K., Jeong, B.-S., Lee, Y.-K., Choi, H.-J., 2012b. Single-pass incremental and interactive mining for weighted frequent patterns. Expert Syst. Appl. 39, 7976–7994.
Amphawan, K., Soulas, J., Lenca, P., 2015. Mining top-k regular episodes from sensor streams. Procedia Comput. Sci. 69, 76–85.
Behera, R.K., Gunasekaran, A., Gupta, S., Kamboj, S., Bala, P.K., 2020. Personalized digital marketing recommender engine. J. Retailing Consum. Serv. 53, 101799.
Berry, L.L., Bolton, R.N., Bridges, C.H., Meyer, J., Parasuraman, A., Seiders, K., 2010. Opportunities for innovation in the delivery of interactive retail services. J. Interact. Market. 24, 155–167.
Braun, P., Cameron, J.J., Cuzzocrea, A., Jiang, F., Leung, C.K., 2014. Effectively and efficiently mining frequent patterns from dense graph streams on disk. Procedia Comput. Sci. 35, 338–347.
Cagliero, L., Cerquitelli, T., Chiusano, S., Garza, P., Attanasio, A., 2018. Characterizing unpredictable patterns in wireless sensor network data. Inf. Sci. 467, 149–162.
Cao, L., Li, L., 2015. The impact of cross-channel integration on retailers' sales growth. J. Retailing 91, 198–216.
Chang, J.H., Lee, W.S., 2006. Finding recently frequent itemsets adaptively over online transactional data streams. Inf. Syst. 31, 849–869.
Cheng, X., Su, S., Xu, S., Tang, P., Li, Z., 2015. Differentially private maximal frequent sequence mining. Comput. Secur. 55, 175–192.
Cuzzocrea, A., Han, Z., Jiang, F., Leung, C.K., Zhang, H., 2015. Edge-based mining of frequent subgraphs from graph streams. Procedia Comput. Sci. 60, 573–582.
Esling, P., Agon, C., 2012. Time-series data mining. ACM Comput. Surv. 45, 1–34.
Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., 1996. From data mining to knowledge discovery in databases. AI Mag. 17, 37–37.
Fournier, V.P., Wu, C., Tseng, V., 2015. Mining partially-ordered sequential rules common to multiple sequences. IEEE Trans. Knowl. Data Eng. 27, 2203–2216.
Fournier-Viger, P., Zhang, Y., Chun-Wei Lin, J., Fujita, H., Koh, Y.S., 2019. Mining local and peak high utility itemsets. Inf. Sci. 481, 344–367.
Fu, X., Zhang, X., Qiao, Z., Li, G., 2019. Estimating the failure probability in an integrated energy system considering correlations among failure patterns. Energy 178, 656–666.
Funde, N.A., Dhabu, M.M., Paramasivam, A., Deshpande, P.S., 2019. Motif-based association rule mining and clustering technique for determining energy usage patterns for smart meter data. Sustain. Cities Soc. 46, 101415.
Gong, H.-Y., Zhang, Y.-Y., Liang, P.-J., Zhang, P.-M., 2010. Neural coding properties based on spike timing and pattern correlation of retinal ganglion cells. Cognit. Neurodyn. 4, 337–346.
Halkidi, M., 2000. Quality assessment and uncertainty handling in data mining process. In: EDBT PhD Workshop, pp. 1–4.
Holzer, P.S., 2020. The effect of time-varying factors on promotional activity in the German milk market. J. Retailing Consum. Serv. 55, 102090.
Kim, H., Choi, D.-W., 2020. Recency-based sequential pattern mining in multiple event sequences. Data Min. Knowl. Discov. https://doi.org/10.1007/s10618-020-00715-7.
Li, H., 2019. Multivariate time series clustering based on common principal component analysis. Neurocomputing 349, 239–247.
Li, H., 2021. Time works well: dynamic time warping based on time weighting for time series data mining. Inf. Sci. 547, 592–608.
Li, H., Wei, M., 2020. Fuzzy clustering based on feature weights for multivariate time series. Knowl. Base Syst. 197, 105907.
Li, H., Wu, Y.J., Chen, Y., 2020. Time is money: dynamic-model-based time series data-mining for correlation analysis of commodity sales. J. Comput. Appl. Math. 370, 112659.
Li, H., Zhang, N., Zhu, J., Wang, Y., Cao, H., 2008. Dsm-fi: an efficient algorithm for mining frequent itemsets in data streams. Knowl. Inf. Syst. 17, 79–97.
Li, H., Zhang, N., Zhu, J., Wang, Y., Cao, H., 2018. Probabilistic frequent itemset mining over uncertain data streams. Expert Syst. Appl. 112, 274–287.
Linardi, M., Zhu, Y., Palpanas, T., Keogh, E., 2018. Matrix profile x: valmod-scalable discovery of variable-length motifs in data series. In: Proceedings of the 2018 International Conference on Management of Data, pp. 1053–1066.
Liu, Q., Ghosh, S., Li, J., Wong, L., Ramamohanarao, K., 2018. Discovering pan-correlation patterns from time course data sets by efficient mining algorithms. Computing 100, 421–437.
Manku, G.S., Motwani, R., 2002. Approximate frequency counts over data streams. In: Proceedings of the 28th International Conference on Very Large DataBases, pp. 346–375.
Moodley, R., Chiclana, F., Caraffini, F., Carter, J., 2020. A product-centric data mining algorithm for targeted promotions. J. Retailing Consum. Serv. 54, 101940.
Mueen, A., Chavoshi, N., 2015. Enumeration of time series motifs of all lengths. Knowl. Inf. Syst. 45, 105–132.
Mueen, A., Keogh, E., Bigdely-Shamlo, N., 2009a. Finding time series motifs in disk-resident data. In: 2009 Ninth IEEE International Conference on Data Mining, pp. 367–376.
Mueen, A., Keogh, E., Zhu, Q., Cash, S., Westover, B., 2009b. Exact discovery of time series motifs. In: Proceedings of the 2009 SIAM International Conference on Data Mining, pp. 473–484.
Nai, K., Xiao, D., Li, Z., Jiang, S., Gu, Y., 2019. Multi-pattern correlation tracking. Knowl. Base Syst. 181, 104789.
Neslin, S.A., Shankar, V., 2009. Key issues in multichannel customer management: current knowledge and future directions. J. Interact. Market. 23, 70–81.
Pujari, S., Mane, R., Ghorpade, V., 2017. Generation of constraint based sequential rules with trulegrowth algorithm. In: International Conference on Computing Communication, Communication, Control and Automation (ICCUBEA).
Ren, H., Yuan, Q., Semba, S., Weng, T., Gu, C., Yang, H., 2020. Pattern interdependent network of cross-correlation in multivariate time series. Phys. Lett. 384, 126781.
Sagare, S., Shirgave, S., Kodavade, D., 2020. A system for predictive data analytics using sequential rule mining. Int. J. Software Innovat. 8, 96–112.
Samee, M.A.H., Bruneau, B.G., Pollard, K.S., 2019. A de novo shape motif discovery algorithm reveals preferences of transcription factors for dna shape beyond sequence motifs. Cell Syst. 8, 27–42.
See-To, E.W., Ngai, E.W., 2018. Customer reviews for demand distribution and sales nowcasting: a big data approach. Ann. Oper. Res. 270, 415–431.
Shakibay Senobari, N., Funning, G., Zimmerman, Z., Zhu, Y., Keogh, E., 2018. Using the similarity matrix profile to investigate foreshock behavior of the 2004 parkfield earthquake. In: AGU Fall Meeting Abstracts, 1–1.
Shen, C., 2019. The influence of a scaling exponent on dcca: a spatial cross-correlation pattern of precipitation records over eastern China. Phys. Stat. Mech. Appl. 516, 579–590.
Shin, S.J., Lee, D.S., Lee, W.S., 2014. Cp-tree: an adaptive synopsis structure for compressing frequent itemsets over online data streams. Inf. Sci. 278, 559–576.
Singh, H., Kaur, M., Kaur, P., 2017. Web page recommendation system based on partially ordered sequential rules. J. Intell. Fuzzy Syst. 32, 3009–3015.
Yeh, C.-C.M., Kavantzas, N., Keogh, E., 2017. Matrix profile vi: meaningful multidimensional motif discovery. In: 2017 IEEE International Conference on Data Mining (ICDM), pp. 565–574.
Yeh, C.-C.M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H.A., Silva, D.F., Mueen, A., Keogh, E., 2016. Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 1317–1322.
Zeng, Y., Yin, S., Liu, J., Zhang, M., 2015. Research of improved fp-growth algorithm in association rules mining. Sci. Program. 15, 1–6.
Zhang, J., Farris, P.W., Irvin, J.W., Kushwaha, T., Steenburgh, T.J., Weitz, B.A., 2010. Crafting integrated multichannel retailing strategies. J. Interact. Market. 24, 168–180.
Zhu, Y., 2018. The Matrix Profile: Scalable Algorithms and New Primitives for Time Series Data Mining. eScholarship, University of California.
Zhu, Y., Zimmerman, Z., Senobari, N.S., Yeh, C.-C.M., Funning, G., Mueen, A., Brisk, P., Keogh, E., 2016. Matrix profile ii: exploiting a novel algorithm and gpus to break the one hundred million barrier for time series motifs and joins. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 739–748.