Personalized Market Basket Prediction with Temporal Annotated Recurring Sequences

Riccardo Guidotti, Giulio Rossetti, Luca Pappalardo, Fosca Giannotti and Dino Pedreschi

Abstract—Nowadays, a hot challenge for supermarket chains is to offer personalized services to their customers. Market basket prediction, i.e., supplying the customer a shopping list for the next purchase according to her current needs, is one of these services. Current approaches are not capable of capturing at the same time the different factors influencing the customer’s decision process: co-occurrence, sequentuality, periodicity and recurrency of the purchased items. To this aim, we define a pattern Temporal Annotated Recurring Sequence (TARS) able to capture simultaneously and adaptively all these factors. We define the method to extract TARS and develop a predictor for next basket named TBP (TARS Based Predictor) that, on top of TARS, is able to understand the level of the customer’s stocks and recommend the set of most necessary items. By adopting the TBP the supermarket chains could crop tailored suggestions for each individual customer which in turn could effectively speed up their shopping sessions. A deep experimentation shows that TARS are able to explain the customer purchase behavior, and that TBP outperforms the state-of-the-art competitors.

Index Terms—Next Basket Prediction, Temporal Recurring Sequences, User-Centric Model, Market Basket Analysis, Data Mining, Interpretable Model.

1 INTRODUCTION

Detecting purchase habits and their evolution in time is a crucial challenge for effective marketing policies and engagement strategies. In this context, one of the most promising facilities retail markets can offer to their customers is basket prediction, i.e., the automated forecasting of the next basket that a customer will purchase. An effective basket recommender can act as a shopping list reminder suggesting the items that the customer could probably need.

A successful realization of this application requires an in-depth knowledge of an individual’s shopping behavior [1]. The purchasing patterns of individuals evolve in time and can experience changes due to both environmental reasons, like seasonality of products or retail policies, and personal reasons, like diet changes or shift in personal preferences. Thus, a satisfactory solution to basket prediction must be adaptive to the evolution of a customer’s behavior, the recurrence of her purchase patterns, and their periodic changes.

We propose the Temporal Annotated Recurring Sequences (TARS), adaptive patterns which model an individual’s purchasing behavior by four main characteristics. First, TARS consider the co-occurrence: a customer systematically purchases a set of items together. Secondly, TARS model the sequentuality of purchases, i.e., the fact that a customer systematically purchases a set of items after another one. Third, TARS consider periodicity: a customer can systematically make a sequential purchase only in specific periods of the year, because of environmental factors or personal reasons. Fourth, TARS consider the recurrency of a sequential purchase during each period, i.e., how frequently that sequential purchase appears during a customer’s period of the year. Modeling these four aspects – co-occurrence, sequentuality, periodicity and recurrency – is fundamental to detect an individual’s shopping behavior and its evolution in time. On one hand, future needs depend on the needs already satisfied: what a customer will purchase depends on what she already purchased. On the other hand, the needs of a customer depend on her specific habits, i.e., recurring purchases she makes over and over. Far from being static, shopping habits are affected by both endogenous and personal factors [2], [3], [4]. For this reason, periodicity is a crucial characteristic of an adaptive model for basket prediction.

We exploit the TARS to construct a parameter-free TARS Based Predictor (TBP) which solves the basket prediction problem and provides a basket recommendation as a list of items to be reminded in the next purchase. We demonstrate the effectiveness of our approach by extracting the TARS for thousands of customers in three large-scale real-world datasets. One of the main properties of TARS is their interpretability [5], [6], which allows retail chains to gain useful insights about the customers’ purchasing patterns. We show that TARS can be used to infer important characteristics of products, like seasonality and inter-purchase times, which can be easily interpreted by both a simple mathematical notation and a visual representation. Then, we compare TBP with a repertoire of state-of-the-art methods and show that: (i) TBP outperforms existing methods, (ii) TBP can predict up to the next 20 baskets, (iii) the quality of TBP’s predictions stabilizes after about 36 weeks. TARS and TBP are user-centric approaches: given a customer, they only use the customer’s individual data to predict her future baskets [7], [8], [9], [10]. This aspect eases the customers’ personal data management and allows for developing tailored recommenders that can run on personal mobile devices [11], [12].

In summary, our contributions are the following: (i) we introduce TARS, a parameter-free algorithm based on transactional data (Section 4); (ii) we develop TBP, a predictor based on TARS which solves the basket prediction problem to produce a shopping list reminder (Section 5); (iii) we
extract TARS from large-scale real-world datasets and show that they are easily interpretable (Section 6); (iv) we characterize TBP and compare it with state-of-the-art methods on real datasets (Section 6). The rest of the paper is organized as follows. Section 2 reviews existing approaches and Section 3 formalizes the problem. Finally, Section 7 concludes the paper suggesting future research directions.

2 RELATED WORK

In this section, we review and categorize the related work on transactional data mining for predictions and recommendations. Next basket prediction is an application of recommender systems based on implicit feedback where only positive observations (e.g., purchases or clicks) are available [14], [15], and no explicit preferences (e.g., ratings) are expressed [16]. The implicit feedback are given in a form of sequential transactional data obtained by tracking the users’ behavior over time [17], e.g. a retail store records the transactions of customers through fidelity cards.

Next basket prediction is mainly aimed at the construction of effective recommender systems (or recommenders). Recommenders can be categorized into general, sequential, pattern-based, and hybrid recommenders. General recommenders are based on collaborative filtering and produce recommendations for a customer based on general customers’ preferences [18], [19]. They do not consider any sequential information (i.e., which item is bought after which) and do not adapt to the customers’ recent purchases. In contrast, sequential recommenders are based on Markov chains and produce recommendations for a customer exploiting sequential information and recent purchases [20]. Pattern-based recommenders base predictions on frequent itemsets extracted from the purchase history of all customers while discarding sequential information [21], [22], [23]. Pattern-based approaches frequently exploit or extend the Apriori algorithm [24] for extracting the patterns.

The hybrid approaches combine the ideas underlying general and sequential recommenders. In [25] the authors use personalized transition graphs over Markov chains and compute the probability that a customer will purchase an item by using the Bayesian Personalized Ranking optimization criterion [26]. HRM [27] and DREAM [28] exploit both the general customers’ preferences and the sequential information by using recurrent neural networks. A different hybrid approach is described in [29]. This probability model merges Markov chain and association patterns.

All the approaches described above suffer from several limitations. For example, general recommenders and pattern-based recommenders do not take into account neither the sequential information (i.e., which item is bought after which) nor the customers’ recency. In contrast, sequential recommenders assume the independence of items in the same basket and do not capture factors like mutual influence. Furthermore, all the approaches require transactional data about many customers in order to make a prediction for a single customer. For this reason, they do not follow the user-centric vision for data protection as promoted by the World Economic Forum [7], [8], [30], which incentives personal data management for every single user of a database service. Cumby et al. [31] propose a predictor which embraces the 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 her personal data. Unfortunately, this approach assumes the independence of items purchased together. Also in [10] is proposed a personalized basket prediction model but it only considers co-occurrence and requires part of the next basket to perform the recommendation.

Finally, the main drawback of the hybrid approaches based on neural networks [27], [28], [29] is that their predictive models are difficult to interpret by humans. The interpretability of a predictive model, i.e., the possibility to understand the mechanisms underlying the predictions [32], is highly valuable for a retail chain manager interested in improving the marketing strategies and the service offered. Moreover, interpretability is also important to customers for gaining insights about their personal purchasing behavior.

We propose an interpretable approach to basket prediction compliant with the user-centric vision, i.e., just the data of a customer are used to make predictions for that customer [9]. In order to do that we model the interactions among items in the same basket as well as the interactions between items in consecutive baskets by considering simultaneously co-occurrence, sequentiality, periodicity and recurrency.

3 MARKET BASKET PREDICTION PROBLEM

We refer to market basket prediction as the prediction of the items a customer will purchase in her next transaction. Let \( C = \{c_1, \ldots, c_z\} \) be a set of \( z \) customers and \( I = \{i_1, \ldots, i_v\} \) be a set of \( v \) items. We indicate with \( B_c = (b_{i_1}, b_{i_2}, \ldots, b_{i_v}) \) the ordered purchase history of the baskets (or transactions) of customer \( c \), where \( b_{i_j} \subseteq I \) is the basket composition and \( t_{c} \in [t_1, t_n] \) the transaction time. Finally, \( B = \{B_{c_1}, \ldots, B_{c_n}\} \) is the set of all customers’ purchase histories.

Given the purchase history \( B_c \) of customer \( c \) and the time \( t_{n+1} \) of the next transaction, market basket prediction consists in providing the set \( b^*_c \) of \( k \) items that customer \( c \) will purchase in the next transaction \( b_{t_{n+1}} \).

Our approach to market basket prediction aims at overcoming the main limitations of existing methods illustrated in Section 2. To this purpose, we propose a hybrid predictor which combines ideas underlying sequential and pattern-based recommenders. The approach consists of two main components. The first one is the extraction of Temporal Annotated Recurring Sequences (TARS) from the customer’s purchase history, i.e., sequential recurring patterns able to capture the customer’s purchasing habits. The second one is the TARS Based Predictor (TBP), a predictive method that exploits the TARS of a customer to forecast her next basket.

4 CAPTURING PURCHASING HABITS

In this section we formalize TARS and describe how to extract them from the purchase history of a customer.

4.1 Temporal Annotated Recurring Sequences

Temporal Annotated Recurring Sequences (TARS) model two aspects: (i) the customer’s recurrent and sequential purchases, i.e., the fact that a set of items is typically purchased...
together and after another set of items; (ii) the recurrence of the sequential purchase, i.e., when and how often such pattern occurs in the customer’s purchase history.

To show how TARS capture these two aspects at the same time, we define their components and clarify their meaning with the help of a real-world example, which refers to a customer’s purchase history reported in Table 1.

**Definition 1 (Sequence).** Given a customer’s purchase history \( B_c = \{b_1, \ldots, b_{n_c}\} \), we call \( S = \langle X, Y \rangle = X \rightarrow Y \) a sequence if the pair of itemsets \( X \subseteq b_h \) and \( Y \subseteq b_f \), \( X, Y \neq \emptyset \), \( t_h < t_f \) and \( \exists S' = X' \rightarrow Y' \), \( X' \subseteq X \subseteq b_{l_h} \) and \( Y' \subseteq Y \subseteq b_{l_f} \) such that \( t_h', t_f' \in (t_h, t_f) \), \( X \) and \( Y \) are called the head and the tail of the sequence, respectively.

We denote with \( T_S = \langle t_1, \ldots, t_m \rangle \) the head time list of \( S \), i.e., the ordered list of the head’s time of all the occurrences of \( S \) in the customer’s purchase history. The support \( |T_S| \) of a sequence \( S \) is the size of its head time list. We call length of a sequence \( |S| = |X| + |Y| \) the sum of sizes of the head and of the tail. We say that a sequence \( S' \) is a subsequence of \( S'' \), \( S' \subseteq S'' \) if \( X' \subseteq X'' \) and \( Y' \subseteq Y'' \). Figure 1 shows the occurrences of sequence \( S = \{a\} \rightarrow \{b\} \) for a customer. We observe that, since by definition it cannot exist a \( S' \subseteq S \) with \( t_h', t_f' \in (t_h, t_f) \), then the first \( a \) is not considered as part of the sequence \( S \), and consequently is also not considered as part of its head time list, hence \( T_S = (01-05, 01-09, 01-13, 01-29, 02-06, 02-14) \).

Beyond the items in a sequence, there are other two crucial aspects needed for capturing re-occurrences: the intra-times between the itemsets \( X \) and \( Y \) of sequence \( S \) and the inter-times between a re-occurrence of sequence \( S \).

**Definition 2 (Intra-Time).** We define \( \alpha_{t_h} = t_f - t_h \) as the intra-time of an occurrence of a sequence \( S \), i.e., the difference between the time of the head and the time of the tail. We denote with \( A_S = (\alpha_1, \ldots, \alpha_m) \) the ordered intra-time list of all the occurrences of \( S \) in \( B \).

**Definition 3 (Inter-Time).** Given the head time list \( T_S \), we define \( \delta_j = t_{h_j} - t_{b_j} \), where \( t_{h_j}, t_{b_j} \in T_S \) and \( t_{b_j} < t_{h_j} \), as the inter-time of a sequence \( S \), i.e., the difference between the times of the heads of two consecutive occurrences of \( S \). We denote with \( \Delta_S = (\delta_1, \ldots, \delta_m) \) the ordered inter-time list of \( S \). We impose \( \delta_m = \alpha_m \) by construction.

In Figure 1, the intra-time list \( A_S \) consists of the differences between the heads and the tails of all the occurrences of \( S \), hence \( A_S = (4, 4, 16, 4, 4, 8) \). The inter-time list \( \Delta_S \) consists of all differences between the head times of two consecutive sequences, hence \( \Delta_S = (4, 4, 16, 8, 8, 8) \). Note that: (i) for each \( t_j \in T_S \) we have that \( \alpha_j \leq \delta_j \), i.e., the intra-time of a sequence is always lower or equal than its inter-time; (ii) for \( S = X \rightarrow X \), we have \( A_S = \Delta_S \).

**Definition 4 (Period).** Given a maximum inter-time \( \delta_{\max} \), a minimum number of occurrences \( q_{\min} \), the head time list \( T_S \) and the inter-time list \( \Delta_S \) of a sequence \( S \), we call period an ordered time list \( P(j) = \{t_{h_j}, \ldots, t_{l_j}\} \subseteq T_S \) such that \( \forall t_w \in P(j), \delta_w < \delta_{\max} \), \( P(j) \) is maximal, i.e., \( \delta_{l_j-1} > \delta_{\max} \), \( \delta_{l_j+1} > \delta_{\max} \) and \( |P(j)| \geq q_{\min} \). We denote with \( P(S) = \{P(1), \ldots, P(m)\} \) the set of periods of \( S \).

The period of a sequence \( S \) captures a temporal interval in which \( S \) occurs at least \( q_{\min} \) times and the time between any two occurrences is at most \( \delta_{\max} \). The support of a period \( |P(j)| \) indicates how many times \( S \) occurs in \( P(j) \). Given the previously observed property \( \alpha_j \leq \delta_j \) of intra- and inter-times, for a given \( \delta_{\max} \) in the definition of period we have that the inter-time also considers the intra-time. In Figure 1, for \( \delta_{\max} = 14 \) and \( q_{\min} = 2 \) we have two periods \( P(1) = (01-05, 01-09) \) and \( P(2) = (01-29, 02-06, 02-14) \) with support 2 and 3 respectively.

**Definition 5 (Recurring Sequence).** Let \( P_S = \{P(1), \ldots, P(m)\} \) be a set of periods, we define \( rec(S) = |P_S| \) as the recurrence of \( S \), i.e., the number of periods \( P_S \) in the customer’s purchase history. Given a minimum number of periods \( q_{\min} \), \( S \) is a recurring sequence if \( rec(S) \geq q_{\min} \).

In the example of Figure 1, for \( q_{\min} = 2 \) we have \( rec(S) = 2 \), meaning that \( S \) is a recurring sequence.

In summary, we have introduced the following basic concepts associated with a customer’s purchase history: (i) a sequence captures items purchased together and after other items; (ii) the period of a sequence is a time list respecting intra- and inter-time constraints; (iii) a recurring sequence is a sequence appearing in a certain number of periods. These four concepts are the components of a TARS, defined as:

**Definition 6 (Temporal Annotated Recurring Sequence).** Given a customer’s purchase history \( B \), a temporally annotated recurring sequence (TARS) is a quadruple \( \gamma = \langle S, \alpha, q, p \rangle \), where \( S = \langle X, Y \rangle = X \rightarrow Y \) is the sequence of itemsets, \( \alpha = (\alpha_1, \alpha_2) \in \mathbb{R}_+^2 \), \( \alpha_1 \leq \alpha_2 \) is the temporal annotation, \( p \) is the number of periods in which the sequence recurs, and \( q \) is the median of the number of occurrences in each period. A TARS will also be represented as follows:

\[
\gamma = X \xrightarrow{\alpha_{p,q}} Y
\]

2. We used the median to aggregate the number of occurrences in each period and as aggregation function in Algorithm 2 in order to obtain a more reliable representative value. Indeed, the median value is less subject than the mean to possible outliers and it is a good representative value also for skewed distributions [33].

### Table 1
Example of customer purchase history \( B_c \).

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Basket</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-01</td>
<td>{a, b, g, h}</td>
</tr>
<tr>
<td>01-05</td>
<td>{a, d}</td>
</tr>
<tr>
<td>01-09</td>
<td>{a, b, e, f, h}</td>
</tr>
<tr>
<td>01-13</td>
<td>{a, b, c, d, h}</td>
</tr>
<tr>
<td>01-17</td>
<td>{c, d, e, f, g}</td>
</tr>
<tr>
<td>01-21</td>
<td>{e, f, g}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Basket</th>
</tr>
</thead>
<tbody>
<tr>
<td>01-29</td>
<td>{a, b, c, g, h}</td>
</tr>
<tr>
<td>02-02</td>
<td>{b, c, d}</td>
</tr>
<tr>
<td>02-06</td>
<td>{a, c, d, e, f, i}</td>
</tr>
<tr>
<td>02-10</td>
<td>{b, e, f, h}</td>
</tr>
<tr>
<td>02-14</td>
<td>{a, b, c, d, e, g, h}</td>
</tr>
<tr>
<td>02-22</td>
<td>{a, b, g, h, i}</td>
</tr>
</tbody>
</table>
Algorithm 1: extractTars(B)
1 S ← extractBaseSequences(B);
2 \{q^\text{max}_S\}, \{q^\text{min}_S\}, \{p^\text{min}_S\} ← parametersEstimation(B, S);
3 S ← sequenceFiltering(B, S, \{δ^\text{max}_S\}, \{q^\text{min}_S\}, \{p^\text{min}_S\});
4 Ψ ← buildTarsTree(\{B, S', \{δ^\text{max}_{S'}\}, \{q^\text{min}_{S'}\}, \{p^\text{min}_{S'}\}\});
5 Γ ← extractTarsFromTree(Ψ);
6 return Γ;

A TARS is based on the concept of sequence, \(S = (X, Y) = X \rightarrow Y\), which intuitively indicates that itemset \(Y\) is typically purchased after another itemset \(X\). The itemsets themselves point out which items are purchased together. For example, a sequence \(\{a\} \rightarrow \{b, c\}\) indicates that \(\{b, c\}\) are purchased together after \(\{a\}\). The temporal annotation \(\alpha = (α_1, α_2)\) indicates the minimum intra-time \(α_1\) and maximum intra-time \(α_2\) of the sequence, i.e., the range of time elapsing between the purchase of \(X\) and the purchase of \(Y\). A sequence can appear in several distinct periods, i.e., time intervals where the sequence occurs continuously. The number of periods \(p\) characterizes these occurrences, that is, how many periods the sequence \(S\) appears. Finally, \(q\) indicates how many times \(S\) typically occurs in a period.

TARSs are an evolution of recurring patterns [34] which model recurrence but not sequentiality and periodicity, and temporally annotated sequences [35] which model sequentiality and periodicity but not recurrence. TARSs, besides co-occurrence, fills the gaps by modeling all the three aspects.

We refer to \(Γ = \{γ_1, \ldots, γ_m\}\) as the set of all the TARS of a customer \(c\). By specifying the maximum inter-time \(δ^\text{max}_{Γ_c}\), the minimum number of occurrences \(q^\text{min}_{Γ_c}\), and the minimum number of periods \(p^\text{min}_{Γ_c}\), we can determine the set \(Γ_c\) of TARSs that can be extracted from the purchase history \(B_c\).

4.2 TARS Extraction Procedure

To extract the TARSs from a customer’s purchase history \(B_c\) we use an extension of the well-known FP-Growth algorithm [36]. Although there are several algorithms that can be used to solve the same task, we adopt FP-Growth for the following reasons. First, FP-Growth produces results that are easily interpretable since it builds an FP-Tree structure, capturing the frequency at which itemsets occur in the dataset, where each node represents an item and each branch a different association. Second, it has been shown in the literature [37], [38], [39] that FP-Growth can be extended by attaching additional information to an FP-tree node in order to calculate the desired type of pattern. In our approach, we extend the FP-tree into a TARS-tree. Every node of a TARS-tree stores a sequence \(S\), the time list \(T_S\); its support \(|T_S|\); the intra-time list \(A_S\); the inter-time list \(Δ_S\); and the periods \(P\) derived from \(T_S\) with respect to \(δ^\text{max}_{Γ_c}\) and \(q^\text{min}_{Γ_c}\).

The TARS extraction procedure is described in Algorithm 1. In the first step, it extracts from the purchase history \(B\) the base sequences \(S\), i.e., the sequences of length 2 (line 1). Then, it estimates a set of parameters \(\{q^\text{max}_S\}, \{q^\text{min}_S\}, \{p^\text{min}_S\}\) for each base sequence \(S\) in \(S\) with respect to \(B\) (line 2). The base sequences \(S\) are then filtered with respect to these parameters and the base recurring sequences \(S^*\) are extracted, while the other base sequences are discarded to reduce the search space (line 3). Finally, the TARS-tree \(Ψ\) is built on the base recurring sequences \(S^*\) (line 4), and the set \(Γ\) of TARS annotated with \(α, p, q\) is extracted from \(Ψ\) (line 5) according to FP-Growth.

In Section 6.5.2 we will show that the TARS procedure overcomes the state-of-the-art in time. With respect to computational complexity the dominant part is the construction of the TARS-tree \(Ψ\) (line 4 of Alg. 1) that is implemented with FP-Growth. Therefore, since FP-Growth is an output-sensitive algorithm [40], the complexity of \(\text{extractTars}(\cdot)\) depends not only on the input \(\mathcal{B}\) but more likely on its output \(Γ\). Regarding the memory consumption, the TARS-tree construction related to FP-Growth is again the dominant part. Besides being easily extensible, in the literature it has been shown that FP-Growth [36] is more efficient than other existing frequent pattern mining algorithms (like Apriori [24]) both in time and space. In [36] is discussed how memory consumption in not a concern when the datasets analyzed are not huge: this is the case of the application of TARS in which the dataset refers to a single individual and it is consequently limited. Moreover, if problems should occur, in [41] is shown how to reduce FP-Growth memory consumption by about an order of magnitude.

4.2.1 Data-Driven Parameters Estimation

In order to make the parameters \(δ^\text{max}_{Γ_c}, q^\text{min}_{Γ_c}, p^\text{min}_{Γ_c}\) adaptive not only to the individual customer [42], but also to the sequences in \(B_c\), we apply two pre-processing steps on the base sequences \(S\) (lines 1–2 Algorithm 1).

Algorithm 2: parametersEstimation(\(S, B\))
1 \(D^\text{max} \leftarrow \emptyset; D^\text{min} \leftarrow \emptyset; D^\text{p} \leftarrow \emptyset;\)
2 foreach \(S \in S\) do
3 \(\text{if } \text{support}(S) \geq \text{median}(\text{Support}(\mathcal{B}));\)
4 \(\text{for } \mathcal{B}_c \in \mathcal{C}_{\text{min}}\) do
5 \(\text{foreach } S \text{assignedTo}(\mathcal{C}_c) \text{ do } q^\text{min}_{S} \leftarrow \text{median}(\mathcal{C}_c);\)
6 \(\text{for } S \in S\) do
7 \(TC_S \rightarrow \text{getTimeCompliantPeriods}(S, \mathcal{B}, \{δ^\text{max}_{S}\});\)
8 \(D^\text{min} \leftarrow D^\text{min} \cup \text{median}(\{q^\text{max}_S(TC^\text{S}_{\text{C}}) \mid TC^\text{S}_{\text{C}} \in TC_S\});\)
9 \(C_{\text{min}} \leftarrow \text{groupSimilar}(D^\text{min});\)
10 \(\text{for } \mathcal{B}_c \in \mathcal{C}_{\text{min}}\) do
11 \(\text{foreach } S \text{assignedTo}(\mathcal{C}_c) \text{ do } q^\text{min}_{S} \leftarrow \text{median}(\mathcal{C}_c);\)
12 \(\text{for } S \in S\) do
13 \(p_S \leftarrow \text{getPeriods}(S, \mathcal{B}, \{δ^\text{max}_{S}\}, \{q^\text{min}_S\});\)
14 \(w_S = \sum_{e = p_S}^{\text{min}(\text{tree}(P_S))} |e_S - w_S|/|P_S|; D^\text{min} \leftarrow D^\text{min} \cup \{e_S\};\)
15 \(C_{\text{min}} \leftarrow \text{groupSimilar}(D^\text{min});\)
16 \(\text{for } \mathcal{B}_c \in \mathcal{C}_{\text{min}}\) do
17 \(\text{foreach } S \text{assignedTo}(\mathcal{C}_c) \text{ do } p^\text{min}_{S} \leftarrow \text{median}(\{\text{tree}(P_S)\} = |P_S| S^\text{assignedTo}(\mathcal{C}_c));\)
18 return \(\{q^\text{max}_S\}, \{q^\text{min}_S\}, \{p^\text{min}_S\}\);
The first pre-processing step is the data-driven estimation of the sets of parameters $\{\delta_{\text{max}}^S\}, \{q_{S}^\text{min}\}, \{p_{S}^\text{min}\}$ described in Algorithm 2. Let $S$ be the set of base sequences and $\delta_S$ be the median of inter-times in $\Delta_S$ (Algorithm 2, line 2). Given a base sequence $S$, we estimate parameter $\delta_{\text{max}}$ as follows: (i) we group the base sequences with similar inter-times $\delta_S$ (line 3) obtaining a set of clusters $C_{\delta_{\text{max}}}=\{C_1, \ldots, C_{C_h}\}$; (ii) if $S \in C_h, C_h \in C_{\delta_{\text{max}}}$, we set $\delta_{S_{\text{max}}}$ as the median of the $\delta_S$ values in cluster $C_h$ (lines 4–5).

Then, we calculate the periods $TC_S$ compliant only with the temporal constraint $\delta_{\text{max}}^S$ (lines 6–8) and we estimate $\{q_{S}^\text{min}\}$: (i) we group the base sequences with similar median number of occurrences per period $q_S$ producing a set of clusters $C_{q_{\text{min}}} = \{C_1, \ldots, C_{C_h}\}$ (line 9); (ii) if $S \in C_h$, $C_h \in C_{q_{\text{min}}}$ we set $p_{S}^\text{min}$ as the median of the $q_S$ in $C_h$ (lines 10–11).

Similarly, we estimate $\{p_{S}^\text{min}\}$ as follows: (i) we compute the sum of the number of occurrences of a base sequence in the periods $w_S$ and we calculate the expected number of occurrences per period $e_S$ as $w_S/P_S$ (lines 12–14); (ii) we group the base sequences with similar $e_S$ producing a set of clusters $C_{p_{\text{min}}} = \{C_1, \ldots, C_{C_h}\}$ (line 15); and (iii) if $S \in C_h$, $C_h \in C_{p_{\text{min}}}$ we set $p_{S}^\text{min}$ as the median of the number of periods of the base sequences in $C_h$ (lines 16–17).

We group the base sequences, $\text{groupSimilart()}$ in Algorithm 2, by dividing the values into equal-sized bins [43]. Each bin corresponds to a group containing similar values.

### 4.2.2 Sequence Filtering

The second pre-processing step consists in selecting the base recurring sequences, i.e., the base sequences satisfying the sets of parameters $\{\delta_{S}^\text{max}\}, \{q_{S}^\text{min}\}, \{p_{S}^\text{min}\}$. We apply this filtering to reduce the search space so that the building of the TARS-tree and the TARS extraction (Algorithm 1) are employed only on the super-sequences of the base recurring sequences. In other words, if $S_1$ is not a base recurring sequence and $S_1 \subset S_2$, then we assume as a heuristic that $S_2$ is not recurring too, and we eliminate it through sequence filtering process. We adopt the sequence filtering heuristic for reducing the search space because the antimonotonic property [46] does not apply to TARS.

Consider $S_1 = \{c\} \rightarrow \{c\}$ and $S_2 = \{c, d\} \rightarrow \{c\}$ in the example of Table 1, we have that $S_1 \subset S_2$. Given $\delta_{\text{max}} = 14$, $q_{\text{min}} = 2$ and $p_{\text{min}} = 2$, we have $\text{rec}(S_1) = 1$ and $\text{rec}(S_2) = 2$. Hence, $S_2$ is recurrent while $S_1$ is not, and the anti-monotonic property is not satisfied. However, it is clear from this example that a TARS like $S_1$ could be useful for the prediction because, despite $\text{rec}(S_1) = 1$ in total it occurs six times $|\hat{P}_{S_1}^{(1)}| = 6$. In real-world, $\{c\}$ could be a fresh product (like milk or salad) that is repeatedly and frequently purchased. Hence, an imposed parameter setting could be not appropriate because (i) it could remove too many TARS which are in fact useful for the prediction; (ii) it could consider too many valid base sequences and not prune enough the search space.

4. The number of bins is estimated as the maximum between the bins suggested by the Sturges [44] and the Freedman-Diaconis methods [45].

5. We point out that, with respect to the final application of TARS, we do not know if the patterns discarded by sequence filtering and by FPGrowth using the sets of parameters $\{\delta_{S}^\text{max}\}, \{q_{S}^\text{min}\}, \{p_{S}^\text{min}\}$ could be potentially useful for the prediction. However, without a filtering of the search space TARS extraction would becomes practically intractable.

### 5 TARS Based Predictor

On top of the set $C_\gamma$ of TARS extracted from a customer’s purchase history $B_c$, we build the TARS Based Predictor (TBP), an approach for market basket prediction that is markedly personalized and user-centric [7], [8]; the predictions for a customer $c$ are performed using only the model build on her purchase history $B_c$, i.e., her TARS $\Gamma_c$.

TBP exploits TARS to simultaneously embed complex item interactions such as co-occurrence (which item is bought with which), sequential relationship (which items are bought after which), periodicity (which item is bought when) and typical times of re-purchase (after when re-purchases happen). These factors enable TBP to observe the customer’s recent purchase history and understand which are the active patterns, i.e., the purchasing patterns that the customer is currently following. In turn, by knowing the active patterns, TBP can provide the items that the customer will need at the time of the next purchase. It is worth noting that TBP is parameter-free: all the parameters of the TARS model $\Gamma_c$ are automatically estimated for each customer on her personal data $B_c$, avoiding the usual case where the same parameter setting is used indiscriminately for all the customers [42].

Given the purchase history $B_c$ of customer $c$, the time $t_{n+1}$ of $c$’s next transaction, and $c$’s TARS set $\Gamma_c$, TBP works in two steps. First, it selects the set $\Gamma_c$ of active TARS. Second, it computes a score $\Omega_{c_i}$ for every item $i$ belonging to an active TARS in $\Gamma_c$, ranks the items according to $\Omega_{c_i}$, and selects the top $k$ items as the basket prediction for $c$.

Algorithm 3 shows TBP’s procedure to select the active TARS $\hat{\Gamma}$ of a customer. First, it sorts the purchase history $B$ from the most recent basket to the oldest one, then it loops on pairs of consecutive baskets (line 2) searching for a set $\hat{\Gamma}$ of potentially active TARS (lines 4–7). When it finds a potentially active TARS $\gamma$, it considers two cases. If the sequence $S$ of $\gamma$ is encountered for the first time, the algorithm adds $\gamma$ to the set $\hat{\Gamma}$ of active TARS and initializes

```
Algorithm 3: getActiveTARS(B, t_{n+1}, \Gamma)

1. $\hat{\Gamma} \leftarrow \emptyset$, $Q \leftarrow \emptyset$, $L \leftarrow \emptyset$, $T \leftarrow \emptyset$
2. for $b_t, b_{t-1} \in \text{sort-desc}(B)$ do
   3.   $\alpha_{j-1} \leftarrow t_j - t_{j-1}$;
   4.   for $X \subseteq b_t$ do
      5.     for $Y \subseteq b_{t-1}$ do
         6.       if $\gamma \in Y \cap \gamma = (S, \alpha, p, q) \land \alpha_1 \leq \alpha_{j-1} \leq \alpha_2 \land S = (X, Y) \rightarrow X \rightarrow Y$
            7.         then $Q_\gamma \leftarrow Q_\gamma + 1; L_\gamma \leftarrow t_j - 1$;
            8.         if $Q_\gamma > q$ then $\hat{\Gamma} \leftarrow \hat{\Gamma} \cup \gamma$;
            9.         $T \leftarrow T / \gamma$;
           10.        if $L_\gamma - t_{j-1} > q \cdot (\alpha_1 - \alpha_2)$ then $T \leftarrow T / \gamma$;
           11.      else $\hat{\Gamma} \leftarrow \hat{\Gamma} \cup \gamma; Q_\gamma \leftarrow 1; L_\gamma \leftarrow t_{j-1}$;
       12.   endfor
   13. endfor
   14. endfor
15. return $\hat{\Gamma}, Q$;
```

For these reasons, we developed the pre-processing steps for parameters estimation described in this section.
 Indeed, if a customer typically purchases baskets with a few items it is useless to predict a basket with a large number of items. On the other hand, if a customer typically purchases baskets with a large number of items, the prediction of a small basket will not cover most of the items purchased. In this paper, we report the evaluation of the predictions made using both a fixed length $k \in \{2, 20\}$ for all the customers and using a customer-specific size $k = k_c^*$, where $k_c^*$ indicates the average basket length of customer $c$.

According to the literature [25], [27], [28], [31], we adopt a leave-one-out strategy for model validation: for each customer $c$ we use the baskets in the purchase history $B_c = \{b_{n_1}, \ldots, b_{n_n}\}$ for extracting the TARS, and the basket $b_{n+1}$ to test the performance. For each customer, we evaluate the agreement of the predicted $b^*$ and the real basket $b$ using the following metrics:

- $F1$-score, harmonic mean of precision and recall [47]:
  \[
  F1\text{-score}(b, b^*) = \frac{2 \cdot \text{Precision}(b, b^*) \cdot \text{Recall}(b, b^*)}{\text{Precision}(b, b^*) + \text{Recall}(b, b^*)}
  \]
  \[
  \text{Precision}(b, b^*) = \frac{|b \cap b^*|}{|b^*|}
  \]
  \[
  \text{Recall}(b, b^*) = \frac{|b \cap b^*|}{|b|}
  \]
- Hit-Ratio, the ratio of customers who received at least one correct prediction (a hit) [48]:
  \[
  \text{Hit-Ratio}(b, b^*) = 1 \text{ if } b \cap b^* \neq \emptyset, 0 \text{ otherwise.}
  \]
- Normalized $F1$-score: the $F1$-score calculated only for the customers having at least one hit.

Furthermore, for each customer we compute both learning and prediction time. The learning time is the amount of time required to extract the model. The prediction time is the amount of time the predictor needs to predict the next basket of a customer. We perform the experiments on Ubuntu 16.04.1 LTS 64 bit, 32 GB RAM, 3.30GHz Intel Core i7.

According to the literature, we report the evaluation metrics by aggregating the quality measures calculated for each customer by using mean, median and percentiles.

It is important to notice that, due to the nature of our problem formulation, and in line with [31], we do not adopt measures of ranking quality like NDCCG and DCG [49]. Such choice is supported by three motivations.

First, since we are dealing with retail transactions we do not have a rating provided by the customers for each item purchased, i.e., an explicit feedback like the voting assigned to movies, songs, restaurants, hotels, etc., that can be used as ground truth for the ranking measures.

Second, we can not use implicit feedback like the individual (or collective) purchase frequency because this would mean to assume that every user would prefer to have in her recommendation the items most frequently purchased - rather than items that are easily forgettable because not very frequent or subjected to seasonality: ranking measures assume that very important items are more useful when appearing earlier in the result list.

Finally, in the market basket prediction problem formulation, both the predicted $b^*$ and the real basket $b$ are set without any order among their items.

6. Experiments on Retail Data

In this section we report the experiments performed on three real-world datasets to show the properties of the TARS and the effectiveness of TBP in market basket prediction. We also highlight an important property of TARS, i.e., their interpretability, showing how crucial aspects like seasonality and inter-purchase times can be easily inferred from TARS.

6.1 Experimental Settings

State-of-the-art methods [25], [27], [28], [31] fix the size of the predicted basket to $k = 5$ or $k = 10$. However, we think that the size $k$ of the predicted basket should adapt to the customer’s personal behavior.

We provide at https://github.com/GiulioRossetti/tbp-next-basket the Python code of TBP and of the baseline methods with open source datasets and an anonymized sample of the private Coop dataset. TARS and TBP code is also indexed within the SoBigData resource catalogue https://goo.gl/N6UhmM. We also provide details about the parameter setting used for the different methods if not specified in the paper. The code of DRM was kindly provided by the authors of [28].
6.2 Datasets

We performed our experiments on three real-world transactional datasets: \textit{Coop-A}, \textit{Coop-C} (both extracted from the private \textit{Coop} repository) and the open source \textit{Ta-Feng} dataset. Table 2 shows the details of the datasets.

The \textit{Coop} repository is provided by Unicoop Tirreno\footnote{7. https://www.unicooptirreno.it/}, a big retail supermarket chain in Italy. It stores 7,407,056 transactions made by 10,000 customers in 23 different shops in the province of Leghorn, over the years 2007-2014. The set of Coop items includes food, household, wellness, and multimedia items. There are 7,690 different articles classified into 520 market categories. From the repository, we extract two datasets: \textit{Coop-A} and \textit{Coop-C}. The two datasets differ in the items categorization. In \textit{Coop-A} (articles) the items are mapped to a more general category: in the example above blood orange and navel orange. In \textit{Coop-C} (categories) the items are mapped to a more general category: in the example above blood orange and navel orange are considered the same generic item (orange). All the customers in \textit{Coop-A} and \textit{Coop-C} have at least one purchase per month.

\textit{Ta-Feng}\footnote{8. http://www.bigdatalab.ac.cn/bronze/bm/dd?data=Ta-Feng} is a dataset covering covers food, stationery and furniture, with a total of 23,812 different items. It contains 817,741 transactions made by 32,266 customers over 4 months. We remove customers with less than 10 baskets and we consider only the remaining 7\% customers.

Since we run experiments on retail data we adopt the \textit{day} as time unit: both the parameters and the TARS annotations are expressed in days.

6.3 Interpretability of TARS

The interpretability of TARS is one of the main characteristics of our approach. Table 3 shows some examples of TARS extracted from \textit{Coop-C}. In the table, we report the median of $\alpha$, $p$ and $q$ across all the customers having the presented TARS. We observe that TARS with a recurring base sequence are the most supported among the customers.

For example \{\textit{milk}\} $\rightarrow$ \{\textit{milk}\} and \{\textit{banana}\} $\rightarrow$ \{\textit{banana}\} are supported by more than 90\% of the customers in \textit{Coop-C}. The two TARS have similar $q$ (6.58 and 7.20 respectively) indicating that they have similar recurrence degrees, i.e., they occur a similar number of times in the respective periods. In contrast \{\textit{banana}\} $\rightarrow$ \{\textit{banana}\} has a higher maximum intra-time ($\alpha_2 = 35$) and a lower average number of recurrences ($p = 14.63$). This indicates that: (i) the time for a banana re-purchase is higher than the time of a milk re-purchase; (ii) the support to have a distinct period is higher for \{\textit{banana}\} than \{\textit{milk}\}.

Moreover, we notice for more than 25\% of the customers the contemporary purchase \{\textit{bread}, \textit{tomato}\} can indicate a future basket with \{\textit{bovine}\} or with \{\textit{banana}, \textit{potato}\} and that these TARS have very different annotations $\alpha$, $p$, $q$. Finally, we highlight that, even if the most common TARS among the customers are those with base sequences, the TARS in $\Gamma_c$ with sequence length greater than two are on average more than the 95\% for each customer.

For better understanding the TARS, in Table 4 we show some TARS made of base recurring sequences with different peculiarities. A base recurring sequence captures the typical repurchasing of the same item within a certain period for a certain number of times.

\textit{Apples} and \textit{bananas} are fruit items available throughout all the year. The associated base TARS \{\textit{banana}\} $\rightarrow$ \{\textit{apple}\} and \{\textit{apple}\} $\rightarrow$ \{\textit{apple}\} have indeed a similar number of periods $p$ and number of typical occurrences in each period $q$.

\textit{Oranges} are a seasonal fruit item, generally available between November and February. The associated base TARS \{\textit{orange}\} $\rightarrow$ \{\textit{orange}\} has a recurrence $p$ significantly lower than the recurrence of banana and apple TARS, while the occurrence inside a period is similar. We observe that ice creams are similar to oranges: the associated TARS \{\textit{ice cream}\} $\rightarrow$ \{\textit{ice cream}\} has a lower $p$ and a higher maximum intra-time $\alpha_2$.\footnote{9. Supported by more than 90\% customers

<table>
<thead>
<tr>
<th>Dataset</th>
<th>cust.</th>
<th># baskets</th>
<th># items</th>
<th>avg basket per cust.</th>
<th>avg basket length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coop-A</td>
<td>10,000</td>
<td>7,407,056</td>
<td>4594</td>
<td>432.4±353.4</td>
<td>9.4±5.8</td>
</tr>
<tr>
<td>Coop-C</td>
<td>10,000</td>
<td>7,407,056</td>
<td>407</td>
<td>432.4±353.4</td>
<td>8.6±4.9</td>
</tr>
<tr>
<td>Ta-Feng</td>
<td>2,319</td>
<td>24,304</td>
<td>5,117</td>
<td>10.4±7.5</td>
<td>1.8±1.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>cust.</th>
<th># baskets</th>
<th># items</th>
<th>avg basket per cust.</th>
<th>avg basket length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coop-A</td>
<td>10,000</td>
<td>7,407,056</td>
<td>4594</td>
<td>432.4±353.4</td>
<td>9.4±5.8</td>
</tr>
<tr>
<td>Coop-C</td>
<td>10,000</td>
<td>7,407,056</td>
<td>407</td>
<td>432.4±353.4</td>
<td>8.6±4.9</td>
</tr>
<tr>
<td>Ta-Feng</td>
<td>2,319</td>
<td>24,304</td>
<td>5,117</td>
<td>10.4±7.5</td>
<td>1.8±1.1</td>
</tr>
</tbody>
</table>

\textbf{TABLE 2} Statistics of the datasets used in the experiments.

\textbf{TABLE 3} Examples of TARS extracted from \textit{Coop-C}.

\textbf{TABLE 4} Periods of TARS with different recurring base sequences from \textit{Coop-C}.

For each TARS is shown how the periods, represented as horizontal single lines, occur along 7 years of observations.
Finally, Strawberries and Easter eggs are items available for just a short period of the year. As result, in the associated TARS we have lower values of both $p$ and $q$ than the other TARS. In particular, among the items considered strawberries’ TARS have the lowest $\alpha_2$ indicating short periods, while Easter eggs have the highest $\alpha_1$ indicating long intra-times.

### 6.4 Properties of TBP

In this section, we present the peculiar properties of TBP: the temporal validity and reliability of the TARS extracted, and the performance improvements yield by parameters estimation. Since these experiments are closely tied to the applicability of TBP in real services, we report the results obtained on Coop dataset where the period of observation (7 years) is much more statistically significant than Ta-Feng.

#### 6.4.1 TARS Temporal Validity

In real-world applications is unpractical, or even unnecessary, to rebuild a predictive model from scratch every time a new basket appears in a customer’s purchase history. This leads to the following question: for how long are TBP predictions reliable? We address this question by extracting TARS on the 70% of the purchase history of every customer and performing the prediction on the subsequent baskets.

As shown in Figure 2, regardless the predicted basket size $k$, F1-score and Hit-Ratio remain stable up to 20 predictions, which suggests a large temporal validity of TBP since the model construction.

#### 6.4.2 TARS Extraction Reliability

How many baskets does TBP need to perform reliable predictions? For each customer, we start from her second week of purchases and we extract the TARS incrementally by extending the training set one week at a time. We then predict the next basket of the customer and we evaluate the performance of TBP in this scenario.

Figure 3 shows the median value and the “variance” (by means of the 10th, 25th, 75th and 90th percentiles) of the F1-score, (top-left), the total number of different items purchased by the customer (top-right), the number of TARS extracted (bottom-left), the number of active TARS during the prediction (bottom-right) as the number of weeks used in the learning phase increases. The average F1-score does not change significantly as the number of weeks increases, while its “variance” reduces as more weeks are used in the learning phase. Differently, the other measures stabilize after an initial setup phase. Thus, this experiment underlines that TBP needs at least 9-12 months of data to produce reliable performances as well as sound, stable, TARS.

#### 6.4.3 Parameter-Free vs. Parameter-Fixed

TARS can be extracted by fixing the same parameters for all the customers and items, as usually done by state-of-the-art methods [25], [27], [28], [31], or by automatically estimating the parameters with a data driven procedure.

In this section we discuss and analyze the impact of fixing the parameters on the predictive performance by comparing the results of parameter-free TBP and a parameter-fixed version of TBP where we set $\delta^{\text{min}}=14$ (e.g., two weeks), $q^{\text{min}}=3$ and $p^{\text{min}}=2$.

Figure 4 shows the distributions of the number of TARS per customer for the parameter-free (left) and parameter-fixed (right) TARS extraction. The bottom line reports a focus of the distributions of the base TARS, i.e., TARS with length equals to 2.

Figure 5 compares the predictive performances of the parameter-free and the parameter-fixed scenarios. For both F1-score and Hit-Ratio, TBP produces better predictions in the parameter-free scenario. In particular, when using the...
average basket size of a customer $k^*_c$ as the size of the predicted basket, the parameter-free approach has F1-score = 0.25 while the parameter-fixed approach has F1-score=0.21. Our results suggest that the adoption of a parameter-free strategy during the extraction of TARS enforces customer behavior heterogeneity and increases prediction accuracy.

6.5 Comparison with Baseline Methods

We compare TBP with several baseline methods on Coop-A, Coop-B and Ta-Feng datasets.

6.5.1 Baseline Methods

We implemented the following user-centric state-of-the-art methods. We recall that these approaches build the predictive model of a customer relying only on her purchase data.

- **LST** [31]: the next basket predicted is the last basket purchased by the customer, i.e., $b_{n+1} = b_n$.

- **TOP** [31]: predicts the top-$k$ most frequent items with respect to their appearance, i.e., number of times that are purchased, in the customer’s purchase history $B_c$.

- **MC** [31]: makes the prediction based on the last purchase $b_n$ and on a Markov chain calculated on $B_c$.

- **CLF** [31]: for each item $i$ purchased by the customer, this method builds a classifier on temporal features extracted from the customer’s purchase history considering two classes: “item $i$ purchased yes/no”. The classifier then predicts the next basket using the temporal features extracted from the customer’s purchase history. Examples of the features extracted from a basket $b_i$ are: the number of days at $t_j$ since item $i$ was bought by $c$, the frequency of purchasing $i$ at time $t_j$, etc.

We also implemented four state-of-the-art methods that are not user-centric, i.e., they require and use purchase data of all customers $B$ to build a collective predictive model:

- **NMF** (Non-negative Matrix Factorization) [50]: is a collaborative filtering method which applies a non-negative matrix factorization over the customers-items matrix. The matrix is constructed from the purchase history of all customers $B$.

- **FMC** (Factorizing personalized Markov Chain) [25]: using the purchase history of all the customers $B$, it combines personalized Markov chains with matrix factorization in order to predict the next basket;

- **HRM** (Hierarchical Representation Model) [27]: 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.

**DRM (Dynamic Recurrent Basket Model)** [28]: it is based on recurrent neural network and can capture both sequential features from all the baskets of a customer, and global sequential features from all the baskets of all the customers $B$.

Theoretically, user-centric methods should perform better than not user-centric methods in solving the market basket prediction problem. Indeed, a user-centric method which is fit on the particular behavior of a customer should be advantaged and should not suffer from the noise generated by the collective shopping behavior. However, not user-centric methods, by exploiting the similarity among various customers, can predict items that a customers has never bought before, and can be employed also for new customers just after one purchase. On the contrary, a user-centric method require a minimum number of purchases in order to provide a reliable prediction.

We do not compare against the methods described in [21], [22], [29] because, even though they employ patterns for producing recommendations, they are designed for web-based services, and because they specifically exploit and use the items’ ratings and not only the occurrences of the items in a basket.

With respect to the not user-centric baseline methods – NMF, FMC, HRM, DRM – we performed preliminary experiments for each dataset in order to tune the dimensionality $d$ used to represent the data. In line with [27], [28], for Ta-feng we set $d=200$ where all the baselines show the best performance. For Coop-A and Coop-C, as consequence of empirical experiments, we set $d=100$ where there is a good balance between the quality of the performance and the learning time. Indeed, we underline that, probably as consequence of both the 7 years of transactions in Coop against the four months of Ta-feng, and of the higher density of Coop dataset, for HRM and DRM we report the results of the test performed on a sample of Coop with 100 customers due to large computational time (see Table 6).

6.5.2 Market Basket Prediction Evaluation

Table 5 reports the average F1-score and Hit-Ratio of TBP against the baseline methods when setting the length of the predicted basket equals to the average basket length for each prediction of each individual customer, i.e., $k=k^*_c$. This kind of evaluation is markedly user-centric and would be a suitable approach in implementing a real personalized basket recommender tailored on the customer behavior. TBP outperforms the baselines both in terms of F1-score and Hit-Ratio and, together with the others user-centric approaches, it outlines how for this particular task a user-centric model is more accurate than a not user-centric one.

<table>
<thead>
<tr>
<th>$k = k^*_c$</th>
<th>TBP</th>
<th>TOP</th>
<th>MC</th>
<th>CLF</th>
<th>LST</th>
<th>NMF</th>
<th>FPM</th>
<th>HRM</th>
<th>DRM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coop-A</td>
<td>.17</td>
<td>.14</td>
<td>.14</td>
<td>.13</td>
<td>.09</td>
<td>.14</td>
<td>.08</td>
<td>.06</td>
<td>.05</td>
</tr>
<tr>
<td>Coop-C</td>
<td>.24</td>
<td>.22</td>
<td>.23</td>
<td>.19</td>
<td>.14</td>
<td>.22</td>
<td>.16</td>
<td>.08</td>
<td>.12</td>
</tr>
<tr>
<td>Ta-Feng</td>
<td>.09</td>
<td>.09</td>
<td>.06</td>
<td>.09</td>
<td>.06</td>
<td>.08</td>
<td>.08</td>
<td>.08</td>
<td>.07</td>
</tr>
<tr>
<td>Coop-A</td>
<td>.62</td>
<td>.58</td>
<td>.58</td>
<td>.56</td>
<td>.40</td>
<td>.59</td>
<td>.44</td>
<td>.35</td>
<td>.33</td>
</tr>
<tr>
<td>Coop-C</td>
<td>.72</td>
<td>.71</td>
<td>.70</td>
<td>.65</td>
<td>.50</td>
<td>.71</td>
<td>.61</td>
<td>.58</td>
<td>.55</td>
</tr>
<tr>
<td>Ta-Feng</td>
<td>.32</td>
<td>.34</td>
<td>.24</td>
<td>.31</td>
<td>.15</td>
<td>.31</td>
<td>.31</td>
<td>.31</td>
<td>.29</td>
</tr>
</tbody>
</table>
To support such findings, the reported results were tested for their statistically significance applying a Friedman test with Bonferroni-Dunn post-hoc evaluation [51]. The test was rejected for both Hit-Ratio and F1-score values with a p-value of 0.05, thus implying that the compared methods do actually behave differently when tested on multiple datasets. Conversely, the post-hoc underlined that TBP significantly outperforms the global approaches under the same confidence interval, and only LST with p-value 0.1.

In Table 6 we report the learning time, i.e., the time needed to build every method. Note that (i) it is expressed in seconds (s) or in hours (h); (ii) for user-centric methods (TBP, MC, CLF) we report the average time per customer while for not user-centric methods (NMF, FPM, FRM, DRM) the total time; (iii) HRM and DRM are tested on a sample* of Coop; (iv) learning time for TOP and LST is always lower than 0.01 seconds. We do not report the prediction time because it is negligible for all the approaches (i.e., less than 0.01 seconds).

We observe that TBP needs more time than existing user-centric methods (5 minutes per customer on average) but, if a prediction is required only for a customer, it is much faster than the not user-centric approaches that require learning the model for all the customers. We believe that such a learning time is acceptable for two reasons: (i) in a real scenario the TARS can be re-computed once every month and still produce reliable predictions; (ii) the computation can be parallelized and personalized with respect to the customer’s behavior, thus the TARS of all the customers can be extracted at the same time by different devices.

To better understand how the performance are affected by the variation of the predicted basket length  \( k \), in Figure 6 we compare the average F1-score (top row) and the average Hit-Ratio (bottom right) produced by TBP and by all the baseline methods while varying  \( k \in [2, 20] \).

**Table 6** Learning time comparison. The learning time for TOP and LST is not reported in the Table because it is always lower than 0.01 seconds. *Test carried on a sample of 100 customers.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>TBP</th>
<th>MC</th>
<th>CLF</th>
<th>NMF</th>
<th>FPM</th>
<th>HRM</th>
<th>DRM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coop-A</td>
<td>351.86s</td>
<td>0.04s</td>
<td>2.38s</td>
<td>244.28s</td>
<td>0.21h</td>
<td>0.84h*</td>
<td>47.53h*</td>
</tr>
<tr>
<td>Coop-C</td>
<td>6.62s</td>
<td>0.01s</td>
<td>1.08s</td>
<td>69.98s</td>
<td>0.11h</td>
<td>0.72h*</td>
<td>34.06h*</td>
</tr>
<tr>
<td>Ta-Feng</td>
<td>0.01s</td>
<td>0.00s</td>
<td>0.00s</td>
<td>803.89s</td>
<td>0.41h</td>
<td>0.34h</td>
<td>4.24h</td>
</tr>
</tbody>
</table>

We observe that TBP considerably overtakes the baseline methods on Coop-A and Coop-B having the highest F1-score and a comparable and competitive Hit-Ratio.

On Ta-Feng TBP has the highest F1-score at the second highest Hit-Ratio. The decrease of the Hit-Ratio of TBP in Ta-Feng is probably due to the very high data sparsity of the dataset. Indeed, as we observe in Table 2, Ta-Feng has a much lower average number of baskets per customer, a much lower average basket length, and a shorter observation period than Coop-A and Coop-C. For this reason, the TARS extracted from Ta-Feng have lower quality than the TARS extracted on the other datasets.

Finally, we underline that a high F1-score, which considers simultaneously precision and recall, is a better indicator than a high Hit-Ratio that only signals that at least an item is predicted. Thus, the improvement of the performance for market basket prediction of TBP with respect to the state of the art are not negligible either using a personal \( k = k^* \) or if a fixed \( k \) is specified for every customer.

Moreover, we notice that the F1-scores can be biased by two extreme scenarios: (i) the F1-score can be low because of a low Hit-Ratio, i.e., for most of the customers no item is predicted even though for some customers we predict most of the items; (ii) the F1-score can be high because for most of the customers just one item is predicted.

Thus, in Figure 7 we show the performance using the normalized F1-score instead of the F1-score. We observe that the positive gap between TBP and the competitors increases: for the customers for which TBP correctly predicts at least one future basket, the baskets predicted by TBP are more accurate and cover a larger number of items than the baskets predicted by the other methods.

![Fig. 6. Performance comparison of TBP against the baselines varying length \( k \): F1-score in the top row, Hit-ratio in the bottom row.](image)

![Fig. 7. Normalized F1-score varying predicted basket length \( k \).](image)
We also investigate to what extent the performances can be affected by the leave-one-out evaluation strategy: the last basket of a customer could depart from her typical behavior affecting the extraction of the TARS.

To cope with this issue we perform the learning process (i.e., extract TARS) by selecting a random subset $B'_c = \{b_{t1}, \ldots, b_{t_n}\}$ of the customers’ purchase history $B_c = \{b_{t1}, \ldots, b_{t_{n'}}\}$, with $t_{n'} < t_n$. We randomly vary the size of the subset $|B'_c|$ among 70% and 90% of $|B_c|$, and we apply TBP on the subsequent basket $b_{t_{n'+1}}$.

Figure 8 presents the results of this experiment for Coop-A and Coop-C and confirms the trends observed in the previous experiments: the leave-one-out evaluation strategy does not affect significantly the performance of the methods.

7 Conclusions

In this work, we have proposed a data-driven, interpretable and user-centric approach for market basket prediction. We have defined Temporal Annotated Recurring Sequences (TARS) and used them to construct a TARS Based Predictor (TBP) for next basket forecasting. Being parameter-free, TBP leverages the specificity of the customers’ behavior to adjust the way TARS are extracted, thus producing more personalized patterns. We have performed experiments on real-world datasets showing that TBP outperforms state-of-the-art methods. Equally important, we have shown that the extraction of TARS provides valuable interpretable patterns that can be used to gather insights on both the customers’ purchasing behaviors and products’ properties like seasonality and inter-purchase times. Our results demonstrate that at least 36 weeks of a customer’s purchase behavior are needed to effectively predict her future baskets. In this scenario, TBP can effectively predict the subsequent twenty future baskets with remarkable accuracy.

Our approach could be adopted by retail market chains to implement an efficient personal cart assistant for reminding to the customers the products that they actually need. Being parameter-free and user-centric, the application could potentially run on private devices or data stores [12], guaranteeing in this way the privacy by design property [52]. Another interesting application for studying consumer behavior is related to detecting the churn from personal purchasing patterns. They can be detected by finding the TARS which are never active during prediction phases.

It is worth highlighting that, being fully user-centric, our approach does not allow the prediction of items that were never bought by a customer, affecting the performance of our predictor for customers having a short purchase history. This regards the so-called cold start problem, which is common to all recommender systems and refers to the fact that if few or no purchases are available for an item the quality of resulting recommendations is poor [53]. In our case, the cold start problem affects predictions in the first 36 weeks of a user’s purchase history, i.e., the time needed for the number of distinct items bought by a user to stabilize (see Figure 3, top right). A strategy to mitigate the cold start problem related to the user-centric approach could be to build a hybrid approach [54] where we combine TBP with collaborative filtering, allowing us to make predictions for items that are not present in a user’s previous baskets.

A future line of research consists of providing to the customers of a living laboratory [12] an app running TBP and observe whether their purchase behaviors are influenced by the recommendations. Furthermore, we would like to exploit TARS for developing analytical services in other domains, such as mobility data, musical listening sessions and health data. Finally, in line with [55], it would be interesting to study if there is an improvement in the quality of the prediction if the user-centric models are exploited for developing a collective or hybrid predictive approach.

Acknowledgment

This work is partially supported by the European Community’s H2020 Program under the funding scheme “INFRAIA-1-2014-2015: Research Infrastructures” grant agreement 654024, http://www.sobigdata.eu, “SoBigData”. We thank UniCoop Tirreno for providing the data.

References
