This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2872587, IEEE Transactions on Knowledge and Data Engineering

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 *sequentiality* 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, sequentiality, 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

- *R. Guidotti, G. Rossetti and D. Pedreschi are with KDDLab, University of Pisa, Italy. E-mail: {name.surname}@di.unipi.it*
- *L. Pappalardo and F. Giannotti are with KDD Lab, ISTI-CNR, Pisa, Italy. E-mail: {name.surname}@isti.cnr.it*

Manuscript received October 6, 2017; revised Month Day, Year.

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[1].

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 data-based 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 $B_c=\langle b_{t_1}, b_{t_2},\ldots,b_{t_n}\rangle$ the ordered *purchase history* of the baskets (or transactions) of customer c, where $b_{t_i}\subseteq I$ is the basket composition and $t_i\in[t_1,t_n]$ the transaction time. Finally, $\mathcal{B}=\{B_{c_1},\ldots,B_{c_z}\}$ 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^* 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 are typically purchased

1. This work extends "Market Basket Prediction using User-Centric Temporal Annotated Recurring Sequences" presented at ICDM'17 [13].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2872587, IEEE Transactions on Knowledge and Data Engineering

TABLE 1
Example of customer purchase history B_c.

timestamp	basket
01-01	a, b, g, h
01-05	a, c, d
01-09	a, b, e, f, h
01-13	a, b, c, d, h
01-17	c, d, e, f, g
01-21	e, f, g
01-29	a, b, c, g, h
02-02	b, c, d
02-06	a, c, d, e, f, i
02-10	b, e, f, h
02-14	a, b, c, d, e, f, g, h
02-22	a, b, g, h, i

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* $B_c = \langle b_{t_1}, \ldots, b_{t_n} \rangle$*, we call* $S = \langle X, Y \rangle = X \rightarrow Y$ *a* sequence *if the pair of itemsets* $X \subseteq b_{t_h}$ *and* $Y \subseteq b_{t_l}$*,* $X, Y \neq \emptyset$*,* $t_h < t_l$ *and* $\nexists\, S' = X' \rightarrow Y'$*,* $X' \subseteq X \subseteq b_{t'_h}$ *and* $Y' \subseteq Y \subseteq b_{t'_l}$ *such that* $t'_h, t'_l \in (t_h, t_l)$*.* X *and* Y *are called the* head *and the* tail *of the sequence, respectively.*

We denote with $T_S = \langle t_{j_1}, \ldots, t_{j_m} \rangle$ 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'', $S' \vec{\subseteq} S''$ if $X' \subseteq X'' \wedge Y' \subseteq Y''$. 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 $t'_h, t'_l \in (t_h, t_l)$, 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 = \langle$01-05, 01-09, 01-13, 01-29, 02-06, 02-14$\rangle$.

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* $\alpha_h = t_l - t_h$ *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* $A_S = \langle \alpha_1, \ldots, \alpha_m \rangle$ *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* $\delta_j = t_{h_i} - t_{h_j}$ *with* $t_{h_i}, t_{h_j} \in T_S$ *and* $t_{h_j} < t_{h_i}$ *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* $\Delta_S = \langle \delta_1, \ldots, \delta_m \rangle$ *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 $A_S = \langle 4, 4, 16, 4, 4, 8 \rangle$. The inter-time list Δ_S consists of all differences between the head times of two consecutive sequences, hence $\Delta_S = \langle 4, 4, 16, 8, 8, 8 \rangle$. 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 $A_S = \Delta_S$.

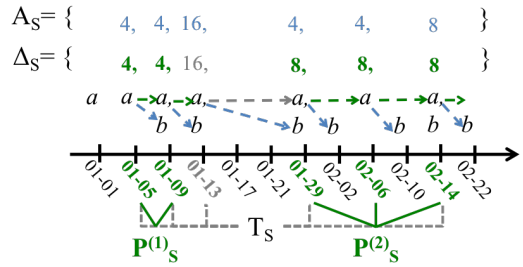


Fig. 1. Head time list T_S, intra-time list A_S, inter-time list Δ_S and periods $P_S^{(1)}$, $P_S^{(2)}$ of sequence $S = \{a\} \rightarrow \{b\}$.

Definition 4 (Period)**.** *Given a maximum inter-time* δ^{max}*, a minimum number of occurrences* q^{min}*, the head time list* T_S *and the inter-time list* Δ_S *of a sequence* S*, we call* period *an ordered time list* $P_S^{(j)} = \langle t_h, \ldots, t_l \rangle \subseteq T_S$ *such that* $\forall t_w \in P_S^{(j)}$*,* $\delta_w \leq \delta^{max}$*,* $P_S^{(j)}$ *is maximal, i.e.,* $\delta_{h-1} > \delta^{max}$*,* $\delta_{l+1} > \delta^{max}$*, and* $|P_S^{(j)}| \geq q^{min}$*. We denote with* $P_S = \{P_S^{(1)}, \ldots, P_S^{(m)}\}$ *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 δ^{max}. The support of a period $|P_S^{(j)}|$ indicates how many times S occurs in $P_S^{(j)}$. Given the previously observed property $\alpha_j \leq \delta_j$ of intra- and inter-times, for a given δ^{max} in the definition of period we have that the inter-time also considers the intra-time. In Figure 1, for δ^{max}=14 and q^{min}=2 we have two periods $P_S^{(1)}$=$\langle$01-05, 01-09$\rangle$ and $P_S^{(2)}$=$\langle$01-29, 02-06, 02-14$\rangle$ with support 2 and 3 respectively.

Definition 5 (Recurring Sequence)**.** *Let* $P_S = \{P_S^{(1)}, \ldots, P_S^{(m)}\}$ *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* p^{min}*,* S *is a* recurring sequence *if* $rec(S) \geq p^{min}$*.*

In the example of Figure 1, for $p^{min} = 2$ 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* $\gamma = (S, \alpha, p, q)$*, 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[2]*. A TARS will also be represented as follows:*

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

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].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2872587, IEEE Transactions on Knowledge and Data Engineering

Algorithm 1: $extractTars(B)$

1 $\mathcal{S} \leftarrow extractBaseSequences(B)$;
2 $\{\delta_S^{max}\}, \{q_S^{min}\}, \{p_S^{min}\} \leftarrow parametersEstimation(B, \mathcal{S})$;
3 $\mathcal{S}^* \leftarrow sequenceFiltering(B, \mathcal{S}, \{\delta_S^{max}\}, \{q_S^{min}\}, \{p_S^{min}\})$;
4 $\Psi \leftarrow buildTars\text{-}Tree(B, \mathcal{S}^*, \{\delta_S^{max}\}, \{q_S^{min}\}, \{p_S^{min}\})$;
5 $\Gamma \leftarrow extractTarsFromTree(\Psi)$;
6 **return** Γ;

A TARS is based on the concept of *sequence*, $S = \langle X, Y \rangle = X \rightarrow Y$, which intuitively indicates that itemset Y is typically purchased after another itemset X. The itemsets themselves point out which items are purchased together[3]. For example, a sequence $\{a\} \rightarrow \{b, c\}$ indicates that $\{b, c\}$ are purchased together after $\{a\}$. The temporal annotation $\alpha = (\alpha_1, \alpha_2)$ 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 recurrences, that is, in how many periods the sequence S appears. Finally, q indicates how many times S typically occurs in a period.

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

We refer to $\Gamma_c = \{\gamma_1, \ldots, \gamma_m\}$ as the set of all the TARS of a customer c. By specifying the maximum inter-time δ^{max}, the minimum number of occurrences q^{min}, and the minimum number of periods p^{min}, we can determine the set Γ_c of TARS that can be extracted from the purchase history B_c.

4.2 TARS Extraction Procedure

To extract the TARS 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_S derived from T_S with respect to δ^{max} and q^{min}.

The TARS extraction procedure is described in Algorithm 1. In the first step, it extracts from the purchase history B the *base sequences* $\mathcal{S}$, i.e., the sequences of length 2 (line 1). Then, it estimates a set of parameters $\{\delta_S^{max}\}, \{q_S^{min}\}, \{p_S^{min}\}$ for each base sequence $S \in \mathcal{S}$ with respect to B (line 2). The base sequences $\mathcal{S}$ are then filtered with respect to these parameters and the base recurring sequences $\mathcal{S}^*$ 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 $\mathcal{S}^*$ (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 $extractTars(\cdot)$ depends not only on the input 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.

Algorithm 2: $parametersEstimation(\mathcal{S}, B)$

1 $D_{\delta^{max}} \leftarrow \emptyset$; $D_{q^{min}} \leftarrow \emptyset$; $D_{p^{min}} \leftarrow \emptyset$;
2 **foreach** $S \in \mathcal{S}$ **do**
  $D_{\delta^{max}} \leftarrow D_{\delta^{max}} \cup \{\hat{\delta_S} = median(\Delta_S)\}$;
3 $\mathcal{C}_{\delta^{max}} \leftarrow groupSimilar(D_{\delta^{max}})$;
4 **for** $C_h \in \mathcal{C}_{\delta^{max}}$ **do**
5 **foreach** S $assignedTo(C_h)$ **do** $\delta_S^{max} \leftarrow median(C_h)$;
6 **for** $S \in \mathcal{S}$ **do**
7 $TC_S \leftarrow getTimeCompliantPeriods(S, B, \{\delta_S^{max}\})$;
8 $D_{q^{min}} \leftarrow D_{q^{min}} \cup \{median(\{\hat{q_S} = |TC_S^{(j)}| \,|\, TC_S^{(j)} \in TC_S\})\}$;
9 $\mathcal{C}_{q^{min}} \leftarrow groupSimilar(D_{q^{min}})$;
10 **for** $C_h \in \mathcal{C}_{q^{min}}$ **do**
11 **foreach** S $assignedTo(C_h)$ **do** $q_S^{min} \leftarrow median(C_h)$;
12 **for** $S \in \mathcal{S}$ **do**
13 $P_S \leftarrow getPeriods(S, B, \{\delta_S^{max}\}, \{q_S^{min}\})$;
14 $w_S \leftarrow \sum_{P_S^{(j)} \in P_S} |P_S^{(j)}|$; $e_S \leftarrow w_S / |P_S|$; $D_{p^{min}} \leftarrow D_{p^{min}} \cup \{e_S\}$;
15 $\mathcal{C}_{p^{min}} \leftarrow groupSimilar(D_{p^{min}})$;
16 **for** $C_h \in \mathcal{C}_{p^{min}}$ **do**
17 **for** S $assignedTo(C_h)$ **do**
18 $p_S^{min} \leftarrow median(\{rec(P_{S'}) = |P_{S'}| S' assignedTo(C_h)\})$;
19 **return** $\{\delta_S^{max}\}, \{q_S^{min}\}, \{p_S^{min}\}$;

4.2.1 Data-Driven Parameters Estimation

In order to make the parameters δ^{max}, q^{min}, p^{min} 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 $\mathcal{S}$ (lines 1–2 Algorithm 1).

3. We consider only sequences with two itemsets (i.e., $X \rightarrow Y$) and not with more itemsets (i.e., $X \rightarrow Y \rightarrow Z$) because of two main reasons. The first one is consequence of the purpose of the definition of TARS: we want to use them for performing prediction, thus in our modeling we only need a head and a tail to use for calculating the items' rank (see Algorithm 4 for details). The second one is that the recursion of a sequence p (number of periods in which it occurs) and the occurrences in each period, try to capture repetitions in the purchasing behavior that last longer than two purchases even considering seasonal trends.

The first pre-processing step is the data-driven estimation of the sets of parameters $\{\delta_S^{max}\}$, $\{q_S^{min}\}$, $\{p_S^{min}\}$ described in Algorithm 2. Let $\mathcal{S}$ be the set of base sequences and $\hat{\delta_S}$ be the median of inter-times in Δ_S (Algorithm 2, line 2). Given a base sequence S, we estimate parameter δ^{max} as follows: *(i)* we group the base sequences with similar inter-times $\hat{\delta_S}$ (line 3) obtaining a set of clusters $\mathcal{C}_{\delta^{max}}=\{C_1,\ldots,C_v\}$; *(ii)* if $S \in C_h$, $C_h \in \mathcal{C}_{\delta^{max}}$, we set δ_S^{max} as the median of the $\hat{\delta_S}$ values in cluster C_h (lines 4–5).

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

Similarly, we estimate $\{p_S^{min}\}$ 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 $\mathcal{C}_{p^{min}} = \{C_1,\ldots,C_d\}$ (line 15); and *(iii)* if $S \in C_h$, $C_h \in \mathcal{C}_{p^{min}}$, we set p_S^{min} as the median of the number of periods of the base sequences in C_h (lines 16–17).

We group the base sequences, $groupSimilar(\cdot)$ in Algorithm 2, by dividing the values into equal-sized bins [43]. Each bin corresponds to a group containing similar values[4].

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 $\{\delta_S^{max}\}$, $\{q_S^{min}\}$, $\{p_S^{min}\}$. We apply this filtering to reduce the search space so that the building of the TARS-tree and the TARS extraction (lines 4–5 Algorithm 1) are employed only on the super-sequences of the base recurring sequences[5]. In other words, if S_1 is not a base recurring sequence and $S_1 \subseteq S_2$, 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 $S_1 \vec{\subseteq} S_2$. Given $\delta^{max} = 14$, $q^{min} = 2$ and $p^{min} = 2$, we have $rec(S_1) = 1$ and $rec(S_2) = 2$. 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 $rec(S_1) = 1$ in total it occurs six times $|P_{S_1}^{(1)}| = 6$. 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 known if the patterns discarded by sequence filtering and by FP-Growth using the sets of parameters $\{\delta_S^{max}\}, \{q_S^{min}\}, \{p_S^{min}\}$ could be potentially useful for the prediction. However, without a filtering of the search space TARS extraction would becomes practically intractable.

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

```
1  Γ̂ ← ∅; Q ← ∅; L ← ∅; ϒ ← Γ;
2  for b_{t_j}, b_{t_{j-1}} ∈ sort-desc(B) do
3      α_{j-1} ← t_j − t_{j-1};
4      for X ⊆ b_{t_{j-1}} do
5          for Y ⊆ b_{t_j} do
6              if ∃ γ ∈ ϒ | γ = (S, α, p, q) ∧
                 α_1 ≤ α_{j-1} ≤ α_2 ∧   S = ⟨X, Y⟩ = X → Y
                 then
7                  if γ ∈ Γ̂ then
8                      Q_γ ← Q_γ + 1; L_γ ← t^{j-1};
9                      if Q_γ > q then Γ̂ ← Γ̂/{γ};
                         ϒ ← ϒ/{γ};
10                     if L_γ − t_{j-1} > q · (α_1-α_2) then
                         ϒ ← ϒ/{γ};
11                 else
12                     Γ̂ ← Γ̂ ∪ {γ}; Q_γ ← 1; L_γ ← t_{j-1};
13             if ϒ = ∅ then return Γ̂, Q;
14 return Γ̂, Q;
```

For these reasons, we developed the pre-processing steps for parameters estimation described in this section.

5 TARS Based Predictor

On top of the set Γ_c 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 Γ_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 Γ_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 Γ_c, TBP works in two steps. First, it selects the set $\hat{\Gamma}_c$ of *active* TARS. Second, it computes a score Ω_{c_i} for every item i belonging to an active TARS in $\hat{\Gamma}_c$, ranks the items according to Ω_{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 Υ of *potentially active* TARS (lines 4–7). When it finds a potentially active TARS γ, it considers two cases. If the sequence S of γ is encountered for the first time, the algorithm adds γ to the set $\hat{\Gamma}$ of active TARS and initializes

Algorithm 4: $calculateItemScore(B, \hat{\Gamma}, Q)$

1 $\Omega \leftarrow \emptyset$; **foreach** $i \in I$ **do** $\Omega_i \leftarrow 0$;
2 **for** $\gamma = (S = \langle X, Y \rangle, \alpha, p, q) \in \hat{\Gamma}$ **do**
3 $\quad$ **foreach** $i \in Y$ **do** $\Omega_i \leftarrow \Omega_i + (q - Q_\gamma)$;
4 **for** $i \in \{i \mid \exists\, \gamma = (S = \langle X, Y \rangle, \alpha, p, q) \in \hat{\Gamma}, i \in Y\}$ **do**
5 $\quad$ $\Omega_i \leftarrow \Omega_i + sup(i)$
6 **return** Ω;

two variables: the number of times γ has been encountered Q_γ and its last starting time L_γ (line 13). In the second case, the algorithm increments Q_γ and updates L_γ (line 9). If $Q_\gamma > q$ the algorithm removes γ from the set of active TARS and from the set of potentially active TARS (line 9). If too much time has passed between the last beginning of TARS γ and its next occurrence (line 11), the algorithm does not look for that TARS γ anymore and removes it from Υ. Algorithm 3 stops either when the set of potentially active TARS is empty (line 14), or when the entire purchase history B has been scanned (line 15). Finally, it returns the set $\hat{\Gamma}$ of active TARS and the number of times Q the sequences of the active TARS have occurred in the last period.

Algorithm 4 shows the procedure of TBP to compute the items' scores. First, it sets to zero the score of each item Ω_i (line 1). Then, for every active TARS γ containing item $i \in Y$, it increases Ω_i with the difference between the typical number of occurrences q of γ and Q_γ indicating the number of times that the sequence of γ occurred in the recent history (lines 2–3). Finally, Ω_i is augmented with the support of item i for the items in the tail of the active TARS (lines 4–5). In other words, each item in the consequent of an active TARS gets an higher score Ω_i if the TARS it belongs to is going to be repeated in its recurring period. The overall importance of an item i for a customer c is also considered augmenting the score with its support. Therefore, intuitively Ω_i is high if i is going to be re-purchased as consequence of previous purchases in the next shopping session. After this procedure, TBP ranks the items' scores Ω_c in descending order and returns the top-k items as prediction.

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[6]. 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.

6. 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/N6UhnM. 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].

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 $B_c = \{b_{t_1}, \ldots, b_{t_n}\}$ for extracting the TARS, and the basket $b_{t_{n+1}}$ 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]:

$$F1\text{-}score(b, b^*) = \frac{2 \cdot Precision(b, b^*) \cdot Recall(b, b^*)}{Precision(b, b^*) + Recall(b, b^*)}$$

$$Precision(b, b^*) = |b \cap b^*|/|b^*|$$

$$Recall(b, b^*) = |b \cap b^*|/|b|$$

- *Hit-Ratio*, the ratio of customers who received at least one correct prediction (a *hit*) [48]:

$$Hit\text{-}Ratio(b, b^*) = 1 \text{ if } b \cap b^* \neq \emptyset, 0 \text{ otherwise.}$$

- *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 NDCG 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.

TABLE 2
Statistics of the datasets used in the experiments.

Dataset	*cust.*	*# baskets*	*# items*	*avg basket per cust.*	*avg basket length*
Coop-A	10,000	7,407,056	4,594	432.4±353.4	9.4±5.8
Coop-C	10,000	7,407,056	407	432.4±353.4	8.6±4.9
Ta-Feng	2,319	24,304	5,117	10.4±7.5	1.8±1.1

6.2 Datasets

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

The *Coop* repository is provided by *Unicoop Tirreno*[7], 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: *Coop-A* and *Coop-C*. The two datasets differ in the items categorization. In ***Coop-A*** (articles) the items of a basket are labeled with a fine-grained categorization which distinguishes, for example, between blood orange and navel orange. In ***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 *Coop-A* and *Coop-C* have at least one purchase per month.

Ta-Feng[8] 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 *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 *Coop-C*. In the table, we report the median of α, 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 $\{milk\} \rightarrow \{milk\}$ and $\{banana\} \rightarrow \{banana\}$ are supported by more than 90% of the customers in *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 $\{banana\} \rightarrow \{banana\}$ 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 $\{banana\}$ than $\{milk\}$.

7. https://www.unicooptirreno.it/
8. http://www.bigdatalab.ac.cn/benchmark/bm/dd?data=Ta-Feng

TABLE 3
Examples of TARS extracted from *Coop-C*.

- Supported by more than 90% customers

$\{\text{milk}\} \xrightarrow[18.87,6.58]{(1,17)} \{\text{milk}\}$ $\{\text{banana}\} \xrightarrow[14.63,7.20]{(2,35)} \{\text{banana}\}$

- Supported by more than 80% customers

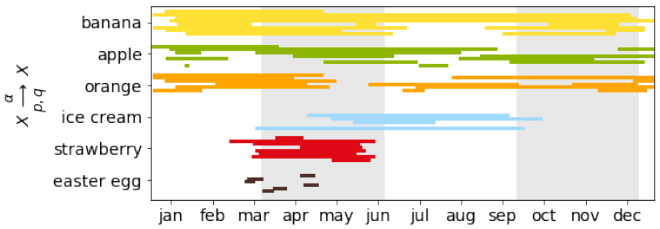
$\{\text{tomato}\} \xrightarrow[13.87,6.58]{(1,17)} \{\text{milk}\}$ $\{\text{tomato}\} \xrightarrow[15.27,5.11]{(1,12)} \{\text{bovine}\}$

- Supported by more than 25% customers

$\{\text{bread, potato}\} \xrightarrow[11.40,8.15]{[2,15]} \{\text{bovine}\}$ $\{\text{bread, potato}\} \xrightarrow[7.25,4.30]{[3,27]} \{\text{banana, potato}\}$

TABLE 4
Periods of TARS with different recurring base sequences from *Coop-C*. For each TARS is shown how the periods, represented as horizontal single lines, occur along 7 years of observations.

X	$\rightarrow$	X	α_1	α_2	p	q
{banana}	$\rightarrow$	{banana}	2	35	14.63	7.20
{apple}	$\rightarrow$	{apple}	2	35	15.90	6.14
{orange}	$\rightarrow$	{orange}	2	33	8.13	6.56
{ice cream}	$\rightarrow$	{ice cream}	2	40	5.90	6.38
{strawberry}	$\rightarrow$	{strawberry}	2	32	3.55	4.69
{easter egg}	$\rightarrow$	{easter egg}	4	20	2.42	3.29



Moreover, we notice for more than 25% of the customers the contemporary purchase $\{bread, tomato\}$ can indicate a future basket with $\{bovine\}$ or with $\{banana, potato\}$ and that these TARS have very different annotations α, p, q. Finally, we highlight that, even if the most common TARS among the customers are those with base sequences, the TARS in Γ_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.

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

In contrast, *oranges* are a seasonal fruit item, generally available between November and February. The associated base TARS $\{orange\} \rightarrow \{orange\}$ 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 $\{ice\ cream\} \rightarrow \{ice\ cream\}$ has a lower p and a higher maximum intra-time α_2.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2872587, IEEE Transactions on Knowledge and Data Engineering

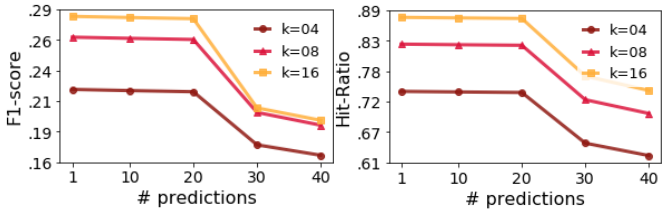


Fig. 2. Evaluation of TARS temporal validity with respect of F1-score (left) and Hit-Ratio (right) on *Coop-C* varying the number k of items.

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 α_2 indicating short periods, while Easter eggs have the highest α_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.

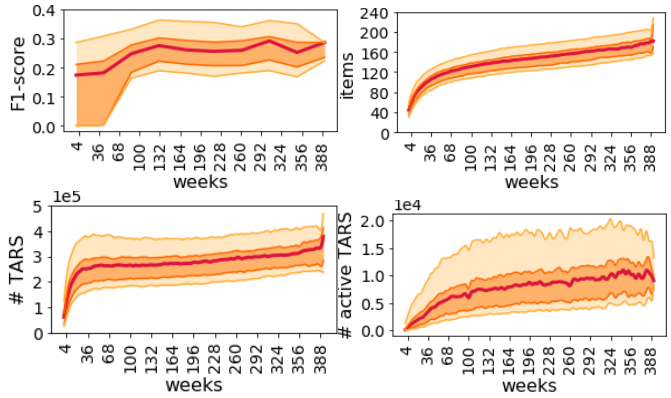


Fig. 3. Evaluation of TARS reliability on *Coop-C* observing F1-score (top left), number of items (top right), number of TARS (bottom left) and number of active TARS (bottom right) by augmenting the size of the purchase history B_c for each customer analyzed. In the plots, the median value (the red line) is shown together with the lower and upper quartiles of the distribution.

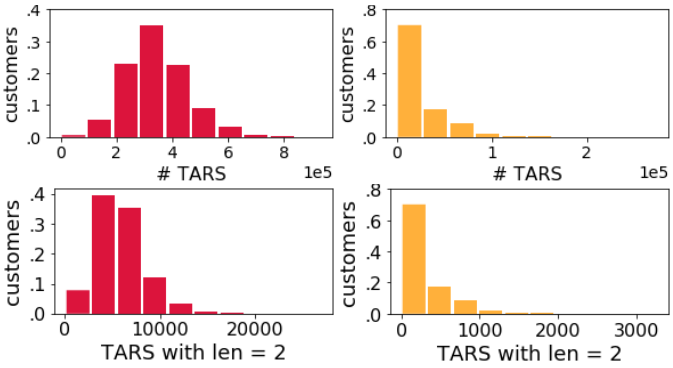


Fig. 4. Number of TARS per customer distribution on *Coop-C*: parameter-free (left) versus parameter-fixed (right) TARS extraction. The bottom line reports a focus of the distributions of the base TARS, i.e., TARS with length equals to 2.

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 δ^{max}=14 (e.g., two weeks), q^{min}=3 and p^{min}=2.

Figure 4 shows the distributions of the number of TARS per customer for the parameter-free (left) and parameter-fixed (right) scenarios. We observe two different distributions: a skewed peaked distribution for the parameter-free scenario and a heavy tail distribution for the parameter-fixed scenario. This suggests that fixing the parameters has a strong impact on the extraction of TARS, leading to a lower average number of TARS per customer than the parameter-free scenario (Figure 4).

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2872587, IEEE Transactions on Knowledge and Data Engineering

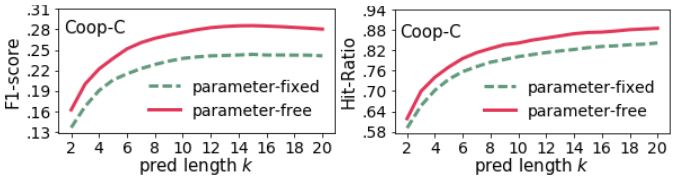


Fig. 5. Next basket prediction performance on *Coop-C* observing F1-score (left) and Hit-Ratio (right) comparing the parameter-free proposed approach versus the same approach with the parameters fixed.

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., $b_{t_{n+1}} = b_{t_n}$;

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_{t_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_{t_j} 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 $\mathcal{B}$ 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 $\mathcal{B}$;

FMC (*Factorizing personalized Markov Chain*) [25]: using the purchase history of all the customers $\mathcal{B}$, 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 $\mathcal{B}$ 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 $\mathcal{B}$.

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 5
F1-score *(F1)* and Hit-Ratio *(HR)* using personalized length $k = k_c^*$. In **bold**, and ***bold-italic*** are highlighted the 1st and 2nd best performer.

$k = k_c^*$		TBP	TOP	MC	CLF	LST	NMF	FPM	HRM	DRM
F1	*Coop-A*	**.17**	***.14***	***.14***	.13	.09	***.14***	.08	.06	.05
	Coop-C	**.24**	.22	***.23***	.19	.14	.22	.16	.08	.12
	Ta-Feng	**.09**	**.09**	.06	**.09**	.06	***.08***	***.08***	***.08***	.07
HR	*Coop-A*	**.62**	.58	.58	.56	.40	***.59***	.44	.35	.33
	Coop-C	**.72**	***.71***	.70	.65	.50	***.71***	.61	.38	.55
	Ta-Feng	**.32**	**.34**	.24	.31	.15	.31	.31	.31	.29

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2872587, IEEE Transactions on Knowledge and Data Engineering

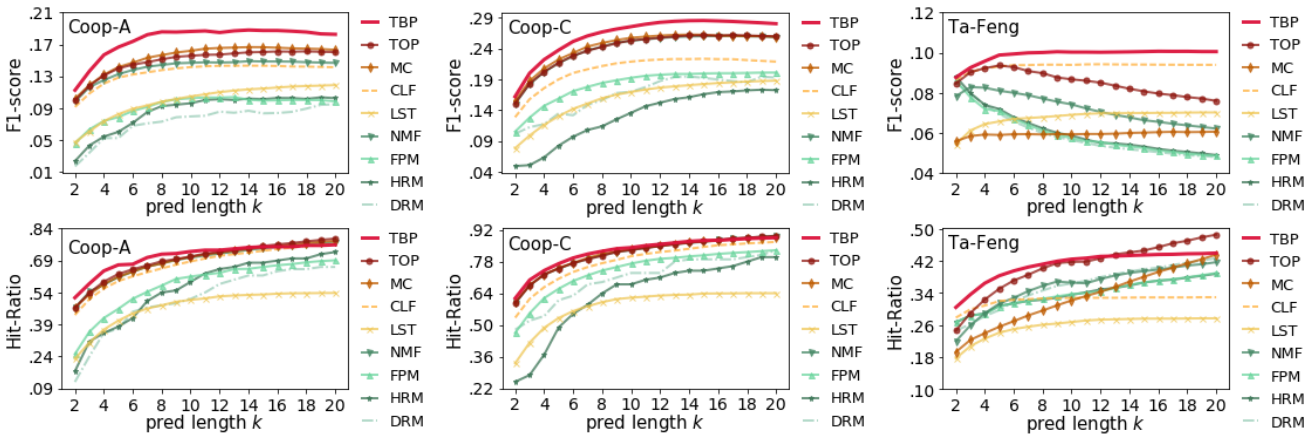


Fig. 6. Performance comparison of TBP against the baselines varying length k: F1-score in the top row, Hit-ratio in the bottom row.

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.

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

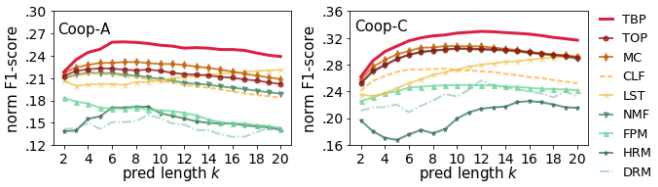


Fig. 7. Normalized F1-score varying predicted basket length k.

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 predicted is correct. 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 $k = k_c^*$ 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.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2872587, IEEE Transactions on Knowledge and Data Engineering

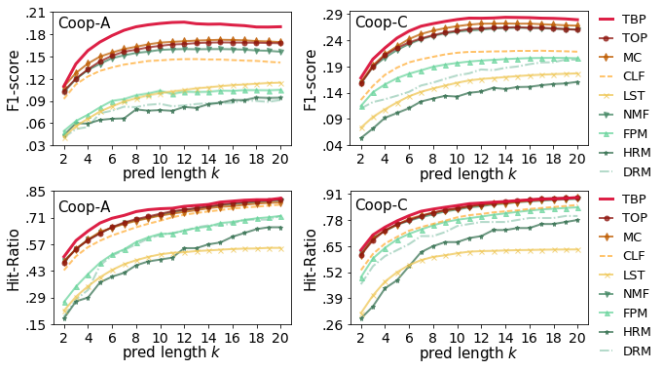


Fig. 8. Performance comparison varying k and using a model built on a subset of B_c with random size between 70% and 90% of $|B_c|$: F1-score in the top row, Hit-ratio in the bottom row.

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 $B'_c = \{b_{t_1}, \ldots, b_{t_{n'}}\}$ of the customers' purchase history $B_c = \{b_{t_1}, \ldots, b_{t_n}\}$, 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 customer 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.

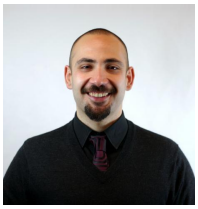
References

[1] B. Mittal and W. M. Lassar, "The role of personalization in service encounters," *Journal of retailing*, vol. 72, no. 1, pp. 95–109, 1996.
[2] A. Dagher, "Shopping centers in the brain," *Neuron*, vol. 53, no. 1, pp. 7–8, 2007.
[3] B. Knutson *et al.*, "Neural predictors of purchases," *Neuron*, vol. 53, no. 1, pp. 147–156, 2007.
[4] R. Guidotti, M. Coscia *et al.*, "Behavioral entropy and profitability in retail," in *DSAA*. IEEE, 2015, pp. 1–10.
[5] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini *et al.*, "A survey of methods for explaining black box models," *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, p. 93, 2018.
[6] R. Guidotti, J. Soldani *et al.*, "Explaining successful docker images using pattern mining analysis," *STAF. Springer*, 2018.
[7] A. Pentland *et al.*, "Personal data: The emergence of a new asset class," in *An Initiative of the World Economic Forum*, 2011.
[8] C. Kalapesi, "Unlocking the value of personal data: From collection to usage," in *World Economic Forum technical report*, 2013.
[9] R. Guidotti, "Personal data analytics: Capturing human behavior to improve self-awareness and personal services through individual and collective knowledge," 2017.
[10] R. Guidotti, A. Monreale, M. Nanni, F. Giannotti, and D. Pedreschi, "Clustering individual transactional data for masses of users," in *SIGKDD*. ACM, 2017, pp. 195–204.
[11] Y.-A. de Montjoye, E. Shmueli, S. S. Wang, and A. S. Pentland, "openpds: Protecting the privacy of metadata through safeanswers," *PloS one*, vol. 9, no. 7, p. e98790, 2014.
[12] M. Vescovi, C. Perentis, C. Leonardi, B. Lepri, and C. Moiso, "My data store: toward user awareness and control on personal data," in *Ubicomp*. ACM, 2014, pp. 179–182.

[13] R. Guidotti, G. Rossetti, L. Pappalardo *et al.*, "Market basket prediction using user-centric temporal annotated recurring sequences," in *ICDM*. IEEE, 2017, pp. 895–900.
[14] A. Mild and T. Reutterer, "An improved collaborative filtering approach for predicting cross-category purchases based on binary market basket data," *JRCS*, vol. 10, no. 3, pp. 123–133, 2003.
[15] K. Christidis *et al.*, "Exploring customer preferences with probabilistic topic models," in *ECML-PKDD*, 2010.
[16] G. Linden *et al.*, "Amazon. com recommendations: Item-to-item collaborative filtering," *IC*, vol. 7, no. 1, pp. 76–80, 2003.
[17] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *ICDM*. IEEE, 2008, pp. 263–272.
[18] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, p. 4, 2009.
[19] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *TKDE*, vol. 17, no. 6, pp. 734–749, 2005.
[20] C. Chand, A. Thakkar, and A. Ganatra, "Sequential pattern mining: Survey and current research challenges," *International Journal of Soft Computing and Engineering*, vol. 2, no. 1, pp. 185–193, 2012.
[21] C.-N. Hsu, H.-H. Chung, and H.-S. Huang, "Mining skewed and sparse transaction data for personalized shopping recommendation," *Machine Learning*, vol. 57, no. 1-2, pp. 35–59, 2004.
[22] E. Lazcorreta *et al.*, "Towards personalized recommendation by two-step modified apriori data mining algorithm," *Expert Systems with Applications*, vol. 35, no. 3, pp. 1422–1429, 2008.
[23] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, "Effective personalization based on association rule discovery from web usage data," in *WIDM*. ACM, 2001, pp. 9–15.
[24] R. Agrawal, R. Srikant *et al.*, "Fast algorithms for mining association rules," in *VLDB*, vol. 1215, 1994, pp. 487–499.
[25] S. Rendle *et al.*, "Factorizing personalized markov chains for next-basket recommendation," in *WWW*. ACM, 2010, pp. 811–820.
[26] ——, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*. AUAI Press, 2009, pp. 452–461.
[27] P. Wang *et al.*, "Learning hierarchical representation model for nextbasket recommendation," in *SIGIR*. ACM, 2015, pp. 403–412.
[28] F. Yu *et al.*, "A dynamic recurrent model for next basket recommendation," in *SIGIR*. ACM, 2016, pp. 729–732.
[29] P. Wang *et al.*, "Modeling retail transaction data for personalized shopping recommendation," in *CIKM*. ACM, 2014, pp. 1979–1982.
[30] J. Rose and C. Kalapesi, "Rethinking personal data: strengthening trust," *BCG Perspectives*, vol. 16, no. 05, p. 2012, 2012.
[31] C. Cumby *et al.*, "Predicting customer shopping lists from point-of-sale purchase data," in *KDD*. ACM, 2004, pp. 402–409.
[32] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *KDD*. New York, NY, USA: ACM, 2016, pp. 1135–1144.
[33] W. Mendenhall, R. J. Beaver, and B. M. Beaver, *Introduction to probability and statistics*. Cengage Learning, 2012.
[34] R. U. Kiran, H. Shang, M. Toyoda, and M. Kitsuregawa, "Discovering recurring patterns in time series." in *EDBT*, 2015, pp. 97–108.
[35] F. Giannotti *et al.*, "Efficient mining of temporally annotated sequences." in *SDM*. SIAM, 2006, pp. 348–359.
[36] J. Han *et al.*, "Mining frequent patterns without candidate generation," in *Sigmod*, vol. 29, no. 2. ACM, 2000, pp. 1–12.
[37] K. Amphawan, A. Surarerks, and P. Lenca, "Mining periodic-frequent itemsets with approximate periodicity using interval transaction-ids list tree," in *WKDD*. IEEE, 2010, pp. 245–248.
[38] P. Fournier-Viger *et al.*, "Phm: mining periodic high-utility itemsets," in *ICDM*. Springer, 2016, pp. 64–79.
[39] R. U. Kiran and M. Kitsuregawa, "Finding periodic patterns in big data," in *ICBDA*. Springer, 2015, pp. 121–133.
[40] W. A. Kosters *et al.*, "Complexity analysis of depth first and fp-growth implementations of apriori," in *MLDM*. Springer, 2003, pp. 284–292.
[41] B. Schlegel, R. Gemulla, and W. Lehner, "Memory-efficient frequent-itemset mining," in *EDBT*. ACM, 2011, pp. 461–472.
[42] E. Keogh, S. Lonardi, and C. A. Ratanamahatana, "Towards parameter-free data mining," in *KDD*. ACM, 2004, pp. 206–215.
[43] K. Pearson, "Contributions to the mathematical theory of evolution," *PTRSL*, vol. 185, pp. 71–110, 1894.
[44] H. A. Sturges, "The choice of a class interval," *Journal of the american statistical association*, vol. 21, no. 153, pp. 65–66, 1926.
[45] D. Freedman and P. Diaconis, "On the histogram as a density estimator: L 2 theory," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 57, no. 4, pp. 453–476, 1981.
[46] R. Agrawal *et al.*, "Mining association rules between sets of items in large databases," in *SIGMOD*, vol. 22. ACM, 1993, pp. 207–216.
[47] P.-N. Tan *et al.*, *Introduction to data mining*. *Pearson Education*, 2006.
[48] G. Karypis, "Evaluation of item-based top-n recommendation algorithms," in *CIKM*. ACM, 2001, pp. 247–254.
[49] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM TOIS*, vol. 20, no. 4, pp. 422–446, 2002.
[50] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, 2001, pp. 556–562.
[51] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *JMLR*, vol. 7, no. Jan, pp. 1–30, 2006.
[52] A. Monreale *et al.*, "Privacy-by-design in big data analytics and social mining," *EPJ Data Science*, p. 10, 2014.
[53] B. Lika *et al.*, "Facing the cold start problem in recommender systems," *ESA*, vol. 41, no. 4, pp. 2065–2073, 2014.
[54] J. Salter and N. Antonopoulos, "Cinemascreen recommender agent: combining collaborative and content-based filtering," *IEEE Intelligent Systems*, vol. 21, no. 1, pp. 35–41, Jan 2006.
[55] R. Trasarti, R. Guidotti, A. Monreale, and F. Giannotti, "Myway: Location prediction via mobility profiling," *Information Systems*, vol. 64, pp. 350–367, 2017.

Riccardo Guidotti Riccardo Guidotti was born in 1988 in Pitigliano (GR) Italy. He received his PhD in Computer Science at University of Pisa with a thesis on "Personal Data Analytics". He is currently a post-doc researcher in the same institution and a member of the KDDLab, a joint research group with the ISTI-CNR in Pisa. He won the IBM fellowship and has been an intern in IBM Research Dublin in 2015. His research interests are in personal analytics, clustering, explainable models, analysis of transactional data.

Giulio Rossetti Giulio Rossetti earned his PhD in Computer Science at University of Pisa with the thesis "Social Network Dynamics". He is a member of the Knowledge Discovery and Data Mining Laboratory, a joint research group with the ISTI-CNR in Pisa. His research interests include big data analytics, dynamic networks analysis and science of success.



Luca Pappalardo Luca Pappalardo earned his PhD in Computer Science at University of Pisa with the thesis "Human Mobility, Social Networks and Economic Development: a Data Science perspective". His research focuses on the analysis of Big Data for the modeling of complex social phenomena, such us human mobility, social networks dynamics and sports performance evaluation.

Fosca Giannotti Fosca Giannotti is a senior researcher at ISTI-CNR, Pisa, Italy, where she leads the KDDLab a joint research initiative with the University of Pisa, founded in 1995. Her current research interests include data mining, knowledge discovery, spatio-temporal data mining, and privacy. She is currently the co-ordinator of the SoBigData Research Infrastructures for Social Mining & Big Data Ecosystem.

Dino Pedreschi Dino Pedreschi is a C.S. Professor at the University of Pisa, and a pioneering scientist in mobility data mining, social network mining and privacy-preserving data mining. He co-leads the Pisa KDDLab, one of the earliest research lab on data mining. His research focus is on big data analytics and their impact on society. He is a founder of the Business Informatics MSc at University of Pisa. In 2009, Dino received a Google Research Award.