Market Basket Prediction using User-Centric Temporal Annotated Recurring Sequences

Riccardo Guidotti*, Giulio Rossetti*§, Luca Pappalardo*§, Fosca Giannotti*, Dino Pedreschi§
* KDD Lab, ISTI - CNR, Via Giuseppe Moruzzi, 1, Pisa, Italy, {name.surname}@isti.cnr.it
§ KDD Lab, University of Pisa, 3, Pisa, Italy, {name.surname}@di.unipi.it

***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 named *Temporal Annotated Recurring Sequence* (*TARS*). 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. A deep experimentation shows that TARS can explain the customers' purchase behavior, and that TBP outperforms the state-of-the-art competitors.**

I. Introduction

Detecting the purchase habits of customers and their evolution in time is a crucial challenge for effective marketing policies and engagement strategies. In such context one of the most promising facilities retail markets can offer to their customers is *market 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 general and recent behavior [1]. In fact, 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 next basket prediction must be *adaptive* to the evolution of a customer's behavior, the recurrence of her purchase patterns and their periodic changes.

In this paper we propose the *Temporal Annotated Recurring Sequences* (*TARS*), adaptive patterns which model the purchasing behavior of an individual by four main characteristics. Firstly TARS consider the *co-occurency*: 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.

We exploit the TARS and the multiple factors they are able to capture for constructing a parameter-free *TARS Based Predictor* (*TBP*). TBP is able to solve the market basket prediction problem and to provide a reliable list of items to be reminded in the next purchase as basket recommendation.

We demonstrate the effectiveness of our approach by extracting the TARS for thousands of customers in three real-world datasets. We show how TARS are easily readable and interpretable, a characteristic which allows gaining useful insights about the purchasing patterns of products and customers. Then, we implement a repertoire of state-of-the-art methods and compare them with TBP. Our results show that *(i)* TBP outperforms the state-of-the-art methods, *(ii)* it is able to predict up to the next 20 baskets, and *(iii)* the quality of its predictions stabilizes after about 36 weeks.

II. Related Work

Next basket prediction is mainly aimed at the construction of effective recommender systems. They can be categorized into *general*, *sequential*, *pattern-based* and *hybrid* recommenders. General recommenders are based on collaborative filtering and produce recommendations with respect to general customers' preferences [2]. Sequential recommenders are based on Markov chains and produce recommendations exploiting sequential information and recent purchases [3]. Pattern-based recommenders base predictions on frequent itemsets extracted from the purchase history of all customers while discarding sequential information [4], [5]. The hybrid approaches combine the ideas underlying general and sequential recommenders. In [6] the authors use personalized transition graphs over Markov chains with Bayesian Personalized Ranking to compute the probability that a customer will purchase an item. HRM [7] and DREAM [8] exploit both general customers' preferences and sequential information by using recurrent neural networks. A different hybrid approach merging Markov chain and association patterns is described in [9].

All the approaches described above suffer from several limitations. General recommenders and pattern-based recommenders do not take into account neither the sequential information nor the customers' recency. On the other hand, sequential recommenders assume the independence of items in the same basket and do not capture factors like mutual influence. Furthermore, all of them require transactional data about many customers in order to make a prediction for a

2374-8486/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDM.2017.111

single customer. For this reason, they do not follow the *user-centric* vision for data protection as promoted by the World Economic Forum [10], [11], which incentives personal data management for every single user of a data-based service. Cumby et al. [12] propose a basket predictor which embraces the user-centric vision by reformulating next basket prediction as a classification problem. However, also this approach also assumes the independence of items purchased together.

Finally, the main drawback of the existing hybrid approaches [8], [7], [9] is that their predictive models are hardly readable and interpretable by humans [13]. The interpretability is valuable both for a retail chain manager, who is interested in interpreting the predictive model to improve the marketing strategies, and the customers who want to gain insights about their personal purchasing behavior.

III. Market Basket Prediction Problem

We refer to *market basket prediction* as the task of predicting which items a customer will purchase in her next transaction. Formally, 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. Given a customer c, $B_c = \langle b_{t_1}, b_{t_2}, \ldots, b_{t_n} \rangle$ is the ordered *purchase history* of her baskets (or transactions), where $b_{t_i} \subseteq I$ represents the basket composition and $t_i \in [t_1, t_n]$ is the transaction time. We indicate with $\mathcal{B} = \{B_{c_1}, B_{c_2}, \ldots, B_{c_z}\}$ 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 II. 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.

IV. Capturing Purchasing Habits

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

Temporal Annotated Recurring Sequences (*TARS*) model recurrent and sequential purchases of a customer – i.e., the fact that a set of items are typically purchased together and that a set of items is typically purchased after another set of items – and the recurrence of the sequential purchase – i.e., when and how often such pattern occurs in the purchase history of the customer. In order to understand how TARS capture all these features at the same time, we need to define its components.

Definition 1 (Sequence)**.** *Given the purchase history of a customer $B_c = \langle b_{t_1}, \ldots, b_{t_n} \rangle$, we call $S = \langle X, Y \rangle = X \to Y$ 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 $\nexists\, S' = X' \to Y'$, $X' \subseteq X \subseteq b_{t'_h}$ 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 B_c. 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 $X' \subseteq X'' \wedge Y' \subseteq Y''$.

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 $\delta_j = t_{l_i} - t_{l_j}$ with $t_{l_i}, t_{l_j} \in T_S$ and $t_{l_j} < t_{l_i}$ 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.*

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 $S = X \to X$, we have $A_S = \Delta_S$.

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 $|P_S^{(j)}|$ of a period indicates how many times S occurs in $P_S^{(j)}$.

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 purchase history. Given a minimum number of periods p^{min}, S is a* recurring sequence *if $rec(S) \geq p^{min}$.*

In summary, a sequence captures items which are purchased together and after other items, the period of a sequence is a time list respecting intra and inter time constraints, and a recurring sequence is a sequence appearing in a certain number of periods. Given these basic components, we define a TARS as:

Definition 6 (Temporal Annotated Recurring Sequence)**.** *Given the purchase history B of a customer, a* temporally annotated recurring sequence *(*TARS*) is a quadruple $\gamma = (S, \alpha, p, q)$, where $S = \langle X, Y \rangle = X \to Y$ 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*

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;

recurs, and q is the median of the number of occurrences in each period. *A TARS will also be represented as follows:*

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

We refer to $\Gamma_c = \{\gamma_1, \ldots, \gamma_m\}$ as the set of all the TARS of a customer c. 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. 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 *intra-time* 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 S appears. Finally, q indicates how many times S typically occurs in a period.

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 a customer c.

To extract the TARS from a customer's purchase history B_c we use an extension of the well-known *FP-Growth* algorithm [14]. FP-Growth builds a *FP-tree* which captures the frequency at which itemsets occur in the dataset. It has been shown in the literature [15], [16], [17] 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 w.r.t. 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, a set of parameters $\{\delta_S^{max}\}, \{q_S^{min}\}, \{p_S^{min}\}$ is estimated 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.

Data-Driven Parameters Estimation. In order to make parameters δ^{max}, q^{min}, p^{min} adaptive not only to the individual customer [18], but also to the sequences in B_c, we apply two pre-processing steps on $\mathcal{S}$ (lines 1–2 Algorithm 1).

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** $\{\delta_S^{max}\}, \{q_S^{min}\}, \{p_S^{min}\}$;

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 (line 2). Given a base sequence S, δ^{max} is estimated 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 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 $\{q_S^{min}\}$ as follows: *(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 $\{p_S^{min}\}$ (lines 12–18).

We group the base sequences by dividing the values into equal-sized bins [19], whose number is estimated as the maximum between the estimated number of bins suggested by the Sturges [20] and the Freedman-Diaconis methods [21].

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. In other words, if S_1 is not a base recurring sequence and $S_1 \vec{\subseteq} S_2$, then we assume as a heuristic that S_2 is not recurring too, and we eliminate it through the sequence filtering process. We adopt the sequence filtering heuristic for reducing the search space because the *antimonotonic property* [22] does not apply to TARS.

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;
```

V. TARS Based Predictor

On top of the set Γ_c of TARS extracted from the purchase history B_c of customer c we build the *TARS Based Predictor* (*TBP*), an approach for market basket prediction that is markedly *personalized* and *user-centric* [11], [10]: 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.

Given the purchasing history B_c of customer c, the time t_{n+1} of c's next transaction, and c's TARS set Γ_c, the TBP approach 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 the procedure of the TBP to select the *active* TARS of a customer $\hat{\Gamma}$. First, it sorts the purchase history B ordering it chronologically 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 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

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

```
1  Ω ← ∅; foreach i ∈ I do Ω_i ← 0;
2  for γ = (S = ⟨X, Y⟩, α, p, q) ∈ Γ̂ do
3      foreach i ∈ Y do Ω_i ← Ω_i + (q − Q_γ);
4  for i ∈ {i | ∃ γ = (S = ⟨X, Y⟩, α, p, q) ∈ Γ̂, i ∈ Y} do
5      Ω_i ← Ω_i + sup(i)
6  return Ω;
```

(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).

After this procedure, TBP ranks the items' scores Ω_c in descending order and returns the top-k items as its prediction.

VI. 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.

A. Experimental Settings

State-of-the-art methods [6], [7], [8], [12] 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. Indeed, 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 [8], [7], [6], [12], we adopt a *leave-one-out* strategy for model validation: for each customer c we use 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*, the harmonic mean of precision and recall [23]:
- *normalized F1-score*: the F1-score calculated only for the customers having at least one item correctly predicted.

B. Datasets

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

The *Coop* repository is provided by *Unicoop Tirreno*[1], a big retail supermarket chain in Italy. It stores transactions made 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. From the

[1] https://www.unicooptirreno.it/

TABLE I
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

TABLE II
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 25% customers

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

repository, we extract two datasets: *Coop-A* and *Coop-C*. 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, for a total of 7,690 different articles. In ***Coop-C*** (categories) the items are mapped to a more general category: e.g., both blood orange and navel orange are considered as orange, generating 520 categories. All the customers in *Coop-A* and *Coop-C* have at least one purchase per month. *Ta-Feng*[2] dataset covers food, stationery and furniture, with 23,812 different items. It contains 817,741 covering over 4 months. We remove customers with less than 10 baskets and we consider the remaining 7%. Since we act experiments on retail data we adopt the *day* as time unit: parameters and annotations are expressed in days.

C. Interpretability of TARS

The interpretability of TARS is one of the main characteristics of our approach. Table II shows some examples of TARS extracted from *Coop-C*. 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 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 (α_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 $\{milk\}$. 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, 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.

D. Temporal Validity and Extraction Reliability of TARS

In this section, we present some peculiar properties of TBP: the temporal validity and reliability of the TARS extracted. Since these experiments are closely tied to the applicability of TBP in real services, we report the results obtained on *Coop*.

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 1 (left), regardless the predicted basket size k the F1-score remain stable up to 20 predictions, which suggests a large temporal validity of TBP since the model construction.

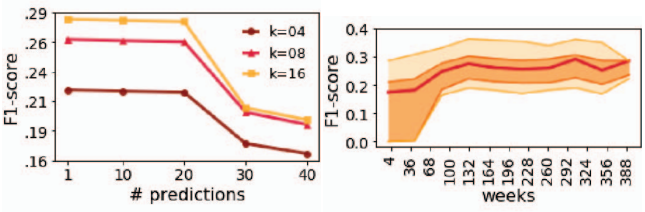

Fig. 1. *Left*: evaluation of TARS temporal validity with respect of F1-score. *Right*: evaluation of TARS reliability by augmenting the purchase history.

How many baskets does TBP need to perform reliable predictions? For each customer, we start from her second week of purchases and extract TARS incrementally by extending the training set one week at a time. We then predict the next basket of the customer and evaluate the performance of TBP in this scenario. Figure 1 (right) shows the median value and the "variance" (by means of the 10th, 25th, 75th and 90th percentiles) of the F1-score. 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. This experiment shows that for a real application that effectively runs TBP reliable performance on sound TARS are expected when from 9 to 12 months of data are required.

E. Comparing TBP with Baseline Methods

We compare the performance of TBP against the following baseline methods. Four user-centric approaches that build the predictive model of a customer using only her purchase data B_c, and four not user-centric methods that require purchase data of all customers $\mathcal{B}$. *LST*: the basket predicted is the *last* basket purchased. *TOP*: predicts the top-k *most frequent* items. *MC*: base the prediction on a *Markov chain* calculated on B_c. *CLF* [12]: for each item it builds a binary *classifier* on temporal features extracted from B_c. *NMF* [24]: *Non-negative Matrix Factorization*. *FMC* [6]: *Factorizing personalized Markov Chain*. *HRM* [7]: *Hierarchical Representation Model DRM* [8]: *Dynamic Recurrent basket Model*[3].

Table III reports the F1-score of TBP against the baselines 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 significantly outperforms the baselines and, together with the others user-centric approaches, it outlines

[2]http://www.bigdatalab.ac.cn/benchmark/bm/dd?data=Ta-Feng

[3]We provide the Python code of TBP, the baseline methods and an anonymized sample of *Coop* dataset at https://github.com/GiulioRossetti/tbp-next-basket. The code of DRM was kindly provided by the authors of [8].

Fig. 2. Performance comparison of TBP against the baselines varying length k.

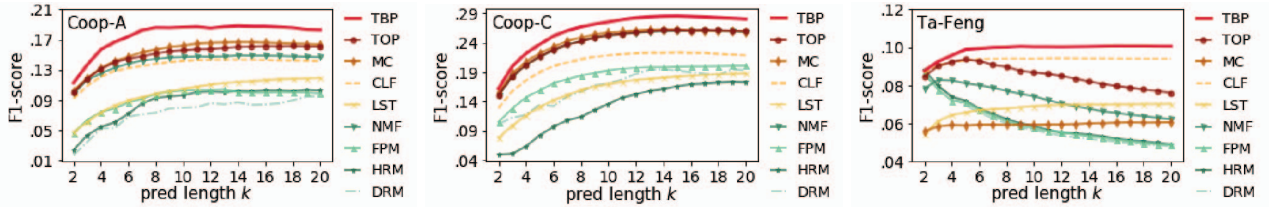

TABLE III
F1-SCORE USING $k{=}k_c^*$. **BOLD** 1ST, ***bold-italic*** 2ND BEST PERFORMER.

$k = k_c^*$		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

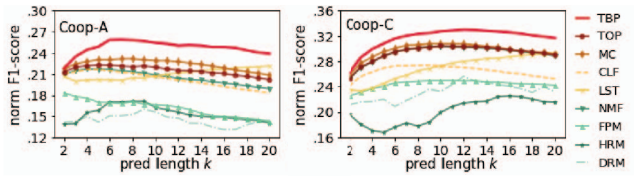

Fig. 3. Normalized F1-score varying predicted basket length k.

how for this particular task a user-centric model is more accurate than a not user-centric one. To understand how the performance are affected by the variation of k, in Figure 2 we compare the F1-score produced by TBP and by the baseline methods while varying $k{\in}[2, 20]$. We observe that TBP considerably overtakes the baseline methods on all the three datasets. Thus, the performance improvement of TBP with respect to the state of the art are not negligible either using $k{=}k_c^*$ or if a fixed k is specified for every customer.

Finally, we notice that the F1-scores can be biased by two extreme scenarios: *(i)* the F1-score can be low because 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 3 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 item, the baskets predicted by TBP are more accurate and cover a larger number of items than the baskets predicted by the other methods.

VII. Conclusions

We have proposed a data-driven and user-centric approach for market basket prediction. Our contribution is twofold. First, we have defined Temporal Annotated Recurring Sequences (TARS). Then we have used TARS to build a TARS Based Predictor (TBP) for forecasting customers' next baskets. We have performed experiments on real-world datasets showing that TBP outperforms state-of-the-art methods and, in contrast with them, it provides interpretable patterns that can be used to gather insights on customers' shopping behaviors.

Acknowledgment

This work is partially supported by the European Communitys H2020 Program, grant agreement 654024, "SoBigData: Social Mining & Big Data Ecosystem", http://www.sobigdata.eu.

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] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, p. 4, 2009.

[3] C. Chand, A. Thakkar, and A. Ganatra, "Sequential pattern mining: Survey and current research challenges," *IJSCE*, pp. 185–193, 2012.

[4] C.-N. Hsu, H.-H. Chung, and H.-S. Huang, "Mining skewed and sparse transaction data for personalized shopping recommendation," *ML*, vol. 57, no. 1-2, pp. 35–59, 2004.

[5] E. Lazcorreta *et al.*, "Towards personalized recommendation by two-step modified apriori data mining algorithm," *ESA*, pp. 1422–1429, 2008.

[6] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *WWW*. ACM, 2010, pp. 811–820.

[7] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for nextbasket recommendation," in *SIGIR*. ACM, 2015, pp. 403–412.

[8] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A dynamic recurrent model for next basket recommendation," in *SIGIR*. ACM, 2016, pp. 729–732.

[9] P. Wang *et al.*, "Modeling retail transaction data for personalized shopping recommendation," in *CIKM*. ACM, 2014, pp. 1979–1982.

[10] C. Kalapesi, "Unlocking the value of personal data: From collection to usage," in *World Economic Forum technical report*, 2013.

[11] A. Pentland *et al.*, "Personal data: The emergence of a new asset class," in *An Initiative of the World Economic Forum*, 2011.

[12] C. Cumby *et al.*, "Predicting customer shopping lists from point-of-sale purchase data," in *SIGKDD*. ACM, 2004, pp. 402–409.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *SIGKDD*. New York, NY, USA: ACM, 2016, pp. 1135–1144.

[14] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *Sigmod Record*, vol. 29, no. 2. ACM, 2000, pp. 1–12.

[15] K. Amphawan, A. Surarerks, and P. Lenca, "Mining periodic-frequent itemsets with approximate periodicity using interval transaction-ids list tree," in *SIGKDDw*. IEEE, 2010, pp. 245–248.

[16] P. Fournier-Viger *et al.*, "Phm: mining periodic high-utility itemsets," in *ICDM*. Springer, 2016, pp. 64–79.

[17] R. U. Kiran and M. Kitsuregawa, "Finding periodic patterns in big data," in *BDA*. Springer, 2015, pp. 121–133.

[18] E. Keogh, S. Lonardi, and C. A. Ratanamahatana, "Towards parameter-free data mining," in *SIGKDD*. ACM, 2004, pp. 206–215.

[19] K. Pearson, "Contributions to the mathematical theory of evolution," *Phil. Trans. R. Soc. Lond.*, vol. 185, pp. 71–110, 1894.

[20] H. A. Sturges, "The choice of a class interval," *JASA*, pp. 65–66, 1926.

[21] D. Freedman and P. Diaconis, "On the histogram as a density estimator: L 2 theory," *Probability Theory and Related Fields*, pp. 453–476, 1981.

[22] R. Agrawal *et al.*, "Mining association rules between sets of items in large databases," in *Sigmod Record*, no. 2. ACM, 1993, pp. 207–216.

[23] P. Tan *et al.*, *Introduction to data mining*. Pearson Education, 2006.

[24] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, 2001, pp. 556–562.