On the Prediction of Possibly Forgotten Shopping Basket Items

Anderson Singh and Patrick Hosein

The University of the West Indies, Trinidad and Tobago
anderson@lab.tt, patrick@lab.tt

Abstract. With the easy availability of shopping transaction data, it is now possible to use historical information to enhance the shopping experience of shoppers. In particular, we focus on the issue of forgotten items. We have all had the experience of shopping at a supermarket and either forgetting to buy certain items or not realizing that certain items are needed. In this paper we focus on predicting such items during a shopper's visit and providing hints to the shopper that they may in fact need certain items that were not purchased. Note that, by providing hints for forgotten items, the shopper benefits by avoiding a return trip to the supermarket and the supermarket benefits because the shopper purchases additional items. We provide examples to illustrate the utility of the proposed approach.

Keywords: Machine Learning, Prediction, Market Basket Analysis, Artificial Intelligence

1 Introduction

In the retail industry, market basket analysis [1, 2] is a popular technique used to identify patterns and trends among customers and their purchased items. It is this data that can be used to influence major corporate decisions. In real-world situations, a customer's shopping history can be considered as a sequence of baskets, where each basket consists of the set of purchased items at a particular visit. We can use this information to predict what the customer should purchase at a future visit through the use of recommender systems [3, 4]. The problem of predicting the customer's next basket has been addressed in different contexts, such as e-commerce websites, which recommend to their customers relevant items, related to their individual lifestyles. It is important to note that we can increase customer satisfaction and total sales by recommending appropriate items to customers.

In this paper, we propose a new method known as Interval Based Predictor (IBP), which solves the next basket recommendation problem in the context of a supermarket environment. IBP provides the customer with an appropriate list of items that they may have forgotten to purchase at their present supermarket visit. IBP is able to learn the consumption time of a particular item and hence be able to predict when next it would be needed. It is worth mentioning that this

approach only uses data for the customer under consideration, unlike traditional approaches which require data from multiple customers, that is, the method is customer-centric.

It should be noted that both the customer and the supermarket benefit from our approach since the customer avoids a return visit to the supermarket, and the supermarket benefits from the potential increase in revenue. To this end, we propose a metric to estimate the potential increase in revenue from each customer, and consequently, for all shoppers of the supermarket. In the evaluation section of our paper, we compare the proposed next basket recommendation method against multiple state-of-the-art methods.

2 Related Work and Contributions

To the best of the authors' knowledge, there has been no previous work on recommending potentially forgotten items to customers. Instead, the literature is focused on next basket recommendation, which provides the customer with a reasonable shopping list of items that they may purchase at their next visit based on historical transactions. On the other hand, our objective is to remind the customer about potentially forgotten items just before the start of their checkout procedure. Since our work is closely related to next basket recommendation, we adopt a similar approach to [5–7] in analyzing the existing literature. Thus, we classify recommender systems based on their type, which can be either general, sequential, or hybrid.

A general recommender system provides recommendations based on the general taste of the customer, that is, it does not consider factors such as the sequence in which the items were purchased or how recent the items were purchased. A commonly used technique in general recommender systems is collaborative filtering. The term collaborative filtering implies that individuals can assist each other in locating content which is of interest to them. Tapestry [8] is one of the first collaborative filtering implementations that recommends documents posted in newsgroups to users. A survey of collaborative filtering including model-based and memory-based techniques are discussed in [9]. Furthermore, [10] discussed the state of research with regard to collaborative filtering and implicit feedback.

A sequential recommender utilizes both historical transactions as well as its accompanying sequential properties to provide a list of recommendations to the customer. A customer's historical transactions can be treated as an ordered sequence of baskets, in which, a basket consists of a set of items purchased at a particular point in time. It is this sequential data that is exploited to provide recommendations. A recommender based on the Markov decision process is proposed in [11], where the generation of recommendations is treated as a sequential decision problem because the recommendations are based on the customer's past behaviour. It should be noted that sequential recommenders are not only applicable to predicting the customer's next basket, for example, [12] investigated real website usage data to compare the effects of contiguous sequential patterns

and less constrained sequential patterns. Another example of sequential recommender applicability is discussed in [13] which proposes a method for generating music playlists based on Markov Chains. In [14], an external memory network that uses hidden states was proposed to capture historical data. Furthermore, a comprehensive overview of sequential mining algorithms based on their efficiency and applicability is provided in [15].

A hybrid recommender system is based on a combination of sequential and general recommendation techniques which leverages the advantages from each method. For example, FPMC [7] is a hybrid recommendation system which combines a general recommendation technique, matrix factorization and a sequential recommendation technique, Markov Chains. HRM [6] is another novel hybrid recommendation method aimed at improving the results of FPMC. The authors of HRM noted that a drawback of FPMC is that the factors that affect a customer's purchasing decisions are combined linearly, however, this assumption is not true. HRM overcame this issue through the introduction of nonlinear aggregation operations, which provided better results. DREAM [16] is yet another hybrid recommendation technique based on recurrent neural networks. The authors of DREAM stated that HRM presents a partial solution to this issue through the inclusion of nonlinear aggregation operations. The authors of DREAM further state that FPMC and HRM do not model the global sequential behavior of a customer, that is, those methods only consider the sequential properties between adjacent baskets. In [5], a novel hybrid recommendation method, Temporal Based Predictor (TBP), which combines both sequential and pattern-based recommender techniques was presented. TBP exploits recurring sequential purchases known as Temporal Annotated Recurring Sequence (TARS) in the customer's historical transactional data. TARS is able to capture multiple factors which affect the purchasing decisions of a customer such as the co-occurrency, sequentiality, periodicity, and recurrency of items in the basket. In [17], the task of predicting a customer's next basket was considered as a machine learning problem, but more specifically, a classification problem. The authors proposed learning separate binary classifiers for every item for each customer. In other words, a single classifier is associated with every customer and item pair which determines whether or not the given customer will purchase the particular item. Let us consider a supermarket that has X customers, where each customer purchases approximately Y items. In this situation, the recommendation system would consist of approximately XY distinct binary classifiers. The authors also explored the application of various machine learning classification methods such as, decision trees, perceptron, Naive Bayes, and Winnow.

3 Problem Definition

We have all had the experience of going grocery (or any) shopping and forgetting to buy one or more items. When this happens we have to make an additional trip to the grocery store for the missed items. In this paper we address this problem. We consider one such visit and attempt to predict any items that may

have been missed. Just before the customer checks out, we determine potentially missed items and inform the customer who can then go back for the items before checking out. The metric we use is the percentage increase in the total number of basket items and percentage increase in revenue because of these additional purchases. This reflects, in some way, the increased revenue achieved by the store and hence the utility of the approach.

We model the problem as follows. Let B represent the set of basket items picked up prior to providing hints to the user. Let F represent the set of forgotten items and finally, let P denote the set of predicted basket items. The set $P \cap B$ represents the set of items that were picked up and were correctly predicted to be purchased. The set $P-(P\cap B)$ represents the set of items that were predicted to be purchased but were not. This is the potential set of forgotten items.

The probability of successfully predicting a basket item is given by $s = \frac{|P\cap B|}{|B|}$. Now consider a forgotten item. If this item had not been forgotten then the probability that it would have been successfully predicted would have also been s. The fact that it was forgotten does not change this probability. Therefore the probability that a forgotten item is correctly predicted should also be s. Therefore the expected number of predicted items that were not bought because they were forgotten is given by $s|F|$. If the customer purchases all forgotten but predicted items then the expected fractional increase in purchased items, ρ, is given by

$$\rho = \frac{s|F|}{|B|} = \frac{|P \cap B||F|}{|B|^2} \tag{1}$$

Let us illustrate with a simple example. Suppose that our predictor has an accuracy of 60% so that $s = 0.6$. Assume that if one goes into a store to purchase 20 items that they tend to forget at least one of the items. We can then assume that $|F| = 0.05|B|$. Therefore we obtain $\rho = 0.03$. In other words there is an expected increase of 3% of purchased items and, if we assume any item is equally likely to be forgotten, this translates into a 3% increase in revenue.

Note that this is an upper bound on revenue increase since the customer may return to the same grocery for the items anyway rather than going somewhere else in which case there is no revenue increase. However, there is still the benefit of convenience since these hints would avoid the customer making a second trip either to the same grocery or another one.

The prediction success probability s can be obtained for each customer based on their prior purchases. A prediction algorithm is used to predict the basket and then this can be compared with the actual basket to determine the success probability s. We assume that the basket items in set B are known. However the set F, the set of forgotten items, is typically not known. One approximation that can be used is the set $P - (P \cap B)$ which can be obtained from the present visit. This is the set of predicted items that were not purchased and is the list of items which will then be provided as hints to the user. If this set only contained forgotten items then it is what we would need. However, in some cases it may also contain items that are neither in the bought or forgotten sets. By using

feedback from the customer (who would reject such items) we can identify the truly forgotten items and use this set to obtain $s|F|$.

4 The Proposed Approach

4.1 Basket Prediction

In this section, we explain the algorithm used to predict a customer's basket at the supermarket. We consider visit i for a single customer which occurs at time t_i, and compute the predicted basket for that visit. We look at a single item j, and determine whether or not it should be included in the predicted basket for visit i. Let T_i represent the average time between visits for the customer, and let G_j represent the average time between visits where the customer purchased item j. Also, we use L_j to represent the last time the customer purchased item j such that $L_j < t_i$. Using this data, $L_j + G_j$ provides an estimated time when the customer would exhaust their stock of item j at home. Hence, $L_j + G_j$ is the ideal time for the customer to repurchase item j. However, the customer may not be shopping at the supermarket at that exact moment. Therefore, a reasonable repurchase time must be computed.

Let us consider the trivial case where $L_j + G_j \leq t_i$. In such a situation, item j should be repurchased by the customer since their stock at home has been exhausted. When $L_j + G_j \geq t_i + T_i$, item j should not be purchased because the customer has sufficient stock at home which will not be exhausted at least until the next visit. In the last case where $t_i < L_j + G_j < t_i + T_i$, item j will be exhausted by the customer some time between the current visit and the next visit. Since the customer uses item j with the time period G_j, then it must be also repurchased with this same time period. If item j is always bought at the current visit, then the customer will continuously increase their stock at home. Furthermore, if item j is always bought at the next visit, the customer will continuously exhaust their stock at home. Therefore, a random decision must be made to purchase item j at the current visit. A random number r is generated between 0 and 1. If $r \geq \frac{L_j+G_j-t_i}{T_i}$, then item j should be purchased at the current visit, otherwise, it should be purchased at the next visit. In this approach, the time period with which item j is repurchased is given by

$$\frac{L_j + G_j - t_i}{T_i}(t_i - L_j + T_i) + \left(1 - \frac{L_j + G_j - t_i}{T_i}\right)(t_i - L_j)$$

which is simply G_j, the desired repurchase time period. If item j is recommended to the customer, a suggested purchase quantity Q_j is also provided.

It is important to note that the above procedure is repeated for each item the customer has purchased in the past to construct the predicted basket, however, we may want to place a constraint on the size of the predicted basket to a given value, say k, to avoid an unreasonably large predicted basket. To achieve this, we select the top k items ranked in descending order by the number of unique visits where each item was purchased by the given customer. We let U_j represent the

number of unique visits where the customer purchased item j. The corresponding pseudo-code is included Algorithm 1. Note that S is computed in Algorithm 2.

Algorithm 1 *predict_basket*$(i(\text{visit}), T_i, t_i, S, k(\text{limit}))$

$P \leftarrow \emptyset$ (predicted basket)
for $((G_j, L_j, U_j, Q_j, j(\text{item})) \in S)$ **do**
  if $L_j + G_j \leq t_i$ **then**
    $P \leftarrow P \cup \{(j, U_j, Q_j)\}$
  else
    if $L_j + G_j > t_i$ **and** $L_j + G_j < t_i + T_i$ **then**
      $r = rand(0, 1)$
      if $r \geq \frac{L_j + G_j - t_i}{T_i}$ **then**
        $P \leftarrow P \cup \{(j, U_j, Q_j)\}$
P ← *top_k_items(P,k)*
return P

4.2 Training the Basket Prediction Algorithm

At the start of the customer's next supermarket visit, for some forgetting factor $0 < \alpha < 1$, we perform an update as follows

$$T_{i+1} = \alpha T_i + (1 - \alpha)(t_{i+1} - t_i). \tag{2}$$

If item j is purchased by the customer at visit i, then we compute

$$G_{j+1} = \alpha G_j + (1 - \alpha)(t_i - L_j). \tag{3}$$

However, if item j is not purchased by the customer at visit i, then we simply set $G_{j+1} = G_j$. In this case, item j will then be purchased at a future visit and the variable will be updated at that time. If the customer purchases quantity q of item j at visit i, where $q > 0$, then we update

$$Q_{j+1} = \beta Q_j + (1 - \beta)(q). \tag{4}$$

where β is a factor $0 < \beta < 1$ used to exponentially smooth the quantity purchased. Also, if item j is purchased at visit i, we update the number of unique visits where the customer purchased this particular item.

$$U_{j+1} = U_j + 1 \tag{5}$$

The updates above are performed for each item the customer has purchased at visit i. Given both the predicted and actual baskets, the success probability and resulting performance metric ρ can be calculated. Algorithm 2 provides the pseudo-code for learning both the time intervals for item repurchases and the customer's time interval for revisiting the supermarket. The update is performed just before the customer begins their shopping at visit $i + 1$.

Algorithm 2 *learn_time_intervals(S, B (basket), t_{i+1}, t_i, α, β)*

$T_{i+1} \leftarrow \alpha T_i + (1-\alpha)(t_{i+1} - t_i)$
for $(j \in B)$ **do**
  if $\nexists\, x : x \in S$ *and* $x = (G_j, L_j, U_j, Q_j, j))$ **then**
    $L_{j+1} \leftarrow t_i$
    $G_{j+1} \leftarrow 0$
    $U_{j+1} \leftarrow 1$
    $Q_{j+1} \leftarrow item\_count(B, item)$
    $S \leftarrow S \cup \{(G_j, L_j, U_j, Q_j, j)\}$
  else
    $G_{j+1} \leftarrow \alpha G_j + (1-\alpha)(t_i - L_j)$
    $Q_{j+1} \leftarrow \beta Q_j + (1-\beta) item\_count(B, j)$
    $U_{j+1} \leftarrow U_j + 1$
    $L_{j+1} \leftarrow t_i$
return S

5 Numerical Results

In this section we provide numerical results for real customer transactions at a major supermarket to show the effectiveness of our approach.

5.1 Dataset and Experimental Approach

The data was supplied by a major supermarket chain. It contains customer transactions for a branch of the supermarket chain and includes data on items from various categories, such as electronics, food, health, and pharmaceuticals. The dataset spans purchases made during 2015, a one-year period. In the data pre-processing phase only those customers who made at least one purchase every month were selected. Furthermore, the items in the dataset were grouped into general classes as opposed to being used as individual items. For instance, *artisan romaine lettuce* and *artisan Boston lettuce* are grouped under the general class of *lettuce*. The rationale for grouping the items under general classes is that the customer may purchase product substitutes in different brands or sizes. Thus, to understand the level of customer stock, it is better to generalize the items. We obtained data for a total of 433 customers and focused on 615 basket items. These customers made an average of 36 visits (baskets) and the average basket size over all customers and visits was 28.

In conducting our experiments, we adopt the *leave one out* strategy for the purpose of model evaluation. In this strategy, the dataset is split into a training set and test set for each customer such that each test set consists of only the last transaction for a given customer. The remaining transactions for each customer are placed into their respective training set. In other words, for each customer, their historical data is used to train their individual predictive model and their last basket is used to check the accuracy of the approach. In terms of the predicted basket's length, we use the average length of the customer's basket, k^*,

based on their historical transactions. However, we also perform experiments where the length of the predicted basket is based on a fixed value $k \in [2, 20]$. With regard to parameters, we set $\alpha = 0.4$ and $\beta = 0.5$, however, we varied these values and found the results to be fairly consistent, which suggests that our method is reasonably stable. We evaluate our model based on the F1-score, which is the harmonic mean of precision and recall [18].

In terms of increase in revenue, the size of the forgotten set, $|F|$, is estimated as $\gamma|B|$, where $|B|$ is the size of the customer's basket and γ is a factor, $0 < \gamma < 1$, which represents the customer's forgetfulness. Let us consider a customer that must purchase 20 items at the supermarket. If the customer forgets to purchase a single item, then 5% of the basket has been forgotten. Therefore, we use a value of $\gamma = 0.05$ which can be interpreted as the customer will forget 5% of their basket. We believe $\gamma = 0.05$ is a reasonable value to use for our experiments.

5.2 Results

In Table 1 we demonstrate the intuition in the item selection process used in our method. It should be noted that the items in Table 1 we both predicted by our algorithm and purchased by the customer. The customer visited the supermarket on the 22-12-2015, and given that the time period between visits is approximately 9 days, their next visit is estimated to be on the 31-12-2015.

Table 1. Examples of Correctly Predicted Items

Item	Past Visits			Visit	Current	Visit	Estimated
	Date 1	Date 2	Date 3	Interval	Date	Interval	Next Visit
Flour	Oct 15	Nov 6	Dec 4	24 days			
Milk	Nov 12	Nov 27	Dec 10	14 days	Dec 22	9 days	Dec 31
Tuna	Nov 19	Nov 27	Dec 4	11 days			

First, let us consider the item, *flour*, which has a consumption time of approximately 24 days for this customer. Thus, given that the customer last purchased this item on the 04-12-2015, we expect that it will be exhausted on 28-12-2015. Therefore, at the current visit date, 22-12-2015, the customer does not need *flour*, since there is still a fraction remaining at home. However, given that the estimated next visit is on 31-12-2015, there would be a 3-day window where the customer will not have *flour*. Hence, the algorithm makes a random decision to recommend the *flour* at the current visit as opposed to the customer foregoing the item for 3 days.

In a similar manner, *milk*, which has a consumption time of approximately 14 days, was last purchased by the customer on 10-12-2015, and at the current visit date, 22-12-2015, would not yet have been exhausted. However, in 2 days,

the *milk* would have been fully consumed, and the customer would be forced to forgo the item for 7 days, that is, until the estimated next visit on 31-12-2015. Consequently, as with *flour*, a random decision is made to recommend *milk* at the current visit, as opposed to the next visit.

Finally, *tuna*, has a consumption time of approximately 11 days for this customer and thus, based on the item's last purchase date, 04-12-2015, the customer's stock is expected to become exhausted on the 15-12-2015. Therefore, at the current visit date, 22-12-2015, the customer has not purchased *tuna* in 18 day and has been deprived of the item for 7 days. As a result, the algorithm recommends that the customer should purchase *tuna* at the current visit.

Intuitively, from Table 1 and the above explanation, the reader should have a clearer understanding of the item selection criteria. Therefore, based on these criteria, we believe the items that were both predicted by our method and were not purchased by the customer are good candidates for forgotten items. However, we do not truly know if these items are forgotten since hints were never supplied to the customers used for training the model.

In Table 2, we show such examples of probable forgotten items for a chosen customer. The customer visited the supermarket on the 14-12-2015, and their next visit is estimated to be on the 27-12-2015, given that the interval between visits is approximately 13 days.

Table 2. Examples of Potentially Forgotten Items

Item	Past Visit Dates			Visit	Current	Visit	Estimated
	Date 1	Date 2	Date 3	Interval	Date	Interval	Next Visit
Pepper	Oct 23	Nov 6	Nov 27	25 days	Dec 14	13 days	Dec 27
Bread	Nov 6	Nov 20	Nov 27	10 days			

First, let us consider the item, *pepper*, that has an approximate consumption time of 25 days for the given customer. Thus, based on the customer's last purchase date, 27-11-2015, we expect that their stock of *pepper* will be exhausted on the 22-12-2015. Now, as at the current visit date, 14-12-2015, the stock of *pepper* is not yet depleted. However, if the customer does not purchase the item at this time, they must forgo it for approximately 5 days, that is, until their next visit. Hence, the algorithm makes a choice to recommend the item to the customer in order to avoid depriving them of it for the 5-day period. Similarly, *bread* has a consumption time of 10 days for the selected customer, and given that the customer last purchased bread on 27-11-2015, we expect that it would be exhausted on 07-12-2015. Thus, on 14-12-2015, the current visit date, the customer has already been deprived of *bread* for 7 days and thus the algorithm recommends that bread be purchased by the customer.

We also computed the average expected increase in revenue for a number of visits. This is plotted in Figure 1. The overall average increase in revenue is approximately 1.74%.

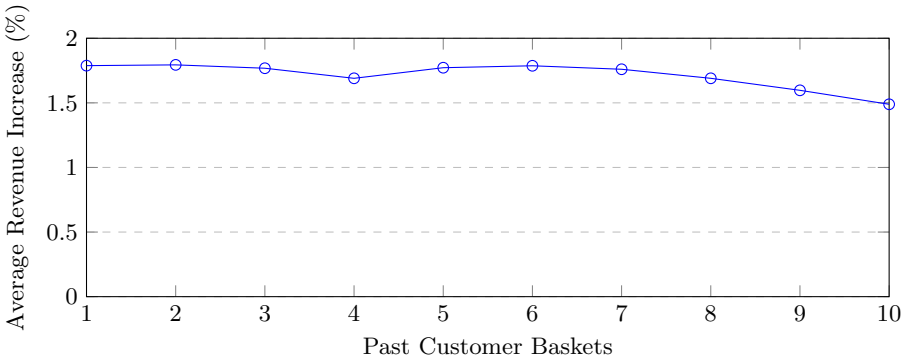

Fig. 1. Expected Revenue Increase for 10 Visits

5.3 Comparison with Prior Methods

Our objective is not to predict the customer's next basket, but rather to provide the customer with a list of items that may have been forgotten. Nevertheless, it is useful to compare the performance of our algorithm against several baseline methods used for next basket prediction. The baseline methods chosen share a central characteristic to our method in that they are customer-centric in nature, which means that each customer's predictive model is based solely on their historical data. The following is a brief description of each of the chosen baseline methods as well as ours.

LST: [17], [5] The predicted basket consists of the same set of items as the last basket bought by the given customer.
TOP: [17], [5] The predicted basket consists of the most frequent items bought by the given customer.
TBP: [5] The predicted basket consists of items that are obtained from recurring patterns or sequences in the given customer's past transactions.
IBP: Our proposed algorithm.

We computed the F1-Scores for each algorithm and obtained values of 0.28 for IBP, 0.30 for TBP, 0.33 for TOP and 0.22 for LST. It should be noted that the size of the predicted basket, k, is equal to the average size of the customer's basket based on their historical data, k^*.

In comparing the methods, we adopt the strategy in [5], where the training time of the methods are reported. We found that for IBP, TOP and LST the

training time was about 0.01 seconds. However for TBP the training time was about 55 seconds. The difference in running time is a clear benefit of using the proposed method over TBP, despite the 2% under performance in the F1-score metric. Also, TBP requires that the model for each customer be retrained as time progresses because the quality of the recommendations will begin to decay. However, the proposed method and TOP both allow a given customer's model to be updated as they make future visits to the supermarket without entirely rebuilding the model.

Although TOP outperforms the proposed method, we believe for the task of recommending potentially forgotten items, our algorithm is more appropriate. In our work, the set of potentially forgotten items is a subset of the items that were recommended but not bought because the customer is not aware that either their stock at home will finish before the next visit, or has already finished. The proposed method takes into account the approximate time interval the given customer takes to return to the supermarket, and also the approximate time it takes for each item to be consumed. Using this data, the method can determine when a customer's stock for an item is about to be exhausted or has already been exhausted, and then, make a recommendation accordingly. However, the TOP algorithm simply recommends the most popular set of items bought by a customer, and this does not reflect items which may have been forgotten. In fact, if an item is popular with a given customer, it is less likely to be forgotten.

Also, TOP is not able to predict items which are bought infrequently, whether because they have a long consumption time or because they are bought in bulk. We believe that these items will generally incur a higher cost to the customer and thus it is in the supermarket's interest to recommend such items to achieve a revenue gain. It was found that the average price of an item that was both recommended and bought by the proposed method is \$23.82, as opposed to TOP, where the average price is \$20.18. Therefore the proposed method recommends items which cost approximately \$3.64 more than those recommended by TOP.

We also report on the F1-scores of the proposed method and the baseline methods aforementioned, where the size of the predicted basket, k, is fixed for all customers in the dataset. We perform these experiments for different values of $k \in [2, 20]$ which is then plotted in Figure 5.3. It can be seen that TOP outperforms both the proposed method and TBP. However, the proposed method outperforms TBP in the range $k \in [2, 11]$ which suggests it is better able to predict smaller baskets. It is also worth mentioning that the proposed method and TBP has approximately the same F1-scores when the size of the predicted basket is in the range $k \in [12, 15]$ after which TBP begins to outperform the proposed method.

6 Conclusions and Future Work

The contributions of this paper are two-fold. Firstly, we introduced a time-based method to recommend items that a customer may have potentially forgotten at their visit to the supermarket. Then, we show the advantages of our work and

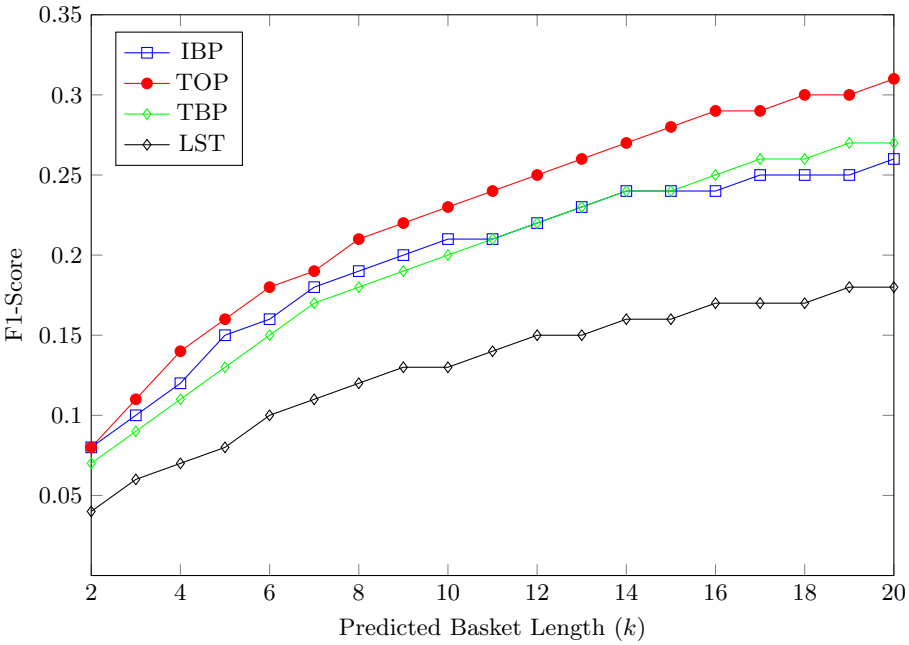

Fig. 2. Performance Comparison

how it compares to existing next basket recommendation methods. Secondly, we demonstrate the effectiveness of our method by estimating its increase in revenue on a real-world dataset. Although the proposed algorithm did not yield the best results in the performance comparison with other algorithms, it took into account individual item costs and therefore, resulted in an increase in revenue stream unlike the cost of items predicted by the other algorithms.

Future work will include cultural habit changes where the consumer starts or stops purchasing a product based on a lifestyle change for example. We also plan to develop and deploy a mobile application that customers may use to obtain hints on potentially missed items when they shop. We believe that the revenue increase, even if it is 2%, will prove to be significant to the supermarket especially since the only cost to them is the development of the application. Consumers also benefit by avoiding return trips to the supermarket for forgotten items.

References

1. Ledolter, J.: Data mining and business analytics with R. John Wiley & Sons (2013)
2. Olson, D.L.: Market basket analysis. In: Descriptive Data Mining, pp. 29–41. Springer (2017)
3. Bobadilla, J., Ortega, F., Hernando, A., Gutirrez, A.: Recommender systems survey. Knowledge-Based Systems **46**, 109 – 132 (2013). DOI https://doi.org/10.

1016/j.knosys.2013.03.012. URL http://www.sciencedirect.com/science/article/pii/S0950705113001044
4. Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G.: Recommender system application developments: A survey. Decision Support Systems **74**, 12 – 32 (2015). DOI https://doi.org/10.1016/j.dss.2015.03.008. URL http://www.sciencedirect.com/science/article/pii/S0167923615000627
5. Guidotti, R., Rossetti, G., Pappalardo, L., Giannotti, F., Pedreschi, D.: Market basket prediction using user-centric temporal annotated recurring sequences. In: 2017 IEEE International Conference on Data Mining (ICDM), pp. 895–900. IEEE (2017)
6. Wang, P., Guo, J., Lan, Y., Xu, J., Wan, S., Cheng, X.: Learning hierarchical representation model for nextbasket recommendation. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval, pp. 403–412. ACM (2015)
7. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th international conference on World wide web, pp. 811–820. ACM (2010)
8. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Communications of the ACM **35**(12), 61–70 (1992)
9. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Advances in artificial intelligence **2009**, 4 (2009)
10. Najafabadi, M.K., Mahrin, M.N.: A systematic literature review on the state of research and practice of collaborative filtering technique and implicit feedback. Artificial Intelligence Review **45**(2), 167–201 (2016). DOI 10.1007/s10462-015-9443-9. URL https://doi.org/10.1007/s10462-015-9443-9
11. Shani, G., Heckerman, D., Brafman, R.I.: An mdp-based recommender system. Journal of Machine Learning Research **6**(Sep), 1265–1295 (2005)
12. Mobasher, B., Dai, H., Luo, T., Nakagawa, M.: Using sequential and non-sequential patterns in predictive web usage mining tasks. In: Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on, pp. 669–672. IEEE (2002)
13. Chen, S., Moore, J.L., Turnbull, D., Joachims, T.: Playlist prediction via metric embedding. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 714–722. ACM (2012)
14. Chen, X., Xu, H., Zhang, Y., Tang, J., Cao, Y., Qin, Z., Zha, H.: Sequential recommendation with user memory networks. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 108–116. ACM (2018)
15. Chand, C., Thakkar, A., Ganatra, A.: Sequential pattern mining: Survey and current research challenges. International Journal of Soft Computing and Engineering **2**(1), 185–193 (2012)
16. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T.: A dynamic recurrent model for next basket recommendation. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 729–732. ACM (2016)
17. Cumby, C., Fano, A., Ghani, R., Krema, M.: Predicting customer shopping lists from point-of-sale purchase data. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, pp. 402–409. ACM, New York, NY, USA (2004). DOI 10.1145/1014052.1014098. URL http://doi.acm.org/10.1145/1014052.1014098
18. Tan, P.N.: Introduction to Data Mining (2nd Edition) (What's New in Computer Science). Pearson (2018). URL https://www.xarg.org/ref/a/0133128903/