Author/s:
Silva, A;Luo, L;Karunasekera, S;Leckie, C

Title:
OMBA: User-Guided Product Representations for Online Market Basket Analysis

Date:
2020

Citation:
Silva, A., Luo, L., Karunasekera, S. & Leckie, C. (2020). OMBA: User-Guided Product Representations for Online Market Basket Analysis. Hutter, F (Ed.) Kersting, K (Ed.) Lijffijt, J (Ed.) Valera, I (Ed.) Machine Learning and Knowledge Discovery in Databases, 12457, pp.55-71. Springer. https://doi.org/10.1007/978-3-030-67658-2_4.

Persistent Link:
https://hdl.handle.net/11343/267644

Accepted as a conference paper at ECML-PKDD 2020

OMBA: User-Guided Product Representations for Online Market Basket Analysis*

Amila Silva ✉, Ling Luo, Shanika Karunasekera, and Christopher Leckie

School of Computing and Information Systems
The University of Melbourne
Parkville, Victoria, Australia
{amila.silva@student., ling.luo@, karus@, caleckie@}unimelb.edu.au

Abstract. Market Basket Analysis (MBA) is a popular technique to identify associations between products, which is crucial for business decision making. Previous studies typically adopt conventional frequent itemset mining algorithms to perform MBA. However, they generally fail to uncover rarely occurring associations among the products at their most granular level. Also, they have limited ability to capture temporal dynamics in associations between products. Hence, we propose OMBA, a novel representation learning technique for Online Market Basket Analysis. OMBA jointly learns representations for products and users such that they preserve the temporal dynamics of product-to-product and user-to-product associations. Subsequently, OMBA proposes a scalable yet effective online method to generate products' associations using their representations. Our extensive experiments on three real-world datasets show that OMBA outperforms state-of-the-art methods by as much as 21%, while emphasizing rarely occurring strong associations and effectively capturing temporal changes in associations.

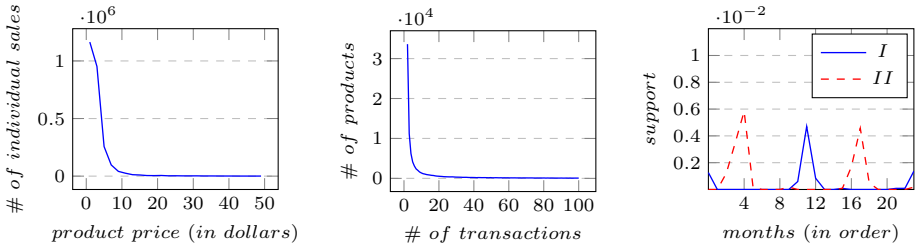
Keywords: Market Basket Analysis · Online Learning · Item Representations · Transaction Data

1 Introduction

Motivation. Market Basket Analysis (MBA) is a technique to uncover relationships (i.e., association rules) between the products that people buy as a basket at each visit. MBA is widely used in today's businesses to gain insights for their business decisions such as product shelving and product merging. For instance, assume there is a strong association (i.e., a higher probability to buy together) between product p_i and product p_j. Then both p_i and p_j can be placed on the same shelf to encourage the buyer of one product to buy the other.

Typically, the interestingness of the association between product p_i and product p_j is measured using *Lift*: the *Support* (i.e., probability of occurrence) of p_i and p_j together divided by the product of the individual *Support* values of p_i

* This research was financially supported by Melbourne Graduate Research Scholarship and Rowden White Scholarship.



(a) (b) (c)

Fig. 1: (a) Number of products' sales with respect to the products' prices; (b) number of products with respect to their appearances in the transactions; and (c) temporal changes in the *Support* of: (I) Valentine Gifts and Decorations; and (II) Rainier Cherries. The plots are generated using Complete Journey dataset

and p_j as if they are independent. MBA attempts to uncover the sets of products with high *Lift* scores. However, it is infeasible to compute the *Lift* measure between all product combinations for a large store in today's retail industry, as they offer a broad range of products. For example, Walmart offers more than 45 million products as of 2018[1]. Well-known MBA techniques [2,7,9] can fail to conduct accurate and effective analysis for such a store for the following reasons.

Research Gaps. First, MBA is typically performed using frequent itemset mining algorithms [7,9,12,2] , which produce itemsets whose *Support* is larger than a predefined *minimum Support* value. However, such frequency itemset mining algorithms fail to detect associations among the products with low *Support* values (e.g., expensive products, which are rarely bought, as shown in Figure 1a), but which are worth analysing. To further elaborate, almost all these algorithms compute *Support* values of itemsets, starting from the smallest sets of size one and gradually increasing the size of the itemsets. If an itemset P_i fails to meet the *minimum Support* value, all the supersets of P_i are pruned from the search space as they cannot have a *Support* larger than the *Support* of P_i. Nevertheless, the supersets of P_i can have higher *Lift* scores than P_i. Subsequently, the selected itemsets are further filtered to select the itemsets with high *Lift* scores as associated products. For instance, *'Mexican Seasoning Mixes'* and *'Tostado Shells'* have 0.012 and 0.005 support values in Complete Journey dataset (see Section 5) respectively. Because of the low support of the latter, conventional MBA techniques could fail to check the association between these two products despite them having a strong association with a $Lift$ score of 43.45. Capturing associations of products that have lower *Support* values is important due to the power law distribution of products' sales [13], where a huge fraction of the products have a low sales volume (as depicted in Figure 1b).

[1] `https://bit.ly/how_many_products_does_walmart_grocery_sell_july_2018`

Second, most existing works perform MBA at a coarser level, which groups multiple products, due to two reasons: (1) data sparsity at the finer levels, which requires a lower *minimum Support* value to capture association rules; and (2) large numbers of different products at the finer levels. Both aforementioned reasons substantially increase the computation time of conventional association rule mining techniques. As a solution to this, several previous works [19,11] attempt to initially find strong associations at a coarser level, such as groups of products, and further analyse only the individual products in those groups to identify associations at the finer levels. Such approaches do not cover the whole search space at finer levels, and thus fail to detect the association rules that are only observable at finer levels. In addition, almost all the conventional approaches consider each product as an independent element. This is where representation learning based techniques, which learn a low-dimensional vector for each unit[2] such that they preserve the semantics of the units, have an advantage. Thus, representation learning techniques are useful to alleviate the cold-start problem (i.e., detecting associations that are unseen in the dataset) and data sparsity. While there are previous works on applying representation learning for shopping basket recommendation, *none of the previous representation learning techniques has been extended to mine association rules between products*. Moreover, user details of the shopping baskets could be useful to understand the patterns of the shopping baskets. Users generally exhibit repetitive buying patterns, which could be exploited by jointly learning representations for users and products in the same embedding space. While such user behaviors have previously been studied in the context of personalized product recommendation [21,15], they have not been exploited in the context of association rule mining of products.

Third, conventional MBA techniques consider the set of transactions as a single batch to generate association rules between products, despite that the data may come in a continuous stream. The empirical studies in [18,13] show that the association rules of products deviate over time due to various factors (e.g., seasonal variations and socio-economic factors). To further illustrate this point, Figure 1c shows the changes in sales of two product categories over time at a retailer. As can be seen, there are significant variations of the products' sales. Consequently, the association rules of such products vary over time. Conventional MBA techniques fail to capture these temporal variations. As a solution to this problem in [18,13], the transactions are divided into different time-bins and conventional MBA techniques are used to generate association rules for different time bins from scratch. Such approaches are computationally and memory intensive; and ignore the dependencies between consecutive time bins.

Contribution. In this paper, we propose a novel representation learning based technique to perform Online Market Basket Analysis (OMBA), which: (1) *jointly learns representations for products and users such that the representations preserve their co-occurrences, while emphasizing the products with higher selling prices (typically low in Support) and exploiting the semantics of the products and users*; (2) *proposes an efficient approach to perform MBA using the learned*

[2] "Units" refers to the attribute values (could be products or users) of the baskets

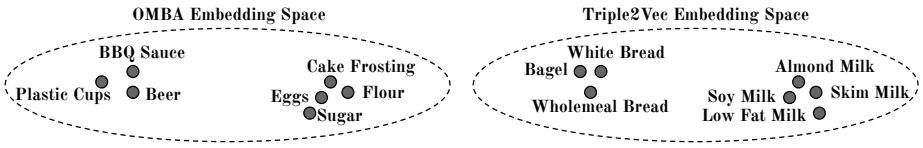


Fig. 2: An illustration of the embedding spaces of OMBA, which preserves complementarity, and TRIPLE2VEC, which only preserves the semantic similarity

product representations, which is capable of capturing rarely occurring and unseen associations among the products; and (3) *accommodates online updating for product representations, which adapts the model to the temporal changes, without overfitting to the recent information or storing any historical records*. The code for all the algorithms presented in this paper and the data used in the evaluation are publicly available via `https://bit.ly/2UwHfr0`.

2 Related Work

Conventional MBA Techniques. Typically, MBA is performed using conventional frequent itemset mining algorithms [7,9,12,2], which initially produce the most frequent (i.e., high *Support*) itemsets in the dataset, out of which the itemsets with high *Lift* scores are subsequently selected as association rules. These techniques mainly differ from each other based on the type of search that they use to explore the space of itemsets (e.g., a depth-first search strategy is used in [12], and the work in [2] applies breadth-first search). As elaborated in Section 1, these techniques have the limitations: (1) inability to capture important associations among rarely bought products, which covers a huge fraction of products (see Figure 1b); (2) inability to alleviate sparsity at the finer product levels; and (3) inability to capture temporal dynamics of association rules. In contrast, OMBA produces association rules from the temporally changing representations of the products to address these limitations.

Representation Learning Techniques. Due to the importance of capturing the semantics of products, there are previous works that adopt representation learning techniques for products [21,5,15,22,6,10]. Some of these works address the *next basket recommendation task*, which attempts to predict the next shopping basket of a customer given his/her previous baskets. Most recent works [22,5] in this line adopt recurrent neural networks to model long-term sequential patterns in users' shopping baskets. However, our task differs from these works by concentrating on the MBA task, which models the shopping baskets without considering the order of the items in each. The work proposed in [15], for *product recommendation tasks*, need to store large knowledge graphs (i.e, co-occurrences matrices), thus they are not ideal to perform MBA using transaction streams. In [21,6,10], word2vec language model [17] has been adopted to learn product representations in an online fashion. Out of them, TRIPLE2VEC [21] is the most similar and recent work for our work, which jointly learns representa-

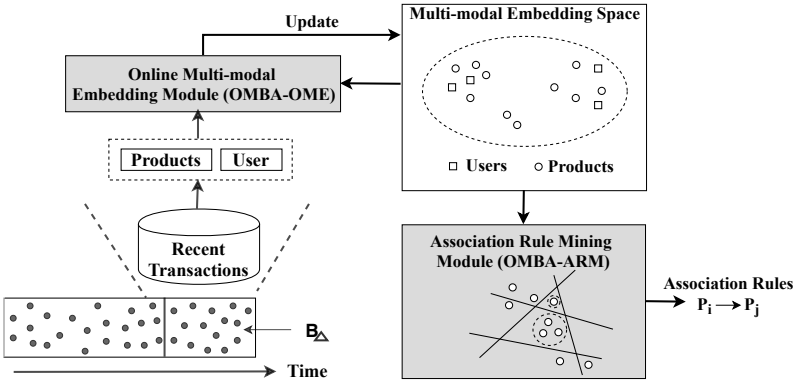


Fig. 3: OMBA consists of: (1) OMBA-OME to learn online product embeddings; and (2) OMBA-ARM to generate association rules using the embeddings

tion for products and users by adopting word2vec. However, Triple2Vec learns two embedding spaces for products such that each embedding space preserves semantic similarity (i.e., second-order proximity) as illustrated in Figure 2. Thus, the products' associations cannot be easily discovered from such an embedding space. In contrast, the embedding space of OMBA mainly preserves the complementarity (i.e., first-order proximity). Thus, the associated products are closely mapped in the embedding space as illustrated in Figure 2. Moreover, none of the aforementioned works proposes a method to generate association rules from products' embeddings, which is the ultimate objective of MBA.

Online Learning Techniques. OMBA learns online representations for products, giving importance to recent information to capture the temporal dynamics of products' associations. However, it is challenging to incrementally update representations using a continuous stream without overfitting to the recent records [23]. When a new set of records arrives, sampling-based online learning approaches in [23,20] sample a few historical records to augment the recent records and the representations are updated using the augmented corpus to alleviate overfitting to recent records. However, sampling-based approaches need to retain historical records. To address this limitation, a constraint-based approach is proposed in [23], which imposes constraints on embeddings to preserve their previous embeddings. However, sampling-based approaches are superior to constraint-based approaches. In this work, we propose a novel adaptive optimization technique to accommodate online learning that mostly outperforms sampling-based approaches without storing any historical records.

3 Problem Statement

Let $B = \{b_1, b_2, ..., b_N, ...\}$ be a continuous stream of shopping transactions (i.e., baskets) at a retailer that arrive in chronological order. Each basket $b \in B$ is a

tuple $< t^b, u^b, P^b >$, where: (1) t^b is the timestamp of b; (2) u^b is the user id of b; and (3) P^b is the set of products in b.

The problem is to learn the embeddings V_P for $P = \bigcup^{\forall b} P^b$ (i.e., the set of unique products in B) such that the embedding v_p of a product $p \in P$:

1. is a d-dimensional vector ($d << |P|$), where $|P|$ is the number of different products in B;
2. preserves the associations (i.e., co-occurrences) among products;
3. is continuously updated as new transactions (B_Δ) arrive to incorporate the latest information.

Market Basket Analysis (MBA) uncovers the associations among products in the form of *association rule*: $P_i \Rightarrow p_j$, "Consumers who buy the products in P_i are likely to buy the product p_j", where $P_i \subset P$ and $p_j \in P \backslash P_i$. The embedding learning module in OMBA is designed such that p_j and the products in P_i are mapped close together in the embedding space.

To quantitatively evaluate the embeddings that are learned for MBA, we focus on the ***intra-basket item retrieval task***, which is to retrieve the true product in a basket given the other attributes (i.e., user and other products) of the same basket. The embeddings should reflect the accurate associations between products to give better performance for this task.

4 OMBA

4.1 Overview of OMBA

OMBA consists of two modules as depicted in Figure 3: (1) Online Multi-Modal Embedding (OMBA-OME) module, which learns the representations for products; and (2) Association Rule Mining (OMBA-ARM) module, which generates the association rules using the representations from OMBA-OME.

To learn representations for products, OMBA-OME jointly embeds all the products and users into the same latent space. We conduct an empirical analysis using three datasets, which verifies that there is a significant similarity of the shopping baskets of a user (due to space limitations, more details of the empirical analysis are given in [4]). Thus, mapping users along with the products into a single embedding space allows us to exploit the user-specific patterns in shopping baskets to generalize the product representations.

The learning process of OMBA-OME proceeds in an online fashion, in which the embeddings are updated incrementally for the arrival of each of the new records B_Δ. Such an approach avoids the unnecessary cost of learning representations from scratch for each new arrival. However, this approach could lead to overfitting to recent information while abruptly forgetting (i.e., catastrophic forgetting) the previously learned information. To address this issue, we propose an online learning method in Section 4.2, which incorporates recent records effectively while alleviating the catastrophic forgetting without storing any historical records. Subsequently, our OMBA-ARM module adopts a clustering approach on the products' embedding space to extract association rules among the products.

4.2 Online Multi-Modal Embedding (OMBA-OME)

OMBA-OME learns the embeddings for units such that the units of a given basket b can be recovered by looking at b's other units. Formally, we model the likelihood for the task of recovering unit $z \in b$ given the other units b_{-z} of b as:

$$p(z|b_{-z}) = \exp(s(z, b_{-z}))/\sum_{z' \in X} \exp(s(z', b_{-z})) \tag{1}$$

X is the set of units (could be user or product) of type z, and $s(z, b_{-z})$ is the similarity score between z and b_{-z}. We define $s(z, b_{-z})$ as $v_z^\top h_z$ where,

$$h_z = \begin{cases} (v_{u^b} + v_{\hat{P^b}})/2 & \text{if } z \text{ is a product} \\ v_{\hat{P^b}} & \text{if } z \text{ is a user} \end{cases} \tag{2}$$

Value-Based Weighting. The representation for the context products ($v_{\hat{P^b}}$) in a basket is computed based on a novel weighting scheme. This weighting scheme emphasizes learning better representations for higher value items from their rare occurrences by assigning higher weight for them considering their selling price. Formally, the term $v_{\hat{P^b}}$ in Equation 2 is computed as:

$$v_{\hat{P^b}} = \frac{\sum_{x \in P_{b_{-z}}} g(x) v_x}{\sum_{x \in P_{b_{-z}}} g(x)} \tag{3}$$

where $g(x)$ function returns a weight for product x based on its selling price, $SV(x)$. The function g is computed as follows:

- Assuming the number of appearances of x follows a power-law distribution with respect to its selling price $SV(x)$ (see Figure 1a), a power-law formula (i.e., $y = cx^{-k}$) is fitted to the curve in Figure 1a, which returns the probability of product x appearing in a basket b, $p(x \in b)$, as $1.3 * SV(x)^{-2.3}$ (the derivation of this formula is presented in [4]);
- Then, the function g is computed as:

$$g(x) = \frac{1}{p(x \in b)} = \frac{1}{1.3 * min(SV(x), 10)^{-2.3}} \tag{4}$$

  The function g is clipped for the products with selling price > 10 (10 is selected as the point at which the curve in Figure 1a reaches a plateau) to avoid the issue of exploding gradient.

Adaptive Optimization. Then, the final loss function is the negative log likelihood of recovering all the attributes of the recent transactions B_Δ:

$$O_{B_\Delta} = -\sum_{b \in B_\Delta} \sum_{z \in b} p(z|b_{-z}) \tag{5}$$

The objective function above is approximated using negative sampling (proposed in [17]) for efficient optimization. Then for a selected record b and unit

$z \in b$, the loss function based on the reconstruction error is:

$$L = -\log(\sigma(s(z, b_{-z}))) - \sum_{n \in N_z} \log(\sigma(-s(n, b_{-z}))) \tag{6}$$

where N_z is the set of randomly selected negative units that have the type of z and $\sigma(.)$ is sigmoid function.

We adopt a novel optimization strategy to optimize the loss function, which is designed to alleviate overfitting to the recent records and the frequently appearing products in the transactions.

For each basket b, we compute the intra-agreement Ψ_b of b's attributes as:

$$\Psi_b = \frac{\sum_{z_i, z_j \in b, z_i \neq z_j} \sigma(v_{z_i}^\top v_{z_j})}{\sum_{z_i, z_j \in b, z_i \neq z_j} 1} \tag{7}$$

Then the adaptive learning rate of b is calculated as,

$$lr_b = \exp(-\tau \Psi_b) * \eta \tag{8}$$

where η denotes the standard learning rate and τ controls the importance given to Ψ_b. If the representations have already overfitted to b, then Ψ_b takes a higher value. Consequently, a low learning rate is assigned to b to avoid overfitting. In addition, the learning rate for each unit z in b is further weighted using the approach proposed in AdaGrad [8] to alleviate the overfitting to frequent items. Then, the final update of the unit z at the t^{th} timestep is:

$$z_{t+1} = z_t - \frac{lr_b}{\sqrt{\sum_{i=0}^{t-1} (\frac{\partial L}{\partial z})_i^2 + \epsilon}} (\frac{\partial L}{\partial z})_t \tag{9}$$

4.3 Association Rule Mining (OMBA-ARM)

The proposed OMBA-OME module learns representations for products such that the frequently co-occurring products (i.e., products with strong associations) are closely mapped in the embedding space. Accordingly, the proposed OMBA-ARM module clusters in the products' embedding space to detect association rules between products. The product embeddings from the OMBA-OME module are in a higher-dimensional embedding space. Thus, Euclidean-distance based clustering algorithms (e.g., K-Means) suffers from the curse-of-dimensionality issue. Hence, we adopt a Locality-Sensitive Hashing (LSH) algorithm based on random projection [3], which assigns similar hash values to the products that are close in the embedding space. Subsequently, the products with the most similar hash values are returned as strong associations. Our algorithm is formally elaborated as the following five steps:

1. Create $|F|$ different hash functions such as $F_i(v_p) = sgn(f_i \cdot v_p^\top)$, where $i \in \{0, 1, \ldots, |F| - 1\}$ and f_i is a d-dimensional vector from a Gaussian distribution $N \sim (0, 1)$, and $sgn(.)$ is the sign function. According to the Johnson-Lindenstrauss lemma [14], such hash functions approximately preserve the distances between products in the original embedding space.

2. Construct an $|F|$-dimensional hash value for v_p (i.e., a product embedding) as $F_0(v_p) \oplus F_1(v_p) \oplus ... \oplus F_{|F|-1}(v_p)$, where $\oplus$ defines the concatenation operation. Compute the hash value for all the products.
3. Group the products with similar hash values to construct a hash table.
4. Repeat steps (1), (2), and (3) $|H|$ times to construct $|H|$ hash tables to mitigate the contribution of bad random vectors.
5. Return the sets of products with the highest collisions in hash tables as the products with strong associations.

$|H|$ and $|F|$ denote the number of hash tables and number of hash functions in each table respectively.

Finding optimal values for $|H|$ and $|F|$. In this section, the optimal values for $|F|$ and $|H|$ are found such that they guarantee that the rules generated from OMBA-ARM have higher *Lift*. Initially, we form two closed-form solutions to model the likelihood to have a strong association between product x and product y based on: (1) OMBA-ARM; and (2) LIFT.

(1) Based on OMBA-ARM, products x and y should collide at least in a single hash table to have a strong association. Thus, the likelihood to have a strong association between product x and product y is[3]:

$$\begin{aligned} p(x \Rightarrow y)_{omba} &= 1 - (1 - p(sgn(v_x \cdot f) = sgn(v_y \cdot f))^{|F|})^{|H|} \\ &= 1 - (1 - (1 - \frac{arccos(v_x \cdot v_y)}{\pi})^{|F|})^{|H|} \end{aligned} \tag{10}$$

where $sgn(x) = \{1 \; if \; x \geq 1; -1 \; otherwise\}$. $p(sgn(v_x \cdot f) = sgn(v_y \cdot f))$ is the likelihood to have a similar sign for products x and y with respect to a random vector. v_x and v_y are the normalized embeddings of product x and y respectively.

(2) Based on Lift, the likelihood to have a strong association between products x and y can be computed as[3]:

$$p(x \Rightarrow y)_{lift} = \frac{Lift(y,x)_{train}}{Lift(y,x)_{train} + |N_z| * Lift(y,x)_{noise}} = \sigma(A * v_x \cdot v_y) \tag{11}$$

where $Lift(y,x)_{train}$ and $Lift(y,x)_{noise}$ are the *Lift* scores of x and y calculated using the empirical distribution of the dataset and the noise distribution (to sample negative samples) respectively. $|N_z|$ is the number of negative samples for each genuine sample in Equation 6, and A is a scalar constant.

Then, the integer solutions for parameters $|F| = 4$ and $|H| = 11$, and real solutions for $A = 4.3$ are found such that $p(x \Rightarrow y)_{omba} = p(x \Rightarrow y)_{lift}$. Such a selection of hyper-parameters theoretically guarantees that the rules produced from OMBA-ARM are higher in *Lift*, which is a statistically well-defined measure for strong associations.

[3] Detailed derivations of Eq. 10 and Eq. 11 are presented in [4]

Table 1: Descriptive statistics of the datasets

Datasets	# Users	# Items	# Transactions	# Baskets
Complete Journey (CJ)	2,500	92,339	2,595,733	276,483
Ta-Feng (TF)	9,238	7,973	464,118	77,202
Instacart (IC)	206,209	49688	33,819,306	3,421,083

The advantages of our OMBA-ARM module can be listed as follows: (1) it is simple and efficient, which simplifies the expensive comparison of all the products in the original embedding space ($O(dN^2)$ complexity) to $O(d|F||H|N)$, where $N \gg |F||H|$; (2) it approximately preserves the distances in the original embedding space, as the solutions for $|F|(= 4)$ and $|H|(= 11)$ satisfy the Johnson–Lindenstrauss lemma [14] ($|F||H| \gg log(N)$); (3) it outperforms underlying Euclidean norm-based clustering algorithms (e.g., K-means) for clustering data in higher dimensional space, primarily due to the curse of dimensionality issue [1]; and (4) theoretically guarantees that the rules have higher *Lift*.

5 Experimental Methodology

Datasets. We conduct all our experiments using three publicly available real-world datasets:

- Complete Journey Dataset(CJ) contains household-level transactions at a retailer by 2,500 frequent shoppers over two years.
- Ta-Feng Dataset(TF) includes shopping transactions of the Ta-Feng supermarket from November 2000 to February 2001.
- InstaCart Dataset(IC) contains the shopping transactions of Instacart, an online grocery shopping center, in 2017.

The descriptive statistics of the datasets are shown in Table 1. As can be seen, the datasets have significantly different statistics (e.g., TF has a shorter collection period, and IC has a larger user base), which helps to evaluate the performance in different environment settings. The time-space is divided into 1-day time windows, meaning that new records are received by the model once per day.

Baselines. We compare OMBA with the following methods:

- POP recommends the most popular products in the training dataset. This is shown to be a strong baseline in most of the domains, despite its simplicity.
- SUP recommends the product with the highest *Support* for a given context.
- LIFT recommends the product that has the highest *Lift* for a given context.
- NMF [16] performs Non-Negative Matrix Factorization on the user-product co-occurances matrix to learn representations for users and products.
- ITEM2VEC [6] adopts word2vec for learning the product representations by considering a basket as a sentence and a product in the basket as a word in the sentence.
- PROD2VEC [10] is the BAGGED-PROD2VEC version, which aggregates all the baskets related to a single user as a product sequence and apply word2vec to learn representations for products.

– Triple2Vec [21] joinlty learns representations for products and users such that they preserve the triplets in the form of $< user, item_1, item_2 >$, which are generated using shopping baskets.

Also, we compare with a few variants of OMBA.

– OMBA-No-User does not consider users. The comparison with OMBA-No-User highlights the importance of users.
– OMBA-Cons adopts SGD optimization with the constraint-based online learning approach proposed in [23].
– OMBA-Decay and OMBA-Info adopt SGD optimization with the sampling-based online learning methods proposed in [23] and [20] respectively.

Parameter Settings. All the representation learning based techniques share three common parameters (default values are given in brackets): (1) the latent embedding dimension d (300), (2) the SGD learning rate η (0.05), (3) the negative samples $|N_z|$ (3) as appeared in Equation 6, and (4) the number of epochs N (50). We set $\tau = 0.1$ after performing grid search using the CJ dataset (see [4] for a detailed study of parameter sensitivity). For the specific parameters of the baselines, we use the default parameters mentioned in their original papers.

Evaluation Metric. Following the previous work [23], we adopt the following procedure to evaluate the performance for the *intra-basket item retrieval task*. For each transaction in the test set, we select one product as the target prediction and the rest of the products and the user of the transaction as the context. We mix the ground truth target product with a set of M negative samples (i.e., products) to generate a candidate pool to rank. M is set to 10 for all the experiments. Then the size-$(M + 1)$ candidate pool is sorted to get the rank of the ground truth. The average similarity of each candidate product to the context of the corresponding test instance is used to produce the ranking of the candidate pool. Cosine similarity is used as the similarity measure of OMBA, Triple2Vec, Prod2Vec, Item2Vec and NMF. POP, Sup, and Lift use popularity, support, and lift scores as similarity measures respectively.

If the model is well trained, then higher ranked units are most likely to be the ground truth. Hence, we use three different evaluation metrics to analyze the ranking performance: (1) Mean Reciprocal Rank (MRR) $= \frac{\sum_{q=1}^{Q} 1/rank_i}{|Q|}$; (2) Recall@k (R@k) $= \frac{\sum_{q=1}^{Q} min(1, \lfloor k/rank_i \rfloor)}{|Q|}$; and (3) Discounted Cumulative Gain (DCG) $= \frac{\sum_{q=1}^{Q} 1/log_2(rank_i+1)}{|Q|}$, where Q is the set of test queries and $rank_i$ refers the rank of the ground truth label for the i-th query. $\lfloor . \rfloor$ is the floor operation. A good ranking performance should yield higher values for all three metrics. We randomly select 20 one-day query windows from the second half of the period for each dataset, and all the transactions in the randomly selected time windows are used as test instances. For each query window, we only use the transactions that arrive before the query window to train different models. Only OMBA and its variants are trained in an online fashion and all the other baselines are trained in a batch fashion for 20 repetitions.

6 Results

Intra-basket Item Retrieval Task. Table 2 shows the results collected for the *intra-basket item retrieval* task. OMBA and its variants show significantly better results than the baselines, outperforming the best baselines on each dataset by as much as 10.51% for CJ, 21.28% for IC, and 8.07% for TF in MRR.

(1) Capture semantics of products. LIFT is shown to be a strong baseline. However, it performs poorly for IC, which is much sparser with a large userbase compared to the other datasets. This observation clearly shows the importance of using representation learning based techniques to overcome data sparsity by capturing the semantics of products. We further analyzed the wrongly predicted test instances of LIFT, and observed that most of these instances are products with significant seasonal variations such as "Rainier Cherries" and "Valentine Gifts and Decorations" (as shown in Figure 1c). This shows the importance of capturing temporal changes of the products' associations.

Out of the representation learning-based techniques, TRIPLE2VEC is the strongest baseline, despite being substantially outperformed by OMBA. As elaborated in Section 2, TRIPLE2VEC mainly preserves semantic similarity between products. This could be the main reason for the performance difference between TRIPLE2VEC and OMBA. To validate that, Table 4a lists the nearest neighbours for a set of products in the embedding spaces from OMBA and TRIPLE2VEC. Most of the nearest neighbours in TRIPLE2VEC are semantically similar to the target products. For example, 'Turkey' has substitute products like 'Beef' and 'Chicken' as its nearest neighbours. In contrast, the embedding space of OMBA mainly preserve complementarity. Thus, multiple related products to fulfill a specific need are closely mapped. For example, 'Layer Cake' has neighbours related to a celebration (e.g., Cake Decorations and Cake Candles).

(2) Value-based weighting. To validate the importance of the proposed value-based weighting scheme in OMBA, Table 4b shows the deviation of the performance for *intra-basket item retrieval* with respect to the selling price of the ground truth products. With the proposed value-based weighting scheme, OMBA accurately retrieves the ground truth products that have higher selling prices. Thus, we can conclude that the proposed value-based weighting scheme is important to learn accurate representations for rarely occurring products.

(3) Online learning. Comparing the variants of OMBA, OMBA substantially outperforms OMBA-NO-USER, showing that users are important to model products' associations. OMBA's results are comparable (except 27.68% performance boost for TF) with sampling-based online learning variants of OMBA (i.e., OMBA-DECAY and OMBA-INFO), which store historical records to avoid overfitting to recent records. Hence, the proposed adaptive optimization-based online learning technique in OMBA achieves the performance of the state-of-the-art online learning methods (i.e., OMBA-DECAY and OMBA-INFO) in a memory-efficient manner without storing any historical records.

Association Rule Mining. To compare the association rules generated from OMBA, we generate the association rules for the CJ dataset using: (1) Apriori algorithm [2] with *minimum Support=0.0001*; (2) TopK Algorithm [9]

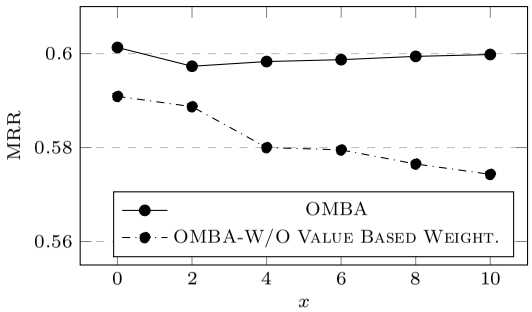
Table 2: The comparison of different methods for *intra-basket item retrieval*. Each model is evaluated 5 times with different random seeds and the mean value for each model is presented. Recall values for different k values are presented in [4] due to space limitations

	Results for *intra-basket item retrieval*									Memory Complexity
Dataset	CJ			IC			TF			
Metric	MRR	R@1	DCG	MRR	R@1	DCG	MRR	R@1	DCG	
POP	0.2651	0.095	0.4295	0.2637	0.07841	0.4272	0.2603	0.0806	0.4247	$O(P + \|B_{max}\|)$
SUP	0.3308	0.1441	0.4839	0.3009	0.1061	0.4634	0.3475	0.1646	0.4972	$O(P^2 + \|B_{max}\|)$
LIFT	0.5441	0.3776	0.6477	0.4817	0.2655	0.6170	0.4610	0.2981	0.5868	$O(P(P+1) + \|B_{max}\|)$
NMF	0.1670	0.0000	0.3565	0.5921	0.3962	0.6922	0.4261	0.2448	0.5591	$O(P(U+P) + \|B_{max}\|)$
ITEM2VEC	0.4087	0.2146	0.5457	0.5159	0.2929	0.6137	0.3697	0.1782	0.5149	$O(k(2 * P) + \|B\|)$
PROD2VEC	0.4234	0.2275	0.5575	0.5223	0.3222	0.6363	0.3764	0.1854	0.5201	$O(k(2 * P) + \|B\|)$
TRIPLE2VEC	0.5133	0.3392	0.6269	0.6169	0.4348	0.7095	0.478	0.3040	0.5990	$O(k(U + 2 * P) + \|B\|)$
OMBA-NO-USER	0.4889	0.3091	0.6004	0.5310	0.3384	0.6405	0.3873	0.2013	0.5298	$O(kP + \|B_{max}\|)$
OMBA-CONS	0.4610	0.2742	0.5843	0.5942	0.3393	0.6998	0.3996	0.2031	0.5324	$O(k(U+P) + \|B_{max}\|)$
OMBA-DECAY	0.5984	0.4221	0.6948	0.7117	0.5442	0.7860	0.402	0.2186	0.5387	$O(k(U+P) + \|B_{max}\|/(1-e^{-\tau}))$
OMBA-INFO	0.5991	0.4275	0.6937	**0.7482**	0.5852	**0.8027**	0.4046	0.2205	0.5421	$O(k(U+P) + \|B_{max}\|/(1-e^{-\tau}))$
OMBA	**0.6013**	**0.4325**	**0.6961**	0.7478	**0.5859**	0.8025	**0.5166**	**0.3466**	**0.6293**	$O(k(U+P) + \|B_{max}\|)$

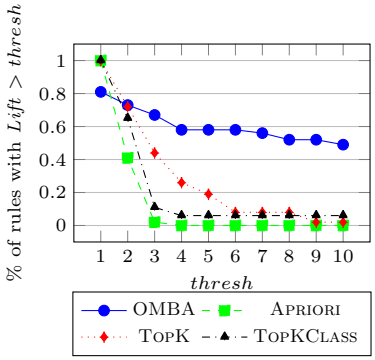
Table 3: (a) 5 nearest neighbours in the embedding space of OMBA and TRIPLE2VEC for a set of target products; (b) MRR for *intra-basket item retrieval* using the test queries, such that the ground truth's price $> x$ (in dollars). All the results are calculated using CJ

Target Product	5 nearest products by OMBA	5 most nearest products by TRIPLE2VEC
Layer Cakes	Cake Candles, Cake Sheets, Cake Decorations, Cake Novelties, Birthday/Celebration Layer	Cheesecake, Fruit/Nut Pies, Cake Candles, Flags, Salad Ingredients
Frozen Bread	Sauces, Eggs, Peanut Butter, Non Carbohydrate Juice, Pasta/Ramen	Frozen Breakfast, Breakfast Bars, Popcorn, Gluten Free Bread, Cookies/Sweet Goods
Turkey	Beef, Meat Sauce, Oil/Vinegar, Cheese, Ham	Ham, Beef, Cheese, Chicken, Lunch Meat
Authentic Indian Foods	Grain Mixes, Organic Pepper, Other Asian Foods, Tofu, South American Wines	Other Asian Foods, German Foods, Premium Mums, Herbs & Fresh Others, Bulb Sets

(a)



(b)



(a)

Produced Rules by OMBA	Lift
(Layer Cake Mix, Frosting)	52.36
(Eggs, Can/Bottle Beverages)	94.33
(Rose Bouquet, Standard 10-15 Stems)	0.0
(Smoke Detectors, Drywall Repair)	0.0
(Facial Creams, Sink Accessories)	0.0
(Facial Soaps, Sink Accessories)	12.41
(Facial Soaps, Facial Creams)	30.62
(Dental Floss/Implements, Toothpaste)	10.82
(Acne Medications, Facial Creams)	22.3
(Mexican Seasoning Mix, Tostado Shells)	43.45
(Mens Skin Care, Razors and Blades)	20.61
(Eggs, Organic Dried Fruits)	161.96
(Cabbage, Spinach)	9.18
(Cabbage, Ham)	7.54

(b)

Fig. 4: (a) The percentage of top 100 rules that have *Lift* scores > *thresh*; and (b) A few examples for the top 100 rules produced by OMBA

Table 4: Association Rules generated by OMBA-ARM on different days

	Mostly Associated Products by OMBA-ARM	
Target Product	At Month 11	At Month 17
Valentine Gifts and Decorations	Rose Consumer Bunch, Apple Juice, Ladies Casual Shoes	Peaches, Dried Plums, Drain Care
Mangoes	Blueberries, Dry Snacks, Raspberries	Dry Snacks, Red Cherries, Tangelos
Frozen Bread	Ham, Fluid Milk, Eggs	Peanut Butter, Eggs, Ham

with *k=100*; and (3) TopKClass Algorithm[4] with *k=100* and *consequent list=* {*products appear in the rules from* OMBA}. Figure 4a shows *Lift* scores of the top 100 rules generated from each approach.

(1) Emphasize rarely occurring associations. As can be seen, at most 20% of the rules generated from the baselines have *Lift* scores greater than 5, while around 60% of the rules of OMBA have scores greater than 5. This shows that OMBA generates association rules that are strong and rarely occurring in the shopping baskets (typically have high *Lift* scores due to low *Support* values), which are not emphasized in conventional MBA techniques.

(2) Discover unseen associations. However, Figure 4a shows that some of the rules from OMBA have *Lift* scores less than 1 (around 20% of the rules). Figure 4b lists some of the rules generated from OMBA, including a few with $Lift < 1$. The qualitative analysis of these results shows that OMBA can even discover unseen associations in the dataset (i.e., associations with *Lift*= 0) as shown in Figure 4b, by capturing the semantics of product names. The associations with *Lift*= 0 like "(Rose Bouquet, Standard 10-15 Stems)" and "(Smoke Detectors, Drywall Repair)" can be interpreted intuitively, but they would be omitted by conventional count-based approaches. The ability to capture the semantics of products enables OMBA to mitigate the *cold-start problem* and recommend products to customers that they have never purchased. The role of

[4] `https://bit.ly/spmf_TopKClassAssociationRules`

the OMBA-OME module on capturing products' semantics can be further illustrated using the rule "(Facial Creams, Sink Accessories)", which is unseen in the CJ dataset. However, both "Sink Accessories" and "Facial Creams" frequently co-occurred with "Facial Soaps", with 12.41 and 30.02 *Lift* scores. Hence, the representations for "Sink Accessories" and "Facial Creams" learnt by OMBA-OME are similar to the representation of "Facial Soaps" to preserve their co-occurrences. This leads to similar representations for "Sink Accessories" and "Facial Creams" and generates the rule. In contrast, conventional count-based approaches (e.g., Apriori, H-Mine, and TopK) are unable to capture these semantics of product names as they consider each product as an independent unit.

(3) Temporal variations of associations. Moreover, Table 4 shows that OMBA adopts associations rules to the temporal variations. For example, most of the associated products for 'Mangoes' at month 11 are summer fruits (e.g., 'Blueberries' and 'Raspberries'); 'Mangoes' are strongly associated with winter fruits (e.g., 'Cherries' and 'Tangelos') at Month 17. However, everyday products like 'Frozen Bread' show consistent associations with other products. These results further signify the importance of learning online product representation.

In summary, the advantages of OMBA are three-fold: (1) OMBA explores the whole products' space effectively to detect rarely-occurring non-trivial strong associations; (2) OMBA considers the semantics between products when learning associations, which alleviates the cold-start problem; and (3) OMBA learns products' representations in an online fashion, thus is able to capture the temporal changes of the products' associations accurately.

7 Conclusion

We proposed a scalable and effective method to perform Market Basket Analysis in an online fashion. Our model, OMBA, introduced a novel online representation learning technique to jointly learn embeddings for users and products such that they preserve their associations. Following that, OMBA adapted an effective clustering approach in high dimensional space to generate association rules from the embeddings. Our results show that OMBA manages to uncover non-trivial and temporally changing association rules at the finest product levels, as OMBA can capture the semantics of products in an online fashion. Nevertheless, extending OMBA to perform MBA for multi-level products and multi-store environments could be a promising future direction to alleviate data sparsity at the finest product levels. Moreover, it is worthwhile to explore sophisticated methods to incorporate users' purchasing behavior to improve product representations.

References

1. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the Surprising Behavior of Distance Metrics in High Dimensional Space. In: Proc. of ICDT (2001)
2. Agrawal, R., Srikant, R., et al.: Fast Algorithms for Mining Association Rules. In: Proc. of VLDB (1994)

3. Andoni, A., Indyk, P.: Near-optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions. In: Proc. of FOCS (2006)
4. Silva, A., Luo, L., Karunasekera, S., Leckie, C.: Supplementary Materials for OMBA: On Learning User-Guided Item Representations for Online Market Basket Analysis. `https://bit.ly/30yGMbx` (2020)
5. Bai, T., Nie, J.Y., Zhao, W.X., Zhu, Y., Du, P., Wen, J.R.: An Attribute-aware Neural Attentive Model for Next Basket Recommendation. In: Proc. of SIGIR (2018)
6. Barkan, O., Koenigstein, N.: Item2vec: Neural Item Embedding for Collaborative Filtering. In: Proc. of MLSP (2016)
7. Deng, Z.H., Lv, S.L.: Fast Mining Frequent Itemsets using Nodesets. Expert Systems with Applications (2014)
8. Duchi, J., Hazan, E., Singer, Y.: Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. In: Proc. of COLT (2010)
9. Fournier-Viger, P., Wu, C.W., Tseng, V.S.: Mining Top-k Association Rules. In: Proc. of CanadianAI (2012)
10. Grbovic, M., Radosavljevic, V., Djuric, N., Bhamidipati, N., Savla, J., Bhagwan, V., Sharp, D.: E-commerce in your Inbox: Product Recommendations at Scale. In: Proc. of SIGKDD (2015)
11. Han, J., Fu, Y.: Discovery of Multiple-Level Association Large Databases. In: Proc. of VLDB (1995)
12. Han, J., Pei, J., Yin, Y., Mao, R.: Mining Frequent Patterns without Candidate Generation: A Frequent-pattern Tree Approach. Data Mining and Knowledge Discovery (2004)
13. Hariharan, S., Kannan, M., Raguraman, P.: A Seasonal Approach for Analysis of Temporal Trends in Retail Marketing using Association Rule Mining. International Journal of Computer Applications (2013)
14. Johnson, W.B., Lindenstrauss, J.: Extensions of Lipschitz Mappings into a Hilbert Space. Contemporary Mathematics (1984)
15. Le, D.T., Lauw, H.W., Fang, Y.: Basket-sensitive Personalized Item Recommendation. In: Proc. of IJCAI (2017)
16. Lee, D.D., Seung, H.S.: Algorithms for Non-negative Matrix Factorization. In: Proc. of NIPS (2001)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In: Proc. of NIPS (2013)
18. Papavasileiou, V., Tsadiras, A.: Time Variations of Association Rules in Market Basket Analysis. In: Proc. of AIAI (2011)
19. Reshu Agarwal, Mittal, M.: Inventory Classification using Multilevel Association Rule Mining. International Journal of Decision Support System Technology (2019)
20. Silva, A., Karunasekera, S., Leckie, C., Luo, L.: USTAR: Online Multimodal Embedding for Modeling User-Guided Spatiotemporal Activity. In: Proc. of IEEE-BigData (2019)
21. Wan, M., Wang, D., Liu, J., Bennett, P., McAuley, J.: Representing and Recommending Shopping Baskets with Complementarity, Compatibility and Loyalty. In: Proc. of CIKM (2018)
22. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T.: A Dynamic Recurrent Model for Next Basket Recommendation. In: Proc. of SIGIR (2016)
23. Zhang, C., Zhang, K., Yuan, Q., Tao, F., Zhang, L., Hanratty, T., Han, J.: React: Online Multimodal Embedding for Recency-aware Spatiotemporal Activity Modeling. In: Proc. of SIGIR (2017)