Singapore Management University
Institutional Knowledge at Singapore Management University

8-2019

Correlation-sensitive next-basket recommendation

Duc Trong LE
Singapore Management University, ductrong.le.2014@phdis.smu.edu.sg

Hady Wirawan LAUW
Singapore Management University, hadywlauw@smu.edu.sg

Yuan FANG
Singapore Management University, yfang@smu.edu.sg

Citation

Correlation-Sensitive Next-Basket Recommendation

Duc-Trong Le , **Hady W. Lauw** and **Yuan Fang**

School of Information Systems, Singapore Management University, Singapore

{ductrong.le.2014, hadywlauw, yfang}@smu.edu.sg

Abstract

Items adopted by a user over time are indicative of the underlying preferences. We are concerned with learning such preferences from observed sequences of adoptions for recommendation. As multiple items are commonly adopted concurrently, e.g., a basket of grocery items or a sitting of media consumption, we deal with a sequence of baskets as input, and seek to recommend the next basket. Intuitively, a basket tends to contain groups of related items that support particular needs. Instead of recommending items independently for the next basket, we hypothesize that incorporating information on pairwise correlations among items would help to arrive at more coherent basket recommendations. Towards this objective, we develop a hierarchical network architecture codenamed *Beacon* to model basket sequences. Each basket is encoded taking into account the relative importance of items and correlations among item pairs. This encoding is utilized to infer sequential associations along the basket sequence. Extensive experiments on three public real-life datasets showcase the effectiveness of our approach for the next-basket recommendation problem.

1 Introduction

To cope with the astounding and escalating number of options facing us, involving the selection of products, news, movies, music, points of interest, etc., a recommender system offers the most, if not the only, pragmatic way for finding an item of interest. In the literature, there are several major bases for recommendation. One is personalization, undergirded by user-specific parameters. Another is association among items, i.e., given items that have been adopted thus far, which other items shall be recommended. Our focus in this work is the latter.

One form of association among items is sequential [Quadrana *et al.*, 2018]. A sequence of items adopted over time carries signals about the underlying preferences that bear clues for future adoptions. For instance, someone who has been listening to a music genre may likely be interested in new songs of that genre. Previous restaurant visits may have a bearing on
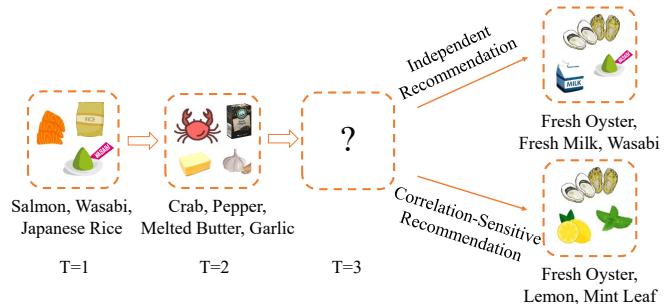

Figure 1: Motivating example for correlation-sensitive next-basket recommendation.

future dining choices. The essence is thus preference driven by sequentiality, rather than personalization per se.

In many scenarios we adopt more than one item at a time. We listen to a few songs in the same sitting, use several tags to label things, run a few errands in the same trip, purchase multiple products in the same shopping cart, etc. We refer to a collection of items adopted concurrently as a "basket". Frequently, some items within a basket are correlated to a certain extent. This is because these items may arise from the same underlying need, e.g., ingredients for the same recipe, tags describing the same object. Hence we are really dealing not with sequences of items, but rather with sequences of baskets.

In this work, we address the problem of *next-basket recommendation*. Given a sequence of baskets adopted by a user as input, our objective is to predict a set of items that are likely to belong in the next basket. Figure 1 illustrates this in the context of grocery shopping. In this case, each time step corresponds to a shopping session. In the first session ($T = 1$), the basket of {*Salmon, Wasabi, Japanese rice*} implies a latent intention of making sushi. The second session ($T = 2$) likely concerns a crab-based recipe with the combination of {*Crab, Pepper, Melted Butter, Garlic*}. The sequentiality hints at an underlying preference for a seafood diet. In Figure 1, the problem is to predict the basket at $T = 3$.

There have been active efforts towards next-basket recommendation. One is to rely only on the most recently purchased basket to predict the next basket [Rendle *et al.*, 2010; Wang *et al.*, 2015; Wan *et al.*, 2018]. This may be applicable in short-term dependency scenarios, but it may not capture underlying preferences as well as a method that looks further

back into history. Hence, another approach is to capture the long-term sequential dependencies using methods such as recurrent neural network (RNN) [Yu *et al.*, 2016]. In any case, these existing approaches arrive at the recommended items for the next basket independently, based only on their respective associations with the *past* basket(s), disregarding the collective associations among items to be recommended.

We postulate that a basket tends to contain coalitions of related items, rather than independent items. Thus, if the objective is to predict items that belong in the next basket, then we should factor in the correlations among those items in our modeling as well as prediction. For example, while independent recommendations in Figure 1 may capture the long-term preference for seafood (predicting oyster), the other recommended items may be unrelated yet popular items such as milk and wasabi. In contrast, taking into account that purchasing an item, e.g., oysters, tends to inspire the purchase of other correlated items, then a *correlation-sensitive* next-basket recommendation may favor items frequently eaten or purchased together with oysters, e.g., lemon and mint.

Contributions. Towards realizing this intuition, we incorporate information on item correlations for next-basket recommendation. To our best knowledge, we are the first to consider correlations among predicted items for this problem, which is our *first contribution*. In Section 3, we formalize this problem, and discuss how item correlations may be obtained. As a *second contribution*, in Section 4 we further describe a novel hierarchical network architecture called *Basket Sequence Correlation Network* (codenamed *Beacon*), which learns the representations of each basket leading to the overall representation of a basket sequence that could be used for next-basket prediction. This model is built on a couple of principles in deriving the representations. For one, individual items in a basket are differentially important, depending on their frequencies as well as efficacies in drawing other items. For another, item pairs in a basket are differentially related, with some having stronger or more exclusive connections. As our *final contribution*, in Section 5 we conduct extensive experiments on three real-life datasets of different domains. The results show that *Beacon*'s modeling of item correlations produce significant improvements over baselines.

2 Related Work

Here we review several classes of previous work related to sequential as well as basket-oriented recommendations.

Item Sequences. One class of approaches are concerned with sequential dependencies among individual items. Some rely on Markov chains to model short-term dependencies using either factorization [Rendle *et al.*, 2010] or Euclidean embedding [Chen *et al.*, 2012] techniques. Others model long-term dependencies using RNN [Hidasi *et al.*, 2016; Li *et al.*, 2017; Villatel *et al.*, 2018], convolutional neural network or CNN [Tang and Wang, 2018], memory networks [Huang *et al.*, 2018], translation-based method [He *et al.*, 2017], or session graphs [Xiang *et al.*, 2010; Wu *et al.*, 2019; Song *et al.*, 2019]. These works are not comparable to ours, as they operate at item level and consider neither basket sequences nor next-basket recommendation.

Symbol	Description
V	the set of items $\{1, 2, \ldots, |V|\}$
S	a temporal sequence of baskets $\langle B_1^{(S)}, \ldots, B_{\ell(S)}^{(S)} \rangle$
C	item correlation matrix
$\boldsymbol{\omega}$	item importance parameters
$B_t, \mathbf{x}_t$	basket at time t and its binary representation
$\mathbf{z}_t, \mathbf{b}_t$	intermediate and latent representations of B_t
$\mathbf{h}_t$	recurrent hidden output at time t
Φ, ϕ	weight and bias parameters in basket encoder
Ψ, Ψ', ψ	weight and bias parameters in sequence encoder
Γ	weight parameters in predictor
$\mathbf{s}^{(S)}$	sequential signal given sequence S
$\mathbf{y}^{(S)}$	predicted item scores given sequence S

Table 1: Summary of Main Notations

Basket Sequences. There have been efforts to model basket-level adoptions for sequential recommendation, but in general they do not incorporate item correlation information within their modeling nor prediction of baskets. For instance, [Yu *et al.*, 2016] encodes each basket and learns the sequence representation via a RNN-based approach. Later, [Bai *et al.*, 2018] improves this approach by incorporating item attributes. In turn [Le *et al.*, 2018] makes use of secondary supporting sequences. To showcase the benefit of item correlation information, we will compare to [Yu *et al.*, 2016] and [Le *et al.*, 2018] (focusing on primary sequence) as baselines. There are also personalized methods [Wang *et al.*, 2015; Ying *et al.*, 2018; Wan *et al.*, 2018], which are not directly comparable as we learn representations from sequences without the presumption of user-specific parameters. For completeness, we will compare to [Wan *et al.*, 2018], focusing the comparison on the sequential and basket effects alone.

Baskets. In an orthogonal direction to ours are a class of techniques focusing solely on basket-level associations. [Sarwar *et al.*, 2000] relies on association rules. [Pathak *et al.*, 2017] seek to recommend bundles. [Le *et al.*, 2017; Wang *et al.*, 2018] attempt basket completion, with existing basket items as context to predict the remaining item. [Li *et al.*, 2009] applies random walks on user-item bipartite graph to generate basket-sensitive item recommendations. There are several works that exploit item-item associations [Ning and Karypis, 2011] and itemset-item associations [Christakopoulou and Karypis, 2014] for similarity-based recommendations, not in the complementary manner as ours.

3 Preliminaries

In this section, we formalize our problem and introduce the formulation of correlation matrix. We summarize the main notations in Table 1, including those to be introduced later.

3.1 Problem Statement

We first introduce some background concepts. Assume a set of items $V = \{1, 2, \ldots, |V|\}$. Several items can form a *basket*, which is essentially a set of items, denoted as

$B = \{i_1, i_2, \ldots, i_{|B|}\}$ where i_k's are distinct integers in $\{1, \ldots, |V|\}$. Note that baskets may have variable sizes. In real-world applications, a basket could be derived from products purchased in a retail transaction, websites surfed in a browser session, or places visited in a trip.

In our problem, as the first input, we assume a set of sequences $\mathcal{S}$. Each sequence $S \in \mathcal{S}$ is a temporally ordered list of basket $S = \langle B_1^{(S)}, B_2^{(S)}, \ldots, B_{\ell(S)}^{(S)} \rangle$ such that $B_t^{(S)}$ happens at time t and $\ell(S)$ is the length of the sequence. Hereafter, when the context is clear, for brevity we will omit the basket superscript $^{(S)}$ which indicates that the basket belongs to sequence S. Note that sequences may have variable lengths, and they are divided into train and test sets.

As the second input, we assume a correlation matrix $C \in \mathbb{R}^{|V| \times |V|}$. If two items i and j tend to co-occur with each other in a basket, C_{ij} should be higher. We will elaborate on the construction of this matrix in Section 3.2.

As output, for a test sequence $S = \langle B_1, \ldots, B_{\ell(S)} \rangle$, we aim to predict the next basket $B_{\ell(S)+1}$ as the recommendation. Ideally, some, if not most, of the predicted items should be related to form a coherent basket. Typically the ground-truth size of $B_{\ell(S)+1}$ is unknown, which is approximated as a basket of a given constant size K [Yu *et al.*, 2016].

3.2 Correlation Matrix

As discussed above, our formulation requires a correlation matrix C, which can be constructed based on the co-occurring items in the observed training baskets. Specifically, let $F \in \mathbb{R}^{|V| \times |V|}$ capture the frequency of co-occurrences, such that F_{ij} is the number of times items i and j appear in a common basket, $\forall i \neq j$. As F contains raw counts that can differ significantly due to the varying popularity of items, we normalize F to obtain the final correlation matrix C based on the Laplacian matrix [Kipf and Welling, 2017]:

$$C = D^{-\frac{1}{2}} F D^{-\frac{1}{2}}, \tag{1}$$

where D is the degree matrix such that $D_{ii} = \sum_j C_{ij}$. Note that by definition, F and C are both symmetric.

Furthermore, in some cases, the correlation matrix could be too sparse to provide useful associations. We may consider higher-order correlations up to the N-th order, i.e., $C + \sum_{n=2}^{N} \mu^{n-1} \text{Norm}(C^n)$, where $\mu \in (0, 1)$ is a discount factor for higher orders, and $\text{Norm}(\cdot)$ sets the diagonal to zero and applies the same normalization in Eq. (1).

4 Basket-Sequence Correlation Networks

In this section, we propose Basket Sequence COrrelation Networks (*Beacon*) for correlation-sensitive next-basket recommendation, and discuss its learning strategy.

4.1 Proposed Framework: *Beacon*

Our framework *Beacon* is outlined in Figure 2, which consists of three main components, namely, correlation-sensitive basket encoder, basket sequence encoder, and correlation-sensitive predictor. Taking a basket sequence and correlation matrix as input, the basket encoder captures *intra-basket item*
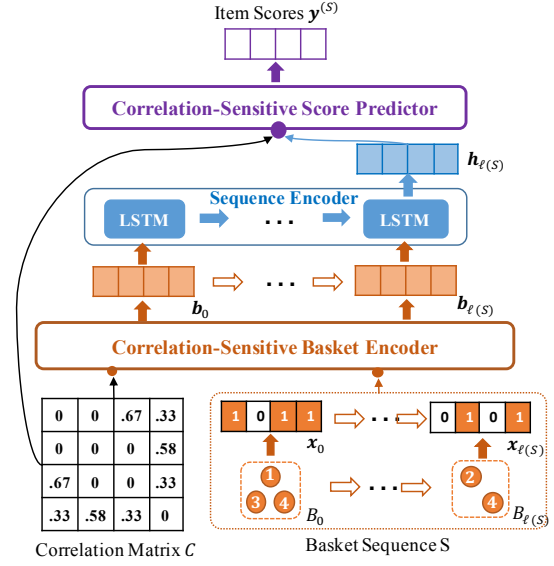

Figure 2: Architecture of the proposed framework *Beacon*.

correlations and produce correlation-sensitive basket representations. The sequence of basket representations is further fed into a sequence encoder to capture *inter-basket sequential associations*. The output from the sequence encoder, together with the correlation matrix, are employed by the predictor to produce the correlation-sensitive next basket. We further elaborate each component in the following.

Correlation-Sensitive Basket Encoder
Given a basket B_t at time t, we can convert it to a binary vector $\mathbf{x}_t \in \{0, 1\}^{|V|}$, whereby its i-th element is zero if and only if $i \in B_t$. There are two primary factors that trigger the presence of an item in the basket B_t, including not only the item's self-importance, but also its correlative associations with other items in B_t. Simultaneously accounting for the two factors may enhance the representation of B_t. Thus, we propose the following intermediate representation $\mathbf{z}_t \in \mathbb{R}^{|V|}$ for the basket B_t:

$$\mathbf{z}_t = \mathbf{x}_t \circ \boldsymbol{\omega} + \mathbf{x}_t C \tag{2}$$

where $\circ$ denotes the Hadamard (i.e., element-wise) product, $\boldsymbol{\omega} \in \mathbb{R}^{|V|}$ entails the learnable item importance parameters and C is the input correlation matrix. Generally, not all correlative associations are useful—weak correlations are more likely to be noises that adversely impact the basket representation. Therefore, we introduce $\eta \in \mathbb{R}^+$, a learnable scalar parameter to filter out weak correlations, into the intermediate representation:

$$\mathbf{z}_t = \mathbf{x}_t \circ \boldsymbol{\omega} + \text{ReLU}(\mathbf{x}_t C - \eta \mathbf{1}), \tag{3}$$

where $\mathbf{1}$ is a vector of ones and ReLU is applied in an element-wise manner. Subsequently, $\mathbf{z}_t$ is fed into a fully-connected layer to infer a latent L-dimension basket representation $\mathbf{b}_t \in \mathbb{R}^L$, as follows:

$$\mathbf{b}_t = \text{ReLU}(\mathbf{z}_t \Phi + \boldsymbol{\phi}), \tag{4}$$

where $\Phi \in \mathbb{R}^{|V| \times L}$ and $\boldsymbol{\phi} \in \mathbb{R}^L$ are weight and bias parameters to be learned, respectively.

Basket Sequence Encoder

The sequence encoder employs a RNN to capture the sequential associations in basket sequences. Given a basket sequence $S = \langle B_1, \ldots, B_{\ell(S)} \rangle$ with corresponding latent basket representations $\langle \mathbf{b}_1, \ldots, \mathbf{b}_{\ell(S)} \rangle$, the recurrent H-dimension hidden output $\mathbf{h}_t \in \mathbb{R}^H$ at time t is computed by:

$$\mathbf{h}_t = \tanh(\mathbf{b}_t \Psi + \mathbf{h}_{t-1} \Psi' + \boldsymbol{\psi}) \qquad (5)$$

where $\Psi \in \mathbb{R}^{L \times H}, \Psi' \in \mathbb{R}^{H \times H}$ and $\boldsymbol{\psi} \in \mathbb{R}^H$ are weight and bias parameters to be learnt. As shown in Figure 2, while *Beacon* adopts LSTM units [Le *et al.*, 2018], it is flexible to plug in other recurrent units, e.g., GRU [Hidasi *et al.*, 2016].

Correlation-Sensitive Score Predictor

The predictor aims to derive a score for each item based on both the inter-basket sequential associations and intra-basket correlation associations. Let $\mathbf{h}_{\ell(S)}$ be the last hidden output of sequence S via the sequence encoder. Thus, the sequential signal $\mathbf{s}^{(S)} \in \mathbb{R}^{|V|}$ for item recommendation given sequence S can be estimated by the following:

$$\mathbf{s}^{(S)} = \sigma(\mathbf{h}_{\ell(S)} \Gamma), \qquad (6)$$

where σ is the sigmoid function applied in an element-wise manner, and $\Gamma \in \mathbb{R}^{H \times |V|}$ is a weight matrix to be learned. In order to recommend a basket with correlated items, we further aggregate the sequential signal with item importance and correlative associations. Similar to Eq. (2), a straightforward solution is $\mathbf{s}^{(S)} \circ \boldsymbol{\omega} + \mathbf{s}^{(S)} C$. However, in this formulation, the intra-basket correlative association often dominates and masks the inter-basket sequential associations. Thus, we adopt the following predictor, such that the trade-off between correlative and sequential associations can be tuned:

$$\mathbf{y}^{(S)} = \alpha(\mathbf{s}^{(S)} \circ \boldsymbol{\omega} + \mathbf{s}^{(S)} C) + (1 - \alpha)\mathbf{s}^{(S)}, \qquad (7)$$

where $\alpha \in [0, 1]$ is a hyperparameter to control the balance between correlative and sequential associations, and $\mathbf{y}^{(S)} \in \mathbb{R}^{|V|}$ contains the predicted scores such that its i-th element, $\mathbf{y}_i^{(S)}$, indicates the score of item i.

Next-Basket Recommendation. Given a test basket sequence $S = \langle B_1, \ldots, B_{\ell(S)} \rangle$, we recommend the next basket $B_{\ell(S)+1}$ based on the predicted scores $\mathbf{y}^{(S)}$. The scores indicate how likely each item could form the next basket, accounting for both intra-basket correlative and inter-basket sequential associations. Since the size of the next basket is unknown and is often noncritical in a recommendation setting [Yu *et al.*, 2016], in practice we form the next basket by taking K items with the highest scores in $\mathbf{y}^{(S)}$, where K is a small constant such as 5 or 10.

4.2 Learning Strategy

For each training sequence S, we remove its last basket to obtain $S' = \langle B_1, \ldots, B_{\ell(S)-1} \rangle$. The goal is to make sure that the predicted scores $\mathbf{y}^{(S')}$ based on S' should align well with the ground truth next basket $B_{\ell(S)}$.

To this end, we favor the adopted items in the ground truth basket $B_{\ell(S)}$, and at the same time penalize other negative items in $V \setminus B_{\ell(S)}$. In particular, we formulate the following

Dataset	#Sequence	#Item	Average length	Average basket size
TaFeng	77 209	9 964	7.0	5.9
Delicious	61 908	6 520	21.4	3.8
Foursquare	100 980	5 527	22.2	1.8

Table 2: Statistics for TaFeng, Delicious and Foursquare datasets

loss for sequence S, where we try to maximize the scores of the adopted items (first term), and minimize the scores of negative items with respect to the minimum score among adopted items (second term). Intuitively, the second term encourages the negative items to be ranked lower than all of the adopted items in $\mathbf{y}^{(S')}$.

$$\mathcal{L}(S) = -\frac{1}{|B_{\ell(S)}|} \sum_{i \in B_{\ell(S)}} \log \sigma(\mathbf{y}_i^{(S')}) \qquad (8)$$
$$-\frac{1}{|V \setminus B_{\ell(S)}|} \sum_{j \in V \setminus B_{\ell(S)}} \log(1 - \sigma(\mathbf{y}_j^{(S')} - \mathbf{y}_m^{(S')})),$$

where $m = \arg\min_{i \in B_{\ell(S)}} \mathbf{y}_i^{(S')}$ is the adopted item with minimum predicted score.

Given the set of training basket sequences $\mathcal{S}_{\text{train}}$, we seek to minimize the total loss to learn our parameter set $\Theta = (\boldsymbol{\omega}, \eta, \Phi, \boldsymbol{\phi}, \Psi, \Psi', \boldsymbol{\psi}, \Gamma)$:

$$\Theta^* = \arg\min_{\Theta} \sum_{S \in \mathcal{S}_{\text{train}}} \mathcal{L}(S) \qquad (9)$$

Complexity Analysis. According to Eq. (3) and Eq. (4), the complexity of the basket encoder is $\mathcal{O}(|V|^2 + |V| \cdot L)$. In the sequence encoder, the complexity of an LSTM unit is $\mathcal{O}(H^2 + H \cdot L)$ [Hochreiter and Schmidhuber, 1997]. Moreover, the correlation-sensitive predictor has the complexity $\mathcal{O}(|V| \cdot H + |V|^2)$. Thus, given a set of training sequences $\mathcal{S}_{\text{train}}$ with an average sequence length of $\bar{S}$, and considering that H, L are generally much smaller than $|V|$, the overall complexity of *Beacon* on a training epoch can be simplified to $\mathcal{O}(|\mathcal{S}_{\text{train}}| \cdot \bar{S} \cdot |V|^2)$.

5 Experiments

We investigate the efficacy of *Beacon* for the next-basket recommendation task, particularly through comparing with a series of classic and state-of-the-art baselines, and conducting both quantitative and qualitative analyses on our model.

5.1 Setup

Datasets. We conduct experiments on three publicly available real-life datasets of three different domains as follows. *TaFeng*[1] is a grocery shopping dataset containing transactions from Nov 2000 to Feb 2001. Each transaction is a basket of purchased items. Each sequence is a user's chronological ordering of baskets. *Delicious*[2] consists of users' sequences of bookmarks. Each bookmark is associated with a basket of tag

---

[1] https://www.kaggle.com/chiranjivdas09/ta-feng-grocery-dataset

[2] https://grouplens.org/datasets/hetrec-2011

assignments. *Foursquare*[3] has users' chronological check-ins from Aug 2010 to Jul 2011 [Yuan *et al.*, 2013]. We define a basket as the set of check-ins within the same day.

Preprocessing. To cater sufficient information about each user and item for modeling, we ensure that each user adopts at least n items and each item is adopted by at least n users, with n being 10, 5, 5 for TaFeng, Delicious and Foursquare respectively. To get a sense of the extent of reduction, only 5.9% were removed out of a total of 817,741 adoptions in TaFeng. For Delicious, 11.8% out of 430,987 adoptions were removed. For Foursquare, 0.1% of 186,804 adoptions were removed. Additionally, we filter out basket sequences with fewer than 2 baskets. To create train/validation/test sets, sequences are chronologically split into three non-overlapping periods ($t_{trainl}, t_{val}, t_{test}$), i.e., (3, 0.5, 0.5) months for TaFeng, (80, 2, 2) months for Delicious and (10, 0.5, 0.5) months for Foursquare. For the train and validation sets, we generate all subsequences of the basket sequences with more than 3 baskets. Anything longer than 30 baskets is truncated with the prefix cut off. To facilitate new-item recommendations, as in [Rendle *et al.*, 2010], we do not consider the item just adopted in the immediately preceding time step. The statistics after preprocessing are described in Table 2.

Correlation Matrix. We construct the input correlation matrix according to Section 3.2. Based on the validation set, we choose the first-order correlation for Delicious and Foursquare whilst adopting the higher-order correlation for TaFeng with $N = 5$ and $\mu = 0.85$.

Evaluation Metrics. Given a test sequence S, we use the preceding baskets $S' = \langle B_1, \ldots, B_{\ell(S)-1} \rangle$ to predict the last basket at time $\ell(S)$. This prediction is then compared to the ground-truth basket $B_{\ell(S)}$ on two well established metrics. One is F-measure (*F1@K*) [Yu *et al.*, 2016], where K is the basket size to be predicted. The second metric, Half-life utility (*HLU*) a.k.a. "Breese score" [Breese *et al.*, 1998]. For both metrics, performances are averaged across test baskets using 10 runs with different random initialization. Comparisons are supported by two-tailed paired-sample Student's t-test at 0.05 significance level.

Learning Details. With the objective of minimizing the log loss in Eq. (9), our model is trained in 15 epochs of batch-size 32. We use the RMSProp optimizer with the learning rate 0.001. The LSTM layer is applied with a 0.3 dropout probability. η is initialized by the mean of non-zero values in C. The model is further tuned on the validation set over the latent dimension $L \in \{8, 16, 32, 64\}$ and recurrent hidden unit $H \in \{16, 32, 64\}$ using a grid search. Lastly, we use $\alpha = 0.5$ as the default to control the trade-off between sequential or correlative associations. We will also vary α and study its impact in Section 5.3. For our experiments on NVIDIA P100 GPU with 16GB memory, each mini-batch takes approximately 0.1 second.

5.2 Comparison to Baselines

We compare *Beacon* to a suite of classic and state-of-the-art baselines, as follows.

---

[3]http://www.ntu.edu.sg/home/gaocong/datacode.htm

Dataset	Model	L	H	F1@K (%) @5	F1@K (%) @10	HLU
TaFeng	*POP*	-	-	4.66	4.02	6.64
	MC	-	-	4.11	3.61	5.78
	MCN	8	-	4.56	4.02	6.34
	DREAM	8	-	5.85	4.90	6.96
	BSEQ	32	16	4.48	4.04	6.34
	triple2vec	64	-	4.66	3.88	4.85
	Beacon	8	64	**6.36**†	**5.26**†	**7.83**†
Delicious	*POP*	-	-	3.88	4.04	6.05
	MC	-	-	4.27	4.59	6.52
	MCN	32	-	4.20	4.59	6.50
	DREAM	32	-	3.13	3.47	4.93
	BSEQ	64	32	3.86	3.97	5.95
	triple2vec	32	-	3.76	4.04	5.16
	Beacon	64	64	**4.93**†	**5.47**†	**7.76**†
Foursquare	*POP*	-	-	2.73	2.90	4.84
	MC	-	-	3.58	3.43	5.53
	MCN	64	-	3.09	2.89	5.08
	DREAM	64	-	2.84	3.00	4.98
	BSEQ	64	32	2.80	2.89	4.82
	triple2vec	64	-	2.73	2.90	4.53
	Beacon	64	64	**3.61**	**3.59**†	**6.32**†

Table 3: Performance comparison between *Beacon* versus baselines on TaFeng, Delicious and Foursquare. † represents statistically significant improvements of *Beacon* over the second best model.

- *POP* ranks items based on their global popularity.
- *MC* ranks items based on first-order Markov-chain transition probabilities from items in the previous basket.
- *MCN* is similar to *MC*, but uses denser Markov-chain dependencies derived using neural networks.
- *DREAM* [Yu *et al.*, 2016] is a dynamic recurrent model, where a basket representation is aggregated by items' embedding via a pooling layer. The most recent basket representation is used to generate the next basket[4].
- *BSEQ* [Le *et al.*, 2018] captures long-term dependencies. Each basket is encoded directly from a binary vector using a fully-connected layer. Next-basket predictions are based on the sequential signal at the last basket.
- *triple2vec* [Wan *et al.*, 2018] infers the embeddings of items and users from (user u, item i, item j) triplets, where i, j co-occur in the same basket. We use the author's implementation[5] with various initial loyalty values to derive sequence representations for a global user to focus the comparison on sequential effects.

All baselines, if applicable, are trained as well as tuned on the validation set in the same manner as *Beacon* outlined in Section 5.1.

Table 3 shows the results in terms of $F1@5$, $F1@10$ and HLU. For TaFeng, popularity seems to be an important factor since *POP* performs better than *MC*, *MCN*, *BSEQ* and *triple2vec*. Beyond popularity, *DREAM* and *Beacon* show

---

[4]https://github.com/LaceyChen17/DREAM
[5]https://github.com/MengtingWan/grocery

Dataset	Model	F1@K (%)		HLU
		@5	@10	
TaFeng	$Beacon_{\text{corr-impt-}}$	3.87	3.44	5.13
	$Beacon_{\text{corr-}}$	$5.78^\dagger$	$4.86^\dagger$	$7.18^\dagger$
	$Beacon$ (full)	$\mathbf{6.36}^\S$	$\mathbf{5.26}^\S$	$\mathbf{7.83}^\S$
Delicious	$Beacon_{\text{corr-impt-}}$	4.02	4.43	6.38
	$Beacon_{\text{corr-}}$	$4.67^\dagger$	$5.10^\dagger$	$7.15^\dagger$
	$Beacon$ (full)	$\mathbf{4.94}^\S$	$\mathbf{5.47}^\S$	$\mathbf{7.76}^\S$
Foursquare	$Beacon_{\text{corr-impt-}}$	2.98	3.29	5.39
	$Beacon_{\text{corr-}}$	$3.58^\dagger$	$3.52^\dagger$	$6.16^\dagger$
	$Beacon$ (full)	$\mathbf{3.61}$	$\mathbf{3.59}^\S$	$\mathbf{6.32}^\S$

Table 4: Performance comparison between *Beacon* and its variants without item importance (impt) or correlation (corr). $\dagger, \S$ represent statistically significant improvements from the previous row.

advantages in capturing associations between basket items. Yet, *Beacon* is the best-performing model. For Delicious, Markov-based models (*MC* and *MCN*) do better than other baselines. It might imply that items in a testing basket are strongly dependent on the most recent basket. The modeling of basket-oriented associations in *DREAM* and *triple2vec* is not helpful to improve the performance. In contrast, *Beacon* shows a significant improvement over these models across the three measurements, which we attribute to the advantage of modeling correlations effectively. For Foursquare, we witness a similar observation as Delicious, where *Beacon* outperforms the baselines significantly.

5.3 Quantitative Model Analysis

We further analyze our model quantitatively in the context of two research questions listed below.

Are item importance and correlation helpful?
Our basket encoder accounts for two primary factors: item importance $\boldsymbol{\omega}$ and correlation C, as shown in Eq. (3). To study the contribution of each factor, we compare two simpler variants with *Beacon*: (i) $Beacon_{\text{corr-}}$, which ignores item correlation by setting C to a zero matrix; and (ii) $Beacon_{\text{corr-impt-}}$, which ignores both item importance and correlation by further setting $\boldsymbol{\omega}$ to a vector of one's.

We report their results in Table 4. Specifically, the full model significantly outperforms $Beacon_{\text{corr-}}$, demonstrating that item correlation plays a crucial role in next-basket recommendation. Likewise, $Beacon_{\text{corr-}}$ significantly beats $Beacon_{\text{corr-impt-}}$ to imply that item importance is another useful factor. In summary, our model benefits from both factors.

What is the effect of hyper-parameter α?
According to Eq. (7), α tunes the relative weights of correlative and sequential associations. Higher α emphasizes endogenous effects within baskets, while lower α favors exogenous effects across baskets. In Figure 3, we plot the performance when varying α. There are some minor variations across datasets, but generally the range of $\alpha \in [0.2, 0.6]$ tends to do relatively well in most scenarios, indicating that some balance is useful.

---

[6] http://www.desarrolloweb.com/manuales/manual-jquery.html
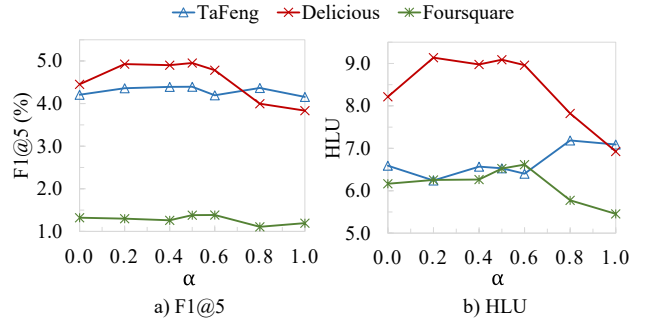
[7] https://articles.uie.com/three_hund_million_button

Figure 3: Impact of α on the performance of *Beacon*.

Target bookmark	Tag basket prediction ($K = 5$)		
	Beacon	*MC*	*POP*
Manual de jQuery [6]	*web*, design, *programming*, *javascript*, *tools*	digital, sociales, *web*, internet, periodismo	art, design, education, video, *tools*
The $300 Million Button [7]	twitter, *ux*, *propinquity*, *critical*, *writing*	*design*, peace, education, blog, *tips*	art, design, education, video, *tools*

Table 5: Illustrations of tag basket prediction by *Beacon*, *MC* and *POP* on Delicious. Italics denote tags relevant to the bookmark.

5.4 Qualitative Analysis

Finally, we perform a qualitative analysis on Delicious, where the objective is to recommend a basket of tags for the next bookmark to visit. The other two datasets only contain item IDs and thus cannot be used for the qualitative study. In Table 5, *Beacon* is compared to the second best model *MC* and the popularity-based method *POP*, illustrating two examples of tag-basket prediction with respect to two bookmarks. *POP* keeps suggesting the same set of tags as it only leverages the global popularity, while *MC* recommends somehow general tags with limited relevance. In contrast, *Beacon* proposes more relevant baskets of correlated tags. The set of tags {*web, programming, javascript, tools*} are descriptive for *jQuery*, a Javascript library. Likewise, the second bookmark refers to a *critical discussion* on how to increase a site's revenue by maximizing user experience (i.e., *ux*) with an efficient design (e.g., *propinquity* between buttons and fields).

6 Conclusion

In this paper, we address the next-basket recommendation problem. Assuming baskets incorporate correlative dependencies among items, we propose *Beacon* that utilizes the correlation information to enhance the representation of individual baskets as well as the overall basket sequence. Experimental results on three public real-life datasets show the benefit of exploiting correlative dependencies.

References

[Bai *et al.*, 2018] Ting Bai, Jian-Yun Nie, Wayne Xin Zhao, Yutao Zhu, Pan Du, and Ji-Rong Wen. An attribute-aware neural attentive model for next basket recommendation. In *SIGIR*, pages 1201–1204, 2018.

[Breese *et al.*, 1998] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI*, pages 43–52, 1998.

[Chen *et al.*, 2012] Shuo Chen, Josh L Moore, Douglas Turnbull, and Thorsten Joachims. Playlist prediction via metric embedding. In *KDD*, pages 714–722, 2012.

[Christakopoulou and Karypis, 2014] Evangelia Christakopoulou and George Karypis. Hoslim: Higher-order sparse linear method for top-n recommender systems. In *PAKDD*, pages 38–49, 2014.

[He *et al.*, 2017] Ruining He, Wang-Cheng Kang, and Julian McAuley. Translation-based recommendation. In *Recsys*, pages 161–169, 2017.

[Hidasi *et al.*, 2016] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *ICLR*, 2016.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Huang *et al.*, 2018] Jin Huang, Wayne Xin Zhao, Hong-Jian Dou, Ji-Rong Wen, and Edward Y. Chang. Improving sequential recommendation with knowledge-enhanced memory networks. In *SIGIR*, pages 505–514, 2018.

[Kipf and Welling, 2017] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Le *et al.*, 2017] Duc Trong Le, Hady W Lauw, and Yuan Fang. Basket-sensitive personalized item recommendation. In *IJCAI*, pages 2060–2066, 2017.

[Le *et al.*, 2018] Duc Trong Le, Hady Wirawan Lauw, and Yuan Fang. Modeling contemporaneous basket sequences with twin networks for next-item recommendation. In *IJCAI*, pages 3414–3420, 2018.

[Li *et al.*, 2009] Ming Li, Benjamin M Dias, Ian Jarman, Wael El-Deredy, and Paulo JG Lisboa. Grocery shopping recommendations based on basket-sensitive random walk. In *KDD*, pages 1215–1224, 2009.

[Li *et al.*, 2017] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In *CIKM*, pages 1419–1428, 2017.

[Ning and Karypis, 2011] Xia Ning and George Karypis. Slim: Sparse linear methods for top-n recommender systems. In *ICDM*, pages 497–506, 2011.

[Pathak *et al.*, 2017] Apurva Pathak, Kshitiz Gupta, and Julian McAuley. Generating and personalizing bundle recommendations on steam. In *SIGIR*, pages 1073–1076, 2017.

[Quadrana *et al.*, 2018] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. Sequence-aware recommender systems. *ACM Computing Surveys (CSUR)*, 51(4):66, 2018.

[Rendle *et al.*, 2010] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, pages 811–820, 2010.

[Sarwar *et al.*, 2000] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Analysis of recommendation algorithms for e-commerce. In *EC*, pages 158–167, 2000.

[Song *et al.*, 2019] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. Session-based social recommendation via dynamic graph a ention networks. In *WSDM*, 2019.

[Tang and Wang, 2018] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*, pages 565–573, 2018.

[Villatel *et al.*, 2018] Kiewan Villatel, Elena Smirnova, Jérémie Mary, and Philippe Preux. Recurrent neural networks for long and short-term sequential recommendation. In *Recsys*, 2018.

[Wan *et al.*, 2018] Mengting Wan, Di Wang, Jie Liu, Paul Bennett, and Julian McAuley. Representing and recommending shopping baskets with complementarity, compatibility and loyalty. In *CIKM*, pages 1133–1142, 2018.

[Wang *et al.*, 2015] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. Learning hierarchical representation model for nextbasket recommendation. In *SIGIR*, pages 403–412, 2015.

[Wang *et al.*, 2018] Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. Attention-based transactional context embedding for next-item recommendation. In *AAAI*, 2018.

[Wu *et al.*, 2019] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *AAAI*, 2019.

[Xiang *et al.*, 2010] Liang Xiang, Quan Yuan, Shiwan Zhao, Li Chen, Xiatian Zhang, Qing Yang, and Jimeng Sun. Temporal recommendation on graphs via long-and short-term preference fusion. In *KDD*, pages 723–732, 2010.

[Ying *et al.*, 2018] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. Sequential recommender system based on hierarchical attention networks. In *IJCAI*, pages 3926–3932, 2018.

[Yu *et al.*, 2016] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. A dynamic recurrent model for next basket recommendation. In *SIGIR*, pages 729–732, 2016.

[Yuan *et al.*, 2013] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. Time-aware point-of-interest recommendation. In *SIGIR*, pages 363–372, 2013.