Interactive Sequential Basket Recommendation by Learning Basket Couplings and Positive/Negative Feedback

WEI WANG and LONGBING CAO, Data Science Lab, University of Technology Sydney, Australia

Sequential recommendation, such as *next-basket recommender systems* (NBRS), which model users' sequential behaviors and the relevant context/session, has recently attracted much attention from the research community. Existing *session-based NBRS* involve session representation and inter-basket relations but ignore their hybrid couplings with the intra-basket items, often producing irrelevant or similar items in the next basket. In addition, they do not predict next-baskets (more than one next basket recommended). *Interactive recommendation* further involves user feedback on the recommended basket. The existing work on next-item recommendation involves positive feedback on selected items but ignores negative feedback on unselected ones. Here, we introduce a new setting—*interactive sequential basket recommendation*, which iteratively predicts next baskets by learning the intra-/inter-basket couplings between items and both positive and negative user feedback on recommended baskets. A *hierarchical attentive encoder-decoder model* (HAEM) continuously recommends next baskets one after another during sequential interactions with users after analyzing the item relations both within a basket and between adjacent sequential baskets (i.e., intra-/inter-basket couplings) and incorporating the user selection and unselection (i.e., positive/negative) feedback on the recommended baskets to refine NBRS. HAEM comprises a basket encoder and a sequence decoder to model intra-/inter-basket couplings and a prediction decoder to sequentially predict next-baskets by interactive feedback-based refinement. Empirical analysis shows that HAEM significantly outperforms the state-of-the-art baselines for NBRS and session-based recommenders for accurate and novel recommendation. We also show the effect of continuously refining sequential basket recommendation by including unselection feedback during interactive recommendation.

CCS Concepts: • **Information systems → Recommender systems**; • **Computing methodologies →** *Learning from implicit feedback*; *Neural networks*;

Additional Key Words and Phrases: Sequential recommendation, interactive sequential basket recommendation, next-basket recommender system, positive feedback, negative feedback, basket coupling, recurrent neural network, factorization machine, attention model

ACM Reference format:
Wei Wang and Longbing Cao. 2021. Interactive Sequential Basket Recommendation by Learning Basket Couplings and Positive/Negative Feedback. *ACM Trans. Inf. Syst.* 39, 3, Article 24 (February 2021), 26 pages.
https://doi.org/10.1145/3444368

This work was partially supported by the Australian Research Council Discovery Grant (DP190101079) and ARC Future Fellowship Grant (FT190100734).

Authors' address: W. Wang and L. Cao, Data Science Lab, University of Technology Sydney, Australia; emails: Wei.Wang-8@alumni.uts.edu.au, longbing.cao@gmail.com.

1046-8188/2021/02-ART24 $15.00
https://doi.org/10.1145/3444368

1 INTRODUCTION

Recent recommendation research has focused increasing attention on modeling user behaviors and interactions with recommended items [3, 38], e.g., modeling sequences of purchased baskets in retail and online businesses and sequential clicks in mobile game and news apps for recommendation. Such recommendation applications involve sequential interactions between users and items (products or services) and generate sequences of user preferences, behaviors, feedback (selecting or not-selecting recommended items), and recommendations. These multi-aspect sequences are all coupled with and influence each other along the interactive sequential recommendation process.

The problem: Interactive sequential basket recommender systems. As illustrated in Figure 1, a typical interactive sequential basket recommender (iSBRS) works as follows. First, at a time point (or period), a basket of items is consumed by a customer. Second, the recommender analyzes the consumed items, the user preferences, and other relevant information and suggests items the customer may like to consume in the next basket. Then, users may choose to select or ignore the recommended items when purchasing their next baskets, which corresponds to *positive/negative feedback*. Here, *negative feedback* refers to user feedback on recommended but unselected items. These unselected items are called *negative items* [7], and their sequences are *negative sequences*, in contrast to *positive items* and *positive sequences.*[1] Both positive and negative feedback reflect user preferences and selection bias [52].

While negative items/sequences are typically ignored in existing recommendation research, other related research [7] shows the importance of involving such nonoccurring but important items in behavior and sequence analysis [9, 46]. Here, we treat the unselections (recommended yet non-occurring in the user's next basket) as negative feedback and combine both positive and negative feedback with user preferences and behaviors during the user-item interactions in making the next-basket recommendation. We further assume an interactive sequential recommendation process as an iterative trial-and-error process, i.e., the iSBRS recommends a next basket to a user at a time point, who then selects or ignores some of the items in the recommended basket. The iSBRS then collects and jointly analyzes the selection-or-not as user feedback (i.e., positive/negative feedback) together with previous sequential baskets, user behaviors and preferences, and then suggests another next basket. iSBRS repeats this interactive recommendation process to continuously recommend next baskets. Such interactive sequential basket recommender systems, or broadly sequential basket recommender systems (SBRS), sequentially recommends baskets of items one after another during the sequential and long-range user-item interactions. They are pragmatic for interactive applications, e.g., sequential health/medical treatments, customer relationship management, and recommending news and entertaining apps and services. Such applications likely involve continuous interactions between consumers and the consumed services and long-range consumption relationships. During the follow-up interactions, feedback is available and consumer circumstances or preferences evolve over time, hence novel and personalized services need to be recommended.

Modeling the above interactive sequential basket recommendation (iSBR) process involves several inherent characteristics and complexities [5, 6] in iSBRS: (1) iSBRS involves multiple sequences including the sequence of baskets, the sequence of user preferences, and the sequence of user interactive feedback on recommended basket items; (2) the items in a basket are coupled with each other owing to various relations and factors (e.g., user preference or item correlation in one transaction), and they may also be coupled with the items in adjacent baskets (e.g., today's purchase of an iphone may lead to a subsequent purchase of iphone accessories). These form hierarchical

[1]These terms of positive/negative items/sequences are consistent with the terminology used in sequence analysis [7, 46]

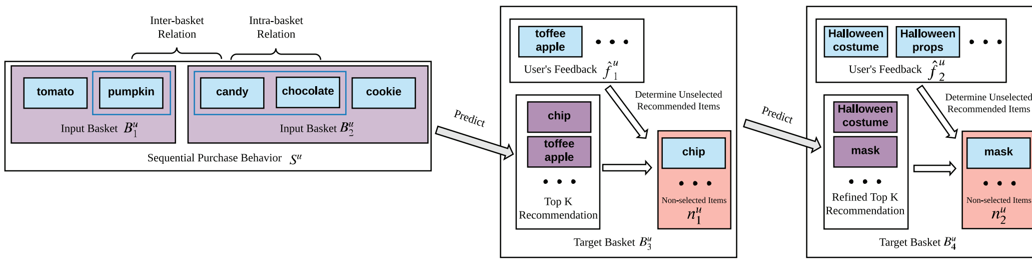


Fig. 1. An example of interactive sequential basket recommendation. Given a user's purchase sequence $<(tomato, pumpkin), (candy, chocolate, cookie)>$, which consists of two consecutive baskets $B_1^u :< (tomato, pumpkin) >$ and $B_2^u :< (candy, chocolate, cookie) >$, a recommender suggests $\{chip, toffee\ apple\}$ as the next basket B_3^u to the user. Once the user receives the recommendation, if he/she selects items $toffee\ apple$ (i.e., selected/positive items) but unselects $chip$ (i.e., unselected/negative item), this selection/unselection-based positive/negative feedback indicates that the user may intend to celebrate Halloween instead of an ordinary party. Accordingly, the system further recommends $Halloween\ costume$ and $mask$ as the next basket B_4^u based on basket B_3^u and user feedback.

item couplings [6] both within baskets and between baskets. Such intra-/inter-basket couplings may determine or influence an item's occurrence in the next basket; (3) both positive and negative feedback and the feedback sequence formed during interactive sequential recommendation should be jointly modeled with user-item/basket interactions, selection/unselection behaviors, and their dynamics along with interactive recommendation to understand user like/dislike preferences and future interest; (4) the above multiple sequences are coupled with each other [8] and together form the iSBRS problem and should be jointly modeled in designing interactive sequential recommenders.

Gap analysis. The aforementioned iSBRS characteristics challenge existing recommendation research including collaborative filtering–(CF) based recommendation [29], context-based recommendation [28], sequential recommendation [17] such as session-based recommendation for next-item recommendation [24, 44, 45] and next-basket recommender systems (NBRS) [1, 12, 16, 26, 36, 40, 41, 50], interactive recommendation [19, 42, 53], and feedback-based recommendation [52]. CF-based recommendation does not fit the iSBRS settings and thus cannot handle the iSBRS problem, in addition to facing many other issues [6, 22, 23]. Context-based recommendation models the context of a recommendation target but does not necessarily satisfy sequential recommendation. Deep neural networks such as hierarchical representation model (HRM) [41] and dynamic recurrent basket model (DREAM) [50], which only handle one specific assumption [24, 45] are widely used in context-based recommendation. Session-based recommendation focuses on recommending a next-item/basket based only on first-order transition (e.g., by Markov chain) from the last basket to the current one but ignores previous baskets [13, 21, 24, 27, 44, 45]. They model the session (or context) relevant to the next item/basket but do not involve intra- or inter-basket relations. Interactive recommendation, which typically only recommends one item per interaction, has rarely been studied. Feedback-based recommendation typically only focuses on positive feedback, namely those items recommended and selected by users. Negative item/feedback is rarely considered, probably because the positive items are more visible while negative ones are implicit and often overlooked (which are also hard to model) [9]. In summary, while inter-basket sessions and positive feedback have been considered in existing work, both intra-basket item relations and negative feedback are often overlooked. No work has comprehensively involved both intra- and inter-basket couplings and positive and negative feedback for sequential basket recommendation in an interactive manner, which motivates this article on iSBRS.

In particular, session-based next-item/basket recommendation [31, 43] can be treated as a special case of iSBRS and SBRS; however, it requires significant effort to extend next-item recommenders for iSBRS. A next-item recommender predicts the next item based on the past session/context consisting of previous items. A NBRS suggests a basket of items at a time point for a user based on the past basket as the session. However, iSBRS sequentially predicts the next basket one after another by successively modeling the sequential interactions between a recommender and a user. We illustrate the difference between next-item/basket recommendation and iSBRS using the example in Figure 1. Given two consecutive baskets $B_1^u :< (tomato, pumpkin) >$ and $B_2^u :< (candy, chocolate, cookie) >$, an NBRS models the relations between all items in these two baskets and the most recent one and suggests the next item, e.g., snacks such as *chips* probably for a party, since all items are food-relevant and the latter basket consists only of snacks. However, if we consider intra-/inter-basket item relations and identify the cross-basket item combination in the sequence, e.g., $< pumpkin, (candy, chocolate) >$, which may indicate that the user may be preparing for Halloween, then an iSBRS would more likely recommend the next item set, e.g., $toffee\ apple$; items $\{chip, toffee\ apple\}$ are then recommended as the next basket B_3^u. Further NBRS does not involve interactive feedback as iSBRS does. With iSBRS, once basket B_3^u is recommended, if the user selects the item $toffee\ apple$ but does not select *chips*, then this could further confirm our guess that the user may intend to celebrate Halloween instead of organizing some other party. Accordingly, *Halloween costume* and *mask* may be further recommended for the next basket B_4^u based on the above positive/negative feedback on basket B_3^u. In this example, items *candy* and *chocolate* are coupled in basket B_2^u for their intra-basket item relation, and the two subsequences *pumpkin* and $(candy, chocolate)$ are coupled for the inter-basket relation between baskets B_1^u and B_2^u.

The above analysis shows that (1) iSBRS may be more realistic than NBRS as continuous user-item interactions can help understand user preferences and feedback on recommended items/baskets and their preference and feedback evolution during continuous interactive sessions. iSBRS downgrades to NBRS when it only involves one session to predict the next basket; (2) existing single session-based NBRS and next-item recommenders cannot directly support iSBRS objectives as they involve the current session and predict the next one item/basket for the next timestamp; (3) it is valuable yet challenging to model hierarchical basket couplings and positive/negative feedback during interactive recommendation. While one could repeat NBRS for iterative next-basket recommendation, each next-basket recommendation is independent of the next basket. Hence, intra-/inter-basket relations are overlooked and the repeated NBRS does not involve the relations between next baskets. Such a practice also does not involve user-item interactions and positive/negative feedback. As shown in the relevant research [5, 6, 38, 52], continuously incorporating hierarchical item relations, user-item interactions, and user feedback in the recommendation process not only characterizes the intricate SBRS problem but can also improve recommendation performance.

Contributions. Inspired by the above perspective on iSBRS, this article proposes a hierarchical attentive encoder-decoder model (HAEM) for interactive sequential basket recommendation by modeling intra-/inter-basket couplings and positive/negative feedback during the interactive recommendation process. HAEM consists of three components: a basket encoder, a sequence encoder, and a prediction decoder. The basket encoder builds a compound basket representation over all items in a basket by exploring multiple intra-basket relations between items. The sequence encoder maps the sequence of the built basket representations into a sequence of annotation representations by exploring the inter-basket relations between baskets. The prediction decoder makes recommendations for target baskets by modeling both basket context representations and positive and negative feedback and take into consideration the unselected items for future basket recommendations. Specifically, we incorporate the factorization machine (FM) mechanism [34, 37] into the basket encoder to model both linear and pairwise intra-basket item couplings within a basket. We introduce the attention mechanism to the basket and sequence encoders to recognize and pay more attention to those highly relevant items and significant baskets.

To the best of our knowledge, this is the first attempt to model interactive sequential basket recommendation by jointly learning continuous sessions, the intra-/inter-basket couplings in sequential purchase behaviors, and positive/negative feedback during continuous interactions. HAEM integrates negative feedback into refining the next basket recommendation during the sequential interactions. Our empirical analysis shows the effect of the HAEM design for interactive recommendation, and HAEM significantly outperforms the state-of-the-art NBRS and session-based methods on two real-life datasets for accurate and novel recommendation.

2 RELATED WORK

In the rapidly growing research area on recommender systems [1, 6, 32], the SBRS techniques related to this article can be broadly categorized into CF-based recommendation, sequential recommendation such as NBRS, session-based recommendation, interactive recommendation, and hybrid methods. Another relevant topic is feedback-based recommendation. In addition to the above gap analysis, we further briefly review these topics in terms of recommending iSBRS.

CF methods can predict next baskets by capturing common (collaborative) user preference in the purchase history of all users but ignore user sequential behaviors. Recommendation is typically made to a user/item by finding the top-k similar users/items per either particular measures or factoring the user-item matrix [41]. For example, a Bayesian personalized ranking (BPR) [35] optimizes the user preference ranking over item pairs instead of using the user preference score on a single item. However, CF methods often discard more sophisticated item couplings (e.g., long-range and attributed item relations), treat items independent of each other [6, 45] and ignore the couplings between user sequential behaviors [8], and also cannot model sequential behaviors. As a result, CF methods tend to produce similar or duplicated recommendations with low diversity and novelty owing to its similarity-based rationale [24, 44]. Their settings are not for SBRS and iSBRS.

Sequential recommendation has become a recent highlight in recommendation, which typically involves the Markov chain mechanism to explore sequential user behaviors and predict the next purchase based on the last basket [10, 17]. For example, a personalized sequential pattern mining-based recommender [49] learns user sequence importance based on the competence score for personalized next-item recommendation. Reference [11] proposed logistic Markov embedding (LME) to treat playlists as a Markov chain and to learn to represent the songs in the latent space for music recommendation. With LME, Reference [47] proposed personalized Markov embedding for next-song recommendation by modeling sequential user singing behaviors in the Euclidean space, and [15] proposed personalized ranking metric embedding for next-POI recommendation by jointly modeling user check-in sequences and individual preferences. A hierarchical recurrent neural network (RNN) [33] models user identifiers without past sessions, which is incapable for iSRBS. However, these methods only model the transition between adjacent behaviors but fail to model intra-basket item couplings and coupled sequential behaviors [6, 39, 41] in sequential recommendation.

Session-based recommendation has attracted a lot of recent attention by incorporating the current session representation into next-item/basket prediction [31, 43]. Typically, neural models and attention mechanisms are applied to represent the context or session of an item or item sequence for next-item/basket recommendation. Examples include GRU4Rec [21], SWIWO [24], NTEM [44], NARM [27], HCA-GRU [13], and ATEM [45]. However, these methods model the influence of the most recent session on the next target in terms of only the items within the session. The adjacent sessions, the intra-/inter-session couplings, and the coupled sequential behaviors between sessions are ignored. This is similar to the way intra-/inter-basket couplings are not considered by iSBRS.

In *interactive recommendation*, the recommender interacts with users over time and iteratively updates the recommendation model for further next-item recommendation by collecting positive user feedback on recommended items. Examples are ICTRTS [42], MAB [19], and ICF [53]. These methods do not address the various characteristics of iSBRS, nor can they model the intra-/inter-basket couplings or negative feedback.

More recent *hybrid methods* involve both general user preference and sequential behaviors for better recommendation [32]. For example, Reference [36] proposed factorized personalized Markov chains (FPMC) to combine Markov chain and matrix factorization for NBRS, which models sequential user behaviors by capturing the item relevance between the last and next baskets and general user preferences. However, FPMC assumes a linear relation between items within a basket, i.e., all items are linearly combined and independently affect a user's next purchase. Such strong assumptions are inconsistent with sophisticated user/item couplings and user-item interactions in real-life recommendation [6], since multiple factors may affect a user's next purchase [50]. In recent years, several network-based models were proposed for NBRS. A HRM [41] involves the representations of both last basket and users. However, both FPMC and HRM only consider the local sequential behaviors between the last and target baskets but disregard the impact of previous baskets on prediction. Reference [50] proposed a DREAM to learn the dynamic representation of users and applied RNN to capture the global sequential relations among baskets. Nevertheless, HRM and DREAM have the same deficiency in that they summarize item relevance using simple pooling operations, which neither pays attention to dominating items nor captures compound intra-basket item relations. Furthermore, the recurrent model on which DREAM is based suffers from the temporal dependency assumption and thus is not sufficient to capture the inter-basket item relations within a local range. It pays more attention to recent behaviors but fails to distinguish significant behaviors [13, 29]. In other words, both HRM and DREAM fail to model the inter-basket item relations between baskets.

In addition, *FM-based models* incorporate the FM mechanism of modeling sophisticated relevance in recommendation. Reference [18] proposed DeepFM to combine FMs with deep networks to model low-order relations using FMs and high-order relations using deep neural networks. In addition, Reference [20] proposed a neural factorization machine to enhance FMs by modeling higher-order and non-linear relations for sparse prediction. Reference [48] proposed an attentional factorization machine to learn attentive relations for better prediction. These methods do not jointly model intra-/inter-basket item relations or embed negative feedback for recommendation refinement.

Last, *user feedback* has been shown to be useful for refining recommendations with existing work typically only involving implicit feedback. For example, Reference [52] proposed a deep reinforcement learning based model, DEERS, to automatically learn optimal recommendation strategies by incorporating both positive and negative feedback and continuously to improve its strategies during interactions with users. DEERS validates the importance of negative feedback for more accurate recommendation. However, it only recommends a single item to a user each time by assuming only one item in a basket. Though it may be extendable by adjusting the Markov decision process for multiple items, the model does not capture the intra-basket relations between items within each basket.

In contrast, our iSBRS recommender HAEM caters for interactions between users and items/baskets along continuous basket generation, learns multi-aspect sequences and hierarchical item and behavior couplings both within and between baskets for iSBRS recommendation, and refines next baskets based on positive/negative feedback. Technically, HAEM achieves this by

Table 1. List of Main Notations

Notation	Description
U	A set of users, i.e., $U = \{u_1, u_2, \ldots, u_{\lvert U \rvert}\}$
I	A set of items, i.e., $I = \{i_1, i_2, \ldots, i_{\lvert I \rvert}\}$
B_t^u	The basket of items consumed by user u at time t
S^u	The basket sequence of user u, i.e., $S^u :=< B_1^u, B_2^u, \ldots, B_{t_u}^u >$ where $B_{t_u}^u$ is the basket at time t_u for user u
$r_{u,t}$	The personalized ranking of basket B_t^u for user u at time t
$\boldsymbol{o}_t^u$	The preference score of user u at time t for the prediction of basket B_t^u
$y_a^{u,t}$	The intra-basket relation between items in basket B_t^u
$\boldsymbol{h}_j^t \in \mathbb{R}^H$	The representation vector of item $i_j^t \in B_t^u$
w_j	The weight of item i_j^t w.r.t. the intra-basket relation $y_a^{u,t}$
$\boldsymbol{v}_j^t \in \mathbb{R}^F$	The latent feature vector of item i_j^t
$\langle \boldsymbol{v}_j^t, \boldsymbol{v}_k^t \rangle$	The inner-product of latent item vectors $\boldsymbol{v}_j^t$ and $\boldsymbol{v}_k^t$, i.e., the factorized interactions between items i_j^t and i_k^t
$\boldsymbol{h}_j^t \odot \boldsymbol{h}_k^t$	The element-wise product of two embedding vectors $\boldsymbol{h}_j^t$ and $\boldsymbol{h}_k^t$
$\hat{w}_{jk}^t$	The weight of pairwise relation between items i_j^t and i_k^t w.r.t. $y_a^{u,t}$
$\boldsymbol{l}_t^u \in \mathbb{R}^H$	The overall linear attentive intra-basket relation between items in B_t^u, i.e., a weighted integration of the embedding vectors $\boldsymbol{l}_t^u = \{\boldsymbol{h}_j^t \mid i_j^t \in B_t^u\}$
$\boldsymbol{p}_{jk}^t$	The pairwise intra-basket relation between a pair of items i_j^t and i_k^t
$\boldsymbol{q}_t^u \in \mathbb{R}^H$	The overall pairwise attentive intra-basket relation between items in B_t^u, i.e., the weighted integration of $\boldsymbol{p}_{jk}^t$ between each pair of items
$\boldsymbol{b}_t^u$	The basket representation of basket B_t^u
$\boldsymbol{s}_t^u$	The annotation representation of user u at time t
$\boldsymbol{p}_{t'}^u$	The RNN hidden state of user u for the target basket $B_{t'}^u$ at a next time point t'
$\hat{\boldsymbol{f}}_{t'}^u$	The user feedback vector of user u for target basket $B_{t'}^u$
$\boldsymbol{c}_{t'}^u$	The context vector for target basket $B_{t'}^u$
$w_j^{t'}$	The integration weight capturing the impact on basket B_j^u for the prediction of target basket $B_{t'}^u$
$\boldsymbol{n}_{t'}^u$	The negative feedback vector of user u at time t'

integrating behavior modeling, neural modeling, attention, coupling learning, and FM-based relevance learning.

3 PROBLEM STATEMENT

Here we formulate the iSBRS problem for interactive sequential basket recommendation and define the basic concepts. The main notations are listed in Table 1.

For the interactive sequential basket recommendation illustrated in Figure 1, assume a user is associated with a sequence of baskets where each basket consists of a set of lexicographically ordered items. Let $U = \{u_1, u_2, \ldots\}$ be the set of users, and $I = \{i_1, i_2, \ldots\}$ be the set of items, where $|U|$ and $|I|$ denote the number of users and items respectively. The sequential baskets of user u are denoted as $S^u :=< B_1^u, B_2^u, \ldots, B_T^u >$, where each basket B_t^u denotes the basket of items consumed by user u at time $t (\in [1, T])$, B_t^u is a non-empty subset of I, i.e., $B_t^u \subseteq I$. Each item $i \in B_t^u$ is associated with a user feedback-based preference state f, which represents user feedback on the item. The positive state f indicates the user consumed the item, while the negative state $\hat{f}$ indicates the user did not consume the item. To simplify the item state in terms of the terminology in negative sequence analysis [7], we use i to represent a positive item that has a positive preference state, while $\neg i$ refers to a negative item, i.e., item i has a negative state. For example, given an item set $I = \{tomato, pumpkin, candy, chocolate, cookie\}$ and a basket $B_t^u = (pumpkin, candy, chocolate)$, which only represents positive items, B_t^u can be further encoded as $(\neg tomato, pumpkin, candy, chocolate, \neg cookie)$ to involve both positive and negative items to form an entire-item-based sequence. This positive/negative items-based representation can reflect scenarios such as u selects items *pumpkin*, *candy*, and *chocolate* but unselects *tomato* and *cookie* at time t.

With all the positive items and baskets purchased by all users, denoted as $D := \{S^{u_1}, S^{u_2}, \ldots, S^{u_{|U|}}\}$, the objective of *interactive sequential basket recommendation* is to iteratively suggest the top-K items in each of the next p sequential target baskets $\{B_{t+1}^u, \ldots, B_{t+p}^u\}$ to user u as a result of predicting a personalized ranking R with its item set starting with $R_{u,t} \subset I^2$ at the initial time t (i.e., for the next time $t+1$). For items $i_\alpha, i_\beta \in B_t^u$, $r_{i_\alpha} < r_{i_\beta}$ represents that the rank r_{i_α} of item i_α is higher that the rank r_{i_α} of item i_β, i.e., user u prefers i_α over i_β at time t.

4 THE HAEM MODEL

Here, we introduce the proposed HAEM model and its modules in detail. We also explain the training and parameterization of the model and in particular how to support sequential basket recommendations with iterative refinement.

4.1 The HAEM Architecture

Figure 2 shows the HAEM architecture and working process. To fulfil the objective of iSBRS described in Section 3, HAEM has three modules: *a basket encoder*, *a sequence encoder*, and *a prediction decoder*. They work together to make sequential basket recommendations by jointly modeling both intra- and inter-basket relations in user sequential behaviors and then refining future recommendations by incorporating the positive/negative feedback received from users on the recommended baskets for each next basket. Given the behavior sequence S^u of user u at time t, the *basket encoder* first learns a basket representation for each basket B_t^u ($\in S^u$) to capture the compound intra-basket relations over the items within B_t^u. The basket encoder generates a sequence of basket representations where each representation $\boldsymbol{b}_t^u$ corresponds to a basket with all items coupled. The *sequence encoder* then learns the inter-basket relations between baskets in S^u to obtain a sequence of basket representations $\boldsymbol{s}_t^u$ built on all the learned basket representations $\{\boldsymbol{b}_t^u\}$. $\boldsymbol{s}_t^u$ captures the interactions and couplings between items across the adjacent baskets and thus the inter-basket couplings between all sequential behaviors of user u. $\boldsymbol{s}_t^u$ also pays higher attention to the baskets surrounding a target basket B_t^u to reflect the usual fact that more recent behaviors may have a higher impact on the target behavior. Finally, the *prediction decoder* predicts the next target basket $B_{t'}^u$ ($t' \in [t+1, t+p]$, $p > 0$ is the number of next baskets to be continuously predicted) by modeling the basket's context representations from the learnt sequence representations and refining the recommendations per the feedback from each user on previously recommended baskets.

4.2 The Basket Encoder

The basket encoder represents the intra-basket couplings. It captures two types of item representations as two sets of vectors and their couplings in terms of item-specific and item-pairwise relations. Each item i is represented in terms of embedding as an embedding vector $\boldsymbol{v}$ and a latent representation as a latent feature vector $\boldsymbol{h}$. The embedding vector $\boldsymbol{v}$ is an informative and

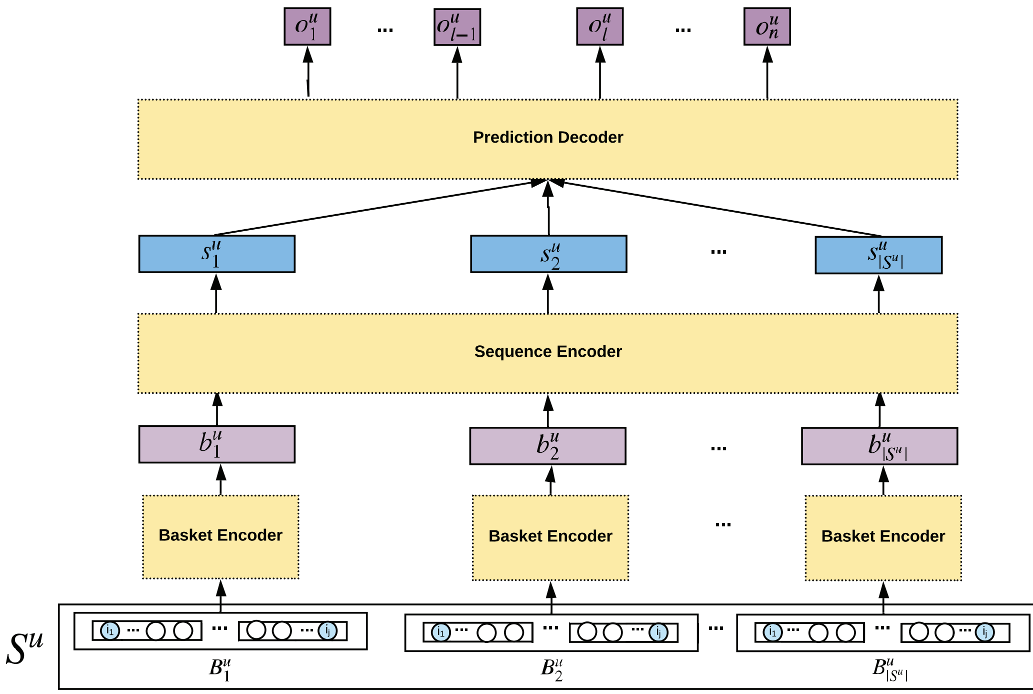


Fig. 2. The HAEM model for interactive sequential basket recommendation. HAEM is composed of a basket encoder, a sequence encoder, and a prediction decoder.

lower-dimensional representation of an item to represent its individual information (per one-hot vector in this work), while latent feature vector $\boldsymbol{h}$ represents the latent interaction information (here in terms of a FM model) to capture the pairwise intra-basket relation between a pair of items. As discussed in Section 1, compared with single items, a pair of relevant items may have higher impact on revealing user preference, and thus the pairwise intra-basket relations identify the significant item pairs to improve recommendation.

Taking the items in basket $B_t^u \in S^u$ of user u as an input example, the basket encoder learns a corresponding basket representation $\boldsymbol{b}_t^u$ by modeling the compound intra-basket item relations $y_a^{u,t}$ within B_t^u. We learn both linear and pairwise intra-basket relations using the FM mechanism, as shown in Figure 3. Inspired by the FM-based work in References [20, 34], the intra-basket relations $y_a^{u,t}$ for basket B_t^u are learned as follows:

$$y_a^{u,t} = \boldsymbol{W}_l \sum_{j=1}^{|B_t^u|} w_j^t \boldsymbol{h}_j^t + \boldsymbol{W}_p \sum_{j=1}^{|B_t^u|} \sum_{k=j+1}^{|B_t^u|} \hat{w}_{jk}^t \langle \boldsymbol{v}_j^t, \boldsymbol{v}_k^t \rangle \boldsymbol{h}_j^t \odot \boldsymbol{h}_k^t. \tag{1}$$

The first term learns the linear intra-basket relations per item in the basket, and the second term learns the pairwise intra-basket relations between every two items. $\boldsymbol{h}_j^t \in \mathbb{R}^H$ is the embedding representation vector of item $i_j^t \in B_t^u$, and w_j is the weight of item i_j^t w.r.t. the intra-basket relation $y_a^{u,t}$, i.e., the contribution rate of i_j^t to $y_a^{u,t}$. Further, $\boldsymbol{v}_j^t \in \mathbb{R}^F$ denotes the latent feature vector of item i_j^t, and $\langle \boldsymbol{v}_j^t, \boldsymbol{v}_k^t \rangle$ is the inner-product of latent vectors $\boldsymbol{v}_j^t$ and $\boldsymbol{v}_k^t$ to capture the factorized interactions between items i_j^t and i_k^t [34, 37]. $\boldsymbol{h}_j^t \odot \boldsymbol{h}_k^t$ is the element-wise product of two embedding vectors, and $\langle \boldsymbol{v}_j^t, \boldsymbol{v}_k^t \rangle \boldsymbol{h}_j^t \odot \boldsymbol{h}_k^t$ captures the second-order relations between i_j^t and i_k^t in

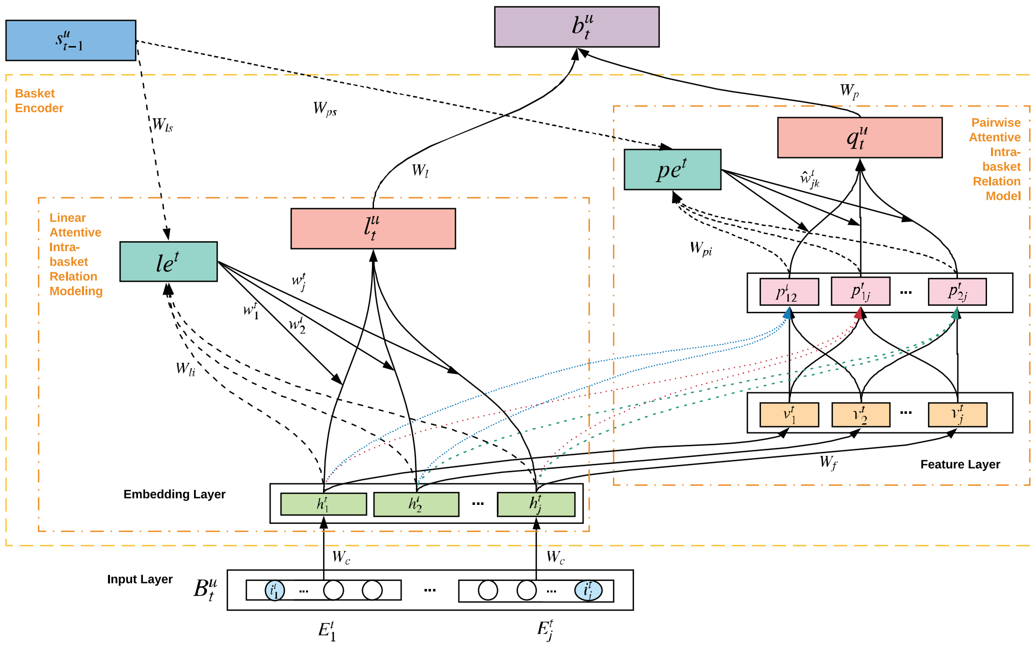


Fig. 3. The basket encoder. It maps a basket to its basket representation and models both linear and pairwise attentive intra-basket relations between items in the basket.

the embedding space [18, 20]. Last, $\hat{w}^t_{jk}$ is the weight of pairwise relation between items i^t_j and i^t_k w.r.t. $y^{u,t}_a$ to reflect the contribution rate of this pair of items to $y^{u,t}_a$. Parameters $\boldsymbol{W}_l \in \mathbb{R}^{B\times H}$ and $\boldsymbol{W}_p \in \mathbb{R}^{B\times H}$ fully connect the linear and pairwise vectors to $y^{u,t}_a$. In the following, we explain their implementation.

First, we convert the input basket B^u_t to an input embedding. Each item $i^t_j \in B^u_t$ is encoded as a *one-hot* vector $\boldsymbol{E}^t_j$, where only the unit at position j is set to 1 while the others are set to 0. The one-hot vectors of all items in B^u_t constitute the input units in the bottom layer of Figure 3. Since $\boldsymbol{E}^t_j$ only represents the meaningless ID information, an embedding layer is created to map the sparse one-hot vector of an item to an informative continuous low-dimensional representation—a H-dimensional vector $\boldsymbol{h_j}^t \in \mathbb{R}^H$ as the embedding representation vector of i^t_j. The weight matrix $\boldsymbol{W}_c \in \mathbb{R}^{H\times|I|}$ fully connects the input layer and embedding layer, namely:

$$\boldsymbol{h}^t_j = \sigma(\boldsymbol{W}_c\boldsymbol{E}^t_j). \tag{2}$$

Further, we obtain the overall linear attentive intra-basket relations $\boldsymbol{l}^u_t \in \mathbb{R}^H$ between the items in B^u_t by a weighted integration of the item embedding vectors $\{\boldsymbol{h}^t_j | i^t_j \in B^u_t\}$,

$$\boldsymbol{l}^u_t = \sum\nolimits_{j=1}^{|B^u_t|} w^t_j \boldsymbol{h}^t_j. \tag{3}$$

The integration weight w^t_j captures the contribution rate of item i^t_j to the linear intra-basket relations in basket B^u_t. It is learned by the linear attention layer, as depicted in the left part of Figure 3. In a basket, the contribution and significance of an item depend not only on the item's contextual items in the basket [45] but also the previous baskets [13]. Therefore, our linear attentive model calculates the integration weights using a softmax layer as shown in Equations (4) and (5),

$$w_j^t = softmax(\boldsymbol{\alpha}^T \boldsymbol{l_e}_j^t) = \frac{exp(\boldsymbol{\alpha}^T \boldsymbol{l_e}_j^t)}{\sum_{k=1}^{|B_t^u|} exp(\boldsymbol{\alpha}^T \boldsymbol{l}_k^t)}. \tag{4}$$

$\boldsymbol{\alpha}$ is a context vector shared by all items, which acts as a high-level representation of the informative factors over the items [25] and is jointly trained with all the other components [2]. In addition, vector $\boldsymbol{s}_{t\text{-}1}^u \in \mathbb{R}^B$ is the annotation (contextual) representation of the previous basket $B_{t\text{-}1}^u$, matrices $\boldsymbol{W}_{ls}$ and $\boldsymbol{W}_{li}$ fully connect vectors $\boldsymbol{s}_{t\text{-}1}^u$ and $\boldsymbol{h}_j^t$ to form a contextual linear intra-basket relation vector $\boldsymbol{l_e}_j^t$ for each item j (or k in the above formula) at time t, and $\boldsymbol{b}_l$ is the bias vector,

$$\boldsymbol{l_e}_j^t = tanh(\boldsymbol{W}_{ls}\boldsymbol{s}_{t\text{-}1}^u + \boldsymbol{W}_{li}\boldsymbol{h}_j^t + \boldsymbol{b}_l). \tag{5}$$

In addition to the above linear embedding, we further capture the latent feature-based pairwise intra-basket relation. Inspired by the FM mechanism, we create a feature layer to map the embedding vector of each item to a latent feature vector, denoted as $\boldsymbol{v_j}^t \in \mathbb{R}^F$. It represents the latent feature interaction information of item i_j^t and is used to capture its pairwise intra-basket relation with another item. Since the items hold homogeneous feature information, a shared weight matrix $\boldsymbol{W_f} \in \mathbb{R}^{F \times H}$ is used to fully connect the embedding layer and the feature layer as depicted in Equation (6), where $\boldsymbol{b_f}$ is the bias vector,

$$\boldsymbol{v}_j^t = \sigma(\boldsymbol{W}_f \boldsymbol{h}_j^t + \boldsymbol{b}_f). \tag{6}$$

By implementing the FM mechanism, the pairwise intra-basket relation $\boldsymbol{p}_{jk}^t$ between a pair of items i_j^t and i_k^t can be captured by the embedding vectors and feature vectors:

$$\boldsymbol{p}_{jk}^t = \langle \boldsymbol{v}_j^t, \boldsymbol{v}_k^t \rangle \boldsymbol{h}_j^t \odot \boldsymbol{h}_k^t. \tag{7}$$

Similarly to the linear attentive intra-basket relation modeling, the overall pairwise attentive intra-basket relation $\boldsymbol{p}_t^u \in \mathbb{R}^H$ between items in B_t^u is captured by the weighted integration of the $\boldsymbol{p}_{jk}^t$ between each pair of items,

$$\boldsymbol{p}_t^u = \sum_{j=1}^{|B_t^u|} \sum_{k=j+1}^{|B_t^u|} \hat{w}_{jk}^t \boldsymbol{p}_{jk}^t. \tag{8}$$

Similarly to the design of the linear attention model, the integration weight $\hat{w}_{jk}^t$ is learned by the pairwise attention model as shown in the right part of Figure 3, namely

$$\hat{w}_{jk}^t = \frac{exp(\boldsymbol{\beta}^T \boldsymbol{p_e}_{jk}^t)}{\sum_{m=1}^{|B_t^u|} \sum_{n=m+1}^{|B_t^u|} exp(\boldsymbol{\beta}^T \boldsymbol{p}_{mn}^t)}. \tag{9}$$

$\boldsymbol{\beta}$ is a context vector shared by all pairs of items, matrices $\boldsymbol{W}_{ps}$ and $\boldsymbol{W}_{pi}$ fully connect vectors $\boldsymbol{s}_{t\text{-}1}^u$ and $\boldsymbol{p}_{jk}^t$ to form the contextual pairwise intra-basket relation vector $\boldsymbol{p_e}_{jk}^t$ for each item pair respectively, and $\boldsymbol{b}_p$ is the bias vector,

$$\boldsymbol{p_e}_{jk}^t = tanh(\boldsymbol{W}_{ps}\boldsymbol{s}_{t\text{-}1}^u + \boldsymbol{W}_{pi}\boldsymbol{p}_{jk}^t + \boldsymbol{b}_p). \tag{10}$$

Finally, the overall basket representation $\boldsymbol{b}_t^u$ is obtained by integrating both linear and pairwise intra-basket couplings as shown in Equation (11),

$$\boldsymbol{b}_t^u = \sigma(y_a^{u,t}) = \sigma(\boldsymbol{W}_l \boldsymbol{l}_t^u + \boldsymbol{W}_p \boldsymbol{p}_t^u). \tag{11}$$

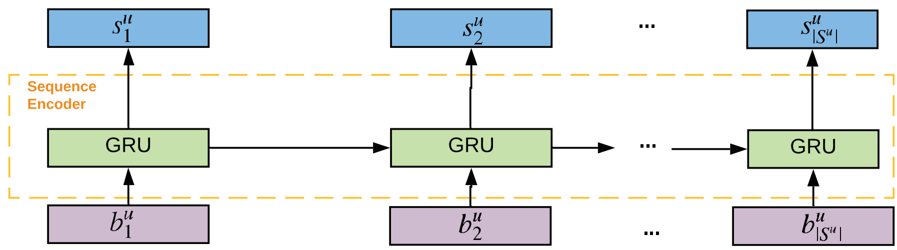


Fig. 4. The sequence encoder.

4.3 The Sequence Encoder

The above basket encoder only captures the intra-basket item relations to learn the intra-basket couplings-oriented representations $< \boldsymbol{b}_1^u, \boldsymbol{b}_2^u, \ldots, \boldsymbol{b}_t^u >$. To learn the inter-basket relations between baskets of the sequential behaviors with the sequential context of each basket, the sequence encoder applies a gated recurrent unit (GRU)-based recurrent architecture on the basket encoder to propagate the sequential information between every two adjacent baskets to capture the global inter-basket couplings w.r.t. the sequential features of all baskets [50], as depicted in Figure 4. Given the annotation representation $\boldsymbol{s}_{t\text{-}1}^u$ of user u at time t-1 and the basket representation $\boldsymbol{b}_t^u$, the contextual representation $\boldsymbol{s}_t^u$ for time t is calculated by Equation (12), which indicates the contextual representation of the preference of user u at time t,

$$\boldsymbol{s}_t^u = GRU(\boldsymbol{s}_{t\text{-}1}^u, \boldsymbol{b}_t^u) = (1 - z_t^{se})\boldsymbol{s}_{t\text{-}1}^u + z_t^{se}\hat{\boldsymbol{s}}_t^{se}. \tag{12}$$

$\hat{\boldsymbol{s}}_t^{se}$ and $\boldsymbol{z}_t^{se}$ are the update state and update gate of the sequence encoder respectively, $\boldsymbol{z}_t^{se}$ allows each annotation representation to maintain the activation of its previous basket,

$$\hat{\boldsymbol{s}}_t^{se} = tanh(\boldsymbol{W}_h^{se}\boldsymbol{b}_t^u + \boldsymbol{U}_h^{se}(\boldsymbol{r}_t^{se} \odot \boldsymbol{s}_{t\text{-}1}^u)), \tag{13}$$

$$\boldsymbol{z}_t^{se} = \sigma(\boldsymbol{W}_z^{se}\boldsymbol{b}_t^u + \boldsymbol{U}_z^{se}\boldsymbol{s}_{t\text{-}1}^u), \tag{14}$$

where $\boldsymbol{r}_t^{se}$ is the reset gate, which controls how much and what information from the previous basket should be reset [2] and is given as follows:

$$\boldsymbol{r}_t^{se} = \sigma(\boldsymbol{W}_r^{se}\boldsymbol{b}_t^u + \boldsymbol{U}_r^{se}\boldsymbol{s}_{t\text{-}1}^u). \tag{15}$$

Subsequently, with the sequence of basket representations $< \boldsymbol{b}_1^u, \boldsymbol{b}_2^u, \ldots, \boldsymbol{b}_{t_{u-1}}^u >$ as input, the sequence encoder generates the corresponding sequence of annotation representations, i.e., $< \boldsymbol{s}_1^u, \boldsymbol{s}_2^u, \ldots, \boldsymbol{s}_{|S^u|}^u >$. This sequence of annotation representations carries both the contextual behavioral information of a user behavior sequence for basket behaviors by learning both intra-basket and inter-basket item relations. It will be used to predict the next baskets.

4.4 The Prediction Decoder

With the sequence of annotation representations learned by the sequence encoder and the user feedback on the recommended previous baskets, the prediction decoder makes a prediction for the next several target baskets sequentially, as depicted in Figure 5. This is motivated by the intuition that the consumption preference of a user u for the next target basket $B_{t'}^u$ is likely influenced by or associated with his/her recent past consumption behaviors S^u as shown in Figure 1 [13, 29] and positive/negative feedback on those previously recommended baskets [52] to refine future recommendations.

Accordingly, the prediction decoder predicts the next basket $B_{t'}^u$ at time $t' \in [t+1, t+p]$ based on the user preference score $\boldsymbol{o}_{t'}^u$ w.r.t. all items. The preference score is generated based on the hidden state $\boldsymbol{p}_{t'}^u$ of an RNN for time t'. $\boldsymbol{p}_{t'}^u$ is calculated by involving its previous hidden state

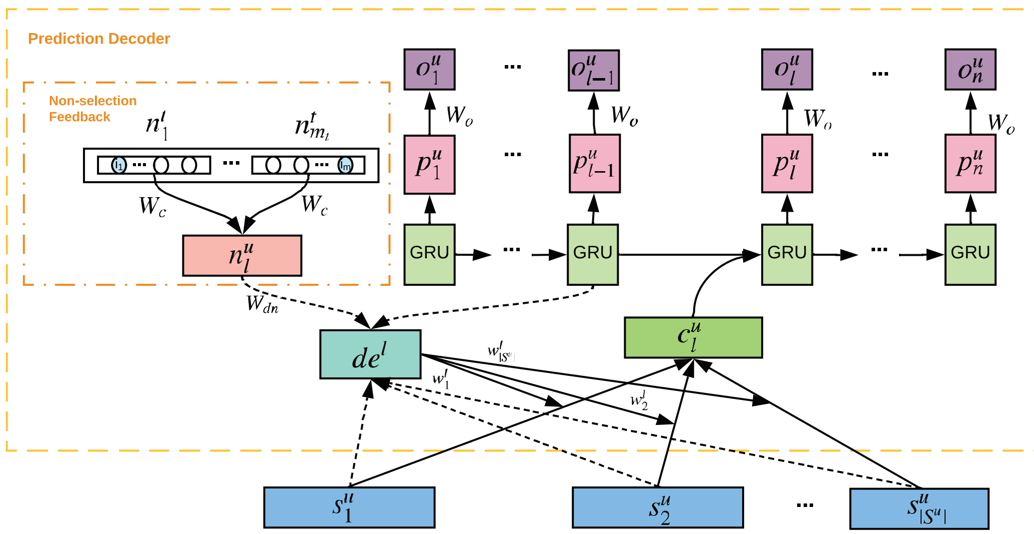


Fig. 5. The prediction decoder. It makes predictions for sequential basket recommendation based on the learned annotation sequence. $\boldsymbol{s}$ refers to the contextual annotated representation of baskets, $\boldsymbol{p}$ refers to the hidden state, $\boldsymbol{n}$ corresponds to unselected item representation, $\boldsymbol{c}$ is the contextual representation, and $\boldsymbol{o}$ refers to the user preference score.

$\boldsymbol{p}^u_{t'\text{-}1}$, the user feedback vector $\hat{\boldsymbol{f}}^u_{t'\text{-}1}$ that captures the feedback on whether a recommended item is selected or unselected by the user, and a context vector $\boldsymbol{c}^u_{t'}$ that consists of the sequence of previous baskets consumed by the user and is used as the context of the current basket,

$$\boldsymbol{o}^u_{t'} = \boldsymbol{W}_o \boldsymbol{p}^u_{t'}, \tag{16}$$

$$\boldsymbol{p}^u_{t'} = RNN(\boldsymbol{p}^u_{t'\text{-}1}, \hat{\boldsymbol{f}}^u_{t'\text{-}1}, \boldsymbol{c}^u_{t'}) = (1 - \boldsymbol{z}^{pd}_{t'})\boldsymbol{p}^u_{t'\text{-}1} + \boldsymbol{z}^{pd}_{t'}\hat{\boldsymbol{p}}^{pd}_{t'}. \tag{17}$$

Here, the weight matrix $\boldsymbol{W_o}$ fully connects the RNN hidden states to the output layer, and $\hat{\boldsymbol{p}}^{pd}_{t'}$ and $\boldsymbol{z}^{pd}_{t'}$ are the update state and update gate of the prediction decoder respectively, given below:

$$\hat{\boldsymbol{p}}^{pd}_{t'} = tanh(\boldsymbol{W}^{pd}_h \hat{\boldsymbol{f}}^u_{t'\text{-}1} + \boldsymbol{U}^{pd}_h(\boldsymbol{r}^{pd}_{t'} \odot \boldsymbol{p}^u_{t'\text{-}1}) + \boldsymbol{C_h}\boldsymbol{c}^u_{t'}), \tag{18}$$

$$\boldsymbol{z}^{pd}_{t'} = \sigma(\boldsymbol{W}^{pd}_z \hat{\boldsymbol{f}}^u_{t'\text{-}1} + \boldsymbol{U}^{pd}_z \boldsymbol{p}^u_{t'\text{-}1} + \boldsymbol{C_z}\boldsymbol{c}^u_{t'}), \tag{19}$$

where $\boldsymbol{r}^{pd}_{t'}$ is the reset gate, which is given as follows:

$$\boldsymbol{r}^{pd}_{t'} = \sigma(W^{pd}_r \hat{\boldsymbol{f}}^u_{t'\text{-}1} + U^{pd}_r \boldsymbol{p}^u_{t'\text{-}1} + C_r \boldsymbol{c}^u_{t'}). \tag{20}$$

Here, the user feedback vector $\hat{\boldsymbol{f}}^u_{t'\text{-}1}$ is constructed as a multi-hot vector for the recommended items in basket $B^u_{t'\text{-}1}$, where the units corresponding to the items recommended and also selected by the user are set to 1 while items that were recommended but not yet selected are set to 0. When we predict target basket $B^u_{t'}$, user feedback at time t'-1 is available and thus is used to construct the feedback vector $\hat{\boldsymbol{f}}^u_{t'\text{-}1}$, which is then fed into Equation (17). In addition, the context vector $\boldsymbol{c}^u_{t'}$ of user u at time t' is modelled as a weighted integration of the annotation representations (i.e., previous baskets), namely

$$\boldsymbol{c}^u_{t'} = \sum\nolimits_{j=1}^{|S^u|} w^{t'}_j \boldsymbol{s}^u_j, \tag{21}$$

where the integration weight $w_j^{t'}$ captures the influence of user preference for each previous basket B_j^u on the target basket B_t^u. Accordingly, we set the weights in terms of two perspectives. On one hand, the weight w_j^t is associated with both the annotation vector $\boldsymbol{s}_j^u$ and the hidden state of previous prediction $\boldsymbol{p}_{t'\text{-}1}^u$. However, as discussed in Section 1, the recommended but unselected items in the user feedback reflect the divergence of user preferred items from the recommendations and are thus valuable to refine future recommendations. Consequently, weight $w_j^{t'}$ is computed as follows:

$$w_j^{t'} = softmax(\boldsymbol{\gamma}^T \boldsymbol{de}_j^{t'}) = \frac{exp(\boldsymbol{\gamma}^T \boldsymbol{de}_j^{t'})}{\sum_{k=1}^{|S^u|} exp(\boldsymbol{\gamma}^T \boldsymbol{de}_k^{t'})}, \tag{22}$$

$$\boldsymbol{de}_j^{t'} = tanh(\boldsymbol{W}_{ds}[\boldsymbol{s}_j^u : \boldsymbol{p}_{t'\text{-}1}^u] + \boldsymbol{W}_{dn}\boldsymbol{n}_{t'\text{-}1}^u + \boldsymbol{b}_d). \tag{23}$$

$\boldsymbol{W}_{ds}$ and $\boldsymbol{W}_{dn}$ are the weight matrices and $\boldsymbol{b}_d$ is the bias vector. In addition, the unselection feedback vector $\boldsymbol{n}_{t'\text{-}1}^u \in \mathbb{R}^H$ is the integration of the embedding representation of $m_{t'\text{-}1}$ recommended but unselected items $\{i_j^{t'\text{-}1}\}$ $(i_j^{t'\text{-}1} \in I)$. Here, $m_{t'\text{-}1}$ is the number of unselected items in the basket recommended for time t'-1. Assume $\mathbb{B}_{t'\text{-}1}^u$ is the set of items recommended for basket $B_{t'\text{-}1}^u$, then the set of negative items is generated via $\{i_j^{t'\text{-}1}\} \equiv \{i | i \in \mathbb{B}_{t'\text{-}1}^u \wedge i \notin B_{t'\text{-}1}^u\}$. For the example in Figure 1, $\mathbb{B}_3^u = \{chip,\ toffee\ apple\}$ while $B_3^u = \{toffee\ apple\}$ and the recommended item *chip* is unselected, thus $m_3 = 1$ and i_1^3 is *chip*. $\boldsymbol{n}_j^{t'\text{-}1}$ is the one-hot embedding vector of unselected item $i_j^{t'\text{-}1}$, the unselection feedback vector $\boldsymbol{n}_{t'\text{-}1}^u \in \mathbb{R}^H$ is calculated by Equation (24), namely:

$$\boldsymbol{n}_{t'\text{-}1}^u = \frac{1}{m_{t'\text{-}1}} \sum_{j=1}^{m_{t'\text{-}1}} \sigma(\boldsymbol{W}_c \boldsymbol{n}_j^{t'\text{-}1}). \tag{24}$$

As a result, the context vector $\boldsymbol{c}_{t'}^u$ can identify and pay attention to the important cross-basket item combinations in sequential behavior S^u for the prediction of target basket $B_{t'}^u$, such as the identification of the sub-sequence $< pumpkin, (candy, chocolate) >$ for target basket B_3^u in Figure 1. This is owing to the integration weight $w_j^{t'}$ in Equation (22), which captures these significant baskets, while the linear integration weight w_j^t in Equation (4) and the pairwise integration weight $\hat{w}_{jk}^t$ in Equation (9) of the basket encoder respectively capture the significant items and item combinations within a basket. Consequently, the prediction decoder generates a user preference score $\boldsymbol{o}_{t'}^u$ for the prediction of target basket $B_{t'}^u$, where the vth element $o_{t',v}^u$ shows the preference score of user u to item i_v at time t', and a higher score indicates a higher preference of i_v.

4.5 The HAEM Learning and Prediction Process

As HAEM involves user feedback and generates the top-K ranking of recommended items for each target basket, we apply the BPR to the HAEM learning process. BPR is a state-of-the-art pairwise ranking method on feedback data [35] and shows good suitability as an objective function for behavior prediction tasks [13, 29, 30, 50]. HAEM implements BPR in terms of a logistic function $\sigma(\cdot)$ by assuming that a selected item v in the basket at a specific time is more preferred by a user than an unselected v'. Accordingly, we maximize the following probability:

$$\arg\max p(u, t, v \succ v') = \arg\max \sigma(o_{t,v}^u - o_{t,v'}^u). \tag{25}$$

By adding up the log-likelihood and regularization, the objective function is minimized:

$$\arg\min J = \arg\min \left(\sum ln \left(1 + e^{-(o_{t,v}^u - o_{t,v'}^u)}\right) + \frac{\lambda}{2} ||\Theta||^2 \right). \tag{26}$$

Table 2. Statistics of Experimental Datasets

Statistics	Ta-Feng	IJCAI-15
#Users $\|U\|$	9,238	24,889
#Items $\|I\|$	7,982	35,272
#Baskets	67,964	183,472
Average basket size	7.4	9.1
Average number of items per User	43.7	67.1

Θ denotes the set of all parameters to be estimated, and λ is the hyper-parameter to control the power of the regularization term. The above objective function is optimized by backpropagation through time and the parameters are updated by stochastic gradient descent until convergence.

After the training, HAEM makes the sequential basket recommendation as follows. Given a sequential behavior S^u, HAEM first produces the preference score vector $\boldsymbol{o}^u_{t'}$ for basket $B^u_{t'}$ according to Equation (16), which indicates the preference ranking over items at time t'. The top-ranked items at time t' are recommended to user u as the next basket for feedback. Second, by collecting the user feedback on these recommendations, the user feedback vector $\hat{\boldsymbol{f}}^u_{t'}$ and unselection feedback vector $\boldsymbol{n}^u_{t'}$ are updated. Third, HAEM generates the preference score vector $\boldsymbol{o}^u_{t'+1}$ for the further next basket $B^u_{t'+1}$ at time $t'+1$ using the prediction decoder. Last, HAEM further produces an item ranking for the target basket $B^u_{t'+1}$. HAEM repeats this process for sequential recommendations of the next baskets.

5 EXPERIMENTS AND EVALUATION

An empirical analysis of HAEM on two real-life datasets is conducted in terms of both accuracy and novelty.

5.1 Datasets and Preparation

Two real-life datasets are used to evaluate HAEM:

- Ta-Feng[2] is a public grocery shopping dataset, covering numerous baskets of purchased products collected from November 2000 to February 2001.
- IJCAI-15[3] is a real-life dataset collected from Tmall.com, containing user shopping logs for the six months before and on the "Double 11" day (November 11th).

Both datasets only record the purchase dates without further specific time points, so we treat the items purchased by the same user on the same day as a basket, and all of a user's baskets are sorted temporally in a sequence. We empirically filter the sparse users/items for those users who bought less than 10 baskets or those items purchased by less than 10 users from the raw data for several reasons. (1) Our iSBRS involves both past baskets as context and follow-up baskets for iterative interactive recommendation. A good number of reasonably long sequences of purchased baskets are required so that we can conduct the iterative prediction of multiple next-baskets based on a good number of input baskets for each user. A smaller number of baskets does not satisfy this modeling requirement, including the p next-baskets recommendation and for the top-K recommendation. (2) Both the real-life datasets are rather sparse in terms of user distributions and item distributions [14, 22, 31]. As shown in Table 2, the average number of baskets in Ta-Feng is 7.4 and

[2]http://stackoverflow.com/questions/25014904/download-link-for-ta-feng-grocery-dataset
[3]https://tianchi.aliyun.com/datalab/dataSet.htm?id=1

in IJCAI-15, it is 9.1. With the above filtering, we still retain a reasonably large set of data for the modeling and experiments (e.g., fitting the five-fold cross validation need) as shown in Table 2.

For each dataset, a sequence of purchased baskets is obtained for each user; each sequence consists of multiple sequential baskets. Each basket contains multiple items, where the daily order of the items in the baskets is retained for each user. In the sequence, some of the last baskets are reserved for next-basket prediction and the previous ones for context learning. At each time, one basket is chosen for next-basket prediction, with its previous baskets as context. Specifically, for user u, we choose his/her last three (i.e., $p = 3$) baskets as the *target baskets* for interactive sequential basket prediction and feedback verification and their previous baskets form the input sequence S^u as the context. The statistics of the datasets for the experiments are shown in Table 2.

Further, we randomly choose 80% of the extracted basket sequences for training and the remainder are used for testing. Fivefold cross-validation [31] is undertaken on the data for training and testing, resulting in the average testing results reported in the following evaluation.

5.2 Baseline Methods

To the best of our knowledge, there are no existing recommenders for sequential basket recommendation that jointly learn both intra- and inter-basket relations and positive/negative feedback in sequential recommendation. We thus compare HAEM with the following representative state-of-the-art NBRS and session-based methods for the experiments to evaluate their quality of next-basket/next-item recommendation. We also test the effect of unselection feedback-based recommendation refinement embedded in HAEM for each next-basket recommendation and sequential basket recommendation.

- *FPMC* [36]: A hybrid model combines the first-order Markov chains with matrix factorization for next-basket recommendation, which factorizes the personal transition matrix between items with a pairwise interaction model.
- *HRM* [41]: A hierarchical representation-based model explores both general user preference and the last basket for next-basket recommendation.
- *DREAM* [50]: An RNN-based model learns a dynamic representation of a user to capture user preference on all baskets for next-basket recommendation.
- *ATEM* [45]: An attention-based model learns an attentive context embedding over all the observed items within a transaction for session-based next-item recommendation.
- *DEERS* [52]: A deep reinforcement learning-based model automatically learns the optimal recommendation strategies by incorporating both positive and negative feedback, which can continuously improve its strategies by modeling the interactions with users.
- *HAEM_L*: A sub-model of HAEM only models the linear intra-basket relations between items in the basket encoder.
- *HAEM_S*: A sub-model of HAEM does not contain the negative feedback-based refinement mechanism in the prediction decoder.
- *HAEM*: The full HAEM model consists of intra-/inter-basket relations and feedback refinement.

HAEM_L, HAEM_S, and HAEM are created for the ablation test. HAEM_L only models linear intra-basket relations and ignores pairwise intra-basket relations in the basket encoder. The comparison between HAEM_L and HAEM is to evaluate the effect and contribution of the pairwise intra-basket relations on modeling sequential basket recommendation. In addition, by replacing the attention layer of the basket encoder in HAEM_L by max/average pooling as well as replacing the context vector of the prediction decoder by the last annotation representations of the sequence encoder, HAEM_L degrades to DREAM for NBRS. The effect and contribution of the attention layers in both the basket encoder and the prediction decoder is evaluated by comparing HAEM_L with DREAM for NBRS. Finally, the significance of the negative feedback-based refinement in the prediction decoder is evaluated by comparing HAEM_S with HAEM.

5.3 Experiment Settings

In the training, the sequential baskets of each user in the training set are fed into the models in batches to learn the sequence of annotation representations. During testing, the context vector is built based on the sequence of annotation representations and is then used to predict the first target basket. The actual first target basket in the source data is used as the ground truth. Those items recommended in the target but nonoccurring in the ground truth are unselected items (negative feedback). The unselected items and the actual first basket are treated as the input of the trained HAEM model to further predict the second target basket. This interactive recommendation-feedback process is repeated until the pth target basket is predicted.

To evaluate the recommendation performance, we select the top-K items as the recommended basket for each user, denoted as $R^u_{t'}$ for time t', where $R^u_{t'}[j]$ represents the jth recommended item. While theoretically K could be any reasonable value, its selection generally corresponds to major perspectives including (1) the design and constraints of a recommender, (2) the specific characteristics of the data to be tested, and (3) practical experience and better practice. In this work, we typically choose $K = 5$ for iSBRS and $k = 5$ and $k = 10$ for NBRS for the following reasons. The first is the limitation of the average basket size in the experimental datasets, where most baskets have a relatively small number of items, so 5 seems to be a fair cutoff to cater for most. Second, as the iSBRS involves negative feedback on unselected items, the length of the unselected items is usually very small with 5 as a good average for most coverage. In fact, a larger K may lead to degraded performance and fluctuations when using different methods, since a larger proportion of the items in the ranking list may more likely mismatch the ground-truth items in a basket, which would lead to unfair comparison, underfitting and poor performance. Further, while technically it is possible to set a suitable K for each basket, different baskets likely involve different top-K items. It would be very time-consuming to personalize K for each basket. In addition, in related research with the top-K setting [36, 41, 50], top-5 [33], or 10 items are often empirically selected for testing that may consider the reality that most customers are only interested in the first few items recommended on the first page [24], also due to limited item observability.

Experiments were undertaken on a cluster node with Intel Xeon W3690 CPU of 3.47 GHz with 12-GB memory. The hyperparameter settings of HAEM and the baselines are tuned as follows: (1) choosing a reasonable step size and value range for each parameter inspired by the parameter roles and sensitivity in the model, (2) traversing the value combinations with multiple loops if there is more than one parameter to test the performance, and (3) inheriting the parameter settings in the original papers for the baseline methods and fine-tuning them on the given datasets in our experiments. We report the parameter settings corresponding to the best performance achieved in our experiments in the following.

The number of factors is set as 5 for FPMC when the best performance is achieved and its accuracy worsens when more factors are adopted in the two datasets, consistent with [24, 44, 45]. For HRM, the number of negative samples is empirically set as 25, and the regularization constant and drop ratio are set as 0.001 and 60%, respectively, inspired by their original design [41]. For DREAM, the regularization constant is set as 0.001, and the number of RNN hidden layers is set as 3 [50]. The aggregation operations for HRM and DREAM are max pooling, because it has an advantage over average pooling. For DEERS, the length of the modelled feedback is set as 10, and the discounted factor and pairwise regularization constant are set as 0.9 and 0.1 empirically [52].

Finally, for the HAEM-based methods, the hyper-parameter λ is empirically set as 0.001 for the best performance. The other parameter settings for specific experiments (e.g., top-K) are introduced in the respective test in the following sections.

5.4 Evaluation Measures

We evaluate the *accuracy* and *novelty* of the recommendations made by HAEM against those of the baseline methods. *Accuracy* is evaluated in terms of two popular measures *F1-score@K* and Normalized Discounted Cumulative Gain (NDCG@K), which compare the predicted baskets with those preferred by users. The final $F1$-*score* and $NDCG@K$ of HAEM are obtained for predicting both a single next basket and the sequential next baskets. The latter evaluates the effect of involving user feedback on refining recommendations.

- *F1-score* is the harmonic mean of precision and recall, which are calculated as follows for each time t':

$$Precision(B^u_{t'}, R^u_{t'}) = \frac{|B^u_{t'} \cap R^u_{t'}|}{|R^u_{t'}|}, \tag{27}$$

$$Recall(B^u_{t'}, R^u_{t'}) = \frac{|B^u_{t'} \cap R^u_{t'}|}{|B^u_{t'}|}, \tag{28}$$

$$F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{29}$$

- *(NDCG@K)* is a cumulative measure of ranking quality and takes into account the order of the recommended items in the list, which is calculated as follows:

$$NDCG@K = \frac{1}{N_K} \sum_{j=1}^{K} \frac{2^{I(R^u_{t'}[j] \in B^u_{t'})} - 1}{log_2(j+1)}, \tag{30}$$

  where $I(\cdot)$ is an indicator function and N_K is a constant that denotes the maximum value of NDCG@K given $R^u_{t'}$.

The *novelty* of the recommended baskets addresses the usual concern with duplicated items and interest in novel items that are recommended. We take the mean context-aware novelty (MCAN) *MCAN@K* [44, 45] to quantify the discrepancy between the recommended target basket and its previous one (i.e., the context), thus indicating the novelty of newly recommended items in terms of its previous local context. Given recommended basket $R^u_{t'}$, if there are less items in $R^u_{t'}$ that also appear in its previous basket $B^u_{t'-1}$, then the novelty is higher. Accordingly, the novelty of $R^u_{t'}$, denoted as *CAN*, is defined as follows:

$$CAN = \frac{1}{p} \sum_{p} \left(1 - \frac{|R^u_{t'} \cap B^u_{t'-1}|}{|R^u_{t'}|}\right), \tag{31}$$

where p refers to the number of interactions (next target baskets) and its prediction time is $t' + p - 1$. The *mean novelty* over all N top-K recommendations, i.e., *MCAN@K* is defined as the mean of the *CAN* over all N (all users) recommendations as follows:

$$MCAN = \frac{1}{N} \sum_{N} CAN. \tag{32}$$

By substituting Equation (31) into Equation (32), we obtain Equation (33) to evaluate the novelty,

$$MCAN = \frac{1}{N} \sum_{N} \left(1 - \frac{|R^u_{t'} \cap B^u_{t'-1}|}{|R^u_{t'}|}\right). \tag{33}$$

Table 3. Accuracy of HAEM against Baselines for Next-basket Recommendation on Ta-Feng

Methods	$d = 50$		$d = 100$		$d = 150$	
	F1-score@5	NDCG@5	F1-score@5	NDCG@5	F1-score@5	NDCG@5
FPMC	0.0543	0.0763	0.0578	0.0781	0.0620	0.0813
HRM	0.0573	0.0805	0.0633	0.0823	0.0653	0.0829
DREAM	0.0630	0.0830	0.0660	0.0837	0.0685	0.0845
ATEM	0.0609	0.0828	0.0654	0.0841	0.0677	0.0848
DEERS	0.0645	0.0835	0.0667	0.0849	0.0691	0.0855
HAEM_L	0.0660	0.0868	0.0686	0.0892	0.0701	0.0897
HAEM	**0.0692**	**0.0887**	**0.0719**	**0.0904**	**0.0721**	**0.0907**

Table 4. Accuracy of HAEM against Baselines for Next-basket Recommendation on IJCAI-15

Methods	$d = 50$		$d = 100$		$d = 150$	
	F1-score@5	NDCG@5	F1-score@5	NDCG@5	F1-score@5	NDCG@5
FPMC	0.0556	0.1248	0.0584	0.1438	0.0593	0.1469
HRM	0.0606	0.1318	0.0618	0.1470	0.0653	0.1550
DREAM	0.0640	0.1515	0.0670	0.1610	0.0695	0.1665
ATEM	0.0628	0.1434	0.0652	0.1560	0.0682	0.1628
DEERS	0.0659	0.1582	0.0692	0.1654	0.0715	0.1701
HAEM_L	0.0736	0.1679	0.0767	0.1759	0.0794	0.1803
HAEM	**0.0793**	**0.1791**	**0.0825**	**0.1897**	**0.0828**	**0.1913**

MCAN@K quantifies the item difference between a recommended basket and its last adjacent basket that has been actually consumed by a user. A higher *MCAN@K* indicates that there are more items in the predicted basket that are different from those already consumed in the previous transactions, i.e., more novel items are recommended.

5.5 Accuracy Evaluation

5.5.1 Performance of Next-basket Recommendation. We first apply HAEM to next-basket recommendation and compare it with the existing NBRS methods as they cannot make sequential basket recommendations. We set three different values of dimensionality d for the embedding representations: $d \in \{50, 100, 150\}$ on Ta-Feng and IJCAI-15. Tables 3 and 4 demonstrate the five-fold cross-validation results of *F1-score*@5 and *NDCG*@5 over the datasets Ta-Feng and IJCAI-15, respectively.

The experiment results show that, of all the methods, FPMC performs the worst. This may be due to its assumption that the intra-basket relations between items are linearly independent, which is inconsistent with real-world cases, and FPMC fails to depict the sophisticated interactions between items and cannot capture the influence of item interactions. In addition, both datasets are sparse, and therefore the matrix constructed by these datasets is quite large and sparse for training this MF-based model. HRM and ATEM lag behind DREAM, because both only learn the embedding representation based on the successive transaction while neglecting all the baskets previously purchased in the behavior history, which may mean information is lost. Moreover, HRM utilizes the max pooling operation on the item representation within a basket to capture the most significant features of the input items for intra-basket relation modeling, but max pooling can neither recognize the relevant items and capture their impact on next baskets nor model any pairwise intra-basket relations. Compared with HRM, ATEM achieves higher accuracy, because it intensifies relevant items and downplays the irrelevant ones for prediction by the attention model. Compared with HRM and ATEM, which only make recommendations based on the last basket, DREAM captures the global sequential information and represents user dynamics through its recurrent structure. It thus achieves a slightly better performance. In both datasets, the *F1-score*@5 of DREAM reveals an improvement of over 6% over HRM and around 2% over ATEM. However, DREAM is weak in capturing the compound intra-/inter-basket relations between items. Thanks to the mechanism of incorporating both selection and unselection feedback to learn the optimal recommendation strategies, DEERS achieves higher recommendation accuracy than DREAM and contributes to an average improvement of over 2% over DREAM. However, similar to the drawback of DREAM, DEERS adopts an RNN with a GRU to capture user sequential preference, which tends to pay more attention to recent behaviors and is less effective in capturing the inter-basket relations between divergent baskets.

In contrast, HAEM achieves much better performance than all the baselines by a maximum of 27.4% and an average of 15.9% w.r.t. *F1-score*@5 and a maximum of 16.3% and an average of 9.5% w.r.t. *NDCG*@5 on Ta-Feng with $d = 50$, and a maximum of 42.6% and an average of 28.9% w.r.t. *F1-score*@5 and a maximum of 43.5% and an average of 27.4% w.r.t. *NDCG*@5 on IJCAI-15 with $d = 50$. HAEM shows its superiority consistently over all the baseline methods on both datasets, which illustrates the effectiveness of capturing compound intra-/inter-basket couplings for NBRS. In particular, compared with DEERS, HAEM contributes to an average improvement of 6.51% *F1-score*@5 and 6.28% *NDCG*@5 ($d = 50$) on Ta-Feng and 18.49% *F1-score*@5 and 13.45% on IJCAI-15, showing that the modelling of the intra-basket couplings helps to enhance the accuracy of next-basket recommendation.

5.5.2 Ablation Study. Here we evaluate the design effect of only modeling intra-basket couplings using HAEM_L and excluding negative feedback-based refinement using HAEM_S against involving both intra- and inter-basket couplings and positive and negative feedback using HAEM.

In Tables 3 and 4, compared with HAEM_L, which only models the linear intra-basket relations in the basket encoder, HAEM contributes to an additional 4.85% *F1-score*@5 and 2.19% *NDCG*@5 ($d = 50$) on Ta-Feng and 7.74% *F1-score*@5 and 6.67% *NDCG*@5 ($d = 50$) on IJCAI-15 over HAEM_L for next-basket recommendation. In addition, compared with DREAM, which models the intra-basket relation by max pooling, HAEM_L captures the linear intra-basket relations to intensify the relative items by the linear attention layer and thus contributes to an additional 4.76% *F1-score*@5 and 4.58% *NDCG*@5 ($d = 50$) on Ta-Feng and 15.0% *F1-score*@5 and 10.8% *NDCG*@5 ($d = 50$) on IJCAI-15 over DREAM for next-basket recommendation. This ablation analysis shows that the attention layer as well as the pairwise intra-basket relation modelling in the basket encoder significantly improves the accuracy of HAEM for NBRS.

In addition, the experiments also show that both HEAM and HAEM_L achieve better accuracy on a higher dimensionality d on two datasets but greater accuracy advantage over the baselines on a lower dimensionality. The highest *NDCG*@5 value also demonstrates that our models can more accurately predict the more highly ranked items in the recommendation list to a user. The reason for this is multifaceted. On one hand, the basket encoder in HAEM makes the learned basket representations more informative: the compound intra-basket couplings between items within a basket are learned and encoded into the basket representation, and the significant items and item combinations are intensified by the attentive mechanism. However, the basket context representations learned by the sequence encoder and prediction decoder capture the inter-basket couplings within the whole behavior sequence and recognize the significant baskets for target

Table 5. Accuracy of Negative Feedback-based Refinement on Ta-Feng

Methods	$p = 1$		$p = 2$		$p = 3$	
	F1-score@5	NDCG@5	F1-score@5	NDCG@5	F1-score@5	NDCG@5
HAEM_L_50	0.0660	0.0868	0.0679	0.0878	0.0720	0.0918
HAEM_S_50	0.0692	0.0887	0.0689	0.0886	0.0686	0.0884
HAEM_50	0.0692	0.0887	0.0702	0.0897	0.0734	0.0932
HAEM_L_150	0.0701	0.0897	0.0724	0.0913	0.0757	0.0941
HAEM_S_150	0.0721	0.0907	0.0718	0.0906	0.0716	0.0903
HAEM_150	0.0721	0.0907	0.0754	0.0950	0.0769	0.0971

Table 6. Accuracy of Negative Feedback-based Refinement on IJCAI-15

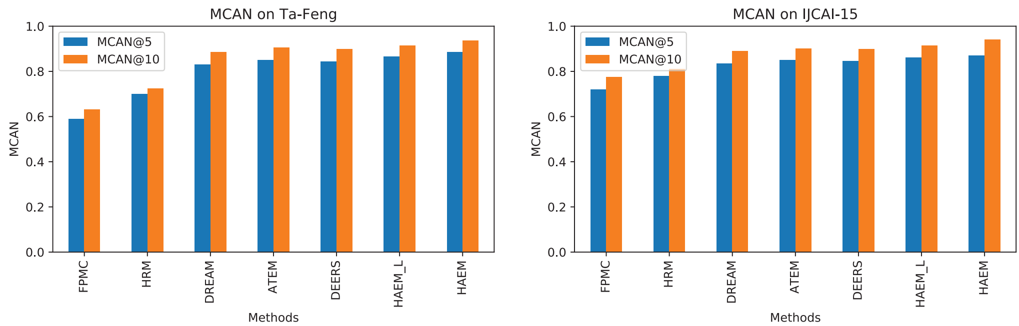
Methods	$p = 1$		$p = 2$		$p = 3$	
	F1-score@5	NDCG@5	F1-score@5	NDCG@5	F1-score@5	NDCG@5
HAEM_L_50	0.0736	0.1679	0.0769	0.1780	0.0803	0.1854
HAEM_S_50	0.0793	0.1791	0.0787	0.1789	0.0771	0.1786
HAEM_50	0.0793	0.1791	0.0810	0.1859	0.0825	0.1901
HAEM_L_150	0.0794	0.1803	0.0837	0.1899	0.0867	0.1971
HAEM_S_150	0.0828	0.1913	0.0827	0.1872	0.0820	0.1850
HAEM_150	0.0828	0.1913	0.0867	0.1978	0.0886	0.2033

recommendation, which is more consistent with the real-world cases compared with those baseline models that only utilize the previous basket or focus more on the recent ones.

In conclusion, the experiment results show that both HAEM_L and HAEM consistently outperform all the baseline methods on both datasets, which illustrates their effectiveness in capturing the compound intra-/inter-basket couplings for NBRS.

5.5.3 Effect of Negative Feedback-based Refinement for Sequential Basket Recommendation. We further apply HAEM_L, HAEM_S, and HAEM to recommending sequential baskets by involving user feedback. Tables 5 and 6 show the results of *F1-score*@5 and *NDCG*@5 for three continuous target baskets on both datasets. We set the dimensionality $d \in \{50, 150\}$ on both Ta-Feng and IJCAI-15, the suffix of each method refers to the adopted dimensionality of the embedding representation on the corresponding datasets. For example, *HAEM_L_50* refers to the HAEM_L method, which sets its dimensionality as 50. Given the user feedback and unselection feedback on previously recommended baskets, both HAEM_L and HAEM with low and high representation dimensionalities achieve higher performance with an increase in the continuous target baskets, showing the significant impact made by negative feedback-based refinement on continuous next-basket recommendations. In contrast to HAEM_L and HAEM, the accuracy of HAEM_S decreases with an increase in the target baskets, showing the accumulation of prediction bias resulting in worse recommendation for iSBRS. Accordingly, the negative feedback-based refinement is significant for more accurate prediction of iSBRS.

Further, HAEM always outperforms HAEM_L by a maximum of 4.85% and an average of 3.39% w.r.t. *F-Score*@5 as well as a maximum of 2.19% and an average of 1.96% w.r.t. *NDCG*@5 ($d = 50$) on Ta-Feng, and a maximum of 7.74% and an average of 5.27% w.r.t. *F-Score*@5 as well as a maximum of 6.67% and an average of 4.55% w.r.t. *NDCG*@5 ($d = 50$) on IJCAI-15 over HAEM_L for the sequential basket recommendation. This benefits from the significant pairwise intra-basket



(a) MCAN@5 on Ta-Feng (b) MCAN@5 on IJCAI-15

Fig. 6. Novelty comparison of methods on two datasets.

relations captured in HAEM for next-basket prediction. In addition, with a negative feedback-based refinement mechanism, HAEM outperforms HAEM_S by a maximum of 7.40% and an average of 4.14% w.r.t. *F1-score*@5 and a maximum of 7.53% and an average of 4.13% w.r.t. *NDCG*@5 on Ta-Feng with $d = 150$, and a maximum of 8.05% and an average of 4.30% w.r.t. *F1-score*@5 and a maximum of 9.89% and an average of 5.18% w.r.t. *NDCG*@5 on IJCAI-15 with $d = 150$. These above observations show that the proposed HAEM mechanisms of modeling both compound intra- and inter-basket relations and enabling negative feedback-based refinement are effective for iSBRS.

5.6 Novelty Evaluation

A critical concern in the area of RS is duplicated or similar items often being recommended by existing methods [45, 51]. Recommending novel items is a way to improve recommendation diversity [24] and actionability [4, 6] for a better user experience, satisfaction and business effect. This has been shown to be particularly important for next-item, next-basket, and session-based recommendations [44], where users may be more interested in novel recommendations rather than items they have previously selected. Accordingly, a good recommender is expected to recommend preferred items, which could be popular, novel, and diverse items, to satisfy both stationary and evolving demand and preferences.

Accordingly, we evaluate the novelty of the recommended next baskets in terms of *MCAN@K* using different methods on both datasets. Figure 6 illustrates the results of the top-5 and top-10 recommendations over Ta-Feng and IJCAI-15 for NBRS. Overall, HAEM achieves the highest novelty compared with the other baselines. FPMC has the lowest novelty, because the sparse characteristics of both datasets make it difficult for FPMC to learn the parameters well thus FPMC only recommends relatively similar items. HRM generates better novelty than FPMC, but it makes recommendations only by modelling the previous basket in the sequential behaviors. This is often not the case in the real-world scenario and may result in some information loss by neglecting the other baskets in the sequential behaviors, which means HRM tends to output items that are similar to the previous basket and therefore incurs low novelty. In addition, benefiting from its RNN-based model, DREAM makes use of all the baskets in the sequential behaviors and captures the global influence of these baskets to make more novel recommendation. Compared with DREAM, DEERS explores the user sequential preference from the perspectives of both positive and negative feedback, which helps DEERS gain more comprehensive knowledge and hence achieves slightly higher novelty for its recommendation.

However, the RNN model on which DREAM and DEERS are based assumes that the temporal dependency changes monotonously along with position in the sequence and recent baskets have a more dominant effect on prediction than previous ones. This assumption does not conform to complex real situations for sequential prediction, where there is no guarantee that one basket has a more or less significant effect than the previous one [29]. Accordingly, DREAM and DEERS cannot distinguish the dominated basket and thus its prediction relies more on the recent baskets, which makes its recommendation similar to its previous basket. Moreover, the max pooling of the basket representation in DREAM only captures the major factors among the items but does not intensify the significant items and item combinations nor does it model their compound relations for prediction. ATEM can generate slightly more novel recommendations by benefitting from its attentive structure, but its prediction is only based on its contextual basket and previous purchase history is neglected.

Compared with the aforementioned baselines, HAEM_L and HAEM not only consider inter-basket couplings to intensify the difference between the last baskets and the next ones along the whole basket sequence but also make full use of the compound intra-basket couplings within each basket to enhance item diversity, which makes the recommendation more diverse and novel. Compared with HAEM_L, HAEM achieves slightly higher novelty, indicating that the modelling of pairwise intra-basket couplings in the basket encoder contributes to more novel recommendations.

6 CONCLUSIONS AND FUTURE WORK

In this work, we discussed the problem of iSBRS and introduced a corresponding method HAEM. HAEM is a hierarchical attentive encoder-decoder model based on deep networks to (1) model both intra-basket item couplings and inter-basket item couplings and (2) continuously incorporate user-recommendation interactions with user selections and unselections of recommended items in the next-baskets into refining further next-basket recommendations. This work represents a step forward in the recent research on sequential recommendation, next-basket recommendation, and session-based recommendation by modeling complex item interactions within/between baskets and interactive positive/negative feedback on recommended items for interactive recommendation. Such interactive sequential recommendation is applicable to many applications including marketing and online e-commerce.

There are many open and further research opportunities on iSBRS. One is to further model the hierarchical and heterogeneous couplings and interactions within/between attributed users, within/between attributed items in baskets, and within/between observable and latent features of users and items, a critical perspective for non-IID recommender systems [6]. Another is to model the unselection feedback on recommended items in sequential modeling, where the research on negative sequence analysis and nonoccurring behavior analytics could inspire the deep modeling of nonoccurring yet important items in sequential recommendation. In addition, during the interactive process, one may change preferences and demand and new items continue to emerge, hence modeling such nonstationary evolving user/item dynamics including their contexts in iSBRS is highly challenging yet pragmatic in real-life businesses. Last, significant opportunities could exist in extending the existing session-based recommenders, in particular, those for next-item and next-basket recommendation.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* 17, 6 (2005), 734–749.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. Retrieved from https://arxiv.org/abs/1409.0473.

[3] Longbing Cao. 2010. In-depth behavior understanding and use: The behavior informatics approach. *Inf. Sci.* 180, 17 (2010), 3067–3085.

[4] Longbing Cao. 2013. Combined mining: Analyzing object and pattern relations for discovering and constructing complex yet actionable patterns. *WIRES Data Min. Knowl. Discov.* 3, 2 (2013), 140–155.

[5] Longbing Cao. 2015. Coupling learning of complex interactions. *J. Inf. Process. Manage.* 51, 2 (2015), 167–186.

[6] Longbing Cao. 2016. Non-IID recommender systems: A review and framework of recommendation paradigm shifting. *Engineering* 2, 2 (2016), 212–224.

[7] Longbing Cao, Xiangjun Dong, and Zhigang Zheng. 2016. eNSP: Efficient negative sequential pattern mining. *Artif. Intell.* 235 (2016), 156–182.

[8] Longbing Cao, Yuming Ou, and Philip S Yu. 2012. Coupled behavior analysis with applications. *IEEE Trans. Knowl. Data Eng.* 24, 8 (2012), 1378–1392.

[9] Longbing Cao, Philip S. Yu, and Vipin Kumar. 2015. Nonoccurring behavior analytics: A new area. *IEEE Intell. Syst.* 30, 6 (2015), 4–11.

[10] Chetna Chand, Amit Thakkar, and Amit Ganatra. 2012. Sequential pattern mining: Survey and current research challenges. *Int. J. Soft Comput. Eng.* 2, 1 (2012), 185–193.

[11] Shuo Chen, Josh L. Moore, Douglas Turnbull, and Thorsten Joachims. 2012. Playlist prediction via metric embedding. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'12)*. 714–722.

[12] Konstantinos Christidis, Dimitris Apostolou, and Gregoris Mentzas. 2010. Exploring customer preferences with probabilistic topic models. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD'10)*.

[13] Qiang Cui, Shu Wu, Yan Huang, and Liang Wang. 2017. A hierarchical contextual attention-based GRU network for sequential recommendation. *arXiv:1711.05114*. Retrieved from https://arxiv.org/abs/1711.05114.

[14] Trong Dinh Thac Do and Longbing Cao. 2018. Metadata-dependent infinite poisson factorization for efficiently modelling sparse and large matrices in recommendation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'18)*. 5010–5016.

[15] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized ranking metric embedding for next new POI recommendation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'15)*. 2069–2075.

[16] Anna Gatzioura and Miquel Sànchez-Marrè. 2015. A case-based recommendation approach for market basket data. *IEEE Intell. Syst.* 30, 1 (2015), 20–27.

[17] Wanrong Gu, Shoubin Dong, and Zhizhao Zeng. 2014. Increasing recommended effectiveness with Markov chains and purchase intervals. *Neural Comput. Appl.* 25, 5 (2014), 1153–1162.

[18] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A factorization-machine based neural network for CTR prediction. *arXiv:1703.04247*. Retrieved from https://arxiv.org/abs/1703.04247.

[19] Negar Hariri, Bamshad Mobasher, and Robin Burke. 2015. Adapting to user preference changes in interactive recommendation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'15)*. 4268–4274.

[20] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'17)*. 355–364.

[21] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv:1511.06939*. Retrieved from https://arxiv.org/abs/1511.06939.

[22] Liang Hu, Longbing Cao, Jian Cao, Zhiping Gu, Guandong Xu, and Jie Wang. 2017. Improving the quality of recommendations for users and items in the tail of distribution. *ACM Trans. Inf. Syst.* 35, 3 (2017), 25.

[23] Liang Hu, Longbing Cao, Jian Cao, Zhiping Gu, Guandong Xu, and Dingyu Yang. 2016. Learning informative priors from heterogeneous domains to improve recommendation in cold-start user domains. *ACM Trans. Inf. Syst.* 35, 2 (2016), 13.

[24] Liang Hu, Longbing Cao, Shoujin Wang, Guandong Xu, Jian Cao, and Zhiping Gu. 2017. Diversifying personalized recommendation with user-session context. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'17)*. 1858–1864.

[25] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *Proceedings of the International Conference on Machine Learning*. 1378–1387.

[26] Jong-Seok Lee, Chi-Hyuck Jun, Jaewook Lee, and Sooyoung Kim. 2005. Classification-based collaborative filtering using market basket data. *Expert Syst. Appl.* 29, 3 (2005), 700–704.

[27] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM'17)*. 1419–1428.

[28] Qiang Liu, Shu Wu, Diyi Wang, Zhaokang Li, and Liang Wang. 2016. Context-aware sequential recommendation. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'16)*. 1053–1058.
[29] Qiang Liu, Shu Wu, and Liang Wang. 2017. Multi-behavioral sequential prediction with recurrent log-bilinear model. *IEEE Trans. Knowl. Data Eng.* 29, 6 (2017), 1254–1267.
[30] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'16)*. 194–200.
[31] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of session-based recommendation algorithms. *User Model. User-Adapt. Interact.* 28, 4–5 (2018), 331–390.
[32] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems. *Comput. Surv.* 51, 4 (2018), 66.
[33] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *Proceedings of the ACM Conference on Recommender Systems (RecSys'17*), Paolo Cremonesi, Francesco Ricci, Shlomo Berkovsky, and Alexander Tuzhilin (Eds.). 130–137.
[34] Steffen Rendle. 2010. Factorization machines. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'10*). 995–1000.
[35] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Uncertainty of Artificial Intelligence (UAI'09)*. 452–461.
[36] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the Annual Conference of the World Wide Web (WWW'10)*. 811–820.
[37] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2011. Fast context-aware recommendations with factorization machines. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11)*. 635–644.
[38] Can Wang, Longbing Cao, and Chi-Hung Chi. 2015. Formalization and verification of group behavior interactions. *IEEE Trans. Syst. Man Cybernet. Syst.* 45, 8 (2015), 1109–1124.
[39] Can Wang, Zhong She, and Longbing Cao. 2013. Coupled clustering ensemble: Incorporating coupling relationships both between base clusterings and objects. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE'13)*. 374–385.
[40] Pengfei Wang, Jiafeng Guo, and Yanyan Lan. 2014. Modeling retail transaction data for personalized shopping recommendation. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM'14)*. 1979–1982.
[41] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning hierarchical representation model for next basket recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'15)*. 403–412.
[42] Qing Wang, Chunqiu Zeng, Wubai Zhou, Tao Li, Larisa Shwartz, and Genady Ya Grabarnik. 2017. Online interactive collaborative filtering using multi-armed bandit with dependent arms. *arXiv:1708.03058*. Retrieved from https://arxiv.org/abs/1708.03058.
[43] Shoujin Wang, Longbing Cao, and Yan Wang. 2019. A survey on session-based recommender systems. arxiv:1902.04864. Retrieved from https://arxiv.org/abs/1902.04864.
[44] Shoujin Wang, Liang Hu, and Longbing Cao. 2017. Perceiving the next choice with comprehensive transaction embeddings for online recommendation. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD'17)*. 285–302.
[45] Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. 2018. Attention-based transactional context embedding for next-item recommendation. In *Proceedings of the AAAI Conference on Artifial Intelligence (AAAI'18)*.
[46] Wei Wang and Longbing Cao. 2019. Negative sequence analysis: A review. *ACM Comput. Surv.* 52, 2 (2019), 32:1–32:39.
[47] Xiang Wu, Qi Liu, Enhong Chen, Liang He, Jingsong Lv, Can Cao, and Guoping Hu. 2013. Personalized next-song recommendation in online Karaokes. In *Proceedings of the ACM Conference on Recommender Systems (RecSys'13)*. 137–140.
[48] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv:1708.04617*. Retrieved from https://arxiv.org/abs/1708.04617.
[49] Ghim-Eng Yap, Xiao-Li Li, and Philip S Yu. 2012. Effective next-items recommendation via personalized sequential pattern mining. In *International Conference on Database Systems for Advanced Applications (DASFAA'12)*. 48–64.
[50] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'16)*. 729–732.

[51] Qian Zhao, Martijn C. Willemsen, Gediminas Adomavicius, F. Maxwell Harper, and Joseph A. Konstan. 2018. Interpreting user inaction in recommender systems. In *Proceedings of the ACM Conference on Recommender Systems (RecSys'18)*. 40–48.
[52] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'18)*. 1040–1048.
[53] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. 2013. Interactive collaborative filtering. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM'13)*. 1411–1420.

Received March 2019; revised October 2020; accepted December 2020