# Taxonomy-Aware Multi-Hop Reasoning Networks for Sequential Recommendation

Jin Huang
School of Information, Renmin University of China
jin.huang@ruc.edu.cn

Zhaochun Ren
Data Science Lab, JD.com
renzhaochun@jd.com

Wayne Xin Zhao*
School of Information, Renmin University of China
batmanfly@gmail.com

Gaole He
School of Information, Renmin University of China
hegaole@ruc.edu.cn

Ji-Rong Wen
Beijing Key Laboratory of Big Data Management and Analysis Methods
jrwen@ruc.edu.cn

Daxiang Dong
Baidu Inc, Beijing, China
dongdaxiang@baidu.com

## ABSTRACT

In this paper, we focus on the task of *sequential recommendation* using taxonomy data. Existing sequential recommendation methods usually adopt a single vectorized representation for learning the overall sequential characteristics, and have a limited modeling capacity in capturing multi-grained sequential characteristics over context information. Besides, existing methods often directly take the feature vectors derived from context information as auxiliary input, which is difficult to fully exploit the structural patterns in context information for learning preference representations.

To address above issues, we propose a novel *Taxonomy-aware Multi-hop Reasoning Network*, named TMRN, which integrates a basic GRU-based sequential recommender with an elaborately designed memory-based multi-hop reasoning architecture. For enhancing the reasoning capacity, we incorporate taxonomy data as structural knowledge to instruct the learning of our model. We associate the learning of user preference in sequential recommendation with the category hierarchy in the taxonomy. Given a user, for each recommendation, we learn a unique preference representation corresponding to each level in the taxonomy based on her/his overall sequential preference. In this way, the overall, coarse-grained preference representation can be gradually refined in different levels from general to specific, and we are able to capture the evolvement and refinement of user preference over the taxonomy, which makes our model highly explainable. Extensive experiments show that our proposed model is superior to state-of-the-art baselines in terms of both effectiveness and interpretability.

## KEYWORDS

Sequential recommendation, multi-hop reasoning, taxonomy

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
*WSDM '19, February 11–15, 2019, Melbourne, VIC, Australia*
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-5940-5/19/02...$15.00
https://doi.org/10.1145/3289600.3290972

**ACM Reference Format:**
Jin Huang, Zhaochun Ren, Wayne Xin Zhao, Gaole He, Ji-Rong Wen, and Daxiang Dong. 2019. Taxonomy-aware Multi-hop Reasoning Networks for Sequential Recommendation. In *The Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19), February 11–15, 2019, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3289600.3290972

## 1 INTRODUCTION

Nowadays, recommender systems have become increasingly important in facilitating the selection of online resources, *i.e.,* items. Instead of seeking static recommendation results at a time, it is usually more common to predict the successive item(s) that a user is likely to interact with given her/his previous interaction records over time [33], called *sequential recommendation.* It has been widely recognized that human behavior is very complex and the accurate prediction for sequential actions is a challenging task [20]. For tackling this task, various approaches have been proposed in the literature [10, 13, 33, 40], including the recent progress with recurrent neural networks (RNN) [13, 21, 45].

With the rapid growth of web content, many studies further leverage rich context information for improving the performance of sequential recommendation [14, 18, 24]. Although these context-aware methods have achieved performance improvement to some extent, there are two major problems for the utilization of context information in sequential recommendation. On one hand, existing studies usually adopt a single vectorized representation for learning the *overall* sequential characteristics reflected in the dependency among adjacent user-item interactions. Hence, the learned preference representation has a limited modeling capacity in capturing multi-grained sequential characteristics corresponding to varying context information. For example, it is difficult to directly infer the sequential characteristics from user behaviors *w.r.t.* a specific category and its sub-categories using previous methods. On the other hand, existing sequential recommendation methods directly take the feature vectors (*e.g.,* a single vector [13] or multiple attribute-based vectors [18]) derived from context data as the auxiliary input. Such a kind of strategy makes it still difficult to exploit the structural patterns in context information (*e.g.,* the data hierarchy) to enhance the preference representations. Overall, there is a lack of the direct association between the data characteristics of context information and the modeling of user preference in existing sequential recommendation methods.

For solving the above problems, we are inspired by a phenomenon that many online platforms have already set up item taxonomies for ease of browsing or selecting resources [48]. A taxonomy provides a hierarchical navigation to instruct users' selection process [43, 53], which is likely to influence users' preference [4]. In Figure 1, we present an illustrative example of such a product selection process by a user on *Amazon*. As we can see, an item taxonomy is a tree-structured category hierarchy in essence, where each non-leaf node corresponds to a category label, representing some attributional or grouping semantics of items. As such, a user can utilize the structured knowledge from the provided taxonomy for her/his clicking or purchasing decision in a recommender system. Moreover, we see that the user's preference is gradually refined through such a hierarchical process with multiple hops.

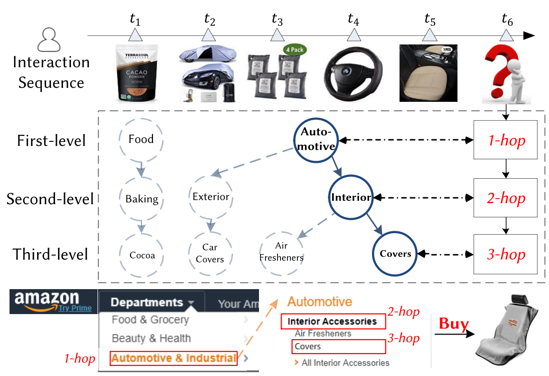



**Figure 1: An illustrative example of the product selection process by a user with the taxonomy from Amazon.**

Motivated by this example, our main idea is to associate the learning of user preference in sequential recommendation with the category hierarchy in the taxonomy. Given a user, for each recommendation, we learn a unique preference representation corresponding to each level in the taxonomy based on her/his overall sequential preference. In this way, the overall, coarse-grained preference representation is gradually refined in different levels from general to specific, which naturally captures the *evolvement* and *refinement* of user preference over the taxonomy in multiple hops.

In this paper, we propose a novel *Taxonomy-aware Multi-hop Reasoning Network* (TMRN) for better understanding and modeling user preference in sequential recommendation. For implementing the multi-hop learning mechanism, we adopt memory-augmented neural networks, which have been shown to be effective in performing complex reasoning tasks with the enhanced inference capacity [23, 27]. We extend the basic GRU-based sequential recommender by integrating it with an elaborately designed memory-based multi-hop reasoning architecture. Specially, we set up a different memory matrix for each hop, and incorporate the corresponding category knowledge from the associated level in the taxonomy. For utilizing the taxonomy data, we propose to learn the representations for the category nodes by using a network embedding approach. At each recommendation, the sequential preference representation from GRUs is first derived as the query to read out the associated content of the corresponding memory matrices. Furthermore, we successively perform the reasoning based on the memory content that has been read out, and obtain the multi-hop preference representations.

As a main technical contribution, we utilize the hierarchical structure of taxonomy data to instruct the learning of user preference. We associate the modeling of multiple fine-grained sequential preference representations with corresponding category levels in the taxonomy. Our reasoning network can characterize both overall and fine-grained sequential characteristics for modeling user preference, which largely improves the learning capacity for complex user behaviors. Interestingly, with such a hierarchical architecture, our model is able to capture the *evolvement* and *refinement* of user preference over the taxonomy, which is useful for understanding the detailed decision-making process for an interaction [17]. Extensive results on three real-world datasets have shown the superiority of the proposed model in sequential recommendation.

## 2 RELATED WORK

In this section, we review the studies related to our work.

Early recommender systems aim to generate a static recommendation list based on either implicit feedback [19] or explicit feedback [32]. Later on, various kinds of context information have been used to enhance the recommendation performance [22, 31, 38, 42, 47, 50, 51] and model interpretability [12, 30, 37, 49].

Different from the original recommendation task, *sequential recommendation* aims to predict the successive item(s) that a user is likely to interact with given her historical data [33]. The pioneering study [33] proposes the FPMC model by combining Markov chain and matrix factorization for next-basket recommendation. The core idea of FPMC is to capture sequential patterns between consecutive user-item interactions. Many existing studies follow the similar idea and adopt different technical solutions [10, 40]. Recently, Recurrent Neural Networks (RNN) together with its variants LSTM and GRU have been widely applied to the sequential recommendations [5, 13, 21, 29, 45, 46]. These RNN based methods typically learn user preference by encoding the historical adoption sequence into a latent vector. For improving the recommendation performance, several studies further propose to incorporate context information into RNN based methods [6, 14, 24]

Taxonomy data is a specific kind of context information utilized in our approach, which is usually a tree-structured hierarchy. Compared with knowledge bases [16, 47, 52], it is more easy to obtain and widely adopted in real-world systems. Previous methods usually use taxonomy data for resolving the sparsity problem in recommendation models [25, 28]. Especially, Kanagal et al. [18] propose to combine the human-induced taxonomies and latent factor models to improve the sequential recommendation. Zhang et al. [48] propose to discover the taxonomy from shopping data automatically and learn a taxonomy-based recommendation system jointly. Wang et al. [41] propose to construct the explainable recommender with tree-enhanced embedding models. Different from them, our focus is to utilize the structural organization of taxonomy data for constructing hierarchical multi-hop reasoning network.

Recently, memory-based neural networks have been widely used in providing the long-term data storage and enhancing the complex reasoning capacity [1, 7, 8, 44]. By endowing memory networks (MN) [44] with more powerful reasoning capacity, many effective variants have been developed for different applications [2, 23, 35]. Especially, our method is related to RUM [2] in which memory networks have been first applied for sequential recommendation. However, RUM does not support the multi-hop reasoning, and it can not directly utilize hierarchical knowledge information.

To our knowledge, our proposed TMRN model is the first sequential recommender which associates the learning of multi-grained user preference with the category hierarchy in the taxonomy. Our model can characterize both overall and fine-grained sequential characteristics for modeling user preference. Besides, it is able to model the evolvement and refinement of user preference over the taxonomy, which largely enhances the reasoning capacity and the model interpretability.

# 3 PRELIMINARIES

We detail the task that we address and describe the GRU-based sequential recommender that our model is built on.

## 3.1 Problem Formulation

A recommender system mainly focuses on the interactions between a user set $\mathcal{U}$ and an item set $\mathcal{I}$. The interaction sequence of a user $u \in \mathcal{U}$ consists of ordered interaction records generated by $u$, *i.e.,* $\{i_1, \cdots, i_t, \cdots, i_{n_u}\}$, where $i_t$ is the item that $u$ interacted with at time $t$ and $n_u$ is the number of interaction records. Following [33], the relative time index is used to number interaction records.

Besides, we assume that an item taxonomy $C$ is also given, which is organized in a tree-structured hierarchy. In $C$, a non-leaf node corresponds to a category label and a leaf node corresponds to an item. All the non-leaf nodes form a category hierarchy, where a higher node corresponds to a more abstractive category. Each item is associated with a category path from the root to the item itself. Furthermore, we assume all the category paths have the same length, denoted by $K$. Given item $i$, the associated categories through the path from the root to $i$ can be given as $c_i^1 \rightarrow \cdots \rightarrow c_i^K$, where each node $c_i^j$ is the only parent node of $c_i^{j+1}$ in the hierarchy. Given the example in Figure 1, the product of "*steering wheel cover*" corresponds to the category path "*Automotive* $\rightarrow$ *Interior Accessories* $\rightarrow$ *Covers*". It can be seen that such a taxonomy provides valuable knowledge information about the items, and forms a hierarchical organization of the entire item set in different levels.

Based on the above notations, we define the task of *sequential recommendation* as follows. Given the item taxonomy $C$ and the interaction sequence $\{i_1, \cdots, i_t, \cdots, i_{n_u}\}$ of user $u$, we aim to infer the item that user $u$ will interact with at time $n_u + 1$.

## 3.2 GRU-based Sequential Recommender

Recently, recurrent neural networks (RNNs) have been widely applied to sequential recommendation for excellent sequence modeling capacity, especially the two improved variants of the long-short term memory (LSTM) networks [15] and the gated recurrent unit (GRU) networks [3]. Considering the simplicity and robustness, we employ a GRU network as the basis of our model in this paper.

For each user $u$, we use $\{i_1, \cdots, i_t\}$ to denote an interaction sequence between $u$ and the items that $u$ has interacted with so far. At time $t$, given $\boldsymbol{h}_{t-1}^u$ calculated at $t-1$, the GRU-based recommender calculates a hidden state vector $\boldsymbol{h}_t^u \in \mathbb{R}^{L_H}$ as follows:

$$\boldsymbol{h}_t^u = \text{GRU}(\boldsymbol{h}_{t-1}^u, \boldsymbol{q}_{i_t}; \Phi), \tag{1}$$

where GRU($\cdot$) is the GRU unit [3], $\boldsymbol{q}_{i_t} \in \mathbb{R}^{L_I}$ is the embedding vector for item $i_t$ (*item embedding* for short), and $\Phi$ denotes all the related parameters of GRU networks. The GRU network can encode the past interaction sequence of $u$ into a hidden vector $\boldsymbol{h}_t^u$, modeling the sequential characteristics of $u$'s preference at time $t$. Hence, we call $\boldsymbol{h}_t^u$ *the overall sequential preference representation* of user $u$.

Given user $u$, at time $t$, for a candidate item $i$, the GRU-based recommender generates a ranking score by calculating an inner product between the representations of $u$ and $i$, formally we have $s_{u,i,t} = \boldsymbol{h}_t^{u\top} \cdot \boldsymbol{q}_i$. According to these ranking scores, the recommender ranks candidate items descendingly, and recommends items with high scores.

# 4 METHOD

In this section, we present the proposed *Taxonomy-aware Multi-hop Reasoning Network* (TMRN) for sequential recommendation.

## 4.1 Overview

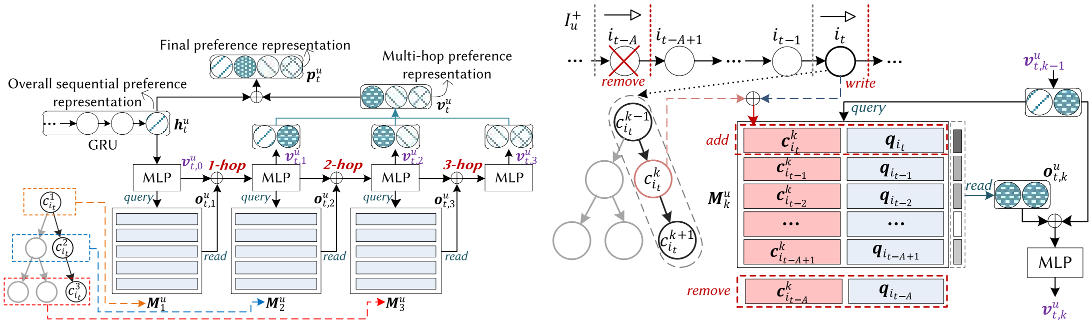
In Fig 2(a), we present an overview of the proposed model. The main contribution lies in the taxonomy-aware multi-hop reasoning mechanism. In our model, we integrate the basic GRU-based sequential recommender (See §3.2) with a novel hierarchical multi-hop memory network. Besides the overall sequential preference representation, TMRN hierarchically learns multiple fine-grained preference representations corresponding to different levels in the item taxonomy, which can capture the evolvement and refinement of user preference during the item selection process.

We detail TMRN in an asymptotical way. In §4.2, we introduce the multi-hop reasoning architecture, whereas in §4.3 we detail the use of taxonomy information for enhancing the reasoning mechanism. §4.4 provides the application of the proposed model in sequential recommendation.

## 4.2 The General Multi-hop Reasoning Architecture

For constructing an effective reasoning network, we adopt memory-augmented neural networks, since they have been shown to be effective in performing complex reasoning tasks with the enhanced learning capacity [23, 27]. Next, we present a novel multi-hop reasoning architecture for sequential recommendation.

*4.2.1 The memory component.* The proposed reasoning network sets up an array of slots as long-term memory for storing information. We assume $K$ hops exist in our model. For a user, instead of setting up a single memory matrix, we incorporate $K$ user-specific memory matrices, which enable the learning of preference representations for multiple hops. Specially, for each user $u$, let $\boldsymbol{M}_k^u = [\boldsymbol{m}_{k,1}^u; \boldsymbol{m}_{m,2}^u; \cdots; \boldsymbol{m}_{k,A}^u]$ denote the corresponding memory matrix for the $k$-th hop, where the subscript of $k$ indicates the hop




(a) The overview of the network architecture.

(b) The memory layout and Write operation for the $k$-th hop.

**Figure 2: The overview of the proposed model TMRN. The left figure presents an illustration of TMRN with three hops, and a unique memory matrix is set up for each hop given a user. The right figure presents the memory layout and the Write operation for the $k$-th hop. We can see that category embeddings have been incorporated corresponding to different hops.**

index, $\boldsymbol{m}^u_{k,a} \in \mathbb{R}^{L_M}$ is the $a$-th vector in the memory matrix, and $\boldsymbol{M}^u_k$ has the size of $L_M \times A$. We equip the memory component in our model with two kinds of operations to obtain and update information, namely the Read and write operations.

*4.2.2 Reasoning.* Given a user $u$, let $\boldsymbol{v}^u_{t,k}$ denote the preference representation of $u$ for the $k$-th hop at time $t$. Initially, we set the preference as the transformed sequential representation from the GRU-based recommender, *i.e.,* $\boldsymbol{v}^u_{t,0} = MLP(\boldsymbol{h}^u_t)$. For the $k$-th hop, we derive the preference representation $\boldsymbol{v}^u_{t,k}$ using the information from the previous hop, including the preference representation $\boldsymbol{v}^u_{t,k-1}$ and the output from the memory component $\boldsymbol{o}^u_{t,k}$

$$\boldsymbol{v}^u_{t,k} = MLP(\boldsymbol{v}^u_{t,k-1} \oplus \boldsymbol{o}^u_{t,k}), \tag{2}$$

where $MLP(\cdot)$ is a multi-layer perceptron consisting of several hidden layers and "$\oplus$" denotes the vector combination operation, *i.e.,* vector concatenation here. To generate the output $\boldsymbol{o}^u_{t,k}$, we set the input of the Read operation as $u$'s preference representation $\boldsymbol{v}^u_{t,k-1}$. Using $\boldsymbol{v}^u_{t,k-1}$ as a query to read memory matrix $\boldsymbol{M}^u_k$, our reasoning network generates the output $\boldsymbol{o}^u_{t,k}$ as follows

$$\boldsymbol{o}^u_{t,k} \leftarrow \textsc{Read}(\boldsymbol{M}^u_k, \boldsymbol{v}^u_{t,k-1}). \tag{3}$$

Following [2, 35, 44], the Read operation can be specified by using an attentive combination of use-specific memory vectors

$$\boldsymbol{o}^u_{t,k} = \sum_{a=1}^{A} w_{t,u,k,a} \cdot \boldsymbol{m}^u_{k,a}, \tag{4}$$

$$w_{t,u,k,a} = \frac{\exp(\sigma(\boldsymbol{v}^u_{t,k-1}{}^\top \cdot \boldsymbol{m}^u_{k,a}))}{\sum_{a'=1}^{A} \exp(\sigma(\boldsymbol{v}^u_{t,k-1}{}^\top \cdot \boldsymbol{m}^u_{k,a'}))}, \tag{5}$$

where we set $w_{t,u,k,a}$ as the attention weight of the $a$-th vector in $\boldsymbol{M}^u_k$, so we have $\boldsymbol{o}^u_{t,k} \in \mathbb{R}^{L_M}$. Note that with the MLP transformation in Eq. 2, $\boldsymbol{v}^u_{t,k}$ is of the same size with $\boldsymbol{m}^u_{k,a}$, so that we can perform successive reasoning from the memory component to obtain $K$ preference representations $\{\boldsymbol{v}^u_{t,k}\}_{k=1}^K$. In Fig. 2(a), we present an illustration example for the above reasoning procedure.

*4.2.3 Updating.* At time $t$, when user $u$ interacts with a new item $i_t$, the memory component performs the Write operation using the information of item $i_t$ to update the content in $K$ memory matrices, *i.e.,* $\{\boldsymbol{M}^u_k\}_{k=1}^K$, which is given in an abstractive form:

$$\{\boldsymbol{M}^u_k\}^{new} \leftarrow \textsc{Write}(\{\boldsymbol{M}^u_k\}^{old}, \tilde{\boldsymbol{q}}_{i_t}), \tag{6}$$

where $\tilde{\boldsymbol{q}}_{i_t}$ denotes all the necessary information related to item $i_t$. We leave the detailed implementation of Write operation in §4.3.3.

## 4.3 Utilizing Taxonomy Data for Instructing the Reasoning Mechanism

In the above, we have introduced a general multi-hop reasoning architecture for hierarchically learning different preference representations in multiple hops. However, it is not clear what kind of characteristics in user preference can be captured at each hop. In this part, we study how to leverage taxonomy data for improving the reasoning mechanism.

*4.3.1 Motivation.* A potential problem in existing memory-based multi-hop reasoning models [35] is that the learning process is difficult to explain and maintain. Our idea is to utilize the structural taxonomy to enhance the multi-hop reasoning mechanism. As shown in Fig 1, the taxonomy organizes the category nodes in three levels. The category nodes in each level correspond to a specific granularity in terms of categorical semantics, which provides a hierarchical navigation for the item set. Such a hierarchy is useful to assist the selection process of a user from general to specific. Hence, we aim to align the reasoning procedure with the category hierarchy. We first infer users' preference regarding general categories, and then gradually refine the learning of user preference by focusing on more specific categories. In this way, each reasoning hop has a more clear explanation, and the learning capacity will be enhanced with external knowledge information.

*4.3.2 Pretraining the embeddings for category nodes.* For utilizing the taxonomy information, we first learn the representations of the category nodes. Our key idea is to preserve the association or relatedness between different category nodes in terms of categorical semantics, and utilize low-dimensional vectors to encode such information. Recall that an item $i$ is associated with a path of $K$ category labels $c_i^1 \to c_i^2 \to \cdots \to c_i^K$. In the path, each child node should be similar to its parent node, and such node similarity is transitive. Hence, we generate an edge for each node pair in the path. We repeat the above process for the paths from all the items. The weight of an edge is set to the occurrence frequency of the corresponding node pair. Then, we apply the widely used network embedding model *LINE* [36] to learn a distributed vector representation for each category node. We use the bold font $\boldsymbol{c}_i^k \in \mathbb{R}^{L_C}$ to denote the learned embedding for $c_i^k$. Note that we do not incorporate item information for the learning of category embeddings, since the major aim is to encode the hierarchical category semantics. As will be shown later, we will combine the item embedding with its associated category embeddings in use. There can be other ways for embedding hierarchical data. Here we adopt the method in [36] for its simplicity and effectiveness in our work (See §6.2.3 for experimental discussions).

*4.3.3 Aligning the reasoning with the category hierarchy.* Our key idea is to align each hop in the reasoning network with a corresponding level in the taxonomy. For achieving this, we specify the setting of memory content and instantiate the WRITE operation in the general reasoning architecture (See §4.2).

A major benefit of memory networks is that they can exactly store information in a longer period. Hence, in [2], the authors propose to fill the memory matrix with $A$ latest items that $u$ has interacted with for sequential recommendation, which is expected to yield a better performance than RNN based models. In our case, we have additional taxonomy data for utilization. At time $t$, the $a$-th memory vector in $\boldsymbol{M}_k^u$ for the $k$-th hop is set as follows

$$\boldsymbol{m}_{k,a}^u \leftarrow \boldsymbol{q}_{i_{t-a+1}} \oplus \boldsymbol{c}_{i_{t-a+1}}^k, \tag{7}$$

where $\boldsymbol{c}_{i_{t-a+1}}^k$ is the embedding of the corresponding category node in the $k$-th level of the taxonomy for item $i_{t-a+1}$, so we have $L_M = L_I + L_C$. With the concatenation, our memory matrix contains the information of the item itself and the associated category on a level.

Another alignment lies in the WRITE operation. When a new item $i_t$ comes, we update the memory content at the $k$-th hop with its associated category embedding

$$\{\boldsymbol{M}_k^u\}^{new} \leftarrow \text{WRITE}(\{\boldsymbol{M}_k^u\}^{old}, \boldsymbol{q}_{i_t} \oplus \boldsymbol{c}_{i_t}^k), \tag{8}$$

where the WRITE operation is implemented by a first-in-first-out mechanism according to [2, 18, 35]

$$[\tilde{\boldsymbol{q}}_{i_{t-1}}; \cdots; \tilde{\boldsymbol{q}}_{i_{t-A+1}}; \underbrace{\tilde{\boldsymbol{q}}_{i_{t-A}}}_{\text{removed}}] \to [\underbrace{\tilde{\boldsymbol{q}}_{i_t}}_{\text{added}}; \tilde{\boldsymbol{q}}_{i_{t-1}}; \cdots; \tilde{\boldsymbol{q}}_{i_{t-A+1}}], \tag{9}$$

where $\tilde{\boldsymbol{q}}_i = \boldsymbol{q}_i \oplus \boldsymbol{c}_i^k$ and the embedding of the earliest item $i_{t-A}$ is removed and the embedding of the new item $i_t$ is added in $\boldsymbol{M}_k^u$. For different memory matrices in $\{\boldsymbol{M}_k^u\}_{k=1}^K$, we update them with the corresponding category embedding in $\{\boldsymbol{c}_{i_t}^k\}_{k=1}^K$. Note that when the memory is not full, the item is directly added without replacement. See Fig. 2(b) for the illustration of the memory layout and update.

*4.3.4 Summary.* We implement the alignment by associating the reasoning procedure with the corresponding category embedding in different hops. In this way, the $k$-hop reasoning is aligned with the $k$-level hierarchy from the taxonomy. With $\{\boldsymbol{v}_{t,k}^u\}_{k=1}^K$, we can capture the sequential characteristics of user preference in different granularities. Note that such a learning architecture follows a hierarchical way, so we can trace the evolvement and refinement of user preference over the hierarchy. Eventually, we calculate $u$'s multi-hop preference representation by summing up these $K$ preference representations,

$$\boldsymbol{v}_t^u = \boldsymbol{v}_{t,1}^u \oplus \boldsymbol{v}_{t,2}^u \oplus \cdots \oplus \boldsymbol{v}_{t,K}^u. \tag{10}$$

In contrast with $\boldsymbol{h}_t^u$ that emphasizes the sequential preference, $\boldsymbol{v}_t^u$ emphasizes the multi-hop preference at the item selection process.

## 4.4 Sequential Recommendation

The working procedure of our reasoning network can be given as follows. At time $t$, we first use the GRU-based recommender (Eq. 1) to learn the overall sequential representation $\boldsymbol{h}_t^u$. Then, we employ $\boldsymbol{h}_t^u$ as the initial query to perform the multi-hop reasoning over the memory matrices. At the $k$-th hop, we obtain a new preference representation $\boldsymbol{v}_{t,k}^u$ corresponding to the $k$-th level in the taxonomy using Eq. 2. After $K$ hops, we obtain a set of $K$ preference representations $\{\boldsymbol{v}_{t,k}^u\}_{k=1}^K$, and derive the combined multi-hop preference representation $\boldsymbol{v}_t^u$ using Eq. 10. Since $\boldsymbol{h}_t^u$ and $\boldsymbol{v}_t^u$ capture the characteristics of user preference in different aspects, we further concatenate $\boldsymbol{h}_t^u$ and $\boldsymbol{v}_t^u$ into a vector $\boldsymbol{p}_t^u = \boldsymbol{h}_t^u \oplus \boldsymbol{v}_t^u$ as the final representation. We utilize an inner product between the transformed representations of $u$ and $i$ to calculate the ranking score:

$$s_{u,i,t} = \text{MLP}(\boldsymbol{p}_t^u)^\top \cdot \text{MLP}(\check{\boldsymbol{q}}_i), \tag{11}$$

where $MLP(\cdot)$ is used to map vectors into the same dimensionality, and $\check{\boldsymbol{q}}_i = \boldsymbol{q}_i \oplus \boldsymbol{c}_1^i \cdots \oplus \boldsymbol{c}_K^i$ is the item representation combined with all associated category embeddings. Overall, the process of feeding the overall sequential user representation to memory component performs like a decoding procedure, where we can gradually distill more fine-grained preference representations with the instruction of the hierarchical knowledge information from the taxonomy.

To learn the model parameters, we use a pairwise loss function:

$$L = \sum_{u \in \mathcal{U}} \sum_{t=1}^{n_u} \sum_{j \in \mathcal{I}_u^-} \log \sigma(s_{u,i_t,t} - s_{u,j,t})), \tag{12}$$

where $n_u$ is the length of interaction sequence of $u$ in the training set, $\mathcal{I}_u^-$ is a small set of sampled items that user $u$ has not interacted with, and $\sigma(\cdot)$ is the sigmoid function. For each item $i$ we pre-train the item embedding $\boldsymbol{q}_i$ using the classic BPR model [32].

In contrast with existing sequential recommendation methods, our model has the following merits: (1) our reasoning network TMRN is able to capture both overall and fined-grained sequential characteristics in user preference; (2) we align the reasoning process with the category hierarchy, which is able to characterize the evolvement and refinement of user preference *w.r.t.* different category levels; (3) our model is endowed with both the excellent reasoning capacity from memory-based neural networks and the interpretability from external structured knowledge.

# 5 EXPERIMENTAL SETUP

In this section, we set up the experiments with the datasets, evaluation metrics and baselines.

## 5.1 Datasets

In our experiments, we collect three real-world datasets from three e-commerce portals respectively, *i.e.,* AMAZON MUSIC [9], JD [39] and LAST.FM [34]. Additionally, we collect or construct triple-level hierarchical category labels from these three portals. Categories in AMAZON dataset contain *music style*, *sub-style*, and *artist*; categories in JD dataset contain *category*, *sub-category*, and *brand*; whereas categories in LAST.FM dataset only contain *artist* and *album*. To extend the concept hierarchy for LAST.FM, we use the clusters of the artists as a new kind of attribute labels (called *music style*). To cluster artists, we first represent each artist by a one-hot vector consisting of user ratings, and then group the artists according to the similarities between their vectors. We treat each cluster as a *music style*, and have 100 music style labels in total. Following [10, 33], we filter out unpopular items and inactive users with fewer than $k$ records. We set $k = 5$ in AMAZON MUSIC dataset and $k = 10$ in other two datasets. We summarize the detailed statistics of the datasets in Table 1. Our datasets and code are shared online via the link https://github.com/RUCDM/TMRN.

**Table 1: Statistics of our datasets. #L1, #L2, and #L3 indicate the number of first-level, second-level and third-level category labels respectively.**

| Datasets | #Interactions | #Items | #Users | #L1 | #L2 | #L3 |
|---|---|---|---|---|---|---|
| LAST.FM | 204,438 | 30,679 | 7,713 | 100 | 3,662 | 10,292 |
| JD | 1,644,953 | 121,976 | 50,000 | 167 | 1,012 | 10,845 |
| AMAZON MUSIC | 74,786 | 16,086 | 4,528 | 91 | 130 | 2,655 |

## 5.2 Task Settings

Following the previous settings [11, 32, 33], we consider two task settings for evaluation, namely next-item recommendation and multi-item recommendation. For next-item recommendation, we hold out the last item of the interaction sequence as the test data; for multi-item recommendation, we hold out the last 20% of the interaction sequence as the test data. Since the item set is very large, it is time-consuming to enumerate all the items as candidates. Hence, following [11], we sample negative cases for each positive record (user-item interaction) in the test set. We pair every positive user record with 100 sampled items that the user has not interacted with, called *negative items*. To sample reliable and representative negative items, 50 items are sampled according to the popularity, while the rest 50 items are sampled randomly.

## 5.3 Comparisons

We propose a novel *Taxonomy-aware Multi-hop Reasoning Network* (TMRN) for sequential recommendation, named **TMRN**. We also implement a variant by removing the external taxonomy information from TMRN, named **MRN**. MRN follows all the details of TMRN except it removes all the category embeddings, where multiple memory matrices degenerate into a shared memory matrix.

Our baselines include related methods on general and sequential recommendation with or without context data:

(1) **BPR** [32] is a latent factor model with implicit feedback, which optimizes a pairwise ranking loss function;

(2) **NCF** [11] is a neural collaborative filtering method utilizing a neural architecture to replace the inner product;

(3) **FM** [31] refers to a generic factorization machine and enhanced with the hierarchical category information;

(4) **FPMC** [33] is a hybrid model to capture both sequential effects and general interests of users for sequential recommendation;

(5) **TF** [18] proposes to combine the taxonomies and latent factor models to improve the sequential recommendation;

(6) **RUM** [2] employs memory networks for implementing a sequential recommender.

(7) **GRU** [13] is a GRU-based sequential recommender with session-parallel mini-batch training.

(8) **GRU**$_F$ [14] proposes to concatenate both item vector and feature vector as the input of GRU networks, which incorporates auxiliary features to improve sequential recommendation.

## 5.4 Evaluation Metrics and Parameter Settings

To assess whether our method can improve the sequential recommendation, we adopt a variety of evaluation metrics following previous work [2, 13, 33]: Precision (P), Recall (R), Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Hit Ratio (HR).

To set the parameters in our experiments, we either follow the reported optimal parameter settings or optimize each model separately using the validation set of 10% training data. For our model, we adopt a one-layer GRU network, the hidden layer size $L_H$ is set to 256, the item embedding size $L_I$ is set to 256, and the hierarchy embedding size $L_C$ is set to 256. For memory matrices, $A$ is set to 20, and $L_M = L_I + L_C$. All the MLP components are of one hidden layer with the *tanh* activation function. We pre-train item embeddings and hierarchical category embeddings. We use SGD to optimize the parameters when we train models. We will discuss the effect of parameter settings in §6.2.

# 6 RESULTS AND ANALYSIS

In this section, we present our experimental results and analysis.

## 6.1 Overall Performance

We first examine the performance of our methods for the two recommendation tasks. Table 2 lists the performance of all methods using different evaluation metrics. We can find that non-sequential recommendation baselines, *i.e.,* BPR and NCF, overall perform worse than knowledge-aware or sequential recommendation methods. Among all sequential recommendation baselines, FPMC performs worst. The method TF can be considered as an enhanced version of FPMC by incorporating taxonomy data for sequential recommendation. As two competitive methods, RUM utilizes the memory network, and GRU adopts the RNN-based architecture, both of which give good performance in all three datasets. Furthermore, GRU$_F$ with additional features achieves an obvious increase over GRU on most of evaluation metrics.

**Table 2: Performance comparison of different methods on next-item and multi-item recommendation. "†" indicates the improvement of the TMRN over the baseline is significant at the level of 0.01. We also report the improvement ratio of our TMRN model over the best performance of all the baselines for each dataset in parentheses.**

| Datasets | Methods | Next-Item Recommendation | | | | | Multi-Item Recommendation | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | P@10 | R@10 | MAP | MRR | HR@10 | P@10 | R@10 | MAP | MRR | HR@10 |
| Last.fm | BPR | 0.051† | 0.514† | 0.260† | 0.260† | 0.514† | 0.081† | 0.215† | 0.155† | 0.224† | 0.419† |
| | NCF | 0.060† | 0.602† | 0.436† | 0.436† | 0.602† | 0.129† | 0.342† | 0.289† | 0.435† | 0.617† |
| | FM | 0.065† | 0.658† | 0.352† | 0.352† | 0.658† | 0.108† | 0.307† | 0.197† | 0.290† | 0.577† |
| | FPMC | 0.056† | 0.558† | 0.382† | 0.382† | 0.558† | 0.114† | 0.307† | 0.252† | 0.365† | 0.530† |
| | TF | 0.065† | 0.648† | 0.388† | 0.388† | 0.648† | 0.119† | 0.329† | 0.235† | 0.308† | 0.548† |
| | RUM | 0.065† | 0.646† | 0.416† | 0.416† | 0.646† | 0.130† | 0.344† | 0.259† | 0.379† | 0.616† |
| | GRU | 0.060† | 0.598† | 0.457† | 0.457† | 0.598† | 0.116† | 0.308† | 0.258† | 0.445† | 0.592† |
| | GRU$_F$ | 0.072† | 0.723† | 0.571† | 0.571† | 0.723† | 0.178† | 0.443† | 0.377† | 0.558† | 0.733† |
| | MRN | 0.068 | 0.682 | 0.509 | 0.509 | 0.682 | 0.155 | 0.392 | 0.326 | 0.490 | 0.671 |
| | TMRN | **0.073** (+1.4%) | **0.730** (+1.0%) | **0.603** (+5.6%) | **0.603** (+5.6%) | **0.730** (+1.0%) | **0.186** (+4.5%) | **0.455** (+2.7%) | **0.405** (+7.4%) | **0.602** (+7.9%) | **0.745** (+1.6%) |
| JD | BPR | 0.038† | 0.379† | 0.231† | 0.231† | 0.379† | 0.053† | 0.134† | 0.106† | 0.207† | 0.316† |
| | NCF | 0.047† | 0.470† | 0.333† | 0.333† | 0.470† | 0.058† | 0.146† | 0.126† | 0.211† | 0.366† |
| | FM | 0.049† | 0.487† | 0.262† | 0.262† | 0.487† | 0.087† | 0.211† | 0.144† | 0.248† | 0.503† |
| | FPMC | 0.056† | 0.561† | 0.327† | 0.327† | 0.561† | 0.074† | 0.160† | 0.126† | 0.289† | 0.466† |
| | TF | 0.059† | 0.592† | 0.347† | 0.347† | 0.592† | 0.114† | 0.279† | 0.189† | 0.326† | 0.612† |
| | RUM | 0.063† | 0.632† | 0.478† | 0.478† | 0.632† | 0.144† | 0.310† | 0.268† | 0.528† | 0.686† |
| | GRU | 0.071† | 0.710† | 0.368† | 0.368† | 0.710† | 0.131† | 0.292† | 0.236† | 0.491† | 0.702† |
| | GRU$_F$ | 0.071† | 0.707† | 0.491† | 0.491† | 0.707† | 0.154† | 0.327† | 0.273† | 0.533† | 0.736† |
| | MRN | 0.071 | 0.708 | 0.508 | 0.508 | 0.708 | 0.166 | 0.337 | 0.291 | 0.579 | 0.750 |
| | TMRN | **0.073** (+2.8%) | **0.732** (+3.5%) | **0.528** (+7.5%) | **0.528** (+7.5%) | **0.732** (+3.5%) | **0.173** (+12.3%) | **0.358** (+9.5%) | **0.306** (+12.1%) | **0.589** (+10.5%) | **0.778** (+5.7%) |
| Amazon Music | BPR | 0.034† | 0.338† | 0.147† | 0.147† | 0.338† | 0.053† | 0.207† | 0.120† | 0.179† | 0.375† |
| | NCF | 0.033† | 0.326† | 0.157† | 0.157† | 0.326† | 0.054† | 0.195† | 0.130† | 0.191† | 0.370† |
| | FM | 0.035† | 0.346† | 0.153† | 0.153† | 0.346† | 0.055† | 0.213† | 0.122† | 0.185† | 0.385† |
| | FPMC | 0.036† | 0.357† | 0.166† | 0.166† | 0.357† | 0.055† | 0.210† | 0.127† | 0.194† | 0.382† |
| | TF | 0.038† | 0.376† | 0.175† | 0.175† | 0.376† | 0.060† | 0.211† | 0.125† | 0.198† | 0.413† |
| | RUM | 0.043† | 0.432† | 0.198† | 0.198† | 0.432† | 0.065† | 0.253† | 0.142† | 0.211† | 0.447† |
| | GRU | 0.037† | 0.373† | 0.171† | 0.171† | 0.373† | 0.054† | 0.206† | 0.116† | 0.179† | 0.376† |
| | GRU$_F$ | 0.043† | 0.432† | 0.204† | 0.204† | 0.432† | 0.068† | 0.257† | 0.151† | 0.230† | 0.459† |
| | MRN | 0.045 | 0.447 | 0.204 | 0.204 | 0.447 | 0.069 | 0.264 | 0.145 | 0.218 | 0.462 |
| | TMRN | **0.045** (+4.7%) | **0.451** (+4.4%) | **0.210** (+2.9%) | **0.210** (+2.9%) | **0.451** (+4.4%) | **0.070** (+2.9%) | **0.265** (+3.1%) | **0.152** (+0.7%) | **0.233** (+1.3%) | **0.470** (+2.4%) |

We compare our proposed model TMRN with all the baselines. Our base architecture MRN is the pre-trained GRU neural network [13] integrated with the multi-hop reasoning mechanism. We find that MRN, the simplification of TMRN, outperforms all the other methods except GRU$_F$. This is because GRU$_F$ utilizes auxiliary features. With hierarchical knowledge, it is clear to see that TMRN outperforms the other methods consistently, including GRU$_F$ and MRN. We conclude that multi-hop reasoning is more capable of learning users' preference for sequential recommendation, and with the incorporation of hierarchical knowledge, TMRN yields a larger performance improvement.

## 6.2 Model Analysis

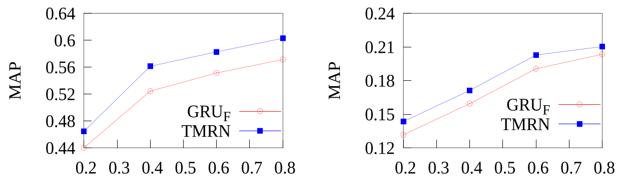
In this section, we perform a series of detailed analysis for our proposed model for further verifying the effectiveness and robustness.

*6.2.1 Varying the depth of hops.* In our reasoning network, the depth of hops is an important parameter for learning user preference. Next, we study the effect of the depth of hops on sequential recommendation. We select the best baseline GRU$_F$ as a reference. We only present the performance comparisons on Last.fm dataset, since the other results are similar and omitted here. Table 3 shows the performance change by varying the depth of reasoning. Note that for a fair comparison, we also vary the features used in GRU$_F$: it only takes the same categorial information that is available to TMRN at each comparison. As shown in Table 3, we find that TMRN outperforms GRU$_F$ consistently. Compared with GRU$_F$, TMRN yields an increase of about 5.0% $\sim$ 5.6% in the next-item recommendation task and 6.6% $\sim$ 8.5% in the multi-item recommendation task by varying the number of hops. The results obtained from both MRN and TMRN improves with the increasing of the depth of hops, which indicates the effectiveness of multi-hop reasoning. Another observation is when we use more categorical information, the performance also improves.

**Table 3: MAP performance with the increasing depth of hops in TMRN on Last.fm dataset.**

| Methods | Next-Item | | | Multi-item | | |
|---|---|---|---|---|---|---|
| | 1-hop | 2-hop | 3-hop | 1-hop | 2-hop | 3-hop |
| GRU$_F$ | 0.525 | 0.552 | 0.571 | 0.332 | 0.355 | 0.377 |
| MRN | 0.494 | 0.502 | 0.509 | 0.314 | 0.321 | 0.326 |
| TMRN | 0.551 | 0.583 | 0.603 | 0.354 | 0.385 | 0.405 |

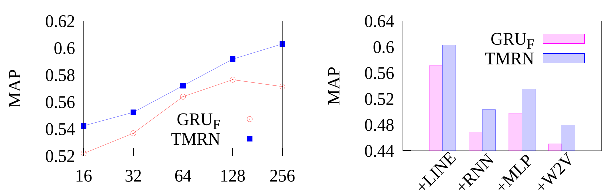
*6.2.2 Varying the amount of training data.* TMRN contains more parameters to learn and theoretically has a higher model complexity than baselines. We study the performance sensitivity of our models by varying the amount of training data. We take 20%, 40%, 60% and 80% from the complete training data to generate four new training sets, respectively. The test set is fixed as original. In Figure 3, we can see that TMRN performs better than GRU$_F$ consistently with four different training sets. Compared with GRU$_F$ Although the model complexity of TMRN is higher, we take several effective techniques to enhance the model learning, *e.g.*, pretraining the item and category embeddings. Furthermore, our reasoning architecture is aligned with external taxonomy data. It effectively utilizes the hierarchical knowledge for developing a stable reasoning mechanism, which is also useful to reduce the complexity for parameter learning in practice.




(a) Last.fm (b) Amazon

**Figure 3: Performance comparison with varying training data.**

*6.2.3 Varying hierarchical concept embeddings.* In our model, we use the network embedding method LINE [36] for learning the representations of category nodes. To see the effect of embedding size on the performance, we vary it in the set $\{16, 32, 64, 128, 256\}$. In Fig. 4(a), we see that an embedding size of 128 gives the best performance for LINE. With the increasing of the embedding size, TMRN gradually improves and is consistently better than GRU$_F$. As discussed previously, there can be other methods for learning category embeddings. Here we consider three alternative methods for comparison. Recall each item is associated with a path consisting of category labels, which can be considered as a node sequence. So we can apply RNN and Word2Vec [26] (W2V for short) for modeling the sequences and learning node embeddings. The third method is to construct a three-layer MLP model for predicting the category labels and meanwhile learning node embeddings. As shown in Fig. 4(b), LINE gives a better performance than the other methods.




(a) Varying the embedding size of LINE. (b) Varying embedding methods.

**Figure 4: Performance tuning *w.r.t.* different embedding sizes and different embedding methods.**

## 6.3 Case Study

Previous experiments have verified that our model is capable to generate high-quality sequential recommendations. Another major benefit is that our recommendations are highly interpretable due to the incorporation of taxonomy information in multi-hop reasoning process. As a case study, Fig. 5 presents an interaction sequence from JD, in which the purchase of the "*Thinkpad laptop*" leads to subsequent purchases. TMRN utilizes the previous five records to recommend a new product (the sixth) to the user. In the given example, each product is associated with a path in the taxonomy, including the information for *category*, *sub-category* and *brand*.

Recall that we will learn a unique preference representation for each level of the taxonomy given a user, which is able to capture more fine-grained sequential characteristics for a user. At the first hop, TMRN has firstly predicted a larger weight on the category of *computer accessories*. Then, the model would focus on the products from *computer accessories*. In the second hop, our model has predicted a larger weight on the subcategory of *mouse pad*, which further narrows down the selection of products. In the third hop, it was on the brand of *Newmen*. The intermediate representations *w.r.t.* each hop are learned in a hierarchical way, which naturally model the selection process from general to specific. After the three-hop reasoning, we combine all the three user representations to obtain the final user representation, which is more accurate to identify the ground-truth product (*i.e., Newmen mouse pad*). This example indicates our model is able to well understand users' preference for sequential recommendation, since it captures more fine-grained preference representations aligned with the category hierarchy.

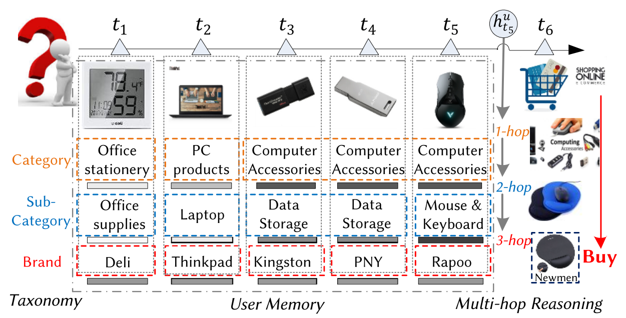



**Figure 5: An interaction sequence from a sample user in JD dataset. We show the intermediate predictions of TMRN on the categories in the reasoning process. We use color darkness to indicate the value of attention weights: darker is larger. We characterize the sequential preference of the user in three hops. We highlight the categories with high probabilities in user memories with colored boxes.**

## 7 CONCLUSIONS

In this paper, we have proposed a novel Taxonomy-aware Multi-hop Reasoning Network (TMRN) for better understanding and modeling user preference for sequential recommendation. We have associated the learning of user preference with the category hierarchy. Our model is able to model the evolvement and refinement of user preference over the taxonomy, which largely enhances the reasoning capacity and the model interpretability. Understanding users' decision-making process for sequential actions has been a challenging topic [17, 20]. Here, we make an attempt by leveraging structured knowledge information from taxonomy data. As future work, we plan to develop more effective reasoning models in capturing complex user behaviors.

**ACKNOWLEDGEMENTS**. The work was partially supported by National Natural Science Foundation of China under the Grant Number 61872369, 61832017 and 61502502.

# REFERENCES


[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[2] X. Chen, H. Xu, Y. Zhang, J. Tang, Y. Cao, Z. Qin, and H. Zha. Sequential recommendation with user memory networks. In *WSDM*, pages 108–116, 2018.

[3] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *EMNLP*, pages 103–111, 2014.

[4] E. Constantinides. Influencing the online consumer's behavior: the web experience. *Internet research*, 14(2):111–126, 2004.

[5] T. Donkers, B. Loepp, and J. Ziegler. Sequential user-based recurrent neural network recommendations. In *RecSys*, pages 152–160, 2017.

[6] J. Gao, T. Zhang, and C. Xu. A unified personalized video recommendation via dynamic recurrent neural networks. In *ACM MM*, pages 127–135, 2017.

[7] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *CoRR*, abs/1410.5401, 2014.

[8] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwinska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, A. P. Badia, K. M. Hermann, Y. Zwols, G. Ostrovski, A. Cain, H. King, C. Summerfield, P. Blunsom, K. Kavukcuoglu, and D. Hassabis. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016.

[9] R. He and J. Mcauley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*, pages 507–517, 2016.

[10] R. He, W. Kang, and J. McAuley. Translation-based recommendation. In *RecSys*, pages 161–169, 2017.

[11] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua. Neural collaborative filtering. In *WWW*, pages 173–182, 2017.

[12] J. L. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *CSCW*, pages 241–250, 2000.

[13] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. Session-based recommendations with recurrent neural networks. *CoRR*, abs/1511.06939, 2015.

[14] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *RecSys*, pages 241–248, 2016.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[16] J. Huang, W. X. Zhao, H. Dou, J.-R. Wen, and E. Y. Chang. Improving sequential recommendation with knowledge-enhanced memory networks. In *SIGIR*, pages 505–514, 2018.

[17] G. H"ubl and V. Trifts. Consumer decision making in online shopping environments: The effects of interactive decision aids. *Marketing Science*, 19(1):4–21, Jan. 2000.

[18] B. Kanagal, A. Ahmed, S. Pandey, V. Josifovski, J. Yuan, and L. G. Pueyo. Supercharging recommender systems using taxonomies for learning user purchase behavior. *PVLDB*, 5(10):956–967, 2012.

[19] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[20] T. Kurashima, T. Althoff, and J. Leskovec. Modeling interdependent and periodic real-world action sequences. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 803–812, 2018.

[21] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma. Neural attentive session-based recommendation. In *CIKM*, pages 1419–1428, 2017.

[22] P. Li, Z. Wang, Z. Ren, L. Bing, and W. Lam. Neural rating regression with abstractive tips generation for recommendation. In *SIGIR*, pages 345–354. ACM, 2017.

[23] F. Liu and J. Perez. Gated end-to-end memory networks. In *EACL*, pages 1–10, 2017.

[24] Q. Liu, S. Wu, D. Wang, Z. Li, and L. Wang. Context-aware sequential recommendation. In *ICDM*, pages 1053–1058, 2016.

[25] A. K. Menon, K. P. Chitrapura, S. Garg, D. Agarwal, and N. Kota. Response prediction using collaborative filtering with hierarchies and side-information. In *SIGKDD*, pages 141–149, 2011.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[27] A. H. Miller, A. Fisch, J. Dodge, A. Karimi, A. Bordes, and J. Weston. Key-value memory networks for directly reading documents. In *EMNLP*, pages 1400–1409, 2016.

[28] A. Mnih. Taxonomy-informed latent factor models for implicit feedback. In *KDD Cup*, pages 169–181, 2012.

[29] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *RecSys*, pages 130–137, 2017.

[30] Z. Ren, S. Liang, P. Li, S. Wang, and M. de Rijke. Social collaborative viewpoint regression with explainable recommendations. In *WSDM*, pages 485–494. ACM, 2017.

[31] S. Rendle. Factorization machines with libfm. *ACM TIST*, 3(3):57:1–57:22, 2012.

[32] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

[33] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, 2010.

[34] M. Schedl. The lfm-1b dataset for music retrieval and recommendation. In *ICMR*, pages 103–110, 2016.

[35] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. End-to-end memory networks. In *NIPS*, pages 2440–2448, 2015.

[36] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. LINE: large-scale information network embedding. In *WWW*, pages 1067–1077, 2015.

[37] N. Tintarev and J. Masthoff. A survey of explanations in recommender systems. In *ICDE*, pages 801–810, 2007.

[38] K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic, and E. Duval. Context-aware recommender systems for learning: A survey and future challenges. *TLT*, 5(4):318–335, 2012.

[39] J. Wang, W. X. Zhao, Y. He, and X. Li. Leveraging product adopter information from online reviews for product recommendation. In *ICWSM*, pages 464–472, 2015.

[40] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng. Learning hierarchical representation model for nextbasket recommendation. In *SIGIR*, pages 403–412, 2015.

[41] X. Wang, X. He, F. Feng, L. Nie, and T. Chua. TEM: tree-enhanced embedding model for explainable recommendation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 1543–1552, 2018.

[42] Z. Wang, Z. Jiang, Z. Ren, J. Tang, and D. Yin. A path-constrained framework for discriminating substitutable and complementary products in e-commerce. In *WSDM*, pages 619–627. ACM, 2018.

[43] L. Weng, Y. Xu, Y. Li, and R. Nayak. Exploiting item taxonomy for solving cold-start problem in recommendation making. In *IEEE*, pages 113–120, 2008.

[44] J. Weston, S. Chopra, and A. Bordes. Memory networks. *Eprint Arxiv*, 2014.

[45] C. Y. Wu, A. Ahmed, A. Beutel, H. Jing, and H. Jing. Recurrent recommender networks. In *WSDM*, pages 495–503, 2017.

[46] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan. A dynamic recurrent model for next basket recommendation. In *SIGIR*, pages 729–732, 2016.

[47] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma. Collaborative knowledge base embedding for recommender systems. In *SIGKDD*, pages 353–362, 2016.

[48] Y. Zhang, A. Ahmed, V. Josifovski, and A. J. Smola. Taxonomy discovery for personalized recommendation. In *WSDM*, pages 243–252, 2014.

[49] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *SIGIR*, pages 83–92, 2014.

[50] W. X. Zhao, Y. Guo, Y. He, H. Jiang, Y. Wu, and X. Li. We know what you want to buy: a demographic-based system for product recommendation on microblogs. In *KDD*, pages 1935–1944, 2014.

[51] W. X. Zhao, S. Li, Y. He, E. Y. Chang, J. Wen, and X. Li. Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. *TKDE*, 28(5):1147–1159, 2016.

[52] W. X. Zhao, G. He, H. Dou, J. Huang, S. Ouyang, and J. Wen. Kb4rec: A dataset for linking knowledge bases with recommender systems. *CoRR*, abs/1807.11141, 2018.

[53] C. Ziegler, G. Lausen, and L. Schmidt-Thieme. Taxonomy-driven computation of product recommendations. In *CIKM*, pages 406–415, 2004.