Multi-Order Attentive Ranking Model for Sequential Recommendation

Lu Yu,[1] **Chuxu Zhang,**[2] **Shangsong Liang,**[1,3] **Xiangliang Zhang**[1]
[1]King Abdullah University of Science and Technology, Thuwal, 23955, SA
[2]University of Notre Dame, IN 46556, USA
[3]School of Data and Computer Science, Sun Yat-sen University, China
Email: {lu.yu,xiangliang.zhang}@kaust.edu.sa, czhang11@nd.edu, liangshangsong@gmail.com

Abstract

In modern e-commerce, the temporal order behind users' transactions implies the importance of exploiting the transition dependency among items for better inferring what a user prefers to interact in "near future". The types of interaction among items are usually divided into individual-level interaction that can stand out the transition order between a pair of items, or union-level relation between a set of items and single one. However, most of existing work only captures one of them from a single view, especially on modeling the individual-level interaction. In this paper, we propose a *Multi-order Attentive Ranking Model* (*MARank*) to unify both individual- and union-level item interaction into preference inference model from multiple views. The idea is to represent user's short-term preference by embedding user himself and a set of present items into multi-order features from intermedia hidden status of a deep neural network. With the help of attention mechanism, we can obtain a unified embedding to keep the individual-level interactions with a linear combination of mapped items' features. Then, we feed the aggregated embedding to a designed *residual neural network* to capture union-level interaction. Thorough experiments are conducted to show the features of *MARank* under various component settings. Furthermore experimental results on several public datasets show that *MARank* significantly outperforms the state-of-the-art baselines on different evaluation metrics. The source code can be found at https://github.com/voladorlu/MARank.

Introduction

Modeling the complicated interactions behind users' temporal preferences over items is very essential for providing personalization service in many domains like e-commerce, social friend suggestion, news/article recommendation, etc. In this work, we focus on exploring users' sequential behavior for predicting next item to visit/buy according to his/her given sequential transaction. One typical example is which book will Mike buy at future time *t* after ordering multiple books? To answer this question, the proposed solution needs to account for a balance between *general* (long-term) and *sequential* (short-term) patterns.

As a representative solution, Markov chain based approaches (Chen et al. 2012; Rendle, Freudenthaler, and Schmidt-Thieme 2010; He, Kang, and McAuley 2017; He and McAuley 2016a) shine light on how to extend general user preference by fusing **individual-level** item-item transition dependency, shown as **Figure 1(a)**, where the arrows indicate the individual-level dependency between a pair of items that don't need to be adjacent. Given a set of previous interacted items (referred as query items), the dependency of a target item on these query items is usually measured as the average or weighted relevance value. Such individual-level dependencies can result in better recommendations, however have limitations on capturing the collective influence between items. For example, buying *eggs*, *milk*, and *butter* together indicates a higher probability of buying *flour* than *eggs*, *milk* or *butter* individually.

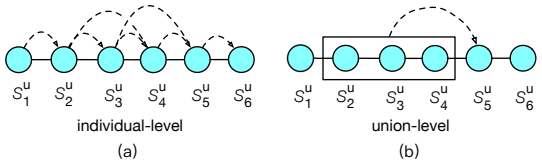


Figure 1: Toy example to illustrate individual- and union-level item relevance model. S^u denotes historical interaction sequence of user *u*, S^u_i represents a specific item.

To tackle this limitation, Tang *et al.* (Tang and Wang 2018) introduce to model **union-level** influence shown as **Figure 1(b)**, which can be measured with the support-confidence for rule $X \rightarrow Y$ of the form $(x_{t-L}, \cdots, x_{t-2}, x_{t-1}) \rightarrow x_t$. Such a rule can reflect that purchase decisions are made based on what collocational products have already been ordered. However, from the evidences as shown in (Tang and Wang 2018), there is a serious sparsity problem as L increases. That is, union-level item-item relation is difficult to model, especially in sparse data. Note that modeling individual-level relationship has no such a problem since only a single pair of items are involved. Despite the success of recent advances on applying *recurrent neural network* (RNN) or *convolutional neural network* (CNN) (Hidasi et al. 2016; Loyola, Liu, and Hirate 2017; Tang and Wang 2018; Wang et al. 2015) to model union-level item dependency, they whittle the ability of discriminating the highly correlated item pairs for sequential recom-

mendation. Therefore, we argue that individual- or union-level influence can be complementary to tackle users' temporal preference learning problem. As we discussed in previous paragraph, pairwise item interaction can be more easily extracted than union-to-single item interaction. In particular, it's more important to utilize individual influence between query and target item when the large length of association rule is not well supported in sparse data.

In this paper, we propose to **unify individual- and union-level** item dependency into a Multi-order Attentive Ranking model (*MARank*) towards top-N sequential recommendation. Modeling union-level influence can be conceptually understood by estimating the probability of associate rule $X \rightarrow Y$, which could be measured by the matching score between a distributional representation of an item set X and single item Y. In this work, we design a model with the combination of attention network and residual network to represent item set X. The abstractive framework is described in **Figure 2**. To better understand users' short-term preference over items, we design a multi-order attention network to highlight the contribution of the encoded sequential items' multi-order features from the intermedia hidden status of a feedforward residual neural network. The idea is inspired by the method to understand an image with features extracted from low to high layer, which act as object representation from different views. Then, we feed the aggregated embedding from the attention network as the input to a multilayer perceptron (MLP) with residual structure (*encoded union-level features*), and simultaneously construct a shortcut connection from input vector to the output of MLP layer. To keep *weighted individual-level features*, an element-wise addition operation is applied on the output from multi-order attention network and MLP layer, then the obtained aggregated *contextual embedding* can be used to capture item-item relation from both individual- and union-level. From experimental analysis on three large-scale real datasets, we can see that the designed structure works very well to outperform the state-of-the-art baselines. The main contributions of this paper can be summarized as:

- We propose to view user's short-term preferences over items from multiple perspectives, and design a corresponding multi-order attentive neural model to extend user's general preference by fusing both individual- and union-level temporal item-item correlations.

- The proposed attention model can learn the contribution of encoded features of sequential items from different order, which keeps the individual-level item-item interaction by augmenting the contribution of relevant items, but downplays the irrelevant ones. MLP encoder with residual structure can help to learn high-order sequential dependency among items from union-level perspective.

- We conduct thorough experiments on studying the impacts of different components on the performance of MARank, and show that it outperforms state-of-the-art baselines in terms of different ranking metrics.

Related Work

Recommendation without Temporal Dynamics: Item recommendation has become an integrated, even leading function in enormous applications involved with products suggestion, hunting excellent musicians, etc. Early works mainly focus on optimizing rating estimation through neighborhood (Linden, Smith, and York 2003) and model based (Koren, Bell, and Volinsky 2009; Salakhutdinov, Mnih, and Hinton 2007) collaborative filtering approaches. Differently hitting users' interests by a short ranking list as much as possible focuses on relative predicted score, other than exact estimation error.

To express the above-mentioned idea, many factorization models are proposed, ranging from point-wise (Hu, Koren, and Volinsky 2008; He et al. 2016), pairwise (Rendle et al. 2009; Yu et al. 2018), to list-wise (Shi, Larson, and Hanjalic 2010; Shi et al. 2012) ranking methods. Along with recent advances on designing deep neural networks (DNNs) for computer vision, text mining etc., Wu *et al.* (Wu et al. 2016) indicated out that factorization models can be represented as a shallow neural network, which makes factorization models very flexible in incorporating encoded auxiliary information by DNNs. He *et al.* (He et al. 2017) focused on capturing highly nonlinear interactions between a user-item pair by unifying factorization models with a multiple perceptron layers.

Next-item Prediction: Temporal dynamics (Tavakol and Brefeld 2014) in recommendation tasks have been demonstrated very effectively in personalized item prediction. In a seminal work for modeling temporal dynamics, Koren (Koren 2009) explored temporal drifting user preferences over items for rating estimation problem. Subsequently several works (Cheng et al. 2013; Yin et al. 2014) are inspired to care about temporal drift effect. However, most of them make use of explicit timestamp difference, and ignore an important fact that users prefer visiting related items in a short period. Chen *et al.* (Chen et al. 2012) utilized Markov chains to explore the transition probability among items via predicting next purchase depending on last actions. Such an idea can reflect users' short-term preferences, however, personalization means beyond that. To better model temporal dynamics, several recent works like FPMC (Rendle, Freudenthaler, and Schmidt-Thieme 2010), HRM (Wang et al. 2015), TranRec (He, Kang, and McAuley 2017), Fossil (He and McAuley 2016b), combine users' general (long-term) tastes with Markov chains (short-term) based on factorization methods. Despite the success of them, it may not be sufficient to capture complicated interactions of user's dynamic transition behaviors via the choice of *shallow relevance metrics* like inner-product (e.g., FPMC, HRM, Fossil) or Euclidean (e.g., TranRec) of low-dimensional embeddings. Recently there are also some works based on recurrent neural networks. (Hidasi et al. 2016; Yu et al. 2016; Li et al. 2018) proposed to use GRU unit to build session-based recommender systems. However, they did not consider users' personalization, nor can tell the contribution scale of each previous item to target one. Both (Chen et al. 2017; Wang et al. 2018) explored the application of attention method to weigh a set of historical items, however

missing the sequential dependency among items.

Proposed Method

Let U and I represent the user and item set, respectively. The whole transaction data can be described as $S = \{S^u : u \in U\}$. Each user $u \in U$ is associated with a transaction sequence $S^u = (S_1^u, S_2^u, ..., S_{|S^u|}^u)$, where $S_t^u \in I$ denotes a user *u* ever interacted with an item at a specific time point. Note that we only care about the relative orders like first, second and so on, but ignore the absolute time point like 1st May, 2018. Sequential recommendation aims at predicting the next item that a user will be interested in with a given historical transaction sequence. Usually, only a very short list of items will be exposed to users. Therefore, hitting users' requirement in a top-N recommendations is a non-trivial sequential prediction problem.

Modeling Long-term User Preference

Understanding the interactions between users and items is the key to provide successful personalization service. Factorization model has gained lots of attention, where proximity between a user *u* and item *j* can be measured as an inner-product of their embeddings:

$$\hat{y}_{u,j} = \mathbf{p}_u^T \mathbf{q}_j, \tag{1}$$

where $\mathbf{p}_u \in \mathbb{R}^d$ and $\mathbf{q}_j \in \mathbb{R}^d$ denote a distributional representation of user *u* and item *j*, respectively. The larger prediction $\hat{y}_{u,j}$ is, the more likely that user *u* will act on item *j* in future. However, such definition of user-item relevance only captures long-term preference, while discards an important fact that human behaviors naturally come along with temporal dependencies. In next section, we will discuss how to augment long-term user interests by unifying both individual- and union-level item-item dependency.

Multi-order Attentive Ranking Model

The proposed model aims to express user interests over items by decomposing user's consuming behavior into long-term and short-term motivations. In previous section, we already show how to capture long-term preference by explicitly measuring user-item similarity with general embedding. In this part, we will discuss how to extend general user's embedding with short-term interests over a small set of the most *n* present items, which can be denoted as $S_{t-1,n}^u = (S_{t-1}^u, S_{t-2}^u, ..., S_{t-n}^u)$. Analogous to the objective scoring function of FPMC (Rendle, Freudenthaler, and Schmidt-Thieme 2010), we formulate the MARank's predictive model as:

$$\hat{y}_{u,j} = \mathbf{p}_u \mathbf{q}_j^T + \mathbf{F}(u, S_{t-1,n}^u)\mathbf{m}_j^T \tag{2}$$

where $\mathbf{m}_j \in \mathbb{R}^d$ denotes additional item embedding to model item-item transition dependency, and $\mathbf{F}(u, S_{t-1,n}^u)$ represents the designed multi-order attentive neural model, including the components *embedding lookup layer*, *multi-order attention network*, and *residual network* (*ResNet*), as shown in **Figure 2(a)**.

Embedding Lookup Layer. The input to the neural encoder $\mathbf{F}(u, S_{t-1,n}^u)$ contains two parts: a user *u*, and a subset $S_{t-1,n}^u$ including the most recent *n* items before time point *t*. The embedding lookup operation retrieves previous *n* items' embeddings, and stacked them together resulting in a matrix $\mathbf{E} \in \mathbb{R}^{n \times d}$:

$$\mathbf{E} = \begin{bmatrix} \mathbf{m}_{S_{t-1}^u} \\ \mathbf{m}_{S_{t-2}^u} \\ \vdots \\ \mathbf{m}_{S_{t-n}^u} \end{bmatrix}. \tag{3}$$

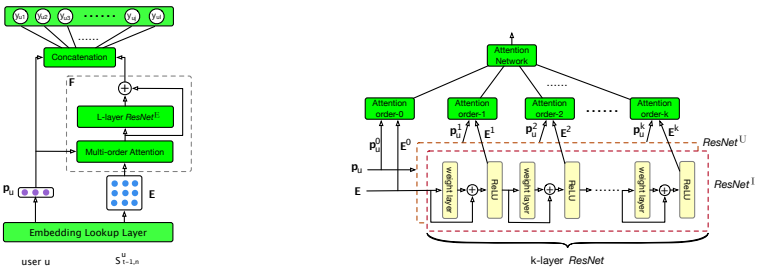
Along with item embeddings, we also obtain the user embedding $\mathbf{p}_u \in \mathbb{R}^d$ to represent general user latent feature. These embeddings are presented by corresponding symbols of the output of embedding lookup layer in the **Figure 2(a)**.

Residual Network (*ResNet*). From the **Figure 2** we can see that *ResNet* plays a key role on learning high-order nonlinear interactions among different elements of input vectors. We initialize residual network with a fully connected multilayer perceptron (MLP) but not CNN, considering the explored problem does not have nature spatial patterns like images. The definition of *L*-layer *ResNet*(*L*, **x**) is as follows:

$$\begin{aligned} \mathbf{h}_1 &= ReLU(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1 + \mathbf{x}) \\ \mathbf{h}_2 &= ReLU(\mathbf{h}_1\mathbf{W}_2 + \mathbf{b}_2 + \mathbf{h}_1) \\ &\cdots\cdots \\ \mathbf{h}_L &= ReLU(\mathbf{h}_{L-1}\mathbf{W}_L + \mathbf{b}_L + \mathbf{h}_{L-1}) \end{aligned} \tag{4}$$

where $\mathbf{W}_L \in \mathbb{R}^{d \times d}, \mathbf{b}_L \in \mathbb{R}^d$ represent weight matrix, and bias vector, respectively. *ReLU* stands for the activation function *rectifier linear unit*. The hyper-parameter *L* denotes the maximum number of residual layers. The output of *L*-layer *ResNet* contains two parts that are a sequence of hidden status $\mathbf{H}_L$=$\{\mathbf{h}_1, \cdots, \mathbf{h}_L\}$ and the last hidden unit $\mathbf{h}_L$. In MARank, we instantiate three different residual networks, *i.e.* $ResNet^I, ResNet^U$ and $ResNet^E$. To capture complicated personal characteristics, we propose to map raw user and item embeddings into high-order features with the help of multilayer $ResNet^U$ and $ResNet^I$, respectively. The idea is inspired by representing vision object with low-to-high level features extracted by deep neural networks. Analogously the sequence of hidden status from $ResNet^U$ or $ResNet^I$ might capture potential high-order interactions between partial fields in user or item embedding, which can help us to discriminate the significance of items in different levels with respect to a given user.

Multi-order Attention Network. With the input embeddings $\mathbf{p}_u$ and $\mathbf{E}$, the multi-order attention network is instantiated with two *k*-layer residual network, *i.e.* $ResNet^U(k, \mathbf{p}_u)$ and $ResNet^I(k, \mathbf{E})$, as shown in **Figure 2(b)**. Since $ResNet^U$ shares the same structure as $ResNet^I$ in **Figure 2(b)**, we only present the feature transformation process of $ResNet^I$. Let $\mathbf{H}_k^U = \{\mathbf{p}_u^1, \cdots, \mathbf{p}_u^k\}$ and $\mathbf{H}_k^I = \{\mathbf{E}^1, \cdots, \mathbf{E}^k\}$ denote the output sequence of hidden statuses of $ResNet^U$ and $ResNet^I$, respectively. In order to distinguish the sources of hidden statuses, we use $\mathbf{p}_u^k$ and $\mathbf{E}^k$ to represent the high-order features generated at *k*-th layer of $ResNet^U$ and $ResNet^I$. Besides the high-order features, we also keep the raw embeddings, resulting in an extended



(a) Overall architecture (b) Multi-order Attention Network

Figure 2: (a) Overall architecture of the proposed model MARank. $\mathbf{p}_u$ and $\mathbf{E}$ denotes the retrieved user embedding and embeddings for prior items in $S^u_{t-1,n}$, respectively. (b) Illustration of the multi-order attention network. As residual network for encoding user and items share the same architecture, we only elaborate the specific structure of $ResNet^I$.

set of encoded features, *i.e.* $\mathbf{H}^U_k = \{\mathbf{p}^l_u | l \in \{0, 1, \cdots, k\}\}$, $\mathbf{H}^I_k = \{\mathbf{E}^l | l \in \{0, 1, \cdots, k\}\}$, where $\mathbf{p}^0_u = \mathbf{p}_u$ and $\mathbf{E}^0 = \mathbf{E}$.

To aggregate the embeddings of sequential items, we utilize a soft attention method. When the high-order features $\mathbf{H}^U_k$ and $\mathbf{H}^I_k$ for both input user u and all sequential items are ready, we can apply the following attentive model to generate an integrated contextual embedding $\mathbf{e}^l_c \in \mathbb{R}^d$ at *l*-layer:

$$\mathbf{e}^l_c = \sum_{i=1}^{n} \alpha^l_{ui} \cdot \mathbf{m}^l_{S^u_{t-i}}; \sum_{i=1}^{n} \alpha^l_{ui} = 1 \tag{5}$$

where weight scale α^l_{ui} is normalized by a softmax layer on the attention scores when given the embedding of user u and the sequential item S^u_{t-i}, and $\mathbf{m}^l_{S^u_{t-i}}$ is the corresponding *i*-th row of transformed matrix $\mathbf{E}^l$. In this work, we use a two-layer network to compute the attention score as follows:

$$\alpha_{ui} = \mathbf{W}^a_1 g(\mathbf{p}^l_u \mathbf{W}^a_u + \mathbf{m}^l_{S^u_{t-i}} \mathbf{W}^a_i + \mathbf{c}_2) + \mathbf{c}_1$$
$$\alpha_{ui} = \frac{exp(\alpha_{ui})}{\sum_{i=1}^{n} exp(\alpha_{ui})},$$

where $\mathbf{W}^a_*$ are the shared weight matrices for attention layer. $g(x)$ is activation function and we use $tanh$ activation unit in this work. Let $\widehat{\mathbf{E}} = \{\mathbf{e}^l_c | l \in \{0, \cdots, k\}\}$ represent the set of aggregated high-order features depending the interactions with given user's features. Based on $\widehat{\mathbf{E}}$, we have many methods to obtain the final *contextual embedding* to represent user's short-term interests with the most recent *n* items. For example, we can employ multiple self-attention blocks *Transformer* (Vaswani et al. 2017) to get much more complicated feature interactions, or directly apply RNN. Though they can provide flexibility to model rich interactions, in this work, we prefer to using a simple soft attention network. The reason is very straightforward that wrapping a attention layer outside $\widehat{\mathbf{E}}$ can help to keep **individual-level** item-item similarity. Specifically, we define the soft attention as follows:

$$\mathbf{e}_c = SoftAtt(\widehat{\mathbf{E}}) = \beta \widehat{\mathbf{E}}$$
$$\beta = softmax(\mathbf{W}^b_1 g(\widehat{\mathbf{E}} \mathbf{W}^b_2 + \mathbf{b}^{att}_2) + \mathbf{b}^{att}_1) \tag{6}$$

where $\mathbf{W}^b_*$ acts as the weight matrices for attention layer, and $\beta \in \mathbb{R}^{k+1}$ is the attention weight vector. We can see that the output of multi-order attention network actually is linear combination of the high-order features inside $\mathbf{H}^I_k$, that is $\mathbf{e}_c = \sum_{l=0}^{k} \sum_{i=1}^{n} \beta_l \cdot \alpha^l_{ui} \mathbf{m}^l_{S^u_{t-i}}$. If we define $\mathbf{F}(u, S^u_{t-1,n})$ in Equation (2) as $\mathbf{e}_c$, then second term $\mathbf{F}(u, S^u_{t-1,n})\mathbf{m}^T_j = \sum_{l=0}^{k} \sum_{i=1}^{n} \beta_l \cdot \alpha^l_{ui} \mathbf{m}^l_{S^u_{t-i}} \mathbf{m}^T_j$ equals to weighted sum of relevance value of item pairs, that is, **individual-level** item-item interaction is captured. In order to model **union-level** item dependency, we instantiate *L*-layer residual network $ResNet^E(L, \mathbf{e}_c)$ and use the last hidden status $\mathbf{h}_L$ as the encoded feature to represent the set of short-tern interacted items by the given user u. Then we have the final definition $\mathbf{F}(u, S^u_{t-1,n}) = \mathbf{e}_c + \mathbf{h}_L$, and the full definition of the prediction value for each item as:

$$\hat{y}_{u,j} = \underbrace{\mathbf{p}_u \mathbf{q}^T_j}_{long-term} + \underbrace{\overbrace{\mathbf{e}_c \mathbf{m}^T_j}^{Individual-level} + \overbrace{\mathbf{h}_L \mathbf{m}^T_j}^{Union-level}}_{short-term} \tag{7}$$

Ranking Optimization: To effectively learn the parameters of the proposed MARank model, we define a pairwise ranking loss based on the assumption that an item (positive sample) this user liked will have a relative larger value than other items (negative samples) that he/she has no interest in. The loss function is formulated as:

$$\arg\min_{\Theta} \sum_{u \in U} \sum_{j \in S^u} \sum_{s \notin S^u} -\ln \sigma(\hat{y}_{u,j,z}) + \frac{\lambda}{2}(\sum_{u \in U} ||\mathbf{p}_u||^2_F + \sum_{j \in I} ||\mathbf{m}_j||^2_F + ||\mathbf{q}_j||^2_F), \tag{8}$$

where $||\cdot||_F$ denotes *Frobenius* norm, and $\hat{y}_{u,j,z} = \hat{y}_{u,j} - \hat{y}_{u,z}$. Item *j* is an item liked by user *u*, in contrast with *z* that s/he shows no interest. Function $\sigma(x) = \frac{1}{1+exp(-x)}$ denotes the probability that a learned model gives a larger score to a positive sample than a negative one. λ is a hyper-parameter for L_2 norm regularization on model parameters.

It's intractable to directly do optimization on the objective function (Eq. (8)) due to the enormous amount of positive-negative sample pairs. Usually, we first uniformly sample a user *u* from U. Then we can sample a positive item *j* from

transaction sequence S^u at time point *t* along with the most *n* recent items $S^u_{t-1,n}$. Negative item $s \notin S^u$ will be uniformly sampled. In this work, we use Adam optimizer to optimize the proposed model. Since each training sample $(u, j, z, S^u_{t-1,n})$ can be constructed independently, we can apply mini-batch SGD to speed up the training efficiency.

Discussion

Temporal Order: One may argue that temporal order is very important for sequential recommendation, in contrast, attention network over items ignores the order. Generally we canset size *n* as small as possible to present user's short-term demands, based on the assumption that user's attention in a small session window will display his/her target motivations or demands at that moment. Different from previous works highlighting the temporal orders of observed interactions, we argue that temporal order is not a must constraint because user usually tend to hover between different items in a small session (like the skip behavior modeled in (Tang and Wang 2018)). The anchor items that drive user to make final decision might not be the closest one, but those viewed before. The experimental evaluations further support this argument, as we will show later. In MARank, we divide user's consuming behavior into long-term and short-term patterns. The long-term preference can store user's general tastes hidden in the whole transaction history, which make it possible to represent short-term preference only with a set of small size of prior interacted items.

Markov Chain (MC): We find that MARank can be viewed as a generalization of classical MC-based models. Rendle *et al.* introduced *Factorized Personalized Markov Chains* (FPMC) based on the observation that users could continuously review two closely related items. The proposed method MARank can be reduced to FPMC by keeping only individual-level contextual embedding and replacing the multi-order attention network with an average pooling operation. Different from FPMC, MARank can automatically weight the contribution of items through matching the relation between user and items.

Experimental Evaluation

In this section, we present our experimental results, with comparison to different kinds of baseline methods to answer the following research questions:

RQ1: Can our proposed method outperform the state-of-the-art baselines for sequential recommendation task?

RQ2: How does data sparsity influence MARank?

RQ3: How MARank is affected by each component?

Evaluation Protocol

To validate the proposed method MARank for top-N sequential recommendation, we use three publicly available datasets, Yelp [1] and Amazon [2], with statistics information in **Table 1** after pre-processing. Amazon data (He, Kang, and McAuley 2017) contains product purchase history ranging from May 1996 - July 2014. We mainly conduct experiments on two subsets, *Movies&TV* composing user review actions on both movie and tv, and *CDs&Vinyl* for actions on CDs and vinyl record. As the original datasets are too sparse, we filter out inactive users with fewer than 10 feedbacks. For each dataset, we convert star-rating into binary feedback regardless of the specific rating values since we care more about the applications without explicit user feedbacks like ratings, following the evaluation in (He, Kang, and McAuley 2017; He et al. 2016). We utilize a widely used evaluation setting, *fold-out* (Tang and Wang 2018; Chen et al. 2018) that is, a number of latest interactions of each user are hold-out as test set and the remaining data is for training. Without special description, we use last 20% of each user's feedbacks as test set.

Table 1: Statistical information of datasets.

Data	# Users	# Items	# Observation	Sparsity
Yelp	25677	25815	731671	99.89%
Movies&TV	35168	51227	1070645	99.94%
CDs&Vinyl	26876	66820	770188	99.95%

Several types of state-of-the-art baselines are taken into consideration, including **BPR-MF** (Rendle et al. 2009), **FPMC** (Rendle, Freudenthaler, and Schmidt-Thieme 2010), **TranRec** (He, Kang, and McAuley 2017), **GRU4Rec** (Hidasi et al. 2016), **NARM** (Li et al. 2017), **ACF** (Chen et al. 2017), **Caser** (Tang and Wang 2018). We evaluate all of algorithms at the following top-*N* ranking metrics: **Precision**@*N* and **Recall**@*N* (Karypis 2001), **NDCG**@*N* (Weimer et al. 2007) and **MRR**@*N* (Shi et al. 2012; Yu et al. 2014). We implement **TranRec** [3], **GRU4Rec** [4], **Caser** [5] with the code given by the authors.

Evaluation Setup : We use grid search to select the best parameter setting for different algorithms. These include embedding size d from $\{8, 16, 32, 64\}$, regularization hyper-parameter λ from $\{0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$. Since Adam is employed to optimize all the algorithms except TranRec, we use Adam's suggested default learning ratio *i.e.,* 0.001. Learning ratio for TranRec is selected from $\{0.1, 0.05, 0.01, 0.005\}$. Besides the commonly shared hyper-parameters, some algorithms have their own special setting. For FPMC, we set the length of Markov chain to one. We also tried other choices, however, the performance does not improve much. Without special mention, we fix the length *N* of ranked list as 20. The model is trained in Tensorflow on a GeForce GTX 1080Ti GPU.

Reproductivity: We study MARank with different number of previous items from $\{2, 4, 6, 8, 10\}$, while $n = 6$ gives the best results. We also try different number of hidden layers L or k from $\{0, 2, 4, 8\}$. From the experimental results, we find that 2 or 4 layers are enough to ensure competitive results. The dropout ratio and gradient clipping norm is set to 0.5

[1] https://www.yelp.com/dataset/challenge

[2] https://www.amazon.com/

[3] https://sites.google.com/a/eng.ucsd.edu/ruining-he/

[4] https://github.com/hidasib/GRU4Rec

[5] https://github.com/graytowne/caser_pytorch

Table 2: Ranking performance comparison (the best results of baseline are marked as * along with underline). The last row shows the improvement of MARank over the best baseline algorithm.

Methods	Yelp				Movies&Tv				CDs&Vinyl			
Measures@20	Rec	Pre	NDCG	MRR	Rec	Pre	NDCG	MRR	Rec	Pre	NDCG	MRR
BPR-MF	0.0618	0.0156	0.0382	0.0132	0.0404	0.0095	0.0248	0.0089	0.0512	0.0121	0.0323	0.0119
GRU4Rec	0.0586	0.0151	0.0373	0.0131	0.0372	0.0098	0.0235	0.0081	0.0358	0.0093	0.023	0.0082
NARM	0.0616	0.0159	0.0401	0.0145	0.0429	0.0111	0.0273	0.0096	0.0426	0.0108	0.028	0.0104
GRU4Rec+	0.0699	0.0179	0.0456	0.0166	0.0526	0.0138	0.0344	0.0124	0.0527	0.0138	0.0352	0.0129
NARM+	*0.0718	*0.0183	*0.0465	*0.0169	0.0587	*0.0145	0.0375	0.0136	0.0678	*0.0169	0.0449	0.0169
ACF	0.0661	0.0169	0.0425	0.0153	0.0458	0.0107	0.0286	0.0106	0.0554	0.0133	0.0357	0.0134
Caser	0.0653	0.0161	0.0415	0.0151	0.0494	0.0122	0.0318	0.0116	0.0495	0.0129	0.0346	0.0132
TranRec	0.0703	0.0176	0.0458	0.0169	0.0606	0.0135	0.0392	0.0155	*0.0704	0.0164	*0.0468	*0.0184
FPMC	0.0713	0.0178	0.0463	0.0169	*0.0608	0.0142	*0.0406	*0.0162	0.0646	0.0156	0.0450	0.0179
MARank	**0.0791**	**0.0199**	**0.0509**	**0.0183**	**0.0680**	**0.0162**	**0.0444**	**0.0170**	**0.0817**	**0.0201**	**0.0552**	**0.0212**
Improvement	10.1%	8.7%	9.4%	8.2%	11.8%	11.7%	9.3%	4.9%	16.0%	18.9%	17.8%	15.2%

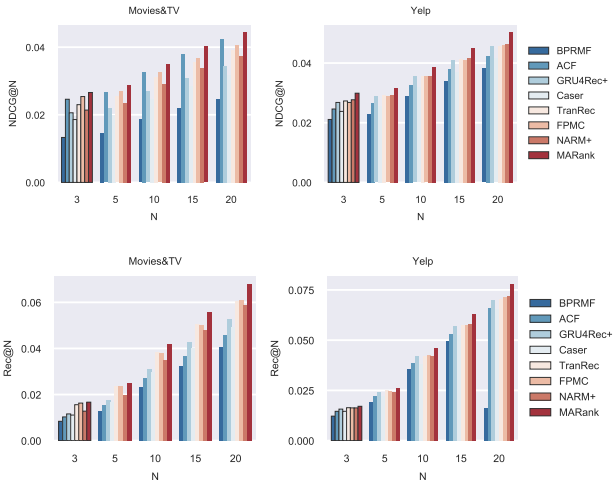


Figure 3: Top-N recommendation evaluation with different values of N on NDCG and Recall.

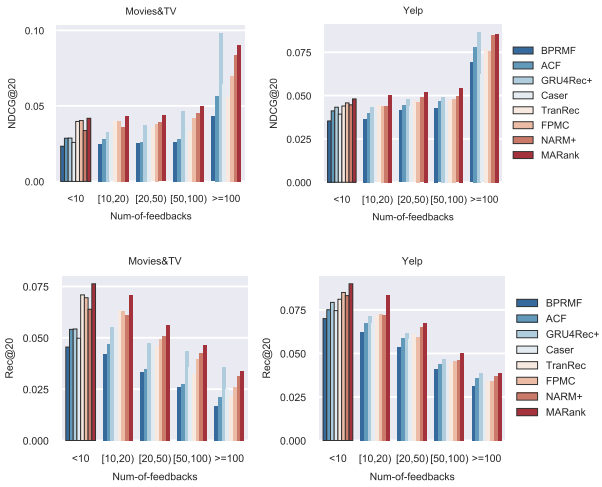


Figure 4: Recommendation evaluation on different user groups, which are separated according to the number of feedbacks.

and 5, respectively.

Overall Comparison (RQ1)

We begin with evaluating the performances of the proposed MARank and all baselines. Results are shown in **Table 2** and we have the following observations.

It's interesting that basic BPR-MF has competitive performance with RNN based method, GRU4Rec or NARM. Here BPR-MF considers user intrinsic preference over items, while GRU4Rec or NARM only models union-level item interaction. Such results may shine lights on the importance of user's general taste. To further validate the effectiveness of RNN-based model, we extend GRU4Rec and NARM to GRU4Rec+ and NARM+ via adding general user-item interaction, and find that GRU4Rec+ or NARM+ can improve BPR-MF after incorporating item's temporal dependency. Different from GRU4Rec+, NARM+ employs the attention network over the sequence of hidden statuses output by GRU. Comparing with GRU4Rec+, attention mechanism makes NRAM+ stand out. ACF also employs attention mechanism to aggregate user's historical items without taking the temporal order into account. Experimental results show that both NARM+ and ACF have better performance than BPR-MF. On considering that ACF and NARM+ share common ideas on fusing item-item relation into prediction function, the observation verifies that modeling item-item dependency can significantly help to improve the recommendation accuracy. Caser is a recently proposed model to apply CNN instead of GRU to compress a sequence of items' embeddings into a low-dimension vector. It has much better performance than GRU4Rec, which is consistent with the performance tendency shown in recent work (Tang and Wang 2018). Similar to NARM+, Caser considers not only union-level item dependency, but also user-item general preference. However, after extending GRU4Rec and NARM with user's general taste, Caser in contrast does not perform as well as GRU-based models.

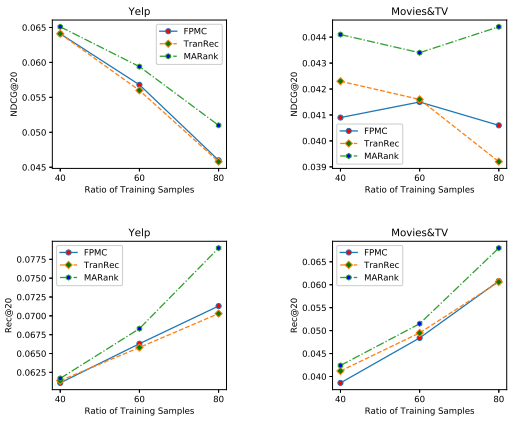


Figure 5: Performance on sparse training set. x axis stands for the proportion of training samples.

Instead of modeling union-level item interaction, TranRec and FPMC are two state-of-the-art approaches based on individual-level item temporal dependency. Interestingly, both of them outperform the other baselines. This observation suggests that keeping directed interaction between a pair of items is essential for sequential recommendation. Encouragingly, we can find that MARank always has better performance than other baselines. In MARank, both union- and individual-level item interaction are captured under the design of attentive neural network with residual structure. These results indicate that dealing with sequential recommendation task has close connection to the fusion of item-item interactions, especially temporal item dependency. Besides overall evaluation of different algorithms, **Figure 3** shows that MARank is stably superior to other baselines with different length of recommendation list.

A Study on Data Sparsity (RQ2)

Performance on Different User Groups In real applications, the number of feedbacks contributed by a user has a large variance. Most of users have a small number of feedbacks, which will raise "cold-start" problem. To gain more insight on the performance of each algorithm, we group users into five classes (*i.e.* <10, [10,20), [20,50), [50,100), >=100) based on the number of observed feedbacks. The results are shown in **Figure 4**. Due to the limited space, we only present evaluation results on Rec@20 and NDCG@20. We find that recall ratio decreases as the growing number of observed feedbacks. Most of the algorithms perform very well on user group with greater than 100 feedbacks, in particular, GRU4Rec+ achieves very high NDCG score on this group. It indicates that RNN-based methods favor long transaction sequence, but not short sequence. The results also demonstrate that MARank can have overall superior performance over different user groups.

Ratio of Training Samples To further validate the capacity of dealing with sparse dataset, we evaluate top-2 baselines (*i.e.* TranRec, FPMC) and MARank with different proportion of training samples. The results are shown in **Figure 5**. We can see that MARank still outperforms baselines in both NDCG@20 and Rec@20 metrics. As increasing the ratio, all of methods achieve better recall score. These observations verify that MARank can cope with data sparsity problem very well.

Table 3: Performance with different level of item-item interactions. MARank-I or MARank-U represents MARank with only individual- or union-level item-item dependency.

Methods	Yelp				Movies&TV			
Metrics@20	Rec	Pre	NDCG	MRR	Rec	Pre	NDCG	MRR
MARank-I	0.0755	0.0185	0.0475	0.0175	0.0635	0.0153	0.0405	0.0158
MARank-U	0.0731	0.0178	0.0456	0.0168	0.0616	0.0148	0.0374	0.0145
MARank	0.0785	0.0195	0.0509	0.0183	0.0680	0.0162	0.0441	0.0170

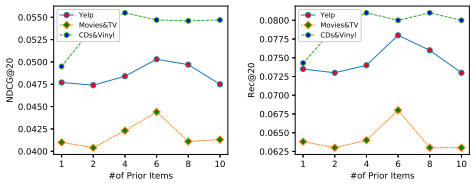


Figure 6: The impact of Markov chain length.

Analyzing Components of MARank (RQ3)

In this section, we will do ablation study on the impact of number of prior items and the contributions of individual- and union-level item-item dependency. Due to the limited space, we only explore the impact of different level item-item relations and the order of Markov chain, *i.e.* hyper-parameter n. In following experiments, the number of residual layers for $ResNet$s are set to 2.

Individual- or Union-level We study the influence of different types of item-item transition dependency. From the experimental results shown in **Table 3**, we can find that keeping only individual- or union-level can reduce the recommendation performance. Comparing with MARank-U, MARank-I has more stable ranking quality, which might give a hint that individual-level item-item relevance has positive impact on forecasting user's preference.

Orders of Markov chain From previous analysis, we have demonstrated that modeling the Markov chain pattern is very essential for next-item recommendation. Generally, MARank can achieve better and better performance on most of dataset when increasing the number of prior items, *i.e.* hyper-parameter n. In this study, we validate different choices of n (from 2 to 10 in our experiments). With results shown in **Figure 6**, $n = 6$ gives the best performance. Such observation happens, presumably, because sequential pattern does not involve a very long sequence.

Conclusions

Building deep model for recommender systems still remains a big challenge. In recommendation tasks, we usually deal

with discrete user behavior data, rather than image data. In this paper, we design a multi-order attentive ranking model, MARank, for sequential recommendation by unifying general and sequential patterns. We conduct extensive experiments on multiple large scale datasets and find that MARank significantly outperforms other baselines. In this work, we only model the dynamic user-item interactions. In real applications, sequential prediction usually involves with heterogeneous information domains. However, exploring temporal evolution of various types of information resources still remains to be a big challenge. In future, we'd like to design deep model to capture heterogeneous item relationship.

Acknowledgement

This work was supported by the funding from King Abdullah University of Science and Technology (KAUST), under award number FCC/1/1976-24-01.

References

Chen, S.; Moore, J. L.; Turnbull, D.; and Joachims, T. 2012. Playlist prediction via metric embedding. In *KDD'12*, 714–722.

Chen, J.; Zhang, H.; He, X.; Nie, L.; Liu, W.; and Chua, T.-S. 2017. Attentive collaborative filtering: Multimedia recommendation with feature- and item-level attention. In *SIGIR'17*.

Chen, X.; Xu, H.; Zhang, Y.; Tang, J.; Cao, Y.; Qin, Z.; and Zha, H. 2018. Sequential recommendation with user memory networks. In *WSDM '18*, 108–116.

Cheng, C.; Yang, H.; Lyu, M. R.; and King, I. 2013. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI'13*, 2605–2611.

He, R., and McAuley, J. 2016a. Fusing similarity models with markov chains for sparse sequential recommendation. In *ICDM'16*.

He, R., and McAuley, J. 2016b. Fusing similarity models with markov chains for sparse sequential recommendation. In *ICDM'16*, 191–200.

He, X.; Zhang, H.; Kan, M.-Y.; and Chua, T.-S. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR'16*, 549–558.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *WWW'17*, 173–182.

He, R.; Kang, W.-C.; and McAuley, J. 2017. Translation-based recommendation. In *RecSys'17*, 161–169.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2016. Session-based recommendations with recurrent neural networks. *arXiv:1511.06939*.

Hu, Y.; Koren, Y.; and Volinsky, C. 2008. Collaborative filtering for implicit feedback datasets. In *ICDM'08*, 263–272.

Karypis, G. 2001. Evaluation of item-based top-n recommendation algorithms. In *CIKM'01*, 247–254.

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

Koren, Y. 2009. Collaborative filtering with temporal dynamics. In *KDD'09*, 447–456.

Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; and Ma, J. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, 1419–1428.

Li, Z.; Zhao, H.; Liu, Q.; Mei, T.; and Cheng, E. 2018. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *KDD'18*.

Linden, G.; Smith, B.; and York, J. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing* 7(1):76–80.

Loyola, P.; Liu, C.; and Hirate, Y. 2017. Modeling user session and intent with an attention-based encoder-decoder architecture. In *RecSys'17*, 147–151.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI'09*, 452–461.

Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW'10*, 811–820.

Salakhutdinov, R.; Mnih, A.; and Hinton, G. 2007. Restricted boltzmann machines for collaborative filtering. In *ICML'07*, 791–798.

Shi, Y.; Karatzoglou, A.; Baltrunas, L.; Larson, M.; Oliver, N.; and Hanjalic, A. 2012. CLiMF: Learning to maximize reciprocal rank with collaborative less-is-more filtering. In *RecSys'12*, 139–146.

Shi, Y.; Larson, M.; and Hanjalic, A. 2010. List-wise learning to rank with matrix factorization for collaborative filtering. In *RecSys'10*, 269–272.

Tang, J., and Wang, K. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM '18*, 565–573.

Tavakol, M., and Brefeld, U. 2014. Factored mdps for detecting topics of user sessions. In *RecSys'14*, 33–40.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *CoRR* abs/1706.03762.

Wang, P.; Guo, J.; Lan, Y.; Xu, J.; Wan, S.; and Cheng, X. 2015. Learning hierarchical representation model for nextbasket recommendation. In *SIGIR'15*, 403–412.

Wang, S.; Hu, L.; Cao, L.; Huang, X.; Lian, D.; and Liu, W. 2018. Attention-based transactional context embedding for next-item recommendation.

Weimer, M.; Karatzoglou, A.; Le, Q. V.; and Smola, A. 2007. COFIRANK maximum margin matrix factorization for collaborative ranking. In *NIPS'07*, 1593–1600.

Wu, Y.; DuBois, C.; Zheng, A. X.; and Ester, M. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *WSDM'16*, 153–162.

Yin, H.; Cui, B.; Chen, L.; Hu, Z.; and Huang, Z. 2014. A temporal context-aware model for user behavior modeling in social media systems. In *SIGMOD'14*, 1543–1554.

Yu, X.; Ren, X.; Sun, Y.; Gu, Q.; Sturt, B.; Khandelwal, U.; Norick, B.; and Han, J. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *WSDM '14*, 283–292.

Yu, F.; Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2016. A dynamic recurrent model for next basket recommendation. In *SIGIR'16*, 729–732.

Yu, L.; Zhang, C.; Liang, S.; and Zhang, X. 2018. Walkranker: A unified pairwise ranking model with multiple relations for item recommendation. In *AAAI'18*.