



Joint Relational Dependency Learning for Sequential Recommendation

Xiangmeng Wang¹, Qian Li², Wu Zhang^{1(✉)}, Guandong Xu^{2(✉)},
Shaowu Liu², and Wenhao Zhu¹

¹ Shanghai University, Shanghai 200444, China
{[chrystal.li](mailto:chrystal.li@shu.edu.cn), [wzhang](mailto:wzhang@shu.edu.cn), [whzhu](mailto:whzhu@shu.edu.cn)}@shu.edu.cn

² University of Technology Sydney, Sydney, Australia
{[qian.li](mailto:qian.li@uts.edu.au), [guandong.xu](mailto:guandong.xu@uts.edu.au), [shaowu.liu](mailto:shaowu.liu@uts.edu.au)}@uts.edu.au

Abstract. Sequential recommendation leverages the temporal information of users' transactions as transition dependencies for better inferring user preference, which has become increasingly popular in academic research and practical applications. Short-term transition dependencies contain the information of partial item orders, while long-term transition dependencies infer long-range user preference, the two dependencies are mutually restrictive and complementary. Although some work investigates unifying both long-term and short-term dependencies for better performance, they still neglect the fact that short-term interactions are multi-folds, which are either individual-level interactions or union-level interactions. Existing sequential recommendations mainly focus on user's individual (i.e., individual-level) interactions but ignore the important collective influence at union-level. Since union-level interactions can reflect that human decisions are made based on multiple items he/she has already interacted, ignoring such interactions can result in the disability of capturing the collective influence between items. To alleviate this issue, we proposed a Joint Relational Dependency learning (JRD-L) for sequential recommendation that exploits both long-term and short-term preferences at individual-level and union-level. Specifically, JRD-L combines long-term user preferences with short-term interests by measuring short-term pair relations at individual-level and union-level. Moreover, JRD-L can alleviate the sparsity problem of union-level interactions by adding more descriptive details to each item, which is carried by individual-level relations. Extensive numerical experiments demonstrate JRD-L outperforms state-of-the-art baselines for the sequential recommendation.

Keywords: Sequential recommendation · Long-term user preference · Short-term user preference · Multi-relational dependency

1 Introduction

Nowadays, abundant user-item interactions in recommender system (RS) are recorded over time, which can be further used to discover the patterns of users'

behaviors [3, 12]. Therefore, sequential recommendation is becoming a new trend in academic research and practical applications, because it is capable of leveraging temporal information among users' transactions for better inferring the user preference.

Dominant approaches aim to modeling long-term temporal information, capturing holistic dependencies of user-item sequence, while short-term temporal information which are essential in capturing partial dependencies are also significant. The long-term interaction is depicted in Fig. 1(a) where arrows indicate the dependency among a user-item interaction sequence. As a representative in long-term dependency modeling for general RS, factorization-based methods plays an important role in long-term dependency sequential recommendation for its remarkable efficiency [12]. Factorization-based methods model the entire user-item interaction matrix into two low-rank matrices. Such measure that aims to deal with the entire user-item interaction matrix is well-suited to train models that capture longer-term user preference profiles, however has limitations on capturing short-term user interests. Two main drawbacks exist in factorization-based methods for sequential recommendation: 1) they failed to fully exploit the rich information of transition dependencies of multiple items; 2) modeling the entire user-item dependencies causes enormous computing cost of growing size of user-item interaction matrix when user has new interactions [8, 9].

As for modeling users' short-term interests, mainstream methods such as Markov chain-based approaches [3] leverage transition dependency of items from the individual-level. The short-term interaction at individual-level is shown as Fig. 1(b). Therefore, individual-level dependencies can capture individual influence between a pair of single item, but may neglect the collective influence [19] among three or more items denoted by union-level dependencies, as shown in Fig. 1(c). Namely, the collective influence is caused by the dependency of a group of items on a single item. To alleviate this issue, Yu et al. [19] leverages both individual and collective influence for better sequential recommendation performance. However, two main drawbacks exist in this methods: 1) the information of individual and collective influence is simply added to the output proximity score of a factorization-based model, leveraging none of the long-term information; 2) The union-level interaction requires a group of items to be joint modeled within a limit length of sequence, which may lead to sparsity problem.

In this paper, we propose a unified framework *joint relational dependency learning* (JRD-L), which exploits long-term temporal information and short-term temporal information from individual-level and union-level for improving sequential recommendation. In particular, a Long Short-Term Memory (LSTM) model [5] is used to encode long-term preferences, while short-term dependencies existing in pair relations among items are computed based on the intermedia hidden states of LSTM on both individual-level and union-level. LSTM hidden states can carry the long-term dependencies information and transmit them to short-term item pairs. Meanwhile, the individual-level relation and union-level relation are modeled together to fully exploit the collective influence among union-level pair relation and to address the sparsity problem. The framework

of JRD-L is described in Fig. 3. Experiments on large-scale dataset demonstrate the effectiveness of the proposed JRD-L. The main contributions of our paper can be summarized as

- JRD-L considers user’s long-term preferences along with short-term pair-wise item relations from multiple perspectives of individual-level and union-level. Specifically, JRD-L involves a novel multi-pair relational LSTM model that can capture both long-term dependency and multi-level temporal correlations for better inferring user preferences.
- A novel attention model is also combined with JRD-L that can augment individual-level and union-level pair relation by learning the contributions to the subsequent interactions between users and items. Meanwhile, the weighted outputs of attention model are fused together, contributing more individual-level information to alleviates the sparse problem in the union-level dependency.

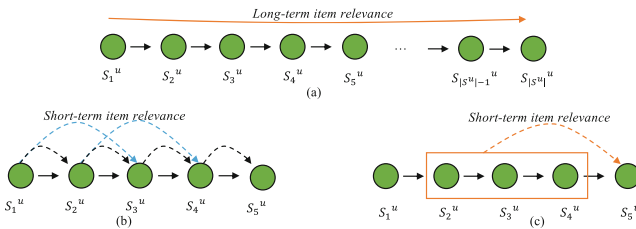


Fig. 1. (a) Long-term user-item interaction; (b) Individual-level item relevance; (3) Union-level item relevance. The dependencies of an item on its’ subsequent item is represented as the transition arrows.

2 Related Works

Many methods consider long-term temporal information to mining the sequential patterns of the users’ behaviors, including factorization-based approaches [12, 14] and Markov chains based approaches [2]. Recently, Deep learning (DL)-based models have achieved significant effectiveness in long-term temporal information modeling, including multi-layer perceptron-based (MLP-based) models [16, 17], Convolutional neural network-based (CNN-based) models [6, 15] and Recurrent neural network-based (RNN-based) models [1]. RNN-based models stand out among these models for its capacity of modeling sequential dependencies by transmitting long-term sequential information from the first hidden state to the last one. However, RNN can be difficult to trained due to the vanishing gradient problem [7], but advances such as Long Short-Term Memory (LSTM) [5] has enabled RNN to be successful. LSTM is considered one of the most successful variant of RNN, with the capability of capturing long-term relationships in a sequence and suffering from the vanishing gradient problem. So far, LSTM models have achieved tremendous success in sequence modelling tasks [20, 21].

With respect to short-term temporal information modeling, existing works on modeling short-term temporal information mainly model pair relations between items. The representative work is Markov Chain (MC)-based models [3]. The objective of such model is to measure the average or weighted relevance values between a given item and its next-interaction item, this only captures dependencies between two single items. Tang et al. [15] propose a method capturing collective dependencies among three or more items. However, the model in [15] suffers from data sparsity problems. Therefore, in order to solve the sparsity problem when merely modeling collective dependencies, Yu et al. [19] add individual (i.e. individual-level) dependencies into collective (i.e. union-level) dependencies, but their work is still insufficient for it does not leverage long-term temporal information.

3 Joint Relational Dependency Learning

Before introducing the proposed method, we provide some useful notations as follows. Let U and I be the user and item set, as shown in Fig. 2. A sequence of interactions between U and I can be represented as $S = \{S_j^{u_i} : u_i \in U\}$, and u_i is associated with a interaction sequence $S_j^{u_i} = (S_1^{u_i}, S_2^{u_i}, \dots, S_{|S_j^{u_i}|}^{u_i})$. The goal of JRD-L method is to predict the likelihood of the user preferred item $e_c^{u_i}$, based on the user’s behavior sequences $S_j^{u_i}$.

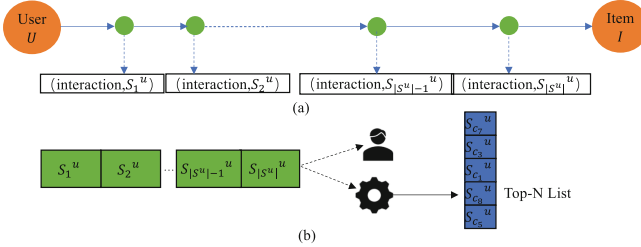


Fig. 2. (a) $S_j^{u_i} = (S_1^{u_i}, S_2^{u_i}, \dots, S_{|S_j^{u_i}|}^{u_i})$ denotes a sequence of interactions between a user u_i and a given item set I . (b) Next-item recommendation aims to generate a ranking list exposed to users by modeling user-item interaction sequence.

The overall architecture of JRD-L is shown in Fig. 3. Generally, JRD-L first models long-term dependency over the whole user-item interaction data $S = \{S_j^{u_i} : u_i \in U\}$ in a LSTM layer. JRD-L takes the most recent n items before time point t of the whole sequence as the short-term interaction sequence. Then, JRD-L computes individual-level and union-level pair relations on the taken short sequence as short-term dependencies modeling. Specifically, with the input of u_i and $S_j^{u_i}$, JRD-L composes u_i and $S_j^{u_i}$ into a single user-item vector via an Embedding layer, and output $e_c^{u_i}$ as the user-item interaction embedding. A LSTM layer is then used to map

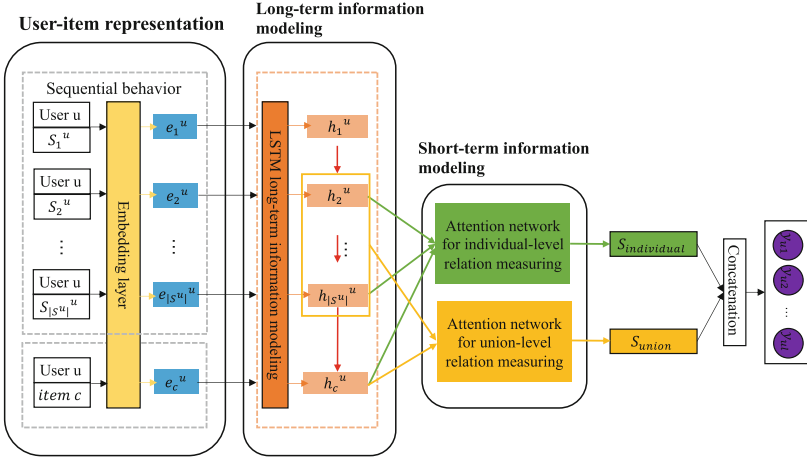


Fig. 3. The overall framework of Joint Relational Dependency Learning (JRD-L).

the whole interaction sequence of user-item vector $e_1^{u_i}, e_2^{u_i}, \dots, e_{|S_j^{u_i}|}^{u_i}$ into a sequence of hidden vectors $h_1^{u_i}, h_2^{u_i}, \dots, h_{|S_j^{u_i}|}^{u_i}$. More importantly, we take one further step from $h_{|S_j^{u_i}|}^{u_i}$ to derive hidden vector $h_{c_i}^{u_i}$ encoding $e_c^{u_i}$ to model long-term sequential information. Based on this, $h_{c_i}^{u_i}$ is paired with the most recent n items before time point t , i.e., $h_{t-1}^{u_i}, h_{t-2}^{u_i}, \dots, h_{t-n}^{u_i}$ ($t - n < |S_j^{u_i}|$), individually. JRD-L then passes the corresponding hidden status pairs of the most recent items to an attention layer, output the correlation likelihood $S_{individual}$ and S_{union} , from which the short-term individual-level and union-level pair relation is modeled, respectively. At Last, $S_{individual}$ is concatenate with S_{union} to obtain the correlation of $e_c^{u_i}$ with the existing items for the next-item prediction task.

3.1 Skip-Gram Based Item Representation

By learning the item similarities from a large number of sequential behaviors over items, we apply skip-gram with negative sampling (SGNS) [10] to generate a unified representation for each item in an given user-item interaction sequence $S_j^{u_i} = (S_1^{u_i}, S_2^{u_i}, \dots, S_{|S_j^{u_i}|}^{u_i})$. Before exploiting users' sequences dependencies, our prior problem is to represent items via embedding layer in a numerical way for subsequent calculations. In the embedding layer, the skip-gram with negative sampling is applied to directly learn high-quality item vectors from users' interaction sequences. The SGNS [10] generate item representations by exploiting the sequence of interactions between users and items. Specifically, given an item interaction sequence $S_j^{u_i} = (S_1^{u_i}, S_2^{u_i}, \dots, S_{|S_j^{u_i}|}^{u_i})$ of user u_i from the user-item interaction sequence S , SGNS aims to solve the following objective function

$$\arg \max_{v_j, w_i} \frac{1}{K} \sum_{i=1}^K \sum_{j \neq i}^K \log(\sigma(w_i^T * v_j)) \prod_{j=1}^E \sigma(-w_i^T * v_k) \quad (1)$$

K is the length of sequence $S_j^{u_i}$, and $\sigma(w_i^T * v_j) \prod_{j=1}^E \sigma(-w_i^T * v_k)$ is computed by negative sampling. $\sigma(x) = 1/(1 + \exp(-x))$, $w_i \in U(\subset \mathbb{R}^m)$ and $v_i \in V(\subset \mathbb{R}^m)$ are the latent vectors that correspond to the target and context representation for items in $S_j^{u_i}$, respectively. The parameter m is the dimension parameter that is defined empirically according to dataset size. E is the number of negative samples per a positive sample. Finally, matrices U and V are computed to generate representation of interaction sequences.

3.2 User Preference Modeling for Long-Term Pattern

To model the long-term temporal information in users' behaviors, we apply a standard LSTM [5] as in Fig. 3 to model the temporal information over the whole user-item interaction sequence. For each user u_i , we first generate an interaction sequence S_{u_i} with embedding items $x_j \in I$ based on U and V calculated by Eq. (1) from embedding layer in Fig. 3, represented as $P_u = e_1^{u_i}, e_2^{u_i}, \dots, e_{|S_j^{u_i}|}^{u_i}$. We use $e_{|S_j^{u_i}|}^{u_i}$ as the d -dimensions latent vector of item x_j . Given the embedding of the user-item interaction sequence $e_1^{u_i}, e_2^{u_i}, \dots, e_{|S_j^{u_i}|}^{u_i}$ and the candidate next-item $e_{c_i}^{u_i} \in e_c^{u_i}$, we generate a sequence of hidden vectors $h_1^{u_i}, h_2^{u_i}, \dots, h_{|S_j^{u_i}|}^{u_i}$ by recurrently feeding $e_1^{u_i}, e_2^{u_i}, \dots, e_{|S_j^{u_i}|}^{u_i}$ as inputs to the LSTM. The inner hidden states in LSTM hidden layer are updated at each time step, which can carry the long-term dependencies information and transmit them to item pairs. At each time step, the next output of computing last hidden status h_i^u is computed by

$$h_i^u = g(e_i^u, h_{i-1}^u, W_{LSTM}) \quad (2)$$

where g is the output function in LSTM and W_{LSTM} are network weights of e_i^u and h_{i-1}^u . Each $e_{c_i}^{u_i}$ is appended separately to $h_1^{u_i}, h_2^{u_i}, \dots, h_{|S_j^{u_i}|}^{u_i}$ calculated by Eq. (2) to obtain the long-term-dependency-sensitive hidden states $h_{c_i}^{u_i}$.

$$h_{c_i}^u = g(e_{c_i}^u, h_{|S_{u_i}|}^u, W_{LSTM}) \quad (3)$$

Through *LSTM long-term information modeling* in Fig. 3, $h_{c_1}^{u_i}, h_{c_2}^{u_i}, \dots, h_{|c_l|}^{u_i}$ is output by Eq. (3) and l is the total number of candidate next items. The sequence $h_1^{u_i}, h_2^{u_i}, \dots, h_{|S_j^{u_i}|}^{u_i}$ is calculated by Eq. (2) for the following multi-relational dependency modeling stage.

3.3 Multi-relational Dependency Modeling for Short-Term Pattern

Long-term dependency models long-range user preferences but neglect important pairwise relations between items, which is insufficient in capturing pairwise relation from multiple level. Therefore our proposed method should unify both short-term sequential dependency (at both individual-level and union-level) and long-term sequential dependency. Inspired by [18], based on $h_{c_1}^{u_i}, h_{c_2}^{u_i}, \dots, h_{|c_l|}^{u_i}$

and $h_1^{u_i}, h_2^{u_i}, \dots, h_{|S_j^{u_i}|}^{u_i}$ output by *LSTM long-term information modeling* stage, we calculate pair relations on $h^{u_i}_{t-1}, h^{u_i}_{t-2}, \dots, h^{u_i}_{t-n}$ ($t-n < |S_j^{u_i}|$) selected from $h_1^{u_i}, h_2^{u_i}, \dots, h_{|S_j^{u_i}|}^{u_i}$. The task is then to learn the correlation between the items in interaction sequence and candidate items. Rather than directly applying the work [18] for modeling the short-term dependency, we introduce an attention mechanism to calculate pair relations from individual-level and union-level to fully modeling the user preferences to different items. This is mainly because the work [18] implies that all vectors share the same weight, discarding an important fact that human naturally have different opinions on items. By introducing attention mechanism, our work can distribute high weights on these items user like more, thus improving recommendation performance.

Individual-Level Pairwise Relations. To capture the individual-level pairwise relations, the input of *attention network for individual-level relation measuring layer* is $h_1^{u_i}, h_2^{u_i}, \dots, h_{|S_j^{u_i}|}^{u_i}$ and $h_{c_1}^{u_i}, h_{c_2}^{u_i}, \dots, h_{|c_i|}^{u_i}$, which is the output vector of *LSTM long-term information modeling layer* in Fig. 3. Specifically, $h_{c_i}^{u_i} \in h^{(2)} = (h_{c_1}^{u_i}, h_{c_2}^{u_i}, \dots, h_{|c_i|}^{u_i})$ (as indicated in Eq. 3) is paired with the hidden states of the most recent n items before time point t , which is $h^{u_i}_{t-1}, h^{u_i}_{t-2}, \dots, h^{u_i}_{t-n}$ ($t-n < |S_j^{u_i}|$) from $h_1^{u_i}, h_2^{u_i}, \dots, h_{|S_j^{u_i}|}^{u_i}$ calculated by Eq. (2). An attention network is used for pairwise relation measuring. Let $H \in \mathbb{R}^{n \times l}$, $H_{ij} = h_i^{(1)} * h_j^{(2)}$ is a matrix consisting of output vectors of last LSTM layer, and n is the size of $h^{(1)} = (h_{t-1}^{u_i}, h_{t-2}^{u_i}, \dots, h_{t-n}^{u_i})$ in Eq. (2) and l is the size of $h^{(2)} = (h_{c_1}^{u_i}, h_{c_2}^{u_i}, \dots, h_{|c_i|}^{u_i})$ in Eq. (3). The attentive weights $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_{t-n})$ of the items in interaction sequence are defined by a weighted sum of these output vectors as $\alpha = \text{softmax}(\omega^T M)$ and $M = \tanh(H)$. We obtain M by a fully connection layer activated by *tanh* activation function. ω^T is a transpose vector of attention network's parameters. $\alpha_i \in [0, 1]$ is the weight of $h_{t-j}^{u_i}$ and $h_{t-j}^{u_i} \in h^{(1)}$. After obtaining the weight α_i of each existing item $h_{t-j}^{u_i}$, the likelihood S_{c_i} , which describe how likely the exiting items in user-item interaction sequence will interact with $e_{c_i}^{u_i}$ in candidate next-interact items set, can be calculated by

$$s_k = \text{softmax}(\beta_1 h_{t-j}^u + \beta_2 h_{c_i}^u + b)$$

$$S_{\text{individual}} = \sum_{i=1}^{n-1} \alpha_i \cdot s_k \quad (4)$$

where s_k is the correlation score of the pair of item $h_{t-j}^u \in h^{(1)}$ with $h_{c_i}^u \in h^{(2)}$, $S_{c_i} \in S_{\text{individual}}$ is the output of *attention network for individual-level relation measuring layer*. β_1 , β_2 and b are LSTM parameters.

Union-Level Pairwise Relations. In order to model short-term union-level pair relation, we predefine a sliding window to determine the length of collective items set in existing user-item sequences. Based on the defined items set, collaborate influence in union-level pair relation can be learned in *attention network for union-level relation measuring layer*. Union-level pairwise relations

learned by our method can capture collective dependencies among three or more items, which complements to the individual-level relation for improving recommendation performance. In the union-level pairwise relation modeling stage, the candidate length of collective items set is defined from $\theta = \{2, 4, 6, 8\}$. To learn the collaborate influence in union-level pair relation, we define a sequence $Q = \{Q_1, \dots, Q_{n-\theta}\}$, $Q_i = (h_i^u, \dots, h_{\theta+i}^u)$. For example, if $\theta = 2$, we have $Q = \{(h_1^u, h_2^u), (h_2^u, h_3^u), \dots, (h_{n-2}^u, h_{n-1}^u, h_n^u)\}$. Then each $Q_i \in Q$ is paired with $h^{(2)} = (h_{c_1}^{u_i}, h_{c_2}^{u_i}, \dots, h_{|c_i|}^{u_i})$ as in Eq. (3). Then union-level pairs pass through the *attention network for union-level relation measuring layer* to obtain the weight α_i of each existing item h_{t-j}^u , and output the correlation likelihood S_{union} by

$$s_m = \text{softmax}(\beta_3 W_i + \beta_4 h_{c_i}^u + b)$$

$$S_{union} = \sum_{i=1}^{n-1} \alpha_i \cdot s_m \quad (5)$$

$h_{c_i}^u \in h^{(2)}$, β_3 , β_4 and b are model parameters. Then, S_{union} is output by *attention network for union-level pair relation measuring layer*. Finally, S_{union} is concatenated with $S_{individual}$ from *attention network for individual-level pair relation measuring layer* to calculate the correlation of $e_c^{u_i}$ with the existing items for the next-item prediction task.

3.4 Optimization

To effectively learn the parameters of the proposed JDR-L model, our training objective is to minimize the loss between the predicted labels and the true labels of candidate items. The optimization setup is, firstly, we define the item that has the latest timestamp among the user-item interaction sequence as the standard subsequent item, and define the rest of items as the non-subsequent items. Secondly, the loss function is therefore based on the assumption that an item (positive samples, i.e. standard subsequent item) 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 then formulated as

$$\arg \min_{\Theta} \sum_{i=1}^N (\text{concatenate}(S_{individual}^{(i)}, S_{union}^{(i)}) - y_i)^2 + \frac{\lambda}{2} \|\Theta\|^2 \quad (6)$$

where the parameter $\Theta = \{W_{LSTM}, \omega, \beta_1, \beta_2, \beta_3, \beta_4, b\}$. $S_{individual}^{(i)}$ in Eq. (4) represents the correlation likelihood output by *attention network for individual-level relation measuring layer*. $S_{union}^{(i)}$ in Eq. (5) represents the correlation likelihood output by *attention network for union-level relation measuring layer*. y_i is the label of the candidate item and λ is a parameter for l_2 regularization. Adaptive moment estimation (Adam) [11] is used to optimize parameters during the training process.

4 Experiments

4.1 Evaluation Setup

We conduct experiments to validate JDR-L for Top-N sequential recommendation task on the real-world dataset, i.e., Movie&TV dataset [19], that belongs to Amazon data¹. Since the original datasets are sparse, we firstly filter out users with fewer than 10 interactions as in [19]. The statistical information of the before-processing and after-processing of Movie&TV dataset is shown in Table 1. Following the evaluation settings in [19], we set train/test with ratios 80/20.

Table 1. Statistical information of dataset.

Movies&TV	Users	Items	Interactions
Before-processing	40929	51510	1163413
After-processing	35168	51227	1070645

We compare JRD-L with three baselines: BPR-MF [12] is a widely used matrix factorization method for sequential RS; TranRec [4] models users as translation vectors operating on item sequences for sequential RS; RNN-based model (i.e., GRU4Rec [6] uses basic Gated Recurrent Unit for sequential RS); FPMC [13] is a typical Markov chain method modeling individual item interactions; Multi-level item temporal dependency model (MARank) [19] models both individual-level and union-level interactions with factorization model.

For fair comparisons, we set the dropout percentage as 0.5 [19]. The embedding size d of Embedding layer is chosen from $\{32, 64, 128, 256\}$, which should be equal to the hidden size h of LSTM. The regularization hyper-parameter λ is selected from $\{0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$. We set the learning rate of Adam as the default number 0.001 [11]. As n is the most recent items for short-term dependency, we choose n from $\{10, 20, 40, 60\}$. The length l of the sliding window of union-level interaction is chosen from $\{2, 4, 6, 8\}$. We define the length N of ranked list as 20. For the hardware settings, JRD-L model is trained on a Linux server with Tesla P100-PCIE GPU.

4.2 Effect of Parameter Selection for JDR-L

This section will discuss how the parameters influence the JRD-L model performance. We first explore the impact of n on the performance of JDR-L, the comparison is set on different n chosen from $\{10, 20, 40, 60\}$. Secondly, we evaluate the influence of the length l , l is chosen from $\{2, 4, 6, 8\}$. We use two metrics

¹ <https://www.amazon.com/>.

to evaluate the model performance, which are MRR (Mean Reciprocal Rank) - the average of reciprocal ranks of the predicted candidate items, and NDCG (Normalized Discounted Cumulative Gain) - a normalized average of reciprocal ranks of the predicted candidate items with a discounting factor, the comparison results of different setups are shown in Fig. 4. Figure 4 show that when other hyperparameters are set equal, $n = 10$ achieves the best performance. These observations, presumably, because sequential pattern does not involve a very long sequence. Besides, $l = 4$ achieves the best performance, indicating that the collective influence of 4 items is informative for the Movie&TV dataset.

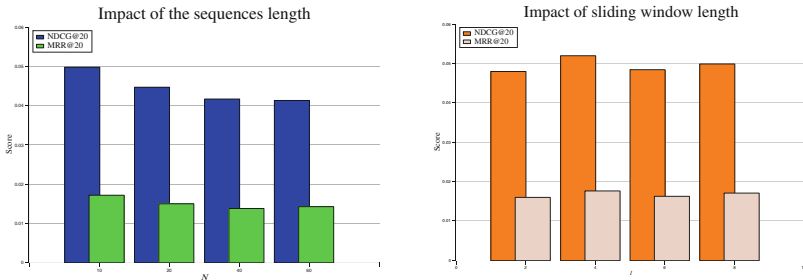


Fig. 4. Results of JDR-L under different settings.

4.3 Ranking Performance Comparison

Ranking performance evaluates how the predicted Top-N lists act on the recommendation system. Table 2 shows the comparison results of JDR-L with baselines. Encouragingly, we can find that JDR-L performs best with the highest MRR and NDCG scores. Besides, baselines may not perform well as JDR-L. Firstly, BPR-MF as matrix factorization-based method obtains less competitive performance when compared with GRU4Rec. This is mainly because BPR-MF considers user intrinsic preference over item while GRU4Rec models union-level item interaction along with users' overall preferences. Secondly, TranRec and FPMC are two state-of-the-art methods exploiting individual-level item temporal dependency. Both of them outperform the other baselines, since they consider individual-level item temporal dependency. This indicates that keeping directed interaction between a pair of items is essential for sequential recommendation. Thirdly, MARank considering individual-level and union-level interactions but neglecting long-term dependencies performs worse than JDR-L. Above all, BPR-MF performs the worst, this is mainly because BPR-MF models only intrinsic preferences within short sequences of user-item interactions, neglecting long-term user preferences and item interactions at individual-level and union-level.

Table 2. Ranking performance.

Methods	Movie&TV	
	MRR	NDCG
BPR-MF	0.0089	0.0248
TranRec	0.0155	0.0392
GRU4Rec	0.0124	0.0344
FPMC	0.0162	0.0406
MARank	0.0170	0.0444
JDR-L	0.0179	0.0518
Improvement	5.2%	16.7%

4.4 Components Influence of JDR-L

JDR-L contains three components as indicated by Fig. 3, i.e. *Long-term user-item interaction modelling*, *individual-level item interaction modeling* and *union-level item interaction modelling*. To analyze the influence of different components to the overall recommendation performance, we set different combinations of components for evaluation, with the results been shown in Table 3. JDR-L with three components performs best compared with other combinations as shown in Table 3, verifying that our proposed JDR-L is optimal. As for other combinations, LSTM-only obtains the lower MRR and NDCG scores compared with JDR-L, this is because LSTM-only models long-term dependencies. LSTM+individual-level item interaction outperforms LSTM+union-level item interaction, the main reason is that union-level item interaction suffers from a sparsity problem as the length of item set increases. Besides, both of LSTM+individual-level item interaction and LSTM+union-level item interaction obtain lower scores compared with JDR-L model. This further indicates that the information in individual-level item interaction should be combined into union-level interaction modeling stage to solve the sparsity problem.

Table 3. Ranking performance on different components in JDR-L.

Methods	Movie&TV	
	MRR	NDCG
Measures@20		
LSTM-only	0.0154	0.0447
LSTM+ individual-level item interaction	0.0147	0.0442
LSTM+ union-level item interaction	0.0142	0.0423
JDR-L	0.0178	0.0518

5 Conclusions

In this paper, we design a Joint Relational Dependency learning (JRD-L) for sequential recommendation. JRD-L builds a novel model to unify both long-term dependencies and short-term dependencies from individual-level and union-level. Moreover, JRD-L can handle the sparsity problem when exploiting the individual-level relation information from the sequential behaviors. Extensive experiments on the benchmark dataset demonstrate the effectiveness of JRD-L.

Acknowledge. This work is supported by the National Key R&D Program of China (Nos: 2017YFB0701501) and Australian Research Council Linkage Projects under LP170100891.

References

1. Bogina, V., Kuflik, T.: Incorporating dwell time in session-based recommendations with recurrent neural networks, pp. 57–59 (2017)
2. Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T.: The YouTube video recommendation system. In: RecSys 2010 - Proceedings of the 4th ACM Conference on Recommender Systems, pp. 293–296 (2010). <https://doi.org/10.1145/1864708.1864770>
3. He, R., Kang, W.C., McAuley, J.: Translation-based recommendation. In: RecSys 2017 - Proceedings of the 11th ACM Conference on Recommender Systems, pp. 161–169. Association for Computing Machinery, Inc., August 2017. <https://doi.org/10.1145/3109859.3109882>
4. He, R., Kang, W.C., McAuley, J.: Translation-based recommendation. In: Eleventh ACM Conference on Recommender Systems (2017)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997). <https://doi.org/10.1162/neco.1997.9.8.1735>
6. Hsu, K., Chou, S., Yang, Y., Chi, T.: Neural network based next-song recommendation. *arXiv: Information Retrieval* (2016)
7. Krause, B., Lu, L., Murray, I., Renals, S.: Multiplicative LSTM for sequence modelling. *arXiv preprint arXiv:1609.07959* (2016)
8. Li, Q., Niu, W., Li, G., Cao, Y., Tan, J., Guo, L.: Lingo: linearized grassmannian optimization for nuclear norm minimization (2015)
9. Li, Q., Wang, Z.: Riemannian submanifold tracking on low-rank algebraic variety. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
10. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. *arXiv: Computation and Language* (2013)
11. Newey, W.K.: Adaptive estimation of regression models via moment restrictions. *J. Econ.* **38**(3), 301–339 (1988)
12. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit Feedback, May 2012. <http://arxiv.org/abs/1205.2618>
13. Rendle, S., Freudenthaler, C., Schmidtthieme, L.: Factorizing personalized Markov chains for next-basket recommendation. In: The Web Conference, pp. 811–820 (2010)

14. Rendle, S., Gantner, Z., Freudenthaler, C., Schmidt-Thieme, L.: Fast context-aware recommendations with factorization machines. In: SIGIR 2011 - Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 635–644. Association for Computing Machinery (2011). <https://doi.org/10.1145/2009916.2010002>
15. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding, pp. 565–573 (2018)
16. Wan, S., Lan, Y., Wang, P., Guo, J., Xu, J., Cheng, X.: Next basket recommendation with neural networks (2015)
17. Wang, P., Guo, J., Lan, Y., Xu, J., Wan, S., Cheng, X.: Learning hierarchical representation model for nextbasket recommendation, pp. 403–412 (2015)
18. Wang, Z., Zhang, Y., Chang, C.Y.: Integrating order information and event relation for script event prediction, pp. 57–67. Association for Computational Linguistics (ACL), January 2018. <https://doi.org/10.18653/v1/d17-1006>
19. Yu, L., Zhang, C., Liang, S., Zhang, X.: Multi-order attentive ranking model for sequential recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 5709–5716, July 2019. <https://doi.org/10.1609/aaai.v33i01.33015709>
20. Zhao, P., Zhu, H., Liu, Y., Li, Z., Xu, J., Sheng, V.S.: Where to go next: A Spatio-temporal LSTM model for next poi recommendation. [arXiv: Information Retrieval](https://arxiv.org/abs/1808.08111) (2018)
21. Zhou, Y., Huang, C., Hu, Q., Zhu, J., Tang, Y.: Personalized learning full-path recommendation model based on LSTM neural networks. *Inf. Sci.* **444**, 135–152 (2018)