Session-based Recommendation with Local Invariance

Tianwen Chen
Dept. of Computer Science and Engineering
The Hong Kong University of Science and Technology
Hong Kong
tchenaj@cse.ust.hk

Raymond Chi-Wing Wong
Dept. of Computer Science and Engineering
The Hong Kong University of Science and Technology
Hong Kong
raywong@cse.ust.hk

***Abstract*—*Session-based recommendation* is a task to predict users' next actions given a sequence of previous actions in the same session. Existing methods either encode the previous actions in a *strict* order or completely *ignore* the order. However, sometimes the order of actions in a short sub-sequence, called the *detailed order*, may not be important, e.g., when a user is just comparing the same kind of products from different brands. Nevertheless, the *high-level* ordering information is still useful because the data is sequential in nature. Therefore, a good session-based recommender should pay different attention to the sequential information in different levels of granularity. To this end, we propose a novel model to automatically ignore the insignificant detailed ordering information in some sub-sessions, while keeping the high-level sequential information of the whole sessions. In the model, we first use a full self-attention layer with Gaussian weighting to extract features of sub-sessions, and then we apply a recurrent neural network to capture the high-level sequential information. Extensive experiments on two real-world datasets show that our method outperforms or matches the state-of-the-art methods.**

I. Introduction

With the rapid development of the internet, users are overwhelmed by a tremendous amount of information. It is inefficient or even impossible to find the most useful and interesting information with only users' active searching. Recommender systems help to solve the information overload problem by letting the information come to users, i.e., by recommending items or services that are likely to be useful or interesting to users. In order to give accurate recommendations, recommender systems need to learn users' preferences from their historical behavior.

In many online services such as e-commerce and media streaming, users' actions are often sequential, e.g., items viewed have temporal correlation among them. Moreover, the sequential actions can be grouped into sessions. Each session consists of actions that occur within a certain time period. The task of *session-based recommendation* is to predict a user's next action giving the user's previous actions in the same session.

Due to the highly practical value in many applications, session-based recommendation has increasingly attracted researchers' attention, and many interesting and effective approaches have been proposed. Most of the approaches achieving state-of-the-art performance follow an encoder-predictor architecture, where the encoder encodes the previous actions into a session embedding, and the predictor generates from the session embedding a probability distribution over the set of all possible actions.

It is easy to understand that the ordering information is useful to predict the next actions in session-based recommendation because session data is sequential in nature. However, not all the ordering information matters. Specifically, we argue that sometimes only the general *high-level* ordering information matters, but the ordering information at the sub-sessional level, called the *detailed ordering*, does not matter so much. For example, when a user wants to purchase a mobile phone along with some accessories on an e-commerce website, the user may need to compare several products from different brands for the mobile phone and its accessories. In this scenario, the order in which the user views different mobile phones or different brands of each accessory does not matter so much, since the user is just comparing the products without any intention to view the products in a specific order. We call the property that the ordering of viewing items within a short period of time does not matter the *local invariance*, which could be found easily in the real world. Moreover, the high-level ordering, namely the ordering that the user views each kind of products, has an impact on the next item. The reason is that if the user searches for mobile phones first, s/he may also want to buy a headphone because it is probable that s/he does not already have one and s/he needs one if s/he buys a mobile phone. On the other hand, if the user searches for headphones first, s/he might just want to buy a headphone. Therefore, a good session-based recommender should consider the ordering information in different levels of granularity.

However, existing studies do not address this local invariance property well. There are two branches of existing studies about session-based recommendation. The first branch includes methods that encode the actions in a *strict* order. Specifically, the recent research work [1, 2, 3] applied recurrent neural networks (RNNs) in the encoder component and achieved promising results. The main advantage of using RNNs is that the (detailed) sequential properties of sessions are naturally modelled by the network structure. Therefore, RNN based methods significantly outperform previous non-RNN based methods. However, these methods still enforce a *strict* order. Moreover, they focus more on users' recent actions and are not good at capturing dependencies in long sessions.

The second branch includes existing studies which completely discard the ordering information. Specifically, inspired by the success of the attention mechanism in computer vision and natural language processing [4, 5, 6], [7] designed the session encoder purely based on attention and obtained state-of-the-art performance. Compared with RNN-based approaches, STAMP [7] is able to capture dependencies between any two items in the same session effectively with the attention mechanism. However, the high-level sequential properties in sessions are overlooked, which can be problematic in some situations.

The above two kinds of approaches of designing the encoder component are two extremes, since they either assume a rigid order between users' actions within sessions, or completely discard the ordering information. It is easy to understand that the ordering information is useful because the session data is sequential. However, in some *local regions* of sessions (called sub-sessions), the order of items may not be important because users do not intend to click items by following a strict order. In other words, sometimes, at the sub-sessional level, it is the occurrence of items that is important, rather than the order.

To tackle the above challenge, we propose a novel model called LINet that takes into account *Local Invariance* by paying different amount of attention to the sequential information at different levels. Specifically, the insignificant detailed ordering information in some sub-sessions is ignored, while the high-level sequential information of the whole sessions is preserved. The high-level idea is to use a full self-attention layer with Gaussian weighting to extract position-invariant features in sub-sessions, and then employ a RNN with the attention mechanism to capture the general sequential information.

The main contributions of this paper are summarized as follows:

- We identify and study the *local invariance* property in session-based recommendation. To the best of our knowledge, we are the first to consider this in the context of session-based recommendation.
- We propose a novel deep learning-based model that takes into account the local invariance property. Unlike previous models that either assume a rigid order between items within sessions, or completely discard the ordering information, our model learns automatically to generate session representations that are invariant to subtle position changes in sub-sessions by following the general order of items.
- Extensive experiments conducted on real-world datasets show that our model outperforms the state-of-the-art methods and the proposed mechanism to capture the sequential information with local invariance plays an important role.

II. Methodology

A. Problem Formulation

Let $V = \{v_1, v_2, \cdots, v_{|V|}\}$ denote the set of all unique items involved in all sessions. A session $s = [v_{s,1}, v_{s,2}, \cdots, v_{s,|s|}]$ is a sequence of items ordered by timestamp, where $v_{s,t} \in V$ denotes the item clicked at time step t in session s. The task of session-based recommendation is to predict the next item, i.e., $v_{s,t+1}$, given a sequence of previously clicked items $s_t = [v_{s,1}, v_{s,2}, \cdots, v_{s,t}]$ for each time step $t, 1 \leq t < |s|$. A typical session-based recommendation model usually computes a probability distribution $p(v|s)$ over the entire item set V. The items with top-k probabilities will be in the candidate set for recommendation.

B. Overview

Like previous state-of-the-art methods [1, 2, 3, 7], our model follows an encoder-predictor architecture.

Let $\mathbb{V} = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{|V|}\}$ denote the embedding vectors with respect to the item set V. Our model learns a d-dimensional real-valued embedding $\mathbf{v}_i \in \mathbb{R}^d$ for each item $v_i \in V$. Given a session sequence $s_t = [v_{s,1}, v_{s,2}, \cdots, v_{s,t}]$, the input to our model is a list of item embeddings $[\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_t]$, where $\mathbf{x}_i$ is the embedding vector of item $v_{s,i}$. The local encoder takes as input the list of item embeddings and generates a sequence of group representations that is invariant to position changes in short sub-sessions. The global encoder is a sequential model that extracts a session representation $\mathbf{c}_h$ from the sequence of group representations. Since [1] and [7] have demonstrated the importance and effectiveness of considering recent interests, we concatenate $\mathbf{c}_h$ and a vector $\mathbf{m}_t$ that represents the recently clicked items to form a hybrid representation $\mathbf{c}$. Finally, the predictor computes a probability distribution of the next item over V from the item embeddings and the session representation.

In the follow, we discuss in detail each component of our model.

C. Local Encoder

The objective of the local encoder is to extract group features that are invariant to position changes inside each group.

Ideally, each group is represented by a group embedding, which is a weighted sum of the embeddings of the items inside the group. For example, a group g can be represented by:

$$\mathbf{x}'_g = \sum_{v \in G} w_{g,v} \mathbf{v} \tag{1}$$

where G is the set of items in group g and $w_{g,v}$ is the contribution of item v towards group g.

However, this ideal objective cannot be achieved since the number of groups are not known. Furthermore, the groups may not have hard borders. Therefore, we extract a group embedding for each original item. The group embedding corresponding to the i^{th} item is defined as:

$$\mathbf{x}_i' = \sum_{1 \le j \le t} w_{ij} \mathbf{x}_j \tag{2}$$

which is simply a weighted sum of all items. The most important part w_{ij} is explained as follows.

Intuitively, groups are formed by similar adjacent items, which means w_{ij} is larger for items more similar and temporally closer to item i. Thus, we define w_{ij} as follows:

$$w_{ij} \propto \alpha_{ij}^{(l)} \cdot f_i(|i-j|) \tag{3}$$

$$\sum_j w_{ij} = 1 \tag{4}$$

$\alpha_{ij}^{(l)}$ is the importance score of $\mathbf{x}_j$ defined using the attention mechanism, which can be viewed as some kind of similarity.

$$e_{ij}^{(l)} = \mathbf{v}_l^T \tanh(\mathbf{W}_l[\mathbf{x}_i, \mathbf{x}_j]) \tag{5}$$

$$\alpha_{ij}^{(l)} = \frac{\exp(e_{ij}^{(l)})}{\sum_k \exp(e_{ik}^{(l)})} \tag{6}$$

(Note: Symbol *l* in notations $e_{ij}^{(l)}, \alpha_{ij}^{(l)}, \mathbf{v}_l, \mathbf{W}_l$ means *local.*)

$f_i(\cdot)$ is the probability density function of a Gaussian distribution with mean 0 and variance σ_i^2. Observe that the variance controls the group size, and groups are formed by similar items. The more similar the adjacent items to $\mathbf{x}_i$, the larger the group size, and more weights should be assigned to adjacent items, and the larger the variance. In other words, the variance is positively correlated to the average similarity between $\mathbf{x}_i$ and its adjacent items. For simplicity, we assume a linear relationship. Thus, the variance is defined as:

$$\sigma_i^2 = k \cdot \frac{1}{2m} \sum_{l, 0 < |i-l| \le m} \mathrm{sim}(\mathbf{x}_i, \mathbf{x}_l) \tag{7}$$

where k is a parameter, and $\mathrm{sim}(\cdot, \cdot)$ is a similarity measure, for which a common choice is cosine similarity. We estimate the average similarity in a context of size m. m should be large enough for an accurate estimation. However, a large m introduces the risk of including the items not likely to be in the group. We suggest setting m to $1 \sim 3$ according to our experiments.

Consider Equation (3). With $\alpha_{ij}^{(l)}$ alone (i.e., if $w_{ij} = \alpha_{ij}^{(l)}$), w_{ij} are actually the same as the weights of a full self-attention layer. Therefore, $\alpha_{ij}^{(l)}$ makes $\mathbf{x}_i'$ a contextualized embedding of $\mathbf{x}_i$. By adding $f_i(|i-j|)$, we make $\mathbf{x}_i'$ focus on a local region. Thus, $\mathbf{x}_i'$ can also be viewed as a local context embedding, and the local region it focuses on is dynamically decided by the group size.

From the perspective of signal processing, the local encoder can also be viewed as a smoothing filter. Similar idea can be found in [8], where shift invariance in convolutional networks is improved by using a smoothing filter. We achieve local position invariance by dynamically adjusting the weights of the filter according to the context.

D. Global Encoder

The global encoder encodes the group features $[\mathbf{x}_1', \mathbf{x}_2', \cdots, \mathbf{x}_t']$ into a session embedding that represents the user's interests in the current session. As the order of groups indicates changes of the user's current focus, the relative order contains useful information for the prediction of the next item. Therefore, we employ a RNN to generate the session representation.

Let the output states of RNN be $[\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_t]$, where $\mathbf{h}_i$ can be viewed as the representation of the previous i groups. To better capture the user's interests, we apply an item-level attention mechanism to dynamically select and linearly combine the representations of previous groups. Specifically, we extract a session representation $\mathbf{c}_h$ as a weighted sum of the output states of RNN:

$$\mathbf{c}_h = \sum_i \alpha_i^{(g)} \mathbf{h}_i \tag{8}$$

where $\alpha_i^{(g)}$ are computed using the attention mechanism:

$$e_i^{(g)} = \mathbf{v}_g^T \tanh(\mathbf{W}_g[\mathbf{h}_i, \mathbf{h}_t]) \tag{9}$$

$$\alpha_i^{(g)} = \frac{\exp(e_i^{(g)})}{\sum_k \exp(e_k^{(g)})} \tag{10}$$

(Note: Symbol g in notations $e_i^{(g)}, \alpha_i^{(g)}, \mathbf{v}_g, \mathbf{W}_g$ means *global.*)

With the combination of the local and global encoders, the session representation $\mathbf{c}_h$ contains the general sequential information of the entire session and is invariant to the unimportant local position changes in some sub-sessions.

E. Predictor

The predictor evaluates the probability distribution of the next item. Since [7] have demonstrated that explicitly considering user's recent interests is effective for predicting the next item, we combine the session representation $\mathbf{c}_h$ with the user's recent interests $\mathbf{m}_t$ (for simplicity, we set $\mathbf{m}_t = \mathbf{x}_t$) into a hybrid representation $\mathbf{c}$ for the current session. Then, the score of a candidate item $v_i \in V$ is defined as:

$$\mathbf{z}_i = \mathbf{v}_i^T g(\mathbf{c}) \tag{11}$$

where $\mathbf{v}_i$ is the embedding of v_i and g is a neural network that transforms $\mathbf{c}$ into a vector with the same dimensionality as $\mathbf{v}_i$.

The scores are then normalized using the *softmax* function to obtain a probability distribution over all items in V:

$$\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{z}) \tag{12}$$

F. Training

For each session sequence s_t, the loss function is defined as the cross-entropy of the prediction and the ground truth:

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i} \mathbf{y}_i \log \hat{\mathbf{y}}_i \tag{13}$$

where $\mathbf{y}$ is a one-hot vector representing the ground-truth distribution for the next item.

Then, all parameters as well as the item embeddings are learned in an end-to-end back-propagation training paradigm.

III. Experiments and Analysis

A. Datasets

We carried out our experiments on two real-world benchmark datasets for session-based recommendation. The first one is the *YooChoose*[1] dataset from RecSys Challenge 2015, which consists of users' click-streams on an e-commerce website within 6 months. The other dataset called *Diginetica*[2] comes from CIKM Cup 2016, where only its transactional data is used. Models need to predict the next item that a user would like to click/buy given the user's click/purchase history in the on-going session. As verified in [9], since the YooChoose training set is quite large and training on the recent fractions yields better results than training on the entire fractions, we only used the most recent fractions 1/64 and 1/4 of the training sessions. Therefore, we generated two datasets from the *YooChoose* dataset. We applied the same preprocessing steps on the datasets as in [1, 2, 7].

B. Compared Models and Evaluation Metrics

Conventional methods: We select the following conventional methods which are commonly used as baselines in session-based recommendation [3].

- Item-KNN recommends items similar to the previous items in the current session, where the similarity is defined as the cosine similarity between the session vectors.
- BPR-MF [10] optimizes a pairwise ranking objective function via stochastic gradient descent.
- FPMC [11] is a state-of-the-art hybrid model for next-basket recommendation.

Neural network based methods: The following models are recent state-of-the-art methods for session-based recommendation.

- GRU4Rec [12] employs ranking-based loss during the session-parallel mini-batch training process.
- NARM [1] employs RNNs with attention to capture users' main purposes and sequential behaviors.
- STAMP [7] purely uses the attention mechanism to capture users general interests and the recent focus.
- RepeatNet [2] considers the repeat-consumption phenomenon using a repeat-explore mechanism.
- SR-GNN [3] converts sessions into graphs and uses a graph neural network to learn session representations.

Following the previous work [1, 2, 3, 7], we use two commonly used metrics **Hit@20** and **MRR@20** to evaluate the performance of models.

- Hit@20: the fraction of test samples in which the desired next item is ranked among the top 20 positions.
- MRR@20: MRR (Mean Reciprocal Rank) is the average of reciprocal ranks of the actual next items. The reciprocal rank is set to 0 if the rank is larger than 20.

C. Comparison with Existing Methods

The performance of all methods is reported in Table I.

Table I
Experimental results (%) on three datasets

Methods	YooChoose 1/64		YooChoose 1/4		Diginetica	
	Hit@20	MRR@20	Hit@20	MRR@20	Hit@20	MRR@20
Item-KNN	51.60	21.81	52.31	21.70	35.75	11.57
BPR-MF	31.31	12.08	3.40	1.57	5.24	1.98
FPMC	45.62	15.01	51.86	17.50	26.53	6.95
GRU4Rec	60.64	22.89	59.53	22.60	40.24	14.37
NARM	68.32	28.76	69.73	29.23	49.70	16.17
STAMP	68.74	29.67	70.44	30.00	48.66	15.55
RepeatNet	69.13	30.24	70.71	31.03	47.79	**17.66**
SR-GNN	70.57	30.94	71.36	31.89	49.90	16.31
LINet	**71.23**	**31.12**	**71.89**	**32.03**	**51.74**	17.53

As mentioned in [2], the scores on the Diginetica dataset differ from the results reported in the original papers [1, 3, 7, 12] because they did not sort the session items according to the "timeframe" field, which means the sequential information is ignored. We ran the code released by [1, 3, 7, 12] on the sorted sessions to obtain the correct scores.

Conventional methods perform poorly compared to neural network models, which proves that conventional methods are no longer suitable for session-based recommendation. One reason for their poor performance is that they consider no or limited sequential information. GRU4Rec, a simple one-layer RNN that can leverage the sequential information of entire sessions, already outperforms all the conventional methods. Therefore, making use of complete sequential information of sessions is essential for session-based recommendation. Nevertheless, we do not necessarily follow a strict order. STAMP completely ignores the ordering information in sessions except for the last item, and SR-GNN may discard some sequential information such as the starts and ends of sessions when encoding sessions into graphs, but the two methods still produce competitive results.

However, the performance of these methods are still inferior to that of the proposed method LINet. The three RNN-based methods GRU4Rec, NAMR and RepeatNet assume a strict order in items of each session, meaning that they can be easily misled by the insignificant ordering in

[1] http://2015.recsyschallenge.com/challenge.html

[2] http://cikm2016.cs.iupui.edu/cikm-cup

local regions. STAMP and SR-GNN blindly discard some important sequential information. On the contrary, LINet can automatically seek a good balance between following the strict order and completely ignoring the order. As shown in Table I, LINet outperforms all the state-of-the-art methods in the YooChoose 1/64 and YooChoose 1/4 datasets, and gains a significant improvement in terms of the metric Hit@20 in all datasets, thus proving the effectiveness and validity of the proposed model.

D. Ablation Experiments

To test the effectiveness of the central modules in the proposed model, i.e., the local and global encoders, we propose and evaluate the following four models:

- LINet-No-LE: LINet with the local encoder removed.
- LINet-No-GE: LINet with the global encoder removed. The average of the extracted group features are treated as the context embedding $\mathbf{c}_h$.
- LINet-No-AW: LINet without the attention weights, i.e., $w_{ij} = f_i(|i-j|)$.
- LINet-No-GW: LINet without the Gaussian weights, i.e., $w_{ij} = \alpha_{ij}^{(l)}$.

Table II
THE PERFORMANCE OF THE COMPLETE AND IN-COMPLETE VERSIONS OF THE PROPOSED MODEL

Method	YooChoose 1/64		Diginetica	
	Hit@20	MRR@20	Hit@20	MRR@20
LINet-No-LE	68.36	28.86	49.54	16.32
LINet-No-GE	68.58	29.27	48.69	15.89
LINet-No-AW	70.13	30.65	50.91	16.84
LINet-No-GW	70.09	30.67	50.63	16.57
LINet	71.23	31.12	51.74	17.53

The results are shown in Table II. We can see that the two in-complete models with either encoder removed have significant performance drops on both datasets compared with the complete model, proving that both the local and global encoders play an important role in the proposed model. With the local encoder removed, LINet-No-LE captures the sequential information by following a strict order without considering the local invariance property, so it may be misled by some useless local ordering and produce wrong predictions. With the global encoder removed, LINet-No-GE merely captures the contextual information in local regions, not being able to consider long-term dependencies between items and utilize the valuable sequential information. Therefore, it is necessary for a session-based recommender to both capture the sequential properties in sessions and consider the local invariance property.

The other two in-complete models with either the attention weights or the Gaussian weights removed also perform worse than the complete model, though the gaps are smaller. Without the attention weights, LINet-No-AW does not consider similarity when computing the scores of adjacent items, so the group embeddings can be easily affected by dissimilar outliers. Without the Gaussian weights, LINet-No-GW simply computes the contextualized embeddings of items, not focusing on local regions. Therefore, it is important for the local encoder to have both kinds of weights when the group embeddings are computed.

E. Capability of Considering Local Invariance

We designed an additional experiment to test the models' capability of considering the local invariance property. The idea is to evaluate their performance on similar pairs of session sequences.

First, we define the similarity between two items using only the training sessions as follows:

$$\mathrm{sim}(v_i, v_j) = \frac{1}{\sum_d n_d^{(ij)}} \sum_d \frac{n_d^{(ij)}}{\log_2(d+1)} \tag{14}$$

where $n_d^{(ij)}$ is the number of training sessions where the distance between v_i and v_j is d, and $\log_2(d{+}1)$ is a term that penalizes long distance, which we borrow from *discounted cumulative gain* [13]. The intuition behind is that $\mathrm{sim}(v_i, v_j)$ measures the probability that v_i and v_j are in the same group. The closer the two items, the higher the probability. A special case is when v_i and v_j are next to each other in every session they co-occur. If we assume their relative order no longer matters in this case, then v_i and v_j are in the same group with probability 1.

We only choose the neural network based methods for comparison because as shown in Table I, methods based on neural network are consistently better than the conventional methods in terms of both evaluation metrics on all datasets. Furthermore, we exclude GRU4Rec and STAMP because NARM can be viewed as an improved version of GRU4Rec and STAMP does not consider any order inside sessions. Therefore, the methods we test are NARM, RepeatNet, SR-GNN, and LINet.

Given a session sequence s_i, let $s_{i,k}$ denote the k^{th} item in s_i, and $s_{i,1:k}$ denote the first k items in s_i. We extract all pairs of session sequences (s_i, s_j) from the test sets that satisfy the following conditions.

1) s_i and s_j have the same length, denoted as l.
2) s_i and s_j have the same last item, i.e., $s_{i,l} = s_{j,l} = v$, so a model needs to predict the next item as v given an input sequence $s_{i,1:l-1}$ or $s_{j,1:l-1}$.
3) There exists a perfect matching M in the bipartite graph G formed by $s_{i,1:l-1}$ and $s_{j,1:l-1}$. Specifically, the items in $s_{i,1:l-1}$ and $s_{j,1:l-1}$ are the two groups of nodes of G, and there is an edge with weight = $\mathrm{sim}(u, v)$ defined by Equation (14) for each pair of (u, v), where $u \in s_{i,1:l-1}$ and $v \in s_{j,1:l-1}$. Furthermore, we require that $\mathrm{sim}(u, v)$ is larger than a threshold θ, and the difference in indices of u and

v is smaller than another threshold β. The reasons are explained below.

Given a pair (s_i, s_j) satisfying all the conditions, we can reorder the first $l-1$ items of one sequence, say s_i, such that the matched items have the same indices. E.g., if $s_{i,1}$ is matched to $s_{j,2}$, $s_{i,1}$ is put at position 2. Let the reordered sequence be $s'_{i,1:l-1}$. If θ is large, the corresponding items of $s'_{i,1:l-1}$ and $s_{j,1:l-1}$ at the same index are very similar, so $s'_{i,1:l-1}$ and $s_{j,1:l-1}$ are very similar. Since $s_{j,1:l-1}$ has a next item $s_{j,l}$, it is likely that given the input sequence $s'_{i,1:l-1}$, the next item is $s_{j,l} = s_{i,l}$ (by Condition 2). Therefore, it is probable that $s'_{i,1:l-1}$ has the same next item as $s_{i,1:l-1}$. If β is small, then the only differences between $s'_{i,1:l-1}$ and $s_{i,1:l-1}$ are some small position changes at the sub-sessional level. and we can conclude that the changes have no effect on the next item. Thus, we can say s_i has a large local invariance. Similarly, we can say it for s_j.

Therefore, the extracted pairs of session sequences form a dataset with large local invariance. Besides, the average weight in the matching M can be used as a good quantitative measure of the local invariance of s_i or s_j. The larger the average weight of M, the larger the local invariance.

We then evaluate each model by the percentage of pairs such that the model ranks the next items among top-k for both sequences in the pairs. For example, given a pair (s_i, s_j), we count the pair only if for both sequences, the ground-truth next item is among the top-k of the prediction. Thus, in order to have a good performance, a model needs to produce consistent and accurate predictions for both sequences. As in previous experiments, k is to 20 in this experiment. The results are shown in Figure 1.

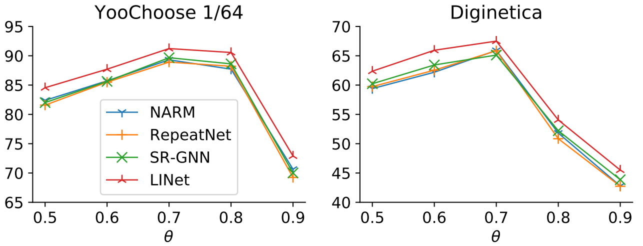

Figure 1. The percentage of similar pairs that each model predicts accurately vs the threshold θ

We can see that the comparative models have similar performance and the proposed model consistently outperforms other models in both datasets. This further proves that our model has a higher capability to handle the local invariance property in sessions. Observe that the performance of all models drops when θ is large. The sequence length may be a possible reason. For a pair of long sequences, it is less likely that Condition 3 is satisfied because more items are involved. Therefore, the sequences in the pairs extracted at a large θ have short lengths, which means they contain less sequential information and thus are harder to predict.

IV. Conclusion

We propose a model with an encoder-predictor architecture to address the *local invariance* property in session-based recommendation. The local encoder incorporates Gaussian weighting into self-attention, capturing contextual information in local regions, and the global encoder extracts high-level sequential information in the whole sessions. With the combination of both, our model can pay different attention to the sequential information in different levels of granularity, and generate session representations that are invariant to subtle position changes in subsequences. Comprehensive experiments conducted on two public benchmark datasets demonstrate the superiority of the proposed model over the state-of-the-art models.

Acknowledgements: We are grateful to the anonymous reviewers for their constructive comments on this paper. The research is supported by ITS/227/17FP.

References

[1] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, “Neural attentive session-based recommendation,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1419–1428.

[2] P. Ren, Z. Chen, J. Li, Z. Ren, J. Ma, and M. de Rijke, “RepeatNet: A repeat aware neural recommendation machine for session-based recommendation,” *National Conference on Artificial Intelligence*, 2019.

[3] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, “Session-based recommendation with graph neural network,” *National Conference on Artificial Intelligence*, 2019.

[4] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” *International Conference on Machine Learning*, pp. 2048–2057, 2015.

[5] T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Neural Information Processing Systems*, pp. 5998–6008, 2017.

[7] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, “STAMP: Short-term attention/memory priority model for session-based recommendation,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 1831–1839.

[8] R. Zhang, “Making convolutional networks shift-invariant again,” in *ICML 2019 : Thirty-sixth International Conference on Machine Learning*, 2019, pp. 7324–7334.

[9] Y. K. Tan, X. Xu, and Y. Liu, “Improved recurrent neural networks for session-based recommendations,” in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 2016, pp. 17–22.

[10] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “BPR: Bayesian personalized ranking from implicit feedback,” *Uncertainty in Artificial Intelligence*, pp. 452–461, 2009.

[11] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, “Factorizing personalized markov chains for next-basket recommendation,” in *Proceedings of the 19th International Conference on World Wide Web*, 2010, pp. 811–820.

[12] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, “Session-based recommendations with recurrent neural networks,” *International Conference on Learning Representations*, 2016.

[13] K. Jrvelin and J. Keklinen, “Cumulated gain-based evaluation of IR techniques,” *ACM Transactions on Information Systems*, vol. 20, no. 4, pp. 422–446, 2002.