STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation

Qiao Liu
University of Electronic Science and Technology of China
Chengdu, China
qliu@uestc.edu.cn

Yifu Zeng
University of Electronic Science and Technology of China
Chengdu, China
ifz@std.uestc.edu.cn

Refuoe Mokhosi
University of Electronic Science and Technology of China
Chengdu, China
refuoemokhosi@yahoo.com

Haibin Zhang
University of Electronic Science and Technology of China
Chengdu, China
herb.zhang@std.uestc.edu.cn

ABSTRACT

Predicting users' actions based on anonymous sessions is a challenging problem in web-based behavioral modeling research, mainly due to the uncertainty of user behavior and the limited information. Recent advances in recurrent neural networks have led to promising approaches to solving this problem, with long short-term memory model proving effective in capturing users' general interests from previous clicks. However, none of the existing approaches explicitly take the effects of users' current actions on their next moves into account. In this study, we argue that a long-term memory model may be insufficient for modeling long sessions that usually contain user interests drift caused by unintended clicks. A novel short-term attention/memory priority model is proposed as a remedy, which is capable of capturing users' general interests from the long-term memory of a session context, whilst taking into account users' current interests from the short-term memory of the last-clicks. The validity and efficacy of the proposed attention mechanism is extensively evaluated on three benchmark data sets from the RecSys Challenge 2015 and CIKM Cup 2016. The numerical results show that our model achieves state-of-the-art performance in all the tests.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; Recommender systems; • **Computing methodologies** → *Neural networks*;

KEYWORDS

Behavior modeling, Session-based recommendation, Attention model, Representation learning, Neural networks

KDD 2018, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3219950

ACM Reference Format:
Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In *KDD 2018: 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3219819.3219950

1 INTRODUCTION

Session-based Recommender systems (SRS) are an important component of modern commercial online systems, usually used for improving user experiences by making suggestions based on user behavior encoded in browser sessions, and the recommender's task is to predict users' next actions (click on an item) based on the sequence of the actions in the current session[5, 21].Recent studies have highlighted the importance of using recurrent neural networks (RNNs) in a wide variety of recommender systems, among which the application of RNNs in session-based recommendation tasks has led to significant progress in the past few years [6, 17]. Although RNN models have been proven useful in capturing users' general interests from a sequence of actions[20], learning to predict from sessions is still a challenging problem to tackle largely due to the uncertainty inherent in user behavior and the limited information provided by browser sessions[18].

Based on existing literature, almost all the RNN-based SRS models only consider modeling the session as a sequence of items, without explicitly taking into account that users' interests drift with time[6], which could be problematic in practice. For example, if a specific digital camera link has just been clicked by a user and recorded in a session, it is highly likely that the user's next intended action is *in response to* the current action. (1) If the current action is to browse the product description before making a purchase decision, then the user is very likely to visit another digital camera brand catalog in the next move. (2) If the current action is to add a camera into the shopping cart, then the user's browsing interest is likely be changed to other peripherals such as memory cards. In this case, to recommend another digital camera to that user would not be a good idea, albeit that the initial intention of this session is to buy a digital camera (as was reflected in the previous actions).

In typical SRS tasks, the session consists of a sequence of named items, and the user interests is hidden in these *implicit feedbacks*(e.g.,clicks). In order to further improve the predictive accuracy of the RNN models, it is important to have the ability to learn both

long-term interests and short-term interests of such implicit feedbacks. As Jannach et al. [7] noted that both the users' short-term and long-term interests are of great importance for recommendation, but traditional RNN architectures are not designed to distinguish and exploit these two types of interests simultaneously [11].

In this study, we consider to solve this problem by introducing a recent action priority mechanism into the SRS model, called Short-Term Attention/Memory Priority (STAMP) model, which can take into account the user's *interests in general* and his/her *current interests* simultaneously. In STAMP, the users' interests in general are captured by an *external memory* built from all the historical clicks in a session prefix (including the *last-click*), and this is where the term "Memory" enters. The term "last-click" denotes the last action (item) of a session prefix, the objective of SRS is to predict the "next click" with regard to this "last-click". In this study, the embedding of the *last-click* is used to represent the user's current interests, and the proposed attention mechanism is built on top of it. Since the *last-click* is a component of the *external memory* it can be regarded as short-term memory of the users' interests. Similarly, the users' attention built on top of the *last-click* can be seen as a short-term attention. To our knowledge, this is the first effort to simultaneously take the long/short term memory into account when constructing a neural attention model for session-based recommendations. The major contributions of this study are as follows:

- We introduce a short-term attention/memory priority model that learns: (a) a uniform embedding space with items across sessions and (b) a novel neural attention model for next-click prediction in session-based recommender systems.
- A novel attention mechanism is proposed for implementation of the STAMP model, in which the attention weights are calculated from the session context and being enhanced with the current interests of the users. The output attention vector is read as a compositional representation of the user's temporal interests, and is more sensitive to user's interests drift with time than other neural attention based solutions. Therefore, it is capable of simultaneously capturing both the users' long-term interests in general (in response to the initial purpose) and their short-term attention (current interests). The validity and efficacy of the proposed attention mechanism is verified through comparison studies.
- The proposed model is evaluated on two real world datasets, the Yoochoose dataset from RecSys 2015, and the Diginetica dataset from CIKM Cup 2016, respectively. Experimental results show that STAMP achieves state-of-the-art, and the proposed attention mechanism plays an important role.

2 RELATED WORK

Session-based recommendation is a subtask of recommender system, in which the recommendations are made according to the implicit feedbacks within the user session. This is a challenging task because the users are usually assumed to be anonymous, and the user preferences (such as ratings) are not provided explicitly, instead, only some positive observations (e.g. purchases or clicks) are available to the decision makers [4]. In the past few years, an increasing amount of research attention has been devoted to the challenge of SRS problem, according to their modeling hypothesis, prevalent approaches can be divided into two categorise: *global models* that focused on identifying users' interests in general [3], and *localized models* that emphasize users' temporal interests [19].

One approach of capturing user's general interests is through collaborative filtering (CF) methods based on users' whole purchase/click history. For example, the Matrix Factorization (MF) approach [8] uses latent vectors to represent general interests, which are estimated through factorizing a user-item matrix consisting of the whole historical transaction data. Another approach is called neighborhood methods [14], which try to make recommendations based on item similarities calculated from the co-occurrences of items in sessions. The third approach is the Markov chain (MC) based models[3, 15], which utilize sequential connections between the user actions to make prediction.

The above models explore either general interests or current interests of users. However, previous current interests based recommenders seldom consider the sequential interactions between items that are not adjacent in the session, although the general interests based recommenders are good at capturing users' general taste, but can hardly adapt its recommendations to users' recent purchases without explicitly modeling the adjacent connections [19]. Ideally, a good recommender should be able to explore the sequential behavior, as well as account for users' general interests for recommendation, because these two factors may interact with each other to influence users' next click. Therefore, some of the researchers tried to improve the SRS models by taking into consideration of both type of user interests. Rendle et al. [13] proposed a hybrid model FPMC, which combined the power of MF and MC to model both sequential behavior and general interests for next basket recommendation, thus achieve better performance than considering either short-term interests or long-term interests alone. Wang et al. [19] proposed a hybrid representation learning model, which employs a two-layer hierarchical structure for modeling of the sequential behavior and general interests of users from their last transactions. However, both of them can only model local sequential behaviours between adjacent actions, without considering the global information conveyed by the session context.

Deep neural networks have proven to be very effective in modeling sequential data recently [9]. Inspired by recent advances in natural language processing area [16], some deep learning based solutions have been developed and some of which represent the state-of-the-art in SRS research field [2, 5, 6, 10]. Hidasi et al.[5] use deep recurrent neural networks with a gated recurrent unit to model session data, which learns session representation directly from previous clicks in the given session and provides recommendations of the next action. This is the first attempt to apply RNN networks for solving the SRS problem, thanks to the sequential modeling capability provided by the RNNs, their model can take into account the users' historical behavior when making predictions of the next move. Tan et al.[17]propose a data augmentation technique to improve the performance of the RNNs for session-based recommendation. Yu et al.[20]propose a dynamic recurrent model, which applies RNN to learn dynamic representation for each basket for user general interests at different times and captures global sequential behavior among baskets.

Most neural network models mentioned above are implemented in SRS by manipulating each context clicked item with the same

operation, allowing the models to capture the relevance between next click and previous clicks in an implicit way. Also the hidden state in the last time step contains information about the sequence with a strong focus on the parts nearest to the next click[1], thus some general interest features of items with a long distance may be forgotten. To solve this problem, a variety of models are introduced to capture relevance between items and more accurate general interests. Hu et al.[6] propose a neural network with wide-in-wide-out structure (SWIWO) to learn user-session context. It constructs the session context via combing all the item embeddings in a current session, which gives each item a fixed weight based on the relative distance with response to the target item. Li et al.[10] propose an RNN based encoder-decoder model (NARM), which takes the last hidden state from the RNN as the sequential behavior, and uses the hidden states of previous clicks for attention computation to capture the main purpose(general interests) in a given session. Another recent related work is the Time-LSTM model[21] which is a variant of the LSTM. Time-LSTM considers both short-term interests and long-term interests by using time gates to control the influence of last consumed item and store time intervals to model users' long-term interest, however the time stamp is not provided in most real-world datasets, so it is not considered here.

Differences: Our model has significant differences with SWIWO and NARM. SWIWO determines the weight of each item in the session in a fixed manner, which we consider is arguable in practice. In STAMP, the proposed attention mechanism can help alleviate this contradiction by explicitly considering correlation between each historical click and the last click, and calculating dynamic weights for given session. Alternatively, NARM combines main purpose and sequential behavior to get the session representation which treats them as equally important complementary features. However, STAMP explicitly emphasizes the current interest reflected by the last click to capture the hybrid features of current and general interests from previous clicks, thus explicitly introducing the importance of last click into the recommender system while NARM only captures the general interests. Short-term interests can be enhanced in STAMP so as to accurately capture the current interest of the user in the case of interest drift, especially in a long session.

3 METHODS

3.1 Symbolic Description

A typical session-based recommender system is built upon historical sessions, and makes prediction based upon current user sessions. Each session, denoted by $S = [s_1, s_2, \ldots, s_N]$, consists of a sequence of actions (items clicked by the user), where s_i represents an item (ID) clicked at time-step i. $S_t = \{s_1, s_2, \ldots, s_t\}$, $1 \le t \le N$ denotes a prefix of the action sequence truncated at time t with regard to session S. Let $V = \{v_1, v_2, \ldots, v_{|V|}\}$ denotes a set of unique items in the SRS system, called *item dictionary*.

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{|V|}\}$ denote the embedding vectors with respect to the item dictionary V. The proposed STAMP model learns a d-dimensional real-valued embedding $\mathbf{x}_i \in \mathbb{R}^d$ for each of the item i in V. Specifically, symbol $\mathbf{x}_t \in \mathbb{R}^d$ represents the embedding of the last click s_t of the current session prefix S_t. The goal of our models is to predict the next possible click (i.e. s_{t+1}) based on given session prefix S_t. To be exact, our models are constructed and trained as a classifier that learns to generate a score for each of the candidates in item dictionary V, let $\hat{\mathbf{y}} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|V|}\}$ denote the output score vector, where $\hat{y}_i$ corresponds to the score of item v_i. After getting this prediction result, the elements in $\hat{\mathbf{y}}$ are ranked in descending order, and the items corresponding to the top-k scores are used for recommendation. For notational convenience, we define the trilinear product of three vectors as:

$$< a, b, c >= \sum_{i=1}^{d} a_i b_i c_i = \mathbf{a}^T (\mathbf{b} \odot \mathbf{c}) \tag{1}$$

where $a, b, c \in \mathbb{R}^d$, and $\odot$ denotes the Hadamard product, i.e. the element-wise product between two vectors $\mathbf{b}$ and $\mathbf{c}$.

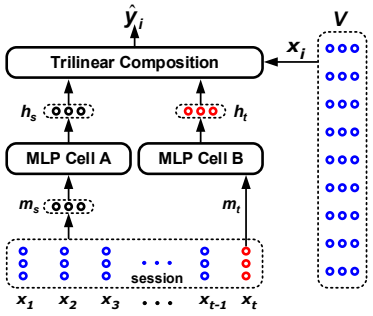

Figure 1: Schematic illustration of the STMP model.

3.2 The Short-Term Memory Priority Model

The proposed STAMP model is built upon a so-called Short-Term Memory Priority model (STMP), as illustrated in Figure 1.

From Figure 1 one can see that the STMP model takes two embeddings ($\mathbf{m}_s$ and $\mathbf{m}_t$) as inputs, where $\mathbf{m}_s$ denotes the user's interests in general with respect to the current session, which is defined as the average of the *external memory* of the session:

$$\mathbf{m}_s = \frac{1}{t} \sum_{i=1}^{t} \mathbf{x}_i \tag{2}$$

where the term *external memory* means the item embedding sequence of the current session prefix S_t. The symbol $\mathbf{m}_t$ denotes the current interests of the user in that session, in this study, the last-click $\mathbf{x}_t$ is used to represent the user's current interests : $\mathbf{m}_t = \mathbf{x}_t$. Since $\mathbf{x}_t$ is taken from the *external memory* of the session, we call it the *short-term memory* of the user's interests. The general interests $\mathbf{m}_s$ and current interests $\mathbf{m}_t$ are then processed with two MLP networks for the purpose of feature abstraction. The network structure of the MLP cells illustrated in Figure 1 are identical to each other, except that they have independent parameter settings. A simple MLP without hidden layer is used for feature abstraction, the operation on $\mathbf{m}_s$ is defined as:

$$\mathbf{h}_s = f(\mathbf{W}_s \mathbf{m}_s + \mathbf{b}_s) \tag{3}$$

where $\mathbf{h}_s \in \mathbb{R}^d$ denotes the output state, $\mathbf{W}_s \in \mathbb{R}^{d \times d}$ is a weighting matrix, and $\mathbf{b}_s \in \mathbb{R}^d$ is the bias vector. $f(\cdot)$ is a non-linear activation function (we use tanh in this study). The state vector $\mathbf{h}_t$ with regard

to $\mathbf{m}_t$ can be calculated similar to $\mathbf{h}_s$. And then, for a given candidate item $\mathbf{x}_i \in V$, the score function is defined as:

$$\hat{\mathbf{z}}_i = \sigma(< \mathbf{h}_s, \mathbf{h}_t, \mathbf{x}_i >) \tag{4}$$

where $\sigma(\cdot)$ denotes the sigmoid function. Let $\hat{\mathbf{z}} \in \mathbb{R}^{|V|}$ denote the vector that consists of the trilinear products $\hat{\mathbf{z}}_i$, in which each $\hat{\mathbf{z}}_i$ ($i \in [1, \ldots, |V|]$) represents the *unnormalized* cosine similarity between the representation of the weighted user interests with regard to the current session prefix S_t and the candidate item $\mathbf{x}_i$. Then it is processed by a softmax function to obtain the output $\hat{\mathbf{y}}$:

$$\hat{\mathbf{y}} = softmax(\hat{\mathbf{z}}) \tag{5}$$

where $\hat{\mathbf{y}} \in \mathbb{R}^{|V|}$ denotes the output vector of the model, which represents a probability distribution over the items $v_i \in V$, each element $\hat{\mathbf{y}}_i \in \hat{\mathbf{y}}$ denotes the probability of the event that item v_i is going to appear as the next-click in this session.

For any given session prefix $S_t \in S$ ($t \in [1, \ldots, N]$), the loss function is defined as the cross-entropy of the prediction results $\hat{\mathbf{y}}$:

$$\mathcal{L}(\hat{\mathbf{y}}) = -\sum_{i=1}^{|V|} \mathbf{y}_i log(\hat{\mathbf{y}}_i) + (1 - \mathbf{y}_i) log(1 - \hat{\mathbf{y}}_i) \tag{6}$$

where $\mathbf{y}$ denotes a one-hot vector exclusively activated by $s_{t+1} \in S$ (the ground truth). For example, if s_{t+1} denotes the i-th element v_i in item dictionary V, then $\mathbf{y}_k = 1$, if $i == k$, and $\mathbf{y}_k = 0$ if $i \neq k$. An iterative stochastic gradient descent (SGD) optimizer is then performed to optimize the cross-entropy loss.

From the definition of the STMP model (Equation 4) one can see that it makes predictions on the next-click based on the inner product of the candidate item and the weighted user interests, where the weighted user interests are represented through bilinear composition of the long-term memory (averaged historical clicks) and the short-term memory (the last-click). The validity of this trilinear composition model is verified in Section 4.5, the experimental results demonstrate that the proposed short-term memory priority mechanism can be very effective in capturing users' temporal interests that benefit the next-click prediction, and it achieves state-of-the-art performance on all the benchmark data sets.

However, as can be seen from Equation 2, when modeling the user's interests in general $\mathbf{m}_s$ from the *external memory* of the current session, the STMP model treats each item in the session prefix as equally important, which we consider would be problematic in capturing the user's interests drift (probably caused by unintended clicks), especially in case of long sessions. Therefore, we propose an attention model to tackle this problem — which has been demonstrated effective in capturing the attention drift in long sequences. The proposed attention model is designed based on the STMP model, and it follows the same idea as STMP in that it also gives priority to short-term attention, hence we call it the Short-Term Attention/Memory Priority Model (STAMP).

3.3 The STAMP Model

The architecture of the STAMP model is illustrated in Figure 2. As can be seen from Figure 2, the only difference between these two models is that in the STMP model the abstract feature vector of user's interests in general (the state vector $\mathbf{h}_s$) is calculated from the average of the external memory $\mathbf{m}_s$, while in STAMP model the $\mathbf{h}_s$ is calculated from an attention based user's interests in general

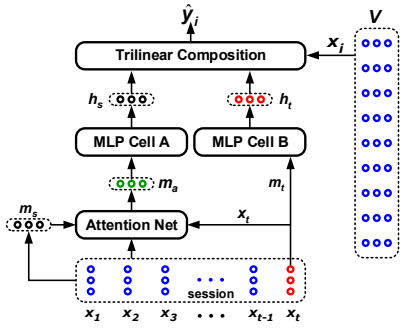

Figure 2: Schematic illustration of the STAMP model.

(a real-valued vector $\mathbf{m}_a$) as depicted in 2, which is produced by the proposed attention mechanism, called *attention net*.

The proposed attention net consists of two components: (1) a simple feed-forward neural network (FNN) that is responsible for generating attention weights for each of the items within the current session prefix S_t, and (2) an attention composite function that is responsible for calculating the attention based user's interests in general $\mathbf{m}_a$. The FNN used for attention computation is defined as:

$$\alpha_i = \mathbf{W}_0 \sigma(\mathbf{W}_1 \mathbf{x}_i + \mathbf{W}_2 \mathbf{x}_t + \mathbf{W}_3 \mathbf{m}_s + \mathbf{b}_a) \tag{7}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the i-th item $s_i \in S_t$, $\mathbf{x}_t \in \mathbb{R}^d$ denotes the last-click, $\mathbf{W}_0 \in \mathbb{R}^{1 \times d}$ is a weighting vector, $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3 \in \mathbb{R}^{d \times d}$ are weighting matrices, $\mathbf{b}_a \in \mathbb{R}^d$ is a bias vector, and $\sigma(\cdot)$ denotes the sigmoid function. α_i represents the attention coefficient of item $\mathbf{x}_i$ within the current session prefix. From Equation 7 one can see that the attention coefficients of the items in a session prefix are calculated based on the embedding of the target item $\mathbf{x}_i$, the last-click $\mathbf{x}_t$ and session representation $\mathbf{m}_s$, therefore, it is capable of capturing the correlations between the target item and the long/short term memory of the user's interests. Note that in Equation 7, the short-term memory is explicitly considered, which is distinctly different from the related works, and this is why the proposed attention model is called the short-term attention priority model.

After obtaining the attention coefficients vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_t)$ with respect to the current session prefix S_t, the attention based user's interests in general $\mathbf{m}_a$ with regard to the current session prefix S_t can be calculated as follows, and then add the $\mathbf{m}_s$ in it:

$$\mathbf{m}_a = \sum_{i=1}^{t} \alpha_i \mathbf{x}_i \tag{8}$$

3.4 The Short-Term Memory Only Model

To evaluate the validity of the basic idea of this study, that is, assigning a priority to the short-term attention/memory of the users' behavior when making decisions according to the session (sequence of actions), in this section, we propose a Short-Term Memory Only (STMO) model, which makes predictions of the next-click s_{t+1} only based on the last-click s_t of the current session prefix S_t.

Similar to the STMP model, a simple MLP without a hidden layer is used for feature abstraction in the STMO model. The MLP takes

the last-click s_t as input, and outputs a vector $\mathbf{h}_t \in \mathbb{R}^d$ just as the "MLP CELL B" in STMP (see Figure 1), defined as:

$$\mathbf{h}_t = f(\mathbf{W}_t \mathbf{x}_t + \mathbf{b}_t) \tag{9}$$

where $\mathbf{h}_t$ denotes the output state, $\mathbf{W}_t \in \mathbb{R}^{d \times d}$ is a weighting matrix, and $\mathbf{b}_t \in \mathbb{R}^d$ is the bias vector. $f(\cdot)$ denotes the activation function tanh. Then for a given candidate item $\mathbf{x}_i \in V$, the score function is defined as the inner product between $\mathbf{x}_i$ and $\mathbf{h}_t$:

$$\hat{\mathbf{z}}_i = \mathbf{h}_t^T \mathbf{x}_i \tag{10}$$

After obtaining the score vector $\hat{\mathbf{z}} \in \mathbb{R}^{|V|}$, one can make predictions based on the ranking list calculated with Equation 5, or optimize the parameters of the model based on Equation 6, just like the situation in STMP model.

4 EXPERIMENTS

4.1 Datasets and Data Preparation

We evaluate the proposed models on two datasets, the first one is called Yoochoose from the RecSys'15 Challenge [1], which consists of six months of click-streams gathered from an e-commerce web site, where the training set only contains session events. Another one is the Diginetica dataset coming from the CIKM Cup 2016[2], for which only the transaction data is used in this study.

Following [5] and [10], we filter out sessions of length 1 and items that appear less than 5 times in both of the datasets. The test set of Yoochoose consists of the sessions of subsequent days with respect to the training set, and we filter out clicks (items) that did not appear in the training set. And for Diginetica, the only difference is that we use the sessions of subsequent week for testing. After the pre-processing phase, there remains 7,966,257 sessions of 31,637,239 clicks on 37,483 items in Yoochoose dataset, and 202,633 sessions of 982,961 clicks on 43,097 items in Diginetica dataset.

Same as [17], we use a sequence splitting preprocess that for an input session $S = \{s_1, s_2, \ldots, s_n\}$, we generate the sequences and corresponding labels($[s_1]$, s_2), ($[s_1, s_2]$, s_3)... ($[s_1, s_2, \ldots, s_{n-1}]$, s_N) for training and testing on both datasets, which proves to be effective. Because the Yoochoose training set is quite large and training on the recent fractions yields better results than training on the entire fractions as per the experiments of [17], we use the recent fractions 1/64 and 1/4 of training sequences. The statistics of the three datasets are shown in Table 1.

4.2 Baselines

The following models, including the state-of-art and closely related work, are used as baselines to evaluate the performance of the proposed STAMP model :

- **POP**: A naive SRS model that always recommends items based on occurrence frequency in the training set.
- **Item-KNN**[14]: An item-to-item model which recommends items similar to the existing items based on cosine similarity between the candidate item and the existing items within the session. A constraint is included to avoid coincidental high similarities between rarely visited items as in [4, 20].

[1] http://2015.recsyschallenge.com/challege.html
[2] http://cikm2016.cs.iupui.edu/cikm-cup

Table 1: Statistics of the experiment datasets.

Dataset	Yoochoose 1/64	Yoochoose 1/4	Diginetica
# train	375,073	5,969,416	719,470
# test	55,898	55,898	60,858
# clicks	565,552	7,980,529	982,961
# items	17,694	30,660	43,097
avg. len.	6.16	5.71	5.12

- **FPMC**[13]: A state-of-the-art hybrid model for next-basket recommendation. In order to make it work on session-based recommendation, we do not consider the user latent representations when computing recommendation scores.
- **GRU4Rec**[5]: An RNN based deep learning model for session based recommendation, which consists of GRU units, it utilizes session-parallel mini-batch training process and also employs ranking-based loss functions during the training.
- **GRU4Rec+**[17]: A improved model based on GRU4Rec which adopts two techniques to improve the performance of GRU4Rec, including a data augmentation process and a method to account for shifts in the input data distribution.
- **NARM**[10]: An RNN based state-of-the-art model which employs attention mechanism to capture main purpose from the hidden states and combines it with the sequential behavior as final representation to generate recommendations.

4.3 Evaluation

We use the following metrics for evaluation of the performance of the SRS models, which are also widely used in other related works.

P@20: The P@K score is widely used as a measure of predictive accuracy in SRS area. P @ K represents the proportion of test cases which has the correctly recommended items in a top k position in a ranking list. In this paper, P@20 is used for all the tests, defined as:

$$P@K = \frac{n_{hit}}{N} \tag{11}$$

where N denotes the number of test data in the SRS system G, n_{hit} denotes the number of cases which have the desired items in top K ranking lists, a *hit* occurs when t appears in the top K position of the ranking list of G.

MRR@20: The average of reciprocal ranks of the desired item t. The reciprocal rank is set to zero if the rank is above 20.

$$MRR@K = \frac{1}{N} \sum_{t \in G} \frac{1}{Rank(t)} \tag{12}$$

The MRR is a normalized score of range $[0, 1]$, an increase in its value reflects that the majority "hits" will appear higher in the ranking order of the recommendation list, which indicates a better performance of the corresponding recommender system.

4.4 Parameters

The hyper-parameters are optimized via extensive grid search on all the data sets, and the best models are selected by early stopping based on the P@20 score on the validation set. Hyper-parameter ranges for the grid search are the following: embedding dimension d in $\{50, 100, 200, 300\}$, learning rate η in $\{0.001, 0.005, 0.01, 0.1, 1\}$,

Table 2: Next-click prediction on 3 benchmark data sets.

Datasets	Yoochoose 1/64		Yoochoose 1/4		Diginetica	
Measures	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
POP	6.71	1.65	1.33	0.30	0.91	0.23
Item-KNN	51.60	21.81	52.31	21.70	28.35	9.45
FPMC	45.62	15.01	–	–	31.55	8.92
GRU4Rec	60.64	22.89	59.53	22.60	43.82	15.46
GRU4Rec+	67.84	29.00	69.11	29.22	57.95	24.93
NARM	68.32	28.76	69.73	29.23	**62.58**	27.35
STMO	64.22	25.81	66.22	26.69	58.62	25.90
STMP	67.79	28.63	69.19	28.94	60.91	25.34
STAMP	**68.74**	**29.67**	**70.44**	**30.00**	62.03	**27.38**

learning rate decay λ in $\{0.75, 0.8, 0.85, 0.9, 0.95, 1.0\}$. According to the averaged performance, in this study we use the following hyper-parameters for all the tests on two datasets : $\{d : 100, \eta : 0.005, \lambda : 1.0\}$. The mini-batch settings are: batch size : 512, epoch : 30. All weighting matrices are initialized by sampling from a normal distribution $N(0, 0.05^2)$, and all biases are set to zeros. All the items embeddings are initialized randomly with a normal distribution $N(0, 0.002^2)$, which are then jointly trained with other parameters.

4.5 The Next-Click Prediction

To demonstrate the overall performance of the proposed model, we compare it with the state-of-the-art item recommendation approaches, and the numerical results on all of the benchmark data sets are illustrated in Table 2, in which the best result of each column is highlighted in boldface. As one can see from Table 2, STAMP achieves state-of-the-art performances in terms of P@20 and MRR@20 on both of the Yoochoose data sets and the Diginetica dataset, which verifies the efficacy and validity of the proposed model. The following observations can be made from table 2:

The performance of traditional methods such as Item-KNN and FPMC are not competitive, as they only outperform the naive POP model. These results help verify the importance of taking the user's behavior (interactions)into consideration in session-based recommendation tasks as the results show that making recommendations solely based on co-occurrence popularity of the items (POP), or simply taking transitions over successive items could be very problematic in making accurate recommendations. In addition, such global solutions can be time and memory consuming, making them not scalable to for large-scale datasets.

All of the neural network baselines significantly outperform conventional models, thus proving the effectiveness of deep learning technology in this field. GRU4Rec+ improves the performances of GRU4Rec by using the data augmentation techniques that split a single session into several sub-sessions for training. While GRU4Rec+ does not modify the model structure of GRU4Rec, they both only take the sequential behavior into account which may encounter difficulties with users' interest drift. NARM achieves the best performances among the baselines, because it not only models the sequential behavior using RNN with GRU units but also uses attention mechanism to capture main purpose, which indicates the importance of main purpose information in recommendations. This is reasonable as part of items in the current session may reflect the user's main purpose and relate to the next item.

Table 3: The results of P@K, MRR@K when K=5,10.

Model	Metrics	Yoochoose 1/64	Yoochoose 1/4	Diginetica
NARM	P@5	44.34	44.34	40.67
	MRR@5	26.21	26.08	25.02
STAMP	P@5	45.69	46.39	41.04
	MRR@5	27.26	27.47	25.21
NARM	P@10	57.50	57.83	51.91
	MRR@10	27.97	28.10	26.53
STAMP	P@10	58.07	59.62	52.07
	MRR@10	28.92	29.24	26.69

Among our proposed models, the STAMP model obtains the highest P@20 and MRR@20 on Yoochoose dataset in 2 experiments and achieves comparable results on the Diginetica dataset. The STMO model cannot capture general interest information from previous clicks in the current session, so it generates the same recommendation whenever it encounters the same last-click, although given different sessions. Unsurprisingly the model has the worst performance in our proposed models, since it cannot take advantage of the general interest information. But compared with traditional machine learning methods such as Item-KNN and FPMC, STMO achieves significantly better performances which demonstrates the ability of our proposed model framework to learn effective uniform item embedding representation. The STMP as an extension to STMO simply uses an average pooling function to generate session representation as the long-term interest and applies last-click information to capture short-term interest. It outperforms STMO in all three experiments and performs comparably with GRU4Rec+ but a little inferior to NARM. As expected, considering both session context information and last click information is suitable for this task as STMP is able to better make the session-based recommendations for a given session. Compared with STMP, STAMP applies item-level attention mechanism and achieves 0.95%, 1.25%, 1.12% improvements on P@20 and 1.04%, 1.06%, 2.04% on MRR@20 in three experiments respectively. The results show that the session representation generated in this way is more effective than average pooling function, which confirms that not all items in the current session are equally important in generating the next recommendation, and part of the important items can be captured by the proposed attention mechanism to model useful features of interest; the state-of-the-art results prove the validity of STAMP.

4.6 Compare STAMP with NARM

Session-based recommender systems have become an indispensable part of many e-commerce systems, helping users to sort out items of interest from large inventories. In fact, there are always more than 10^5 items in an e-commerce website and most users are only interested in viewing recommendations on the first page of real-world recommender systems [6]. In order to verify the performance of our proposed STAMP model and the recent state-of-the-art NARM model in real production environment, where recommendation systems can only suggest a few items at once, the relevant item should be amongst the first few items in the recommendation list[12]. We

Table 4: Runtime of each training epoch.

Method	Dataset	Time (seconds)
NARM	Yoochoose 1/64	155.3
	Yoochoose 1/4	961.4
	Diginetica	99.6
STAMP	Yoochoose 1/64	33.3
	Yoochoose 1/4	356.1
	Diginetica	52.0

therefore evaluate the recommendation quality in terms of P@5, MRR@5, P@10 and MRR@10 in trying to simulate the practical situation. The results are summarized in Table 3, and argue that the experimental results may to some extent reflect their performance in the real production environment. We can observe that STAMP performs well on this mission and much more competitively than NARM when evaluated under stricter rules in an simulated production environment. Our model performs consistently better than NARM and shows obvious advantages in three experiments which demonstrates the effectiveness of considering both general interests and short-term interests, and the validity of the learned item embeddings. The results prove that the proposed STAMP tends to make more accurate recommendations as seen in the above experimental results and the main results in subsection 4.5.

We also record the runtime of the recurrent neural model NARM and the proposed STAMP approach. We implement both models with the same 100-dimensional embedding vectors, and test them on the same GPU server. The training time of each epoch on three datasets is given in Table 4,which illustrates that STAMP is more efficient than NARM. We argue that this is because the NARM model contains a lot of complex operations in each GRU unit, and our proposed model is simpler and faster as it introduces a simplified neural model to save the cost of recurrent calculations in dealing with sequential inputs. All above results imply that STAMP may be more suitable for practical application since computational efficiency is crucial in real-world session-based recommender systems, which always comprise of large amounts of sessions and items.

4.7 Effects of the last click

In this section, we design a series of contrast models to verify the validity of applying the last click information on the basis of session context to make session-based recommendations:

- **STMP-**: On the basis of STMP, not using last click item embedding in the trilinear layer.
- **STMP**: The STMP model proposed in the paper.
- **STAMP-**: On the basis of STAMP, not using last click item embedding in the trilinear layer.
- **STAMP**: The STAMP model proposed in the paper.

The numerical results in Table 5 show that all the models in which the last click is combined with the session context vector have better performance than those without. The results prove that employing the last click positively contributes to recommendations of a given session. Our models are based on simultaneously capturing long-term and short-term interest and enhancing the last click information, which we believe is advantageous in handling

Table 5: Impacts of the last-click.

Datasets	Yoochoose 1/64		Yoochoose 1/4		Diginetica	
Measures	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
STMP-	60.59	21.70	62.92	24.52	57.20	21.55
STMP	67.79	28.63	69.19	28.94	60.91	25.34
STAMP-	65.19	24.95	67.96	26.67	60.15	24.47
STAMP	**68.74**	**29.67**	**70.44**	**30.00**	**62.03**	**27.38**

long sessions as users' interest may change during a long browsing period and the user's next action may be more related to last click that reflects a short-term interest. In order to verify the effects of last click, we investigate the P@20 with different session lengths and the results on Yoochoose 1/64 dataset are shown in Figure 3.

We first present experimental results varying the length of sessions on STMP, STAMP and NARM as shown by Figure 3(a). We can observe that when the length of sessions is above 20 the performance of NARM quickly decreases in contrast with STMP and STAMP. This suggests that short-term interests priority based models may be more powerful in handling long sessions than NARM. On the other hand, in Figure 3(b) we find that the P@20 results of STMP and STAMP when the lengths are between 1 to 30 are significantly higher than each corresponding model without feeding last click into the trilinear layer, respectively. The reason is that with current interests captured in last click or session representation, STMP and STAMP may better model the user interest for the next click recommendation. For longer sessions lengths, the performance margins between STMP- and STMP and between STAMP- and STAMP become larger. This proves that although it is important to capture general interests from the session context, explicitly taking advantage of temporal interests can enhance the quality of recommendations. Moreover, STAMP- outperforms STMP- which results from the hybrid interests captured by the attention mechanism in STAMP- while STMP- only considers the general interests;this demonstrates the importance of the last click information in the session-based recommendation task.

4.8 Comparison among Proposed Models

To further verify the efficacy and validity of different proposed models being those that capture user interests from only last click, those that combine last click with session context, and lastly those that apply attention mechanism; we compare the models by making comparison studies on different session lengths to show their performances and the advantages in different situations. To achieve this purpose, we partition sessions into two groups: 'Short' indicates that the length of sessions is 5 events or less while 'Long' represents sessions having more than 5 events, where 5 is almost an average length of total sessions in all original data sets. The statistics on the percentage of sessions belonging to Short group are 70.10% and 76.40%, and to Long are 29.90% and 23.60% for both test datasets of Yoochoose and Diginetica. For each approach, we compute the results of P@20 and MRR@20 for each length group on each data set. Experimental results are illustrated in Figure 4 (a) and (b) for Yoochoose and Diginetica respectively.

Figure 4 (a) shows the results on Yoochoose. We can see that all methods obtain lower P@20 and MRR@20 results in Long group in

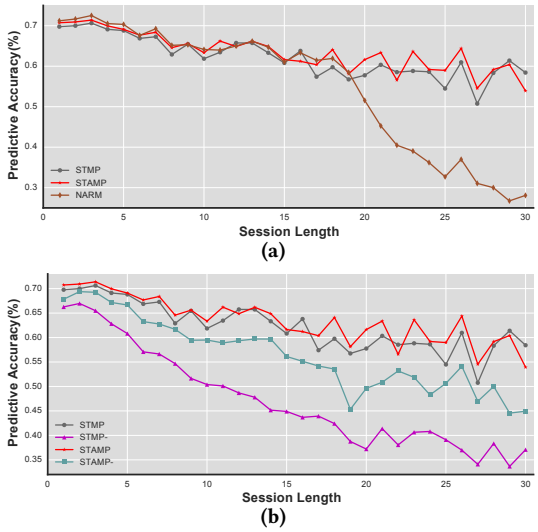

Figure 3: The P@20 evaluated on different lengths of sessions' test cases in Yoochoose 1/64.

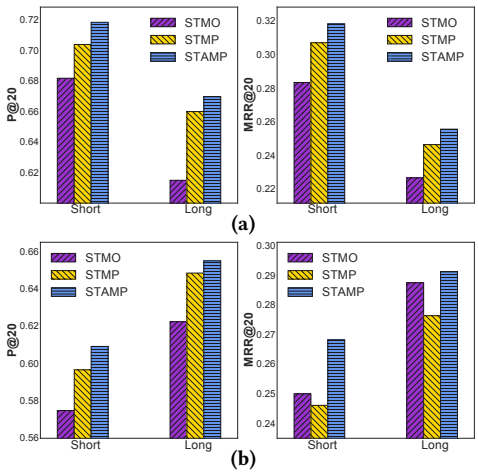

Figure 4: P@20 and MRR@20 of different session lengths. (a) on Yoochoose, (b) on Diginetica.

comparison to Short group, highlighting the challenge of making session-based recommendations for long sessions on this dataset. We suspect it may because of difficulty in capturing users' interest drift as the session grows in length. In addition both STMP and STAMP outperform STMO in two groups and the margin becomes wider as the session length increases, meaning that a model considering both general and current interests may be more powerful in handling long sessions, in comparison to only applying the last click information for recommendations. This confirms our intuition that session context and last click information can simultaneously and effectively be used to learn user interests and predict the next selected item in session-based recommendations.

Figure 4 (b)shows the results on Diginetica. STMO has better MRR@20 results than STMP, and the gap grows from 0.38% to 1.11% with increasing session length. This performance probably indicates that average aggregation in STMP has its disadvantages which influence the rank of correct items in recommendations, also the results of STMO may imply the validity of the short-term interests for making accurate recommendations. Overall, STAMP is still the best performing model which also highlights the need for effective session representation to obtain hybrid interests, this proves the advantages of the proposed attention mechanism.

Table 6: Statistics of sessions have repeated items.

Dataset	2	3	4	5	>5
Diginetica-train	0.1839	0.3272	0.4374	0.5229	0.7016
Diginetica-test	0.1880	0.3304	0.4420	0.5351	0.7149
YooChoose-train	0.1796	0.3298	0.4272	0.5091	0.7181
YooChoose-test	0.1770	0.3166	0.4139	0.5015	0.7563

Furthermore, Figure 4 shows that the trend between Short and Long group on the Yoochoose dataset is much different from that on the Diginetica dataset. To explain this phenomenon, we analyze the two datasets and show the ratio of sessions which have repeated clicks(i.e. the click appears at least twice within a session) in the two datasets with respect to the session length. From Table 6 we can see that the ratio of sessions which have repeated clicks in Yoochoose is smaller on Short group and larger on Long group than those in Diginetica dataset. From these results, we find that repeated clicks in the session have an impact on the recommendations, which have an inversely proportional ratio to model performance. It may be because repeated clicks may emphasize invalid information from unimportant items and make it difficult to capture user interests associated with the next action. In STAMP, we model the user interests using short-term attention priority, whereby the attention mechanism selects important items from the given session to model user interests. Both of these can effectively mitigate the impact of repeated clicks in a session. Conversely, only last click or average click information is used in other approaches, these models usually lose important information and are unable to overcome problems associated with repeated clicks. This proves the validity of short-term attention priority and the proposed attention mechanism.

4.9 Further Investigation

In this section, we repeatedly selected random multiple sets of examples from the Yoochoose test sets for analysis, and they consistently showed the same patterns. Figure 5 illustrates the attention results of the proposed item-level attention mechanism and its advantage.

In Figure 5, the depth of the color indicates the importance of an item, the darker the color the more important an item is. Because it's hard to directly evaluate the association between each context item and the target item in the absence of item specific information, the validity of the attention mechanism can be partially explained based on the category of an item. For example, in session 11255991 we can observe that the items which have the same category with the target item have larger attention weights than other items. The category of item can reflect the interest of the user to a certain extent, and the higher weight of the item with the same category as the target item can partially prove that the attention mechanism can capture user interests for the next action.

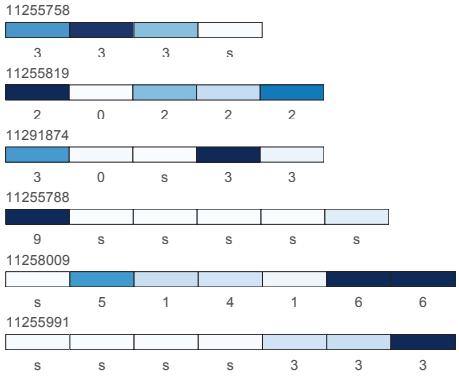

Figure 5: Attention visualization. Attention weights is used for color-coding, the depth of the color indicates the importance of an item. The numbers above the bar are session IDs, and the category ID of each item is given below the item.

Our method is capable of highlighting a number of factors in determining the next action as shown in Figure 5. Firstly, not all items are important in determining the next action and our method is able to pick important items and ignore unintended clicks. Secondly, although some important items are not near the current action in a session they can be flagged as important by our method, we believe that this demonstrates that our model is capable of capturing the users' interests in general in response to the initial or main purpose. Thirdly, items whose position is close to the end of the session often have larger weights, especially the last click item in a session with a long length. This proves our intuition that the user's intended action may be more in response to the current action. It shows that the proposed attention mechanism is sensitive to interests drift in a given session and correctly captures the current interests which is one of the reasons why STAMP can outperform other models which mainly focus on long-term interest. Moreover, the results illustrate that important items can be captured regardless of their position(i.e beginning or end of session) in a given session(e.g. session 11255788, 11255819). This proves our conjecture that the proposed item-level attention mechanism can capture pivotal items from a global perspective to construct hybrid features of general interests and current interests. Therefore based on the visualization results, we argue that the proposed item-level attention mechanism captures important parts for predicting next action in a session by computing attention weights, enabling the model to consider both long-term interest and short-term interest and make more accurate and effective recommendations.

5 CONCLUSION

In this paper, we propose a short-term attention/memory priority model for session-based recommendations. Two important findings can be made from the study: (1) The next move of a user is mostly affected by the last-click of a session prefix, and our model can effectively utilize such information through the temporal interests representation. (2) The proposed attention mechanism can effectively capture long-term and short-term interests of a session, empirical results prove that with the help of the attention mechanism, our model achieves state-of-the-art performance on all datasets.

6 ACKNOWLEDGMENTS

We thank the anonymous reviewers for taking time to read and make valuable comments on this paper. This work was supported by NSFC under grant 61133016 and 61772117, the General Equipment Department Foundation (61403120102), and the Sichuan Hi-Tech industrialization program (2017GZ0308).

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR'15*. CoRR, Scottsdale, USA.
[2] Hidasi Balázs, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel Recurrent Neural Network Architectures for Feature-rich Session-based Recommendations. In *Proceedings of ACM RecSys'16*. ACM, Boston, Massachusetts, USA, 241–248.
[3] Wanrong Gu, Shoubin Dong, and Zhizhao Zeng. 2014. Increasing recommended effectiveness with markov chains and purchase intervals. *Neural Computing and Applications* 25, 5 (2014), 1153–1162.
[4] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of ACM SIGIR'16*. ACM, Pisa, Italy, 549–558.
[5] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. In *Proceedings of ICLR'15* (May 2 - 4). CoRR, San Juan, Puerto Rico.
[6] Liang Hu, Longbing Cao, Shoujin Wang, Guandong Xu, Jian Cao, and Zhiping Gu. 2017. Diversifying Personalized Recommendation with User-session Context. In *Proceedings of IJCAI'17*. IJCAI, Melbourne, Australia, 1858 – 1864.
[7] Dietmar Jannach, Lukas Lerche, and Michael Jugovac. 2015. Adaptation and Evaluation of Recommendations for Short-term Shopping Goals. In *Proceedings of ACM RecSys'15* (September 16 - 20). ACM, Vienna, Austria, 211–218.
[8] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).
[9] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.
[10] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *Proceedings of ACM CIKM'17*. Singapore, Singapore, 1419–1428.
[11] Minh-Thang Luong, Hieu Pham, and Christopher D.Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of EMNLP'15* (September 17 - 21). Association for Computational Linguistics, Lisbon, Portugal, 1412–1421.
[12] Massimo Quadrana, Alexandros Karatzoglou, Hidasi Balázs, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. In *Proceedings of ACM RecSys'17*. ACM, Como, Italy, 130–137.
[13] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of WWW'10*. ACM, Raleigh, North Carolina, USA, 811–820.
[14] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of WWW'01*. ACM, 285–295.
[15] Guy Shani, David Heckerman, and Ronen I Brafman. 2005. An MDP-based recommender system. *JMLR* 6, Sep (2005), 1265–1295.
[16] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of NIPS'14* (December 08 - 13). MIT Press, Montreal, Canada, 3104–3112.
[17] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved Recurrent Neural Networks for Session-based Recommendations. In *Proceedings of DLRS'16* (September 15 - 15). ACM, Boston, MA, USA, 17–22.
[18] Bartlomiej Twardowski. 2016. Modelling Contextual Information in Session-Aware Recommender Systems with Neural Networks. In *Proceedings of ACM RecSys'16* (September 15 - 19). ACM, Boston, MA, USA, 273–276.
[19] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning Hierarchical Representation Model for NextBasket Recommendation. In *Proceedings of ACM SIGIR'15*. ACM, Santiago, Chile, 403–412.
[20] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A Dynamic Recurrent Model for Next Basket Recommendation. In *Proceedings of ACM SIGIR'16* (July 17 - 21). ACM, Pisa, Italy, 729–732.
[21] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2017. What to Do Next: Modeling User Behaviors by Time-LSTM. In *Proceedings of IJCAI'17* (August 19 - 25). IJCAI, Melbourne, Australia, 3602–360.