Knowledge-Based Systems 220 (2021) 106920

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

TPGN: A Time-Preference Gate Network for e-commerce purchase intention recognition

Yanan Liu [a], Yun Tian [a,*], Yang Xu [a], Shifeng Zhao [a], Yapei Huang [a], Yachun Fan [a], Fuqing Duan [a], Ping Guo [b]

[a] School of Artificial Intelligence, Beijing Normal University, Beijing, China
[b] School of Systems Science, Beijing Normal University, Beijing, China

ARTICLE INFO

Article history:
Received 29 May 2020
Received in revised form 28 February 2021
Accepted 2 March 2021
Available online 5 March 2021

Keywords:
Intention recognition
Purchase prediction
Recurrent neural network
LSTM
Online shopping data

ABSTRACT

The studies on users' purchase intentions based on e-commerce data are of great significance to marketers, buyers, and society. Current studies on users' intentions with traditional machine learning methods usually focus on unique features and are time-consuming. Due to the characteristics of user behaviors and the importance of time sequence, deep learning methods are increasingly applied in relevant studies. In the study, in order to predict online user's purchase intentions, based on Long-Short Term Memory (LSTM) model, we proposed Time-Preference Gate Network (TPGN). A pair of preference gates and a pair of time interval gates are added to the model. The preference gates are used to capture the users' category preferences at different time and the time interval gates are used to capture the users' long-term interest. In our model, through coupling the input gate with the forget gate, the parameters of the model are reduced and the performance is improved. In addition, the difference in user gender leads to the difference in the performance. Extended experiments with two real data sets confirmed that our model performed better than other baseline models.

1. Introduction

The studies on users' purchase intentions based on online shopping data are important for both merchants and customers. From related study results, merchants can deeply understand the demands of customers, improve the product recommendation interface and obtain multiple purchases of users. Based on related study results, users can discover new things, find their favorite products quickly, make confident purchase decisions, and gain the better purchase experiences [1,2].

In modern e-commerce, online shopping behaviors of users are multitudinous. For example, online shopping involves many steps such as browsing webpages and purchase. Some users firstly browse and click a commodity for many times, then add it into the shopping cart or mark as favorites, and finally buy it in the end. From the complex online shopping behaviors, some regular patterns can be obtained. Users' behaviors have been extensively explored [3–5] Users' behaviors contain a wealth of information on online purchase [6], such as consumption habits and the dynamics of preferences [7]. At present, users' intentions are generally explored through traditional machine learning methods, but most of existing studies are realized via acquiring special features. In traditional machine learning competitions at home and abroad, machine learning methods are commonly used to solve the problems of prediction and recommendation [8–11]. These methods rely heavily on the extracted features of research objects. In recent years, due to the characteristics of long-term and short-term interests of user behaviors, the methods based on deep learning models have been widely applied. Y. Zhu et al. [12] proposed a variant of LSTM, namely Time-LSTM, in which time gates were added into long short-term memory to simulate time relations and establish a time interval model for recommendation. T. Bai et al. [13] proposed a novel self-attentive Continuous-Time Recommendation model (CTRec) to dynamically capture the demands of users. They used the convolutional neural network (CNN) and the self-attention mechanism to respectively capture the short-term demand and the regular purchase cycle of the long-term demand. In existing methods, based on the consideration of the effect of time, long-term and short-term interests of users were captured, but the influences of users' preferences on purchase prediction were not considered. W. Gan et al. [14] extracted purchase behaviors with an encode-decode measure, but they did not consider the relations between time and items. P. Covington et al. [15] described a successful recommendation which used deep network to generate candidate and ranked. C. Li et al. [16] proposed a Behavior-to-Interest (B2I) dynamic routing model, however it ignored the item positions. N.

* Corresponding author.
E-mail address: tianyun@bun.edu.cn (Y. Tian).

https://doi.org/10.1016/j.knosys.2021.106920
0950-7051/

Zhu et al. [2] treated the user's category as users' intention, but this expression was not clear enough. Inspired by this idea [2], specific commodities are treated as the intentions of users in this study and the time features and category preference features of users are considered. In this way, the time-preference gate model is constructed to explore the user's intentions.

The experiments with real data sets showed that our method performed better than some current baseline methods. The contributions of the study are summarized below. Firstly, the Time-Preference Gate Network (TPGN) is proposed to solve users' intentions problem. A pair of time interval gates and a pair of preference gates are added to improve the performance of users. Secondly, another version of the model, named Time-Preference Gate Network (TPGN1), is proposed and the input gate is combined with the forget gate to decrease the parameters of the model. Thirdly, the prediction performance of the model is further improved.

2. Related works

The TPGN model solves the users' intentions problem through learning history series of users. This problem is the same to recommendation to some degree. The next-item recommendation problem has been explored by some scholars.

S. Wang et al. [17] designed an attention-based transactional context embedding model to realize the next-item recommendation. S. Zhang et al. [18] employed the self-attention mechanism to infer the relationship between the products from the users' real-time interaction. Based on the consideration of long-term intentions, the self-attention model could estimate the relative weight of each product and better represent the users' short-term interests. Z. Li et al. [7] proposed a behavior-intensive neural network (BINN) for the next-item recommendation, which involved the embedding of neural terms and discriminative behavior learning and could learn the historical preferences and current motivation of target users differently. F. Yuan et al. [19] proposed a network architecture consisting of a bunch of convolutional layers with holes and increased the receiving field without the pooling operation. T. Bai et al. [13] proposed a long-short demand model (LSDM), which exploited the users' long-term demands and different clusters and simulated each sequence of the cluster on a time scale with a recurrent neural network. Finally, they summarized the cooperative purchase demands on multiple time scales according to joint learning strategies.

In recent years, deep learning has achieved satisfactory results in user prediction. F. Yu et al. [20] proposed a Dynamic REcurrent bAsket Model DREAM, which could not only learn the dynamic representation of users, but also capture local sequential features in the shopping basket. Q. Cui et al. [21] designed a Hierarchical Contextual Attention-based GRU network (HCA-GRU) for Sequential Recommendation. Q. Cui et al. [22] proposed a multi-view recurrent neural network model MV-RNN, which could alleviate the cold start problem of goods by combining visual data with text information. Based on a recurrent neural network, M. Jiang et al. [23] proposed a terminal model to predict the next most probable user's song through the similarity. Based on unique features in the recommendation system, T. Donkers et al. [24] extended the recurrent neural network and demonstrated the way that in addition to the sequences of consumed commodity items, individual users could be represented in a new type of Gated Recurrent Unit for the generation of the personalized next-item recommendation.

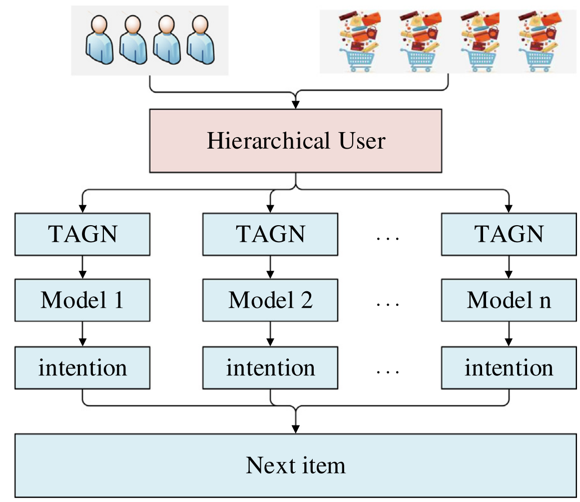


Fig. 1. Framework of our model.

3. Proposed methodology

3.1. Problem descriptions

U, I, T, and C represent the sets of users, products, time and categories, respectively. $S = \{U, I, C, T\}$ is the set of users' sessions.

For a special user $u \epsilon U$, he/she has several corresponding sequences: item sequence $I_u = \{i_1, i_2, i_3, \ldots, i_n\}$ and time sequence $T_u = \{t_1, t_2, t_3, \ldots, t_n\}$. As for the item sequence, each of the items belongs to a category, so the corresponding category sequence $C_u = \{c_1, c_2, c_3, \ldots, c_n\}$ may be generated. For the time sequence, the corresponding time interval sequence will be generated: $\triangle T_u = \{\triangle t_1, \triangle t_2, \triangle t_3, \ldots, \triangle t_n\}$, where $\triangle t_1 = 0$, $\triangle t_k = \triangle t_k - \triangle t_{k-1}$. At a specific time t, when a user click or buy a product, it will generate a session $s^u = \{i, c, t\}$, where, s^u reflects the short-term intention and preference of user u, and $S^u = \bigcup s^u, T = \{1, 2, \ldots, t\}$ reflects the long-term interest of users. The application of time interval sequence is introduced in Section 3.4.

This study aims to explore users' intentions and preferences by giving the users' action sequences and further predict the next commodity for the user.

The proposed method in the study is based on a recurrent neural network and considers the influences of time, category preference, and user's personal profile on the network. Fig. 1 shows the overall framework of the model. The framework consists of two components: a hierarchical phase and a Time-Preference Gate Network model. With the basic profiles and historical behaviors of users as the initial inputs, effective user features are selected to stratify the users and different hierarchies of users are then generated (Model 1 to Model n in Fig. 1). Then, the users' preferences of separate hierarchies are input into TPGN model to train the model and the user's intentions and preferences are obtained. Finally, from the results, the items that match the user's intentions are further searched as the next item.

3.2. Feature embedding

In traditional machine learning prediction algorithms, the performance is often improved by constructing special features [8–11]. Here, we introduce some basic discrete features of users as the embedding of the neural network, such as the user's gender and age, so as to capture the preferences of users at different hierarchies.

The reason for the consideration of the above features may be interpreted as follows. In daily life, users of different genders have different shopping interests and preferences [25]. As for the time feature, when boys have categories in their minds, they are apt to directly access relevant categories, quickly find the products that they like, and make purchasing decisions faster. Girls spend more time finding the items that they like and make the final decision slower. As for the feature of access targets, boys are prone to access some items that are only needed, whereas girls probably access categories that are inconsistent with the targets even if they have categories in their minds before they visit the online shopping platform.

3.3. LSTM

LSTM [26] is a variant of recurrent neural network, which can learn the long-term dependencies between data. Storage units of LSTM include three kinds of gates: input gate, output gate and forget gate, and their update formula are defined as follows:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{1}$$

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{2}$$

$$\tilde{c_t} = tanh(W_c[h_{t-1}, x_t] + b_c) \tag{3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c_t} \tag{4}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \odot tanh(c_t) \tag{6}$$

where i_t, f_t and o_t respectively denote the input gate, output gate and forget gate of the unit at time t; c_t is the activation vector of this unit and composed of two parts: the previous unit's state and the current unit state, which can indicate users' long-term interest and be respectively controlled by forget gate and input gate; $\tilde{c_t}$ indicates the short-term interests of users to some degree; x_t and h_t indicates the input feature vector and hidden feature vector; σ is the sigmoid function and its value is between 0 and 1 (0 indicates that the information is not retained at all and 1 indicates that the information is fully retained; W_i, W_f, W_c and W_o respectively indicates weights of different gates); b_i, b_f, b_c and b_o respectively indicates the bias of different gates.

3.4. Time-preference gate network

LSTM applications in series prediction or personal recommendation have been reported [12,27]. In LSTM, x_t in Eq. (1) represents the product that a user has visited in the last time. In other words, it can represent the short-term interest of a user. c_{t-1} in Eq. (4) represents the historical products that a user has visited and can represent the long-term interest of a user. The gradient problem of RNN is solved to a certain degree in LSTM, but the degree is far from enough for the data with longer sequences. Intuitively, if x_t was accessed by a user long ago, it would be difficult to reflect the current target. The Time-Preference Gate Network (TPGN) employs the user's preference and time to control the impact of the last visited item on the next-item prediction. In addition, the time gate and preference gate can also store the user's time interval and preference features in c_t, thereby enhancing the influences of long sequences and short sequences on the LSTM. In this way, we not only consider the items currently accessed by a user, but also include the corresponding time interval and preference features when the user accesses the items.

The unit state of the Time-Preference Gate Network is shown in Fig. 2. A pair of time interval gates and a pair of preference gates are added into the traditional LSTM and respectively represented by T_1, T_2, P_1 and P_2. The Time-Preference Gate Network

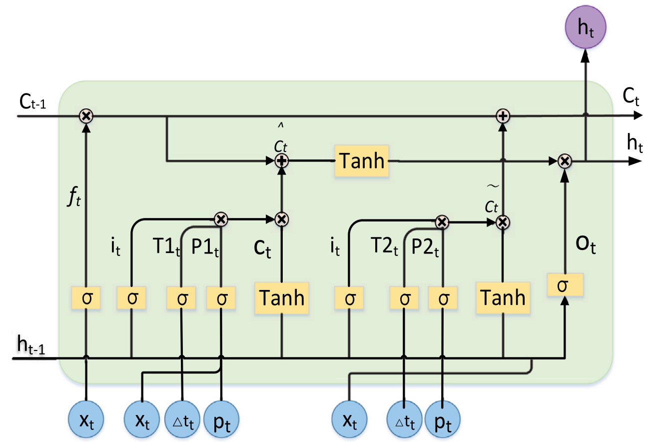


Fig. 2. State of TPGN cell.

consists of an input gate, an output gate, a forget gate, a pair of time interval gates, and a pair of preference gates. Among them, T_1 and P_1 respectively represent the time interval and category of the user, and are used to capture the short-term interest; T_2 and P_2 are used to simulate the user's long-term time interval and preference features in order to capture the user's long-term interest. Corresponding formulas are defined as follows:

$$T1_t = \sigma(x_t W_{x_{t1}} + \sigma(\Delta t_t W_{t1}) + b_{t1}), s.t. W_{x_{t1}} \leq 0 \tag{7}$$

$$T2_t = \sigma(x_t W_{x_{t2}} + \sigma(\Delta t_t W_{t2}) + b_{t2}) \tag{8}$$

$$P1_t = \sigma(x_t W_{x_{p1}} + \sigma(p_t W_{p1}) + b_{p1}) \tag{9}$$

$$P2_t = \sigma(x_t W_{x_{p2}} + \sigma(p_t W_{p2}) + b_{p2}) \tag{10}$$

$$\hat{c_t} = f_t \odot c_{t-1} + i_t \odot T1_t \odot P1_t \odot \tilde{c_t} \tag{11}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot T2_t \odot P2_t \odot \tilde{c_t} \tag{12}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + \Delta t_t W_{to} + p_t W_{po} + b_o) \tag{13}$$

$$h_t = o_t \odot tanh(\hat{c_t}) \tag{14}$$

where $\triangle t_t$ represents the time interval and p_t represents the preference of a user at time t. Except the input gate i_t, $T1_t$ and $P1_t$ can be considered as new filters. $T1_t$ is equivalent to a filter considering the time interval. $P1_t$ is equivalent to a filter considering information of preference. $\hat{c_t}$ is used to store intermediate results. $\hat{c_t}$ is a filter on the time gate $T1_t$, the preference gate $P1_t$, and the current input gate i_t and can affect the next-item recommendation via the hidden state h_t. As for the time gates $T1_t$ and $T2_t$, the following problem is considered. As for user u, corresponding history record is $\{(i_1^u, c_1^u, t_1^u), (i_2^u, c_2^u, t_2^u), \ldots, (i_n^u, c_n^u, t_n^u)\}$, where n represents the total number of user u interacting with the items during the observation period. Then if there is a small time interval between t_{k-1}^u and $t_k^u (1 < k < n)$, it is significant to infer i_k^u with i_{k-1}^u. However, if t_{k-1}^u and $t_k^u (1 < k < n)$ is separated by a long time (more than one month), it is basically meaningless to infer i_k^u with i_{k-1}^u. The difference shows the importance of time intervals.

The category gates $P1_t$ and $P2_t$ can reflect the user's recent and general intentions. The demands of users vary with time. For example, for user u, if time interval between t_{k-1}^u and $t_k^u (1 < k < n)$ is short, c_{k-1}^u may be consistent with c_k^u. However, if t_{k-1}^u and $t_k^u (1 < k < n)$ are separated by a long time, the category in user preferences may change. In other words, c_{k-1}^u and c_k^u may belong to different categories. Together with the time interval gate, the weight of the category can be used to infer the next item of a user. The result can reflect the user's intention and be used to predict

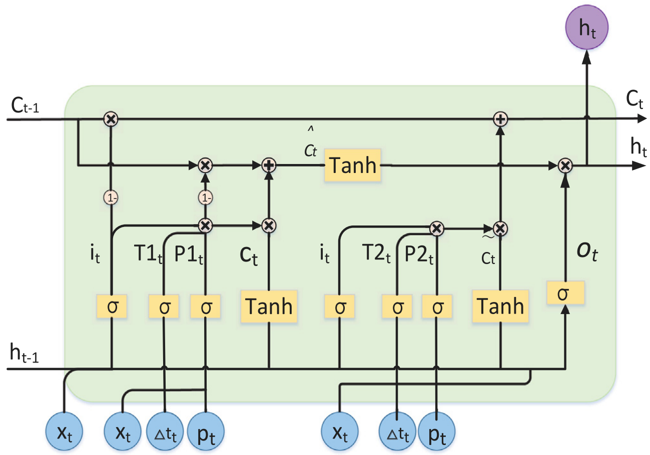


Fig. 3. State of TPGN1 cell.

the user's desired item based on the user's intention. The result considers the importance of user categories.

In Eq. (7), a constraint $W_{x_{t1}} \leq 0$ is applied. Under the constraint, the smaller $\triangle t_t$ corresponds to the larger values of $T1_t$ and $\hat{c}_t$, whereas the larger $\triangle t_t$ corresponds to the smaller values of $T1_t$ and $\hat{c}_t$.

c_t is used to capture the long-term interest of a user. It involves the time interval gate $T2_t$ and the category gate $P2_t$. $T2_t$ stores $\triangle t_t$, and acts on c_t. Similarly, $P2_t$ stores p_t at the first time and then acts on c_t. c_{t-1} not only makes the input of this unit work, but also fuses with the interest of the previous unit. In this way, updating c_t one by one in each unit can improve the simulation of the long-term interests of users.

3.5. Variant of TPGN

Inspired by previous reports [12,28], we proposed a variant of the temporal intention gate model based on the TPGN. In the model, the forget gate is removed. In this paper, we call it TPGN1. Its neural unit is shown in Fig. 3. In TPGN1, Eq (11) and Eq (12) are rewritten as Eq (15) and Eq (16):

$$\hat{c}_t = (1 - i_t \odot T1_t \odot P1_t) \odot c_{t-1} + i_t \odot T1_t \odot P1_t \odot \tilde{c}_t \tag{15}$$

$$c_t = (1 - i_t) \odot c_{t-1} + i_t \odot T2_t \odot P2_t \odot \tilde{c}_t \tag{16}$$

Similar to the input gate i_t, $\hat{c}_t$ and c_t are redefined since the time gate $T1_t$ and the preference gate $P1_t$ are used as the input filters. As for $\hat{c}_t$, the input gate is correlated with the time interval gate and preference gate and the original forget gate is replaced by $1 - i_t \odot T1_t \odot P1_t$. As for c_t, since $T2_t$ is used for storing the time interval, the forget gate is replaced by $1 - i_t$.

4. Experiments and analysis

4.1. Datasets

In this experiment, two datasets were used: the JData dataset.[1] and the T-mall dataset[2] JData dataset is provided by JD.com, a famous online shopping platform in China. It contains the historical behavior data of 100,000 users from February 1, 2016 to April 15, 2016, and user personal profile data, such as gender and age. T-mall dataset is an IJCAI-15 contest dataset recording the historical behavior of users in T-mall, China's largest online shopping platform. The data in both datasets were merged into the form of {user_id, item_id, timestamp, category}. Based on the computing resources, the data were randomly sampled from the original data. In the sampled data, 75% of the data were used as the training data and the remaining 25% of the data were used as the testing data. Table 1 lists the details of the two datasets.

Table 1
Two datasets used in the experiment.

Datasets	JData	T-mall
User num	105,180	121,670
Item num	28710	1,090,390
Cate num	8	1,658
Action type	1: brown 2: add to cart 3: delete from cart 4: buy 5: like 6: click	0: click 1: add to cart 2: buy 3: favorite
Timestamp	–	–

4.2. Experimental settings

In the proposed model, the experimental settings are given as follows. In the training and test sets, we set 3 layers network, and the numbers of hidden layer units were set as {128, 256, 512} to test the influences of different parameters on experimental results. We used the AdaGrad to update the parameters, and considering the sample size and training speed, we chose 0.01 as the initial learning rate. The batch size was set as 20.

4.3. Evaluation method

In order to evaluate the performance of the Time-Preference Gate Network and compare the differences between the results of different models, we employed Recall@K and MRR@K as evaluation indicators. Recall@K was used as the evaluation indicator of original samples. If the actually visited next item appeared in the first K prediction results, Recall@K was determined as 1, otherwise it was determined as 0. MRR@K is a measure of the order of the prediction results. In the first K prediction results, the earlier the next product was actually visited, the larger the value of MRR@K, otherwise the smaller the value of MRR@K.

4.4. Baselines

Long-short term networks (LSTM) [26], a special type of RNN that can learn long-term dependencies and solve the problem of gradient disappearing or explosion of RNN.

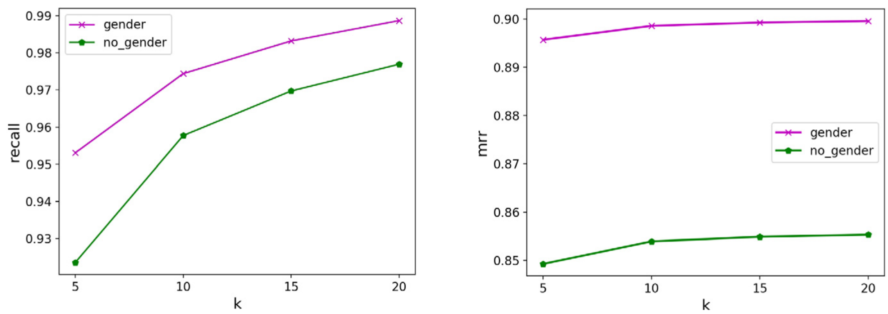
In PLSTM [29], a time gate is added based on classic LSTM and controlled by parameterized oscillations with a frequency range.

Time-LSTM [12] includes three models, in which two models are equipped with a pair of time interval gates based on the classic LSTM and named Time-LSTM2 and Time-LSTM3. In Time-LSTM3, the forget gate is removed. In the study, the best experimental results of these models were selected for the comparison.

YouTube DNN [15] is one of the most classic deep learning models for industrial recommender systems. Its success depends on a deep candidate generation model and a separate deep ranking model.

MIND [16] designs a multi-interest extractor layer based on the capsule routing mechanism, which is applicable for clustering past behaviors and extracting diverse interests.

1 https://jdata.jd.com/html/detail.html?id=1.
2 https://tianchi.aliyun.com/dataset/dataDetail?dataId=47.



(a) Results of recall on feature of genders and without feature of genders

(b) Results of mrr on feature of genders and without feature of genders

Fig. 4. Roles of feature embedding in TPGN..

4.5. Experimental results

4.5.1. Comparison of various models

The model comparison results in Table 2 are summarized below. Here, the results of K={5, 10, 20} are provided.

(1) Our proposed method performed better than all baseline methods, including the classic LSTM, PLSTM, Time-LSTM, DNN and MIND. And the second best performance was Time-LSTM. When K = {5, 10, 20}, as for JData dataset, the recall rates of TPGN were respectively 0.29%, 0.04% and 0.01% higher than those of Time-LSTM. The MRRs of TPGN were 0.11%, 0.82%, and 0.81% higher than those of Time-LSTM. The recall rates of TPGN1 was respectively 0.13%, 0.41% and 0.14% higher than those of Time-LSTM and the MRR of TPGN1 were respectively 0.11%, 0.12% and 0.10% higher than those of Time-LSTM. As for T-mall dataset, the recall rates of TPGN were respectively 0, 0.69%, 3.72% higher than those of Time-LSTM. The MRRs of TPGN were 0.04%, 0.11%, and 0.27% higher than those of Time-LSTM. The recall rates of TPGN1 were respectively 1.63%, 1.77% and 0.82% higher than those of Time-LSTM and the MRR of TPGN1 were respectively 1.08%, 1.06% and 1.00% higher than those of Time-LSTM. Due to not considered the importance of time series, PLSTM and LSTM performed worse than TPGN, TPGN1 and Time-LSTM. MIND ignored the item positions lead to poor results. And DNN showed the worst results, which may be related to the average pooling and ignore the time and category factors.

(2) The results of TPGN1 were slightly better than those of TPGN because of the less parameters after removing the forget gate from the TPGN. 3) When K from 5 to 20, the recall rates and the MRRs on JData dataset growth steady than those on T-mall dataset because the user history behaviors of the JData dataset were more detailed in a continuous time series. For example, the user with ID "200001" clicked and browsed on the item 120208 for multiple times at 12:44:11 on March 13, 2016 and the continuous behaviors resulted in an increase in the values of the evaluation indicators.

4.5.2. Roles of features

In order to further understand the role of features, we compared the results of the model without feature embedding and the model with embedded features. As for JData data, when gender is the only feature, the comparison result is given in Fig. 4. Fig. 4(a) and Fig. 4(b) respectively show Recall values and Mrr values obtained after and before embedding gender feature in the model. After the user's gender was embedded, the performance of the model was much higher than that of the model before the gender feature was embedded regardless of the value of K. When K = 5, the recall rate of the embedded model was 2.95% higher than that of the non-embedded model and Mrr of the embedded model was 4.64% higher than that of the non-embedded model. With the increase in K, although the performance of the two models gradually became close, the performance of the model with embedded gender feature was still better than that of the model without any feature, thus verifying the role of feature embedding.

4.5.3. Roles of preference gates

In order to examine the role of preference gates, we compared the model with only time interval gates with the model with preference gates (TPGN). Due to space limitations, only the results of the T-mall dataset are shown (Fig. 5). Fig. 5(a) shows the comparison results of Time-LSTM2 and TPGN. Fig. 5(b) shows the comparison results of two models with simplified parameters (Time-LSTM3 and TPGN1). As for the T-mall dataset, irrespective of the value of K, the performance of our model was significantly better than that of the model without the preference gate. With the increase in the value of K, the performance of the model with the preference gate was higher. With the increase in K, the recall rate of TPGN showed the more linearly increasing trend, whereas than the recall rate of Time-LSTM2 increased more slowly. As for the two models of TPGN1 and Time-LSTM3, although the recall rate of Time-LSTM3 was increased faster than that of Time-LSTM2, the recall rate of Time-LSTM3 was increased slower than that of TPGN1. In addition, the performances of the two models with reduced model parameters were better than that of the original model (Fig. 5(b)).

4.5.4. Parameter comparison

In order to train the model better, we selected different numbers of parameters from the hidden layer. Fig. 6 shows the results of TPGN1 model on the T-mall dataset. With the increase in the number of hidden layers, the performance of the model became better.

4.5.5. Time performance analysis

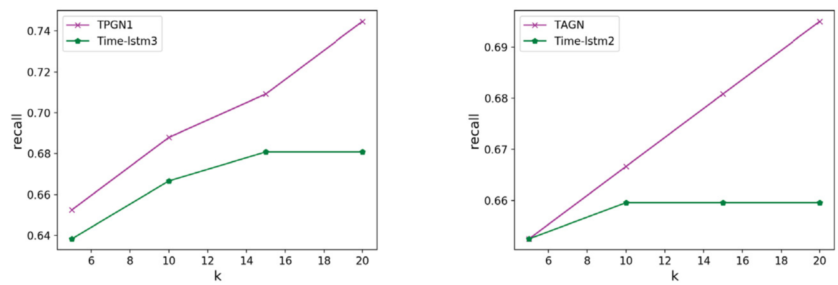
In this paper, we evaluated the time performance between Time3-LSTM and TPGN1. Compared with Time-LSTM, TPGN1 adds feature embedding and preference gates, However, in terms of time consumption, it does not consume more than Time-LSTM. Specifically, on the T-mall dataset, when we set the epoch num to 300, the time consumption of Time-LSTM3 is 6018 s, and the time consumption of TPGN1 is 6028 s.

5. Conclusion

We propose a Time-Preference Gate Network model to solve the users' intention problem in the study. In TPGN, a pair of time gates are adopted to capture the time interests of users,

Table 2
Comparison results of various models.

	JData						T-mail					
	Rec@5	Rec@10	Rec@20	mrr@5	mrr@10	mrr@20	Rec@5	Rec@10	Rec@20	mrr@5	mrr@10	mrr@20
LSTM	0.9146	0.9493	0.9669	0.8398	0.8431	0.8444	0.4938	0.5922	0.7088	0.3880	0.3899	0.4066
PLSTM	0.8562	0.9078	0.9500	0.7461	0.7531	0.7560	0.4681	0.5738	0.6163	0.2965	0.3247	0.3278
Time-LSTM	0.9240	0.9536	0.9755	0.8495	0.8535	0.8551	0.5061	0.5945	0.6925	0.3954	0.4075	0.4143
DNN	0.3097	0.4729	0.6545	0.1550	0.1767	0.1893	0.2306	0.2847	0.3100	0.1501	0.1584	0.1599
MIND	0.3506	0.5413	0.7541	0.1750	0.2003	0.2151	0.4085	0.4679	0.4938	0.2447	0.2530	0.2549
TPGN	**0.9269**	0.9540	0.9756	**0.8506**	**0.8617**	**0.8632**	0.5061	0.6014	**0.7252**	0.3958	0.4086	0.4170
TPGN1	0.9253	**0.9577**	**0.9769**	**0.8506**	0.8547	0.8561	**0.5224**	**0.6122**	0.7007	**0.4062**	**0.418**	**0.4243**



(a) Results of recall on Time-lstm2 and TPGN

(b) Results of recall on Time-lstm3 and TPGN1

Fig. 5. Role of preference gates.

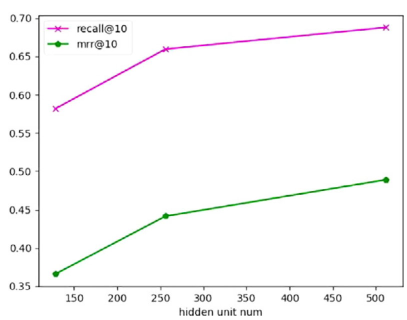


Fig. 6. Comparison results of various models.

and a pair of preference gates are added to capture the category preferences. Similar to filters, both gates can fit long-term and short-term habits of users better. In addition, the model parameters are decreased by coupling the forget gate with the input date. Unlike traditional machine learning methods and deep learning methods, the basic features of users are refined in order to further improve the performance of intention prediction. The experiments with two real datasets proved that our proposed model had certain advantages. In the future, we will apply the attention mechanism to further improve the model performance.

CRediT authorship contribution statement

Yanan Liu: Conceptualization, Methodology, Writing. **Yun Tian:** Reviewing and editing. **Yang Xu:** Writing. **Shifeng Zhao:** Methodology, Hardware support. **Yapei Huang:** Methodology, Hardware support. **Yachun Fan:** Conceptualization. **Fuqing Duan:** Conceptualization. **Ping Guo:** Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Key R&D Program of China (No. 2017YFB1002804). The experimental data came from JD Initiative and Alibaba Initiative. We would like to thank Yukuo Cen from Tsinghua University, and Tao Liu and Xiaoying Huang from our team for help. We also would like to express appreciation to the anonymous reviewers and editors for their helpful comments that improved the paper.

References

[1] G. Zhou, N. Mou, Y. Fan, Q. Pi, W. Bian, C. Zhou, X. Zhu, K. Gai, Deep interest evolution network for click-through rate prediction, 2018, arXiv: Machine Learning.
[2] N. Zhu, J. Cao, Y. Liu, Y. Yang, H. Ying, H. Xiong, Sequential modeling of hierarchical user intention and preference for next-item recommendation, in: Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 807–815, http://dx.doi.org/10.1145/3336191.3371840.
[3] X. Tang, Q. Miao, Y. Quan, J. Tang, K. Deng, Predicting individual retweet behavior by user similarity: A multi-task learning approach, Knowl.-Based Syst. 89 (2015) 681–688, http://dx.doi.org/10.1016/j.knosys.2015.09.008.
[4] Z.Z. Alp, S.G. Oguducu, Identifying topical influencers on twitter based on user behavior and network topology, Knowl.-Based Syst. 141 (2018) 211–221, http://dx.doi.org/10.1016/j.knosys.2017.11.021.
[5] Z. Ming, C. Hancheng, C. Min, L. Yong, User behaviour modeling, recommendations, and purchase prediction during shopping festivals, Electron. Mark. 29 (2) (2019) 263–274, http://dx.doi.org/10.1007/s12525-018-0311-8.
[6] W. Gan, C.W. Lin, P. Fournier-Viger, H.C. Chao, Extracting non-redundant correlated purchase behaviors by utility measure, Knowl.-Based Syst. (2017) 30–41, http://dx.doi.org/10.1016/j.knosys.2017.12.003.
[7] Z. Li, H. Zhao, Q. Liu, Z. Huang, T. Mei, E. Chen, Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 1734–1743, http://dx.doi.org/10.1145/3219819.3220014.
[8] H. Dong, L. Xie, Z. Zhang, Research on statistics-based model for E-commerce user purchase prediction, in: Proceedings of the 10th International Conference on Computer Science and Education (ICCSE), 2015, pp. 553–557, http://dx.doi.org/10.1109/ICCSE.2015.7250308.
[9] Y. Zhang, L. Pang, L. Shi, B. Wang, Large scale purchase prediction with historical user actions on B2C online retail platform, 2014, arXiv:1408.6515.

[10] Z. Yi, D. Wang, K. Hu, Q. Li, Purchase behavior prediction in M-commerce with an optimized sampling methods, in: Proceedings of the 2015 IEEE International Conference on Data Mining Workshop (ICDMW), 2015, pp. 1085–1092, http://dx.doi.org/10.1109/ICDMW.2015.69.
[11] G. Liu, T.T. Nguyen, G. Zhao, W. Zha, J. Yang, J. Cao, M. Wu, P. Zhao, W. Chen, Repeat buyer prediction for E-commerce, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2016, pp. 155–164, http://dx.doi.org/10.1145/2939672.2939674.
[12] Y. Zhu, H. Li, Y. Liao, B. Wang, Z. Guan, H. Liu, D. Cai, What to do next: Modeling user behaviors by time-LSTM, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2017, pp. 3602–3608, http://dx.doi.org/10.24963/ijcai.2017/504.
[13] T. Bai, L. Zou, W.X. Zhao, P. Du, W. Liu, J. Nie, J. Wen, CTRec: A long-short demands evolution model for continuous-time recommendation, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2019, pp. 675–684, http://dx.doi.org/10.1145/3331184.3331199.
[14] C. Park, D. Kim, H. Yu, An encoder–decoder switch network for purchase prediction, Knowl.-Based Syst. 185 (2019) 104932, http://dx.doi.org/10.1016/j.knosys.2019.104932.
[15] P. Covington, J. Adams, E. Sargin, Deep neural networks for youtube recommendations, in: Proceedings of the 10th ACM Conference on Recommender Systems, 2016, pp. 191–198, http://dx.doi.org/10.1145/2959100.2959190.
[16] C. Li, D.L. Lee, Z. Liu, M. Wu, W. Li, Multi-interest network with dynamic routing for recommendation at tmall, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 2615–2623, http://dx.doi.org/10.1145/3357384.3357814.
[17] S. Wang, L. Cao, L. Hu, Attention-based transactional context embedding for next-item recommendation, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 2018, pp. 2532–2539.
[18] S. Zhang, Y. Tay, L. Yao, A. Sun, Next item recommendation with self-attention, 2018, arXiv:1808.06414.
[19] F. Yuan, A. Karatzoglou, I. Arapakis, J.M. Jose, X. He, A simple convolutional generative network for next item recommendation, in: Proceedings of the 12th ACM International Conference on Web Search and Data Mining (WSDM), 2019, pp. 582–590, http://dx.doi.org/10.1145/3289600.3290975.
[20] Y. Feng, L. Qiang, W. Shu, W. Liang, T. Tan, A dynamic recurrent model for next basket recommendation, in: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2016, pp. 729–732, http://dx.doi.org/10.1145/2911451.2914683.
[21] C. Qiang, W. Shu, H. Yan, W. Liang, A hierarchical contextual attention-based GRU network for sequential recommendation, Neurocomputing (2017) http://dx.doi.org/10.1016/j.neucom.2019.04.073.
[22] Q. Cui, S. Wu, Q. Liu, W. Zhong, L. Wang, MV-RNN: A multi-view recurrent neural network for sequential recommendation, IEEE Trans. Knowl. Data Eng. 32 (2) (2020) 317–331, http://dx.doi.org/10.1109/TKDE.2018.2881260.
[23] M. Jiang, Z. Yang, C. Zhao, What to play next? A RNN-based music recommendation system, in: Proceedings of the 51st Asilomar Conference on Signals, Systems, and Computers, 2017, pp. 356–358, http://dx.doi.org/10.1109/ACSSC.2017.8335200.
[24] T. Donkers, B. Loepp, J. Ziegler, Sequential user-based recurrent neural network recommendations, in: Proceedings of the 11th ACM Conference on Recommender Systems (RECSYS), 2017, pp. 152–160, http://dx.doi.org/10.1145/3109859.3109877.
[25] R. Davis, B. Lang, J.S. Diego, How gender affects the relationship between hedonic shopping motivation and purchase intentions, J. Consum. Behav. 13 (1) (2014) 18–30, http://dx.doi.org/10.1002/cb.1450.
[26] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, http://dx.doi.org/10.1007/978-3-642-24797-2_4.
[27] P. Zhao, H. Zhu, Y. Liu, J. Xu, Z. Li, F. Zhuang, V.S. Sheng, X. Zhou, Where to go next: A spatio-temporal gated network for next POI recommendation, AAAI Tech. Track: Mach. Learn. 33 (01) (2019) 5877–5884, http://dx.doi.org/10.1609/aaai.v33i01.33015877.
[28] K. Greff, R.K. Srivastava, J. Koutnik, B.R. Steunebrink, J. Schmidhuber, LSTM: A search space odyssey, IEEE Trans. Neural Netw. 28 (10) (2017) 2222–2232, http://dx.doi.org/10.1109/TNNLS.2016.2582924.
[29] M. Bouaziz, M. Morchid, R. Dufour, G. Linares, R. De Mori, Parallel long short-term memory for multi-stream classification, in: Proceedings of the IEEE Spoken Language Technology Workshop (SLT), 2016, pp. 218–223, http://dx.doi.org/10.1109/SLT.2016.7846268.