ISSN: 2664-9640

In-depth recommendation model of session-based potential intention and time information

Chaoju Hu [1], Tonghong Yu [2]

[1]School of Control and Computer Engineering, North China Electric Power University, Baoding 071000, China;

[2]School of Control and Computer Engineering, North China Electric Power University, Baoding 071000, China;

Abstract

The session-based recommendation system aims to predict the user's next click behavior based on the anonymous session information of the current user. The existing session-based recommendation system based on recurrent neural network does not consider the user's staying time on the item and the user's potential intention at the same time when the user is modeled. In response to this problem, This paper proposes a deep recommendation model that integrates temporal information and user potential intentions (TAGSR). The Time-LSTM model is used to model the time interval that affects user behavior, and the attention mechanism and CRU gating modeling are used to capture the user's potential intentions. And experiments on two public data sets, the results of Recall@10 and MRR@10 two evaluation indicators have improved, which proves the effectiveness of the model.

Keywords

Session-based, potential intentions, time factor, Attention mechanism.

1. Introduction

In recent years, with the rapid development of technologies such as cloud computing, big data, and the Internet of Things, various applications in the Internet space have emerged in an endless stream, triggering an explosive growth in the scale of data [1]. Various online interactive platforms such as e-commerce, short videos, etc. have been developing rapidly. These have brought transformative development to human society, but at the same time they have also brought serious "information overload" problems. As one of the main tools to solve information overload [2], recommendation systems have been widely used. However, in many cases, the user identity information may be unknown, and only user behavior data in the current session is available. Therefore, the classic collaborative filtering and matrix factorization recommendation methods are not applicable.

The purpose of the session-based recommendation system is to mine the user's interest preferences based on the user's click sequence record contained in the ongoing session, and to recommend items that may be of interest to the user in the next click. In spite of these years, many researchers have achieved good results in session recommendation.

But these algorithms have the following problems: 1) Modeling only the click sequence of the user, ignoring the detailed information of the clicked item and the sequence conversion of the clicked item, it is difficult to guess the user's main intention in this session. 2) There is a time interval between user behaviors, and people always stay on the items they are interested in for a longer time; The duration of user behavior has a very important impact on the connection between user behaviors. 3) The user's behavior sequence often contains a lot of useless click

behaviors, and these irrelevant behaviors will obscure the user's intention, thereby affecting the recommendation results.

This paper proposes an attention neural model of latent intention and temporal information (TAGSR). Combining time factor and click sequence information, infer the user's potential intention through the attention layer and sequence modeling of click items, and then incorporate time interval factors to jointly predict the user's next click.

2. Related work

2.1. Traditional session recommendation method

The traditional session-based recommendation methods are mainly project-based domain methods and Markov chain-based sequence methods. The method of the project domain is mainly based on the similarity of the project during the session to recommend to the user. The similarity is calculated based on the co-occurrence of items in the same session. This method does not consider the order of items, and only generates forecasts based on the last click.

Subsequently, the researchers proposed a sequence recommendation method based on Markov chain. The main idea is to model the user's sequence behavior based on the Markov chain, and predict the user's next behavior based on previous clicks. Shani [3] et al. proposed a session-based recommendation method based on Marko Decision Processes (MDPs). MDPS is a model for sequential random decision problems. Based on Markov chain, Zimdars [4] proposed the use of probabilistic decision tree model to extract sequence patterns. Jiang [5] et al. proposed the Mixed Variable Memory Markov Model (MVMM) to predict the user’s next item of interest, and chose the N-gram model as the benchmark model. He [6] et al. proposed a personalized sequence recommendation model based on the Markov chain, combining matrix factorization and Markov chain. Although this method can theoretically model user sequence behavior well, when the user set is large enough, the required state space will be very large, and the state space will quickly become difficult to control.

2.2. Session-based recommendation method for deep learning

Recurrent Neural Network (RNN) has the advantages of memory and parameter sharing, and has certain advantages in learning the nonlinear characteristics of the sequence. RNN has achieved remarkable results in the field of natural language. At the same time, it also plays an important role in the field of conversational recommendation systems. Hidasi [7] and others successfully applied RNN to a session-based recommendation system for the first time, taking a series of click behaviors in the session as a sequence to predict the next most likely item to click. Tan [8] et al. improved it through data enhancement technology and privilege information method on the basis of RNN. Literature [9] integrates the user's stay time into the RNN sequence modeling, and improves the method of Hidasi et al. Later, Sheil [10] and others proposed the Time-LSTM model. This model adds two time gates T1 and T2 to the standard LSTM to process the time interval information of the user’s click sequence separately. Li [11] et al. applied the attention mechanism to conversation recommendation for the first time, using a hybrid encoder to simultaneously model the user's sequence behavior characteristics to infer the user's potential intentions.

Aiming at the existing problems, this paper proposes a new session-based recommendation system by combining the time factor and the user's potential intentions at the same time. This model is mainly divided into two layers: encoding and decoding.

Among them, the coding layer combines the Time-LSTM framework and the attention mechanism to jointly predict the main purpose of the user in this session. Time-LSTM is used to model the influence of time interval on user behavior, and GRU gate with attention mechanism is used to capture the most important intention of the user. Finally, the results of

the two are combined, and a unified session representation is output to realize the recommendation for the user's next click.

3. Model frame

Hidasi et al. have proved that GRU can flexibly control long and short distance dependent information and is suitable for characterizing sequence data. In terms of processing user intent, this paper uses GRU model to model the user's behavior sequence, combined with the attention mechanism to speculate the user's potential intent. In the processing time interval, we use the Time-LSTM model proposed by Sheil[10].

3.1. GRU

GRU input is the previous hidden layer h_{t-1} and the current input x_t ,the output is the next hidden layer h_t. GRU contains two doors reset door r_t and update door z_t , Among them, r_t is used to calculate the candidate hidden layer $\tilde{h}$, and what the control is to keep the information of the previous hidden layer h_{t-1}; z_t is used to control how many candidate hidden layers $\tilde{h}$ are added to get the output h_t. σ is the Sigmoid function.

$$\begin{aligned} r_t &= \sigma(W_z \cdot [h_{t-1}, x_t]) \\ \tilde{h} &= \tanh(W \cdot [r_t * h_{t-1}, x_t]) \\ h_t &= (1 - z_t) * h_{t-1} + z_t * h^t \end{aligned} \tag{1}$$

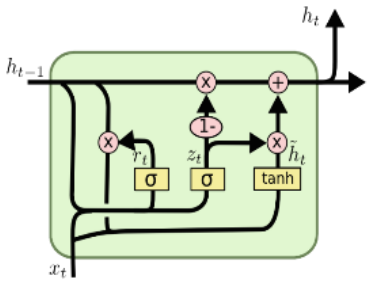

Figure 1 : GRU model diagram

3.2. Attention mechanism

The attention mechanism is derived from the study of human vision. However, as the attention mechanism has achieved better results in natural language processing, the attention mechanism has also been widely used in recommendation systems. The attention mechanism can solve the problem of dynamically calculating the user's current main intention in the user's current session. When the attention mechanism is applied in RNN, when the hidden state h1, h2, ..., hi is given at each time, when calculating the output at t+1, the attention mechanism not only considers the influence of h_t on the output at t+1, And consider the impact of each time between the previous 1-t. The degree of influence of each moment on the output at time t+1 is determined by the attention weight coefficient:

$$\alpha_i = \frac{\exp(f(h_t, h_i))}{\sum_{j=1}^{t} \exp(f(h_t, h_j))} \tag{2}$$

$$c = \sum_{i=1}^{t} \alpha_i h_i \tag{3}$$

ISSN: 2664-9640

In the above formula: $h_i(1 \le i \le t)$ is the hidden state at each moment; α_i is the attention weight coefficient; $f(h_t, h_i)$ is the function for calculating the similarity of the hidden state at each moment, generally calculating the dot product of two parts of the vector or the similarity of the two parts of the Cosine; c is the weighted sum vector of each hidden state.

3.3. Time-LSTM

There are three improved models of Time-LSTM. Time-LSTM3 adopts coupling input and forgetting gate to reduce the parameters of the model without affecting the effect of the algorithm. So this article also uses Time-LSTM3. Unless otherwise specified, the Time-LSTM hereinafter refers to the third type of Time-LSTM3. The Time-LSTM model is proposed on the basis of the standard LSTM. It draws on the gate mechanism in LSTM, and adds two time gates T1 and T2 on the basis of it to realize the impact of the time interval on the current and long-term behavior of users Modeling. Among them, T1 dynamically calculates the influence of the current behavior on the next behavior according to the time interval between behaviors, and the time gate T2 is used to calculate the long influence of the time interval on user behavior; as shown in the following formula

$$\begin{aligned}
\tilde{c}_m &= (1 - i_m \odot T1_m) \odot c_{m-1} + i_m \odot T1_m \odot \delta_c(x_m W_{xc} + h_{m-1} W_{hc} + b_c) \\
c_m &= (1 - i_m) \odot c_{m-1} + i_m \odot T2_m \odot \delta_c(x_m W_{xc} + h_{m-1} W_{hc} + b_c) \\
i_m &= \delta_i(x_m W_{xi} + h_{m-1} W_{hi} + w_{ci} \odot c_{m-1} + b_i) \\
o_m &= \delta_o(x_m W_{xo} + \Delta t_m W_{to} + h_{m-1} W_{hf} + w_{co} \odot \tilde{c}_m + b_0) \\
h_m &= o_m \odot \delta_h(\tilde{c}_m)
\end{aligned} \tag{4}$$

3.4. TAGSR

The basic idea of the model proposed in this paper is to use Time-LSTM to model the influence of time interval on user behavior and the attention mechanism to capture the user's potential intentions. Using the encoder and decoder structure[12], the encoder consists of two parts, one is the user sequence behavior coding layer, using Time-LSTM to encode the user sequence behavior, the other part is the user intention encoder layer, using GRU gate Control and attention mechanisms to encode user intentions. Finally, the decoder combines the two parts to realize the prediction of the next click. The structure diagram of the model is shown below.

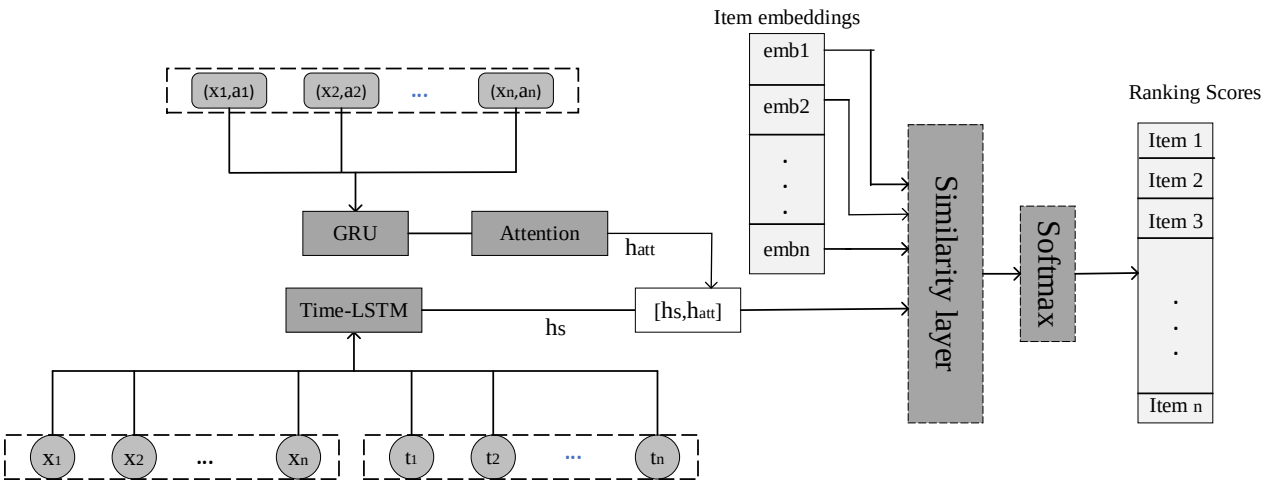

Figure 2: Overall structure of TAGSR model

The input of the model is a user's click, scoring sequence and corresponding time interval. The item x_m^{ui} clicked by the user and the corresponding score a_m^{ui} are coded by one-hot. When calculating the time interval Δt_m^{ui}, this paper uses formula

$\Delta T^u := [(x_1^{ui}, x_2^{ui} - t_1^{ui}), (x_2^{ui}, t_3^{ui} - t_2^{ui}), ..., (x_{nu}^{ui}, t_q^{ui} - t_{nu}^{ui})]$ to calculate. What Time-LSTM learns is the user's sequence behavior characteristics, that is, the long-term and short-term interests of the user's sequence form, so the model uses $(x_m^{ui}, \Delta t_m^{ui})$ as the corresponding input of the sequence behavior encoder layer. In this paper, users' different ratings for different items are converted into user interest weights for different items, so (x_m^{ui}, a_m^{ui}) is used as the input of the user's intention coding layer

The sequence behavior of Time-LSTM coding users is called the sequence behavior coding layer, and the output of this layer is the last hidden state representation h^s of Time-LSTM. The user intention coding layer is composed of the GRU and the attention layer. The item-level user sequence behavior representation is obtained through the GRU, and the user's main intention is learned through the attention layer. The calculation formula is as follows.

$$e_j = v^T \tanh(A_1 h_t + A_2 h_j) \tag{5}$$

$$\alpha_j = \frac{\exp(e_j)}{\sum_{j=1}^{t} \exp(e_j)} \tag{6}$$

$$h^{att} = \sum_{j=1}^{t} \alpha_j h_j \tag{7}$$

Among them, v、A_1 and A_2 are parameter matrices; h_t represents the hidden state of the user's last click generated by the CRU; e_j is to calculate the similarity between each hidden state $[h_i, h_j]$. Then calculate the weight coefficient α_j of the attention mechanism based on these similarities. Calculate the output of the coding layer of the user's intention according to the similarity coefficient. Finally, through the combination of the output of the user sequence behavior encoder and the output of the user's main intention coding layer, a mixed dynamic representation I of the current session is obtained, and the decoder will decode according to the representation to reflect the user's intention. The model uses a cross-entropy loss function, the calculation formula is as follows.

$$L(p, q) = -\sum_{i=1}^{m} p_i \log(q_i) \tag{8}$$

In the formula, q represents the probability distribution of the predicted results of the model; p represents the true distribution.

4. Experiment

4.1. Data set and data processing

This article conducts experiments on the public dataset Movielens and dataset LastFM. For the Movielens dataset, this article extracts tuples (user_id, movie_id, rate, timestamp). Each tuple indicates that a certain user user_id gave the movie movie_id a rating rate at the time timestamp. Since the timestamp is accurate to the day, that is, a user may have scored multiple movies on the same day. In order to facilitate the calculation of the time interval, this article only keeps the first one. Movielens discretizes the user's rating (0.5 to 5.0) for each movie into 10 levels from 0 to 9. For LastFM, this article extracts tuples (user_id, song_id, timestamp). Each tuple indicates that a certain user user_id clicked on the music song_id at the time timestamp. Since

the LastFM data set itself does not include user ratings, it is considered in the experiment that each user on the data set has the same rating for each piece of music.

The statistical data of the number of users, the number of items, and the number of user interactions included in the above two data sets are shown in Table 1. For each data set, this paper randomly selects 80% of it as the training set, and the remaining 20% as the training set.

Table 1: Data set statistics

Content	LastFM 1	Movielens 2
User number	987	600
Number of items	5 000	9 000
Number of user item interactions	818 767	100 000

4.2. Evaluation Index

In the recommendation system, the number of items recommended by most session-based recommendations is affected by certain factors, so most recommendation problems are still TOP-k recommendations. When evaluating the model in this article, two evaluation indicators, recall rate Recall and average reciprocal ranking MRR, are used. These two indicators are also the most commonly used evaluation indicators in current session-based recommendation scenarios.

Recall rate: Mix each target item (true value) that needs to be predicted with other random 100 items. Then the 101 items are sorted according to the model proposed in this article, and the top ten items are selected to generate a recommendation list. The algorithm is defined as follows.

$$\mathrm{Re}call@K = \frac{n_{hit}}{N} \tag{9}$$

Among them, n_{hit} represents the number of samples with correct items in the first K recommended items, and N represents the total number of samples in the test set.

Mean reciprocal ranking MRR: The average value of the reciprocal ranking of the target item in the recommended list. It can reflect the ranking of the target item in the recommended list. The algorithm is defined as follows.

$$MRR@K = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank(i)} \tag{10}$$

Among them, N represents the total number of samples in the test set, and $rank(i)$ represents the arrangement position of the correct item in the recommended list of the test sample.

This article uses $K=10$ when evaluating the model, because in most practical scenarios of session-based recommendation systems, most users only pay attention to the recommended items that appear on the first page.

4.3. Parameter setting

The parameter setting in this article is that the initial learning rate is 0.01, the learning rate attenuation coefficient is 0.1, the training mini-Batch size is 10, and the number of iterations is 30. All weight matrices are initialized with random numbers of Gaussian distribution of N(0,0.1). The loss function of the model uses the cross-entropy loss function and the Adam optimization method is used to solve the model parameters.

ISSN: 2664-9640

4.4. Experimental comparison method

The experimental comparison in this article is mainly carried out from the time interval and main intentions, so the baseline selection is as follows.

LSTM. Recommended method based on LSTM. This method only considers the user's click sequence, without time interval and attention mechanism.

Time-LSTM. This model considers the user's click sequence while also considering the time interval, without adding an attention mechanism to capture the main intent.

NARM. This model is a hybrid encoder that models the user's sequence behavior characteristics and main intention recommendation method at the same time, but does not consider the time interval information.

4.5. Experimental results

The experimental results of the model in this paper and the current three different methods on two data sets are shown in Figure 2. Due to the limitation of the experimental conditions, the NARM model cannot be run, and the experimental results come from the original results of the author.

Table 2: Comparison of experimental results of different methods on two data sets

Method	LastFM		Movielens	
	Recall@10	MRR@10	Recall@10	MRR@10
LSTM	0.244 8	0.09 3	0.539 9	0.257 1
Time-LSTM	0.380 2	0.178 5	0.676 8	0.355 3
NARM	0.220 4	0.103 1	—	—
TAGSR	0.400 0	0.223 6	0.706 3	0.384 1

Through the comparison of experimental results, the TAGSR model has a good performance on the two indicators of recall rate Recall@10 and average reciprocal ranking MRR@10. This shows that in the modeling of user behavior sequence, considering the time interval information and predicting the user's potential intention, it has a better effect on conversation recommendation.

5. Conclusion

For session-based recommendation, this paper proposes a neural network model based on deep learning. When modeling user sequence behavior, taking into account the time interval between user clicks and the influence of user intention on the next click, it can be concluded that this model does play an important role through experiments.

At present, most of the session-based recommendation methods use sequence information instead of other detailed information of the item, which leads to insufficient learning information. In the next work, more coarse-grained item information will be introduced to improve the generalization ability of the model and improve the accuracy of recommendation.

References

[1]. Marz N, Warren J. Big Data: Principles and best practices of scalable realtime data systems[M]. Manning Publications Co., 2015.

[2]. Woods D D, Patterson E S, R oth E M. Can we ever escape from data overload? A cognitive systems diagnosis [J] . Cognition Technology & Work, 2002, 4(1) :22- 36.

[3]. Shani G , Heckerman D , Brafman R I , et al. An MDP-Based Recommender System[J]. Journal of Machine Learning Research, 2005, 6(1):1265-1295.
[4]. Zimdars A , Chickering D M , Meek C . Using Temporal Data for Making Recommendations[J]. Proc. conf. on Uncertainty in Ai, 2001.
[5]. He Q, Jiang D, Liao Z, et al. Web query recommendation via sequential query prediction［C］/ /International Conference on Data Engineering, IEEE, 2009:1443- 1454.
[6]. He R , Mcauley J . Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation[C]// 2016 IEEE 16th International Conference on Data Mining (ICDM). IEEE, 2016.
[7]. Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk.2016. Session-based Recommendations with Recurrent Neural Networks. International Conference on Learning Representations (2016).
[8]. Yong Kiam Tan, Xinxing Xu, and Yong Liu. Improved Recurrent Neural Networks for Session-based Recommendations. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS 2016). ACM, New York, NY,USA, 17–22.
[9]. Bogina, Veronika and T. Kuflik. “Incorporating Dwell Time in Session-Based Recommendations with Recurrent Neural Networks.” RecTemp@RecSys (2017).
[10]. Sheil H, Ｒana O. Classifying and recommending using gradient boosted machines and vector space models［C］/ /UK Workshop on Computational Intelligence, 2017:214 －221.
[11]. Li J, Ｒen P, Chen Z, et al. Neural attentive session- based recommendation［ C］/ /2017 ACM on Conference on Information and Knowledge Management, 2017:1419 －1428.
[12]. Cho K, Van Merri nboer B, Gulcehre C, et al. Learning phrase representations using ＲNN encoder- decoder for statistical machine translation［ EB］ . arXiv:1406. 1078, 2014.