

Graph hierarchical dwell-time attention network for session-based recommendation

Honghui Rong, Wenjun Zhu*, and Cui Zhu

College of Computer Science and Technology, Beijing University of Technology, Beijing, China

Abstract. Session-based recommendation (SBR) is making item recommendations based on anonymous click behavior. SBR based on graph neural networks has shown great power in recent years. It can enhance the representation of items in a session. The aggregation of items is then used to generate a session vector for the recommendation. However, existing SBR models rarely consider the impact of dwell-time in session data when performing session item aggregation. The dwell-time contains the implicit behavior of anonymous users in the session sequence. In order to obtain a more accurate session embedding and take into account the impact of multiple perspectives, we propose a new model, graph hierarchical dwell-time attention network. This approach uses a modified graph neural network to learn session items by extracting loss information from graph modeling. We also design a hierarchical dwell-time attention module that uses the effect of dwell-time to generate long-term preferences for sessions. Experimental results show that GHDAN outperforms the state-of-the-art session-based recommendation methods.

Keywords: Session-based recommendation, Graph neural network, Hierarchical dwell-time.

1 Introduction

In the rapid development of big data, recommendation system plays a vital role in various platforms. They solve the problem of data overload by recommending content to users that meet their needs. Most recommendation algorithms need to get the user's profile to make recommendations. However, user information is often not accessible due to privacy issues. These existing problems have brought significant challenges to the recommendation system. The session-based recommendation system solves this problem by making recommendations without obtaining user configuration information.

Graph neural networks (GNNs) have emerged, with the rapid development of the chip industry and performance improvement in recent years. They capture graph dependencies by transmission between graph nodes and achieve good results in various downstream tasks. Session-based recommendations can also be modeled as graph structures if items are treated as nodes and sequential information is constructed as edges. SR-GNN[1] and other GNN

* Corresponding author: zhuwenjun@bjut.edu.cn

models can learn more relative content information through graph learning. GCE-GNN[2] distinguishes different interaction relationships between session items, a supplementary means of additional information for the relationship between session items. However, this approach does not distinguish the impact of different item interactions within the same relationship in a session. The frequency of interactions between session items also reflects the priority of interactions between items.

The existing models are mainly based on the sequence of contents in a session. Features of recent items are used as the overall expression of the whole session. In this method, the model cannot capture the long-distance preference. The recommendation results mainly rely on the recently clicked content, without much long-distance content information.

Each session is a click action of an anonymous user in a short period. Therefore, the dwell-time of a user in the project can effectively indicate the user's interest preferences. The longer a user stays in a project, the more interested the user is. Dwell-time is an excellent way to capture relatively accurate long-term preferences.

Some residence time-based models[3,4] mainly use unprocessed time data, which is more continuous. However, different time data affect project attention for different session sequences. The influence of temporal data on the user's attention is mainly co-determined by all temporal data of the current session. For example, in the two-time data series, {200,100,50,7} and {50, 30, 25, 4}, the item with a residence time of 50 should be the unconcerned part in the first sequence and the critical part in the second sequence. Previous models rarely took these issues into account.

This paper proposes a Graph hierarchical dwell-time Attention Network(GHDAN).In the model, we first use a weighted multi-relation graph attention network to distinguish the relationships between items to learn item embedding from the session graph. The hierarchical dwell-time attention module is then used to learn long-term dependency representations in the session. Finally, the embedding representation of a single session is generated using the long-term preference embedding with the last-click item embedding short term. The main contributions of the proposed model are as follows:

1. We introduce the user's dwell-time to capture the long-term dependencies. To generalize the effect of time, we design hierarchical methods for dwell-time and use an attention mechanism to aggregate the long-term dependencies of items in a session.
2. We introduced interaction frequencies between items in a graph attention network capable of distinguishing item relationships, which enhanced the representation between items with high relatedness.
3. Experiments conducted on real-world datasets show that GHDAN achieves state-of-the-art performance.

2 Related work

2.1 Markov chain-based model

The Markov chain approach converts the current session into a Markov chain and then infers the user's following action based on the current information. Shani[5] et al. proposed heuristic Markov chain models for the recommendation. Hossein-Zadeh[6] et al. relied on hidden Markov models (HMM) to overcome the limitations of ordinary Markov chain models. Rendle[7] et al. proposed FPMC, which is well adapted to the SBR task by ignoring latent user representations. It combines Markov chains with matrix decomposition to capture long-term preferences in a session. However, these methods mainly model the adjacent items, and it is difficult to capture the relationship between distant items in a session.

2.2 Neural network-based model

Recurrent neural networks can consider the order of historical information and aggregate historical information to represent the current node. GRU4Rec [8] is the first model to apply RNNs to session recommendation tasks. It achieves good results by using multiple layers of GRUs to represent the items in a session for aggregation. Although RNNs show good performance for aggregation of sequential sequences, RNNs cannot explicitly distinguish between the interests of sessions. NARM [9] builds on this by adding attention to items after GRU encoding to capture the importance of items in a session for the recommendation. Liu [10] et al. propose STAMP, which generates attention for items by creating the current session interest and combining it with the last node for the recommendation.

GNNs have been applied in effective recommender systems and have shown compelling results. SR-GNN [1] first proposed to model session sequences through graphs and perform aggregated embeddings between items by graph convolution. GC-SAN [11] added a self-attention mechanism to the work of SR-GNN, allowing better aggregation of session items. FGNN- WGAT [12] introduces a graph attention algorithm and considers the frequency of item interactions between graph nodes. GCE-GNN [2] recommends introducing a global session graph and a session graph, which can organically combine the contents of the sessions with the same session items as the current session. It also classifies and aggregates the links of items in a session through the multi-relation GAT.

3 Preliminaries

3.1 Problem definition

The session-based recommendation task predicts the user's next click based on the click behavior in the current session. Denote by $V = \{v_1, v_2, v_3, \dots, v_m\}$ all m items appearing in all sessions, and denote by $S = \{v_1^s, v_2^s, v_3^s, \dots, v_l^s\}$ the l groups of these sessions, and each session can be represented by $S = (v_1, v_2, v_3, \dots, v_n)$, v is the item of each click, and n is the length of the current session. The goal of session recommendation is to predict the next click behavior corresponding to S based on each given S .

3.2 Hierarchical dwell-time modeling

For a sequence of click items in the dataset $X = \{x_1, x_2, x_3, x_2, x_4\}$, it will include the click occurrence time $T = \{t_1, t_2, t_3, t_4, t_5\}$ of each item, and its dwell-time will be $D = \{t_2 - t_1, t_3 - t_2, t_4 - t_3, t_5 - t_4, 0\} = \{b_1, b_2, b_3, b_4, 0\}$.

In order to obtain more discrete data for a hierarchical representation, we represent the inter-item dwell-time in each group of sessions hierarchically according to equation (1), and obtain a hierarchical hierarchy a_i .

$$a_i = \frac{b_i}{\sum_{k=1}^l b_k} * l \tag{1}$$

To enable consistent representation of the most popular items for each group of sessions, inspired by GCE-GNN, we reverse the obtained hierarchical encoding to achieve an embedded fixed representation of the most popular items.

4 Method

We propose a new session-based recommendation model: GHDAN. The framework of

GHDAN is shown in Figure 1. Next, we will describe the following components in detail.

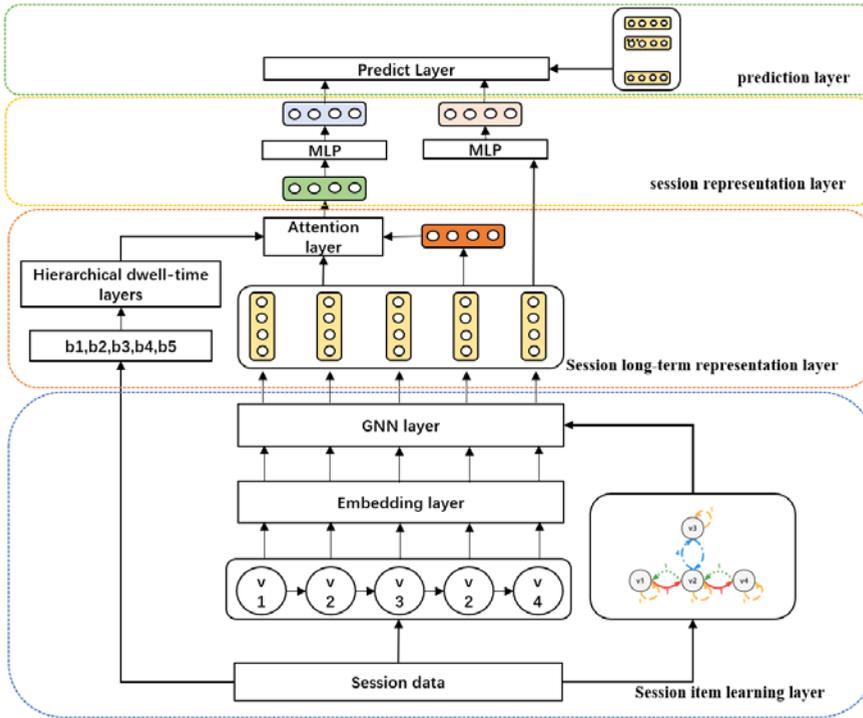


Fig. 1. GHDAN uses a multi-relational graph neural network with weights to implement messages between items and fits the long-term session embedding of a session by the dwell-time between items to make recommendations. It consists of the following four main components: Session item learning layer. Session long-term representation layer. Session representation layer. Prediction layer.

4.1 Session item learning layer

We introduce the interaction frequency between session items in a multi-relational graph attention network to distinguish the information transfer between items in the case of the same relation.

First, items are embedded in a d -dimensional vector space for session S .

We will session graph modeling by the method of Figure 2. For each item in a session, the surrounding neighbors contribute differently to the current node, so we use the attention mechanism to learn the importance of different nodes to the current node.

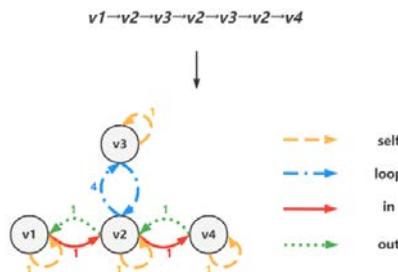


Fig. 2. This figure describes the process of session sequences to graphs. The obtained graph model can distinguish items with different relationships and record the frequency of interactions.

For different types of relational edges, in order to be able to distinguish the difference of aggregation of different relations, the model train four weight vectors $R_{in}, R_{out}, R_{self}, R_{loop}$, respectively.

Thus, four weight matrices corresponding to different types of relations are obtained.

$$r_{ij} = a_r^T (h_i \cdot h_j) \quad (2)$$

Where $a_r \in R^d$ is the learnable parameter.

To enhance the effect of item interaction frequency on aggregation, the model splices the relationship weighting matrix ω_{ij} of the conversation graph with the node importance coefficient r_{ij} . The concern coefficients of surrounding neighbors are obtained by activating nonlinear variances through LeakyReLU. Finally, the weights are normalized by the softmax function.

$$e_{ij} = Attention(h_i, h_j, \omega_{ij}) = LeakyReLU(q^T [r_{ij} || \omega_{ij}]) \quad (3)$$

$$\alpha_{ij} = Softmax(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})} \quad (4)$$

Where $q \in R^2$ is the learnable parameter.

After that, we use the linear combination of the attention weights obtained by equation (5) and the corresponding neighboring features to obtain a neighboring aggregated feature embedding representation for each node.

$$h'_i = \sum_{j \in N_i} \alpha_{ij} h_j \quad (5)$$

4.2 Session long-term representation layer

Firstly, by the method of 3.2, we model the inter-item dwell-time hierarchically and represent it by reverse coding. And embed it into a 2d-dimensional vector space.

$$d_i = Embedding(a_i) \quad (6)$$

a_i indicates the dwell-time of the i-th item in the current session.

To be able to introduce the effect of dwell-time on aggregation in the calculation of attention. The model splices the hierarchical dwell-time embedding obtained by equation (7). item embedding $h'_{v_i^s}$ learned by the graph neural network with a linear transformation and activation using the tanh activation function.

$$z_i = tanh(W_1 [h'_{v_i^s} || d_i] + b) \quad (7)$$

Where $W_1 \in R^{3d \times d}$ is the learnable parameter.

After obtaining the item representation z_i containing the dwell-time, the item attention score for that period is learned through a soft attention mechanism combined with the last clicked item embedding $h'_{v_i^s}$ and activated by the sigmoid to obtain the attention score β_i .

$$\beta_i = \sigma(W_2 z_i + W_3 h'_{v_i^s} + b_4) \quad (8)$$

Where $W_2 \in R^d$ and $W_3 \in R^d$ is the learnable parameter.

Finally, the embedding representation of the session is obtained by weighted summation of the attention scores β_i with the embedding of items in the session.

The session embedding representation, obtained by weighting with the introduction of dwell-time, mainly contains the features of items in the current session. We denote as the long-term preference representation S_l of the session.

$$S_l = \sum_{i=1}^l \beta_i h'_{v_i^s} \quad (9)$$

The model passes the long-term preferences through the dropout layer for processing to prevent overfitting.

$$S'_l = Dropout(S_l) \quad (10)$$

4.3 Session representation layer

The last click behavior in a session often implies the user's current interest. Eventually, the long-term preference embedding and the last click item embedding are jointed together and linearly transformed to represent the final embedding of the session.

$$S = W_4 [h'_{v_i^s} || S'_l] \quad (11)$$

where $W_4 \in R^{2d \times d}$ is the learnable parameter.

4.4 Prediction layer

The task of session-based recommendation is to predict the following item to click for each session. We dot product the session embedding S obtained above with each item and then used the softmax function to obtain the output y_i .

$$y_i = softmax(S^T h_{v_i}) \quad (12)$$

We use the cross-entropy function as the loss function of the model. The objective function to be optimized in the model is as follows:

$$L(\hat{\mathbf{y}}) = - \sum_{i=1}^m y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \quad (13)$$

where \hat{y}_i is the predicted value and y_i is the ground truth.

5 Experiments

5.1 Data preprocessing

We evaluate the model approach using two real datasets representing the session recommendation domain, Diginetica and Yoochoose. The Yoochoose dataset is from the ResSys Challenge 2015, including users' click behavior on the website over six months. The Diginetica dataset is from CIKM Cup 2016, including transaction data from anonymous users. The statistical information of the two datasets after data preprocessing are listed in Table 1.

Following the paper [13], we filtered out all sessions and items with length one appearing less than five times in both datasets for a fair comparison [14]. Sequences are also processed in the same way as in SR-GNN [1]. For the Yoochoose dataset, we use its subset Yoochoose_1/64 for experimental validation.

Table 1. Statistical information of the two datasets used in the experiments.

Statistics	Yoochoose 1_64	Diginetica
# of clicks	557248	982961
# of training sessions	369859	719470
# of test sessions	15324	15963
# of items	37484	43098
Average session length	6.21	4.85

5.2 Evaluation methodology

Following previous work [1], we used two ranking-based metrics commonly used in SBR tasks: Predict@20 and MRR@20.

5.2 Baseline methods

We compared the methods of the GHDAN model with classical methods and the latest models. The following eight baseline models were evaluated.

POP recommends the highest-ranked items in the training set.

Item-KNN [15] makes recommendations by calculating the cosine similarity of items to previous items in the session.

FPMC [5] is a sequential prediction method based on Markov chains.

GRU4REC [6] feeds session sequences into RNN and uses GRU to model the user sequences for the recommendation.

NARM [7] introduces RNN with an attention mechanism to capture the key information in the session for the recommendation.

STAMP [8] uses a soft attention mechanism with the last click sequence to encode the session and make recommendations

SRGNN [1] models sessions as graph sequences and uses gated graph neural networks to obtain item embeddings and computes session-level embedding representations by a soft-attention mechanism.

GC-SAN [9] replaces the soft-attention mechanism with a self-attention mechanism for a recommendation based on SRGNN.

5.3 Hyperparameter setup

Following the previous approach [1], the dimensionality of the embedded items was fixed at 100, and the minimum lot size was set to 100 for all models. For our models, all parameters are initialized using a Gaussian distribution with a mean of 0 and a standard deviation of 0.1. We use an Adam optimizer with an initial learning rate of 0.001, which decays by 0.1 every three cycles.

5.4 Performance comparison

Table 2 shows the experimental results for eight baseline models in the session recommendation domain as well as our proposed model, where the best results in each column are highlighted in bold. For the two evaluation metrics we use, the best performance of our proposed GHDAN is achieved on both datasets. The experimental data results also show that our proposed method is effective.

Table 2. Performances of different methods on the two datasets.

Method	Yoochoose 1 64		Diginetica	
	P@20	MRR@20	P@20	MRR@20
POP	6.71	1.65	0.89	0.20
Item-KNN	51.60	21.81	35.75	11.57
FPMC	45.62	15.01	26.53	6.95
GRU4REC	60.64	22.89	29.45	8.33
NARM	68.32	28.63	49.70	16.17
STAMP	68.74	29.67	45.64	14.32
SR-GNN	70.57	30.94	50.73	17.59
GC-SAN	70.89	30.98	50.80	17.63
GHDAN	71.30	31.44	53.23	18.45

POP leads to the worst performance of traditional methods due to recommending the most frequent items. FPMC can search items using first-order Markov chains and matrix decomposition in comparison and thus outperforms POP on both datasets. While Item-KNN cannot perform learning between items because of the limitation of its model to consider the order of clicks in a session and thus cannot perform interactions.

Among some models of neural networks, NARM and STAMP outperform GRU4REC. STAMP, compared to GRU4REC, does not use RNN to achieve learning between preceding and following items in a session but uses an attention mechanism that eventually aggregates into a session vector representation. Experimental results show that the attention-based neural network model is significantly better than the recurrent neural network-based model. It can be seen that in the session-based recommendation, the importance of the order of items in a session is not high due to the short sequence. NARM, compared to STAMP, adds an RNN to learn the interaction of items sequentially based on the introduction of attention and obtains higher performance on Diginetica, and proves some feasibility of RNN in the field of session recommendation systems.

SR-GNN and GC-SAN demonstrate the feasibility of using graph neural networks in session-based recommendations by modeling the session as a graph model and applying graph neural networks to capture some possible interactions between items at long distances in the session. Furthermore, FGNN improves performance compared to the previous two graph neural network models by increasing the frequency of item interactions in the session data. The above experimental results illustrate the effectiveness of increasing prior knowledge for session-based recommendation systems based on graph models.

Our method GHDAN outperforms other baseline graph modeling methods on both datasets. Unlike them, our approach achieves performance gains by introducing item interaction frequencies in multi-relation and amplifying their effects when aggregated graph attention. On top of this, the performance is improved by capturing the implicit information in the user session by effectively exploiting the user's dwell-time information.

5.5 Effects of introducing dwell-time and adding interaction frequencies to the multi-relation graph attention network

Next, we conducted experiments in two datasets to illustrate the effectiveness of introducing a multilateral type graph attention layer for item interaction frequency and a soft attention layer for hierarchical dwell-time. We designed two comparison models.

GHDAN without item interaction frequency(GHDAN-WIF): using only polygonal relational graphs, graph learning of items.

GHDAN without the dwell-time(GHDAN-WDT): learning the attention weights of items in a session through the items.

Table 3. Performances of different comparison methods on the two datasets.

Method	Yoochoose 1_64		Diginetica	
	P@20	MRR@20	P@20	MRR@20
GHDAN-WIF	70.62	31.04	50.82	17.75
GHDAN-WDT	71.27	31.34	53.09	18.24
GHDAN	71.30	31.44	53.23	18.45

Table 3 shows the results under different comparison models. It can be seen that in both datasets, better performance is obtained for the model with a polygon-type graph attention layer containing item interaction frequencies and a soft attention layer introducing hierarchical dwell-time. In contrast, the model without item interaction frequency cannot learn the transition relationships between different items of the same type of edge in a session, thus reducing the performance. In contrast, the model without introducing the dwell-time soft attention mechanism cannot determine the user's preferences from the prior knowledge in the session data, and thus cannot make better recommendations.

5.6 Analysis on the treatment of dwell-time

This paper aims to enable better aggregation of sessions through effective use of dwell-time. However, the dwell-time used in this paper is a hierarchical dwell-time representation after reverse hierarchical representation. We designed a series of comparative models to verify the validity of our proposed hierarchical dwell-time.

Using forward hierarchical dwell-time (GHDAN-DT): Using forward hierarchical dwell-time and embedding it in d-dimension, we compute the session representation by soft attention.

Using unprocessed dwell-time (GHDAN-UHDT): the original dwell-time is used, and soft attention computed the session representation.

Table 4. Performances of different comparison methods on the two datasets.

Method	Yoochoose 1_64		Diginetica	
	P@20	MRR@20	P@20	MRR@20
GHDAN-DT	70.71	31.16	52.82	18.18
GHDAN-UHDT	70.97	31.25	53.01	18.22
GHDAN	71.30	31.44	53.23	18.45

Table 4 shows the performance analysis of the comparison experiments. We found the worst performance when using unprocessed dwell-time for attention aggregation, this is because the point of using dwell-time information is not to obtain the effect of exact time on it, if there are items with long dwell-time in the session, the items with relatively short user dwell-time should have similar attention scores. Using the forward hierarchical dwell-time for attention calculation significantly improves this situation. However, it does not allow for effective fixation of the hierarchy of items in the session that require the most attention.

Compared to the two variants, the reverse dwell-time hierarchical embedding is superior to the other two comparison models. The above experiments demonstrate that our proposed method is reasonable and feasible.

6 Conclusion

The problem of session-based recommendation is challenging, where user identities and complete historical interactions are often inaccessible due to user privacy issues. This paper proposes a novel framework GHDAN, for session-based recommendations. Specifically, it first enhances the aggregated learning of items by converting session sequences into session graphs and prior knowledge before graph modeling in sessions. Subsequently, it introduces

user dwell-time and session information to enable long-term preferences better to fit users' implicit behavioral information better. Finally, it combines the long-term preferences of sessions with short-term preferences to predict users' following actions better. The combined experiments show that the proposed model approach consistently outperforms the nine baseline models in both datasets.

References

1. Wu S, Tang Y, Zhu Y, et al. Session-based recommendation with graph neural networks[C]//Proceedings of the AAAI Conference on Artificial Intelligence. (2019)
2. Wang Z, Wei W, Cong G, et al. Global context enhanced graph neural networks for session-based recommendation[C]//Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. (2020)
3. Zhang J, Ma C, Zhong C, et al. MBPI: Mixed behaviors and preference interaction for session-based recommendation[J]. Applied Intelligence. (2021)
4. Dorman L I 1975 Variations of Galactic Cosmic Rays (Moscow: Moscow State University Press)
5. Shani G, Heckerman D, Brafman R I, et al. An MDP-based recommender system[J]. Journal of Machine Learning Research. (2005)
6. Hosseinzadeh Aghdam M, Hariri N, Mobasher B, et al. Adapting recommendations to contextual changes using hierarchical hidden Markov models[C]//Proceedings of the 9th ACM Conference on Recommender Systems. (2015)
7. Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized markov chains for next-basket recommendation[C]//Proceedings of the 19th international conference on World wide web. (2010)
8. Hidasi B, Karatzoglou A, Baltrunas L, et al. Session-based recommendations with recurrent neural networks[J]. arXiv preprint arXiv:1511.06939. (2015)
9. Li J, Ren P, Chen Z, et al. Neural attentive session-based recommendation[C]//Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. (2017)
10. Liu Q, Zeng Y, Mokhosi R, et al. STAMP: short-term attention/memory priority model for session-based recommendation[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. (2018)
11. Xu C, Zhao P, Liu Y, et al. Graph Contextualized Self-Attention Network for Session-based Recommendation[C]//IJCAI. (2019)
12. Qiu R, Li J, Huang Z, et al. Rethinking the item order in session-based recommendation with graph neural networks[C]//Proceedings of the 28th ACM International Conference on Information and Knowledge Management. (2019)
13. Li J, Ren P, Chen Z, et al. Neural attentive session-based recommendation[C]//Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. (2017)
14. Tan Y K, Xu X, Liu Y. Improved recurrent neural networks for session-based recommendations[C]//Proceedings of the 1st workshop on deep learning for recommender systems. (2016)
15. Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th international conference on World Wide Web. (2001)