Annular-Graph Attention Model for Personalized Sequential Recommendation

Junmei Hao, Yujie Dun, *Member, IEEE*, Guoshuai Zhao, Yuxia Wu, and Xueming Qian, *Member, IEEE*

***Abstract*—Sequential recommendations aim to predict the user's next behaviors items based on their successive historical behaviors sequence. It has been widely applied in lots of online services. However, current sequential recommendations use the adjacent behaviors to capture the features of the sequence, ignoring the features among nonadjacent sequential items and the summarized features of the sequence. To address the above problems, in this paper, we propose an annular-graph attention based sequential recommendation (AGSR) model by exploring user's long-term and short-term preferences for the personalized sequential recommendation. For user's short-term preferences, AGSR builds an annular-graph on the sequence of user behavior. Then, AGSR proposes an annular-graph attention applying on the sub annular-graph to explore local features and applying annular-graph attention on entire annular-graph to explore the global features and the skip features. For user's long-term preferences, the latent factor model are introduced in AGSR. The experimental results on two public datasets show that our model outperforms the state-of-the-art methods.**

***Index Terms*—Attention mechanism, graph attention, personalized recommendation, sequential recommendation, user preferences.**

I. Introduction

With the development of the internet, watching movies online, and recording the shopping online have become very prevailing. However, the explosive information has given rise to a serious problem called information overload. Users need to spend a lot of time to choose valuable information from the mass of information. Recommender systems are introduced to address the above problem. Recommender systems can filter information to help users find products and content they are interested in. Most recommender systems recommend items just based on the user's general preferences. However, general preferences ignore the short-term variation of user's preferences. Therefore, the sequential recommendation is needed and plays an important role in the recommender systems.

Sequential recommendations model the user behaviors sequence to learn the change of user preferences in a short time, and use the user behaviors sequence to predict the user's next behaviors. There are two types of user preferences exploring from user information, namely the long-term and short-term preferences from user historical information. The former is represented by static behaviors. For example, some users always prefer the dress to the trousers, where the static behaviors show these users have long-term preferences for the dress. The latter is based on the user's recent history behaviors which reveal the dynamic and fluctuate preferences for users. Besides, history behaviors have a strong effect on the next behaviors of the user. For example, some users are likely to buy bags or shoes to go with their recently purchased dress. Thus, we should not only consider long-term preferences but also take short-term dynamic preferences into consideration.

As for recommender systems, the models based on collaborative filtering are widely used. Due to the intuitive and simple characteristic, the methods based on the matrix factorization [1], [2] become the first choice for the recommender systems. Although these methods can explore features of the user and items, they don't consider the influence of user sequential behaviors.

Thus, researchers pay attention to the sequential recommendation and put forward some models. Several models based on the Markov chain [3]–[5] for the sequential recommendation. The personalized transfer matrix based on the Markov chain and the matrix factorization model are fused in the recommender system to capture both the short-term preferences and the long-term preferences. However, with the development of big data, the computational complexity of these methods is also explosive growth. Some deep learning models are introduced in the recommender systems to solve this problem, such as the recurrent neural network (RNN) [6] and the convolutional neural network (CNN) [7]. RNN [6], [8], [9] are utilized to model the sequence dependency of the item to obtain short-term preferences of the user. CNN can also be utilized for sequence embedding [7] to model the adjacent items.

However, there are also some challenges in the recommender system. First, these models mentioned above ignore the global feature in the sequential behaviors. In this paper, local features

Manuscript received April 13, 2021; revised June 15, 2021; accepted July 3, 2021. Date of publication July 14, 2021; date of current version July 12, 2022. This work was supported in part by NSFC under Grants 61732008 and 61772407 and in part by Microsoft Research Asia and Pazhou Lab, Guangzhou. The Associate Editor coordinating the review of this manuscript and approving it for publication was Dr. Wen-Huang Cheng. *(Corresponding authors: Yujie Dun; Guoshuai Zhao; Xueming Qian.)*

Junmei Hao and Yuxia Wu are with the School of Electronics, and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: haojunmei1996@stu.xjtu.edu.cn; wuyuxia@stu.xjtu.edu.cn).

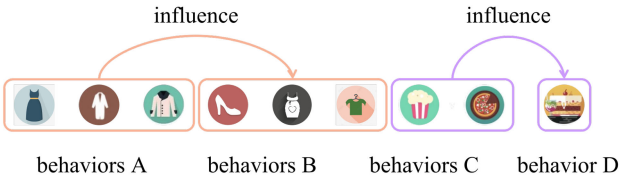
Yujie Dun is with the School of Information, and Communication, Xi'an Jiaotong Universitity, Xi'an 710049, China (e-mail: dunyj@mail.xjtu.edu.cn).

Guoshuai Zhao is with the School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: guoshuai.zhao@xjtu.edu.cn).

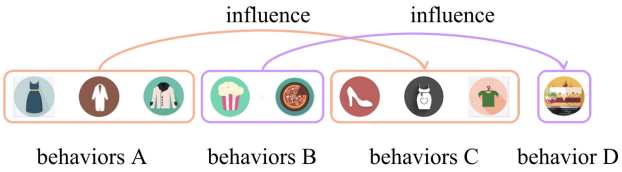
Xueming Qian is with the Key Laboratory for Intelligent Networks, and Network Security, Ministry of Education, and the Smiles Laboratory, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: qianxm@mail.xjtu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMM.2021.3097186.

Digital Object Identifier 10.1109/TMM.2021.3097186



(a) An example of sequential influence. The behaviors B are influenced by the behaviors A and the behavior D are influenced by behaviors C



(b) An example of skip features. The behaviors C are influenced by the behaviors A and the behavior D are influenced by behaviors B

Fig. 1. An example of sequential influence and skip features.

are the features modeling the adjacent items, skip features represent the features modeling nonadjacent sequential items, and global features represent the features modeling summarized information of sequential items. The models mentioned above consider the short-term preferences modeling most based on the local features. For example, in Fig. 1(a), the behaviors B are influenced by the behaviors A, and the behavior D are influenced by behaviors C. They use the adjacent behaviors to capture the features of the sequence, ignoring the summarized features of the sequence. Secondly, these models also ignore the skip features. That the item may be influenced by the nonadjacent items can be called skip features in the recommender systems. For example, in Fig. 1(b), the behaviors C are influenced by the behaviors A, and the behavior D are influenced by behaviors B. As for the application, for example, in the online shopping scene, a user buy a skirt first. Then for going with this dress's color, the user may buy a lipstick. Next, the user buy some cakes. And later, the user buy another skirt. Obviously, there is no direct influence between buying a skirt and buying a cake, but there is a certain correlation between the behavior of buying a skirt this time and the behavior of buying a skirt last time. Furthermore, the lipstick may also have influence on the skirt. However, these behaviors are not adjacent. Thus, extracting global features and skip features to cover this situation is a major challenge in sequential recommendation.

Thus in this paper, we propose an annular-graph attention based sequential recommendation model for exploring global features and skip features. First, we introduce an annular-graph to model sequence patterns. Then, we apply annular-graph attention on the sub annular-graph and entire annular-graph to get the short-term preference for users. For sub-annular graph, we explore local features and we explore both the global features and the skip features for entire annular-graph. At the same time, we employ latent factor models (LFM) [10] to capture the long-term preferences besides the sequential model. Finally, the recommended list is calculated by combining user short-term and long-term preferences with the item features. Thus, there are two aspects about the personalization: (1) We use each user's historical behavior to represent short-term preferences. (2) We use latent factor model to generate long-term preferences for each user. The experiments prove that AGSR is very effective for the sequential recommendation.

The contributions of this work are summarized as follows:

- We propose an annular-graph attention based sequential recommendation (AGSR) model by exploring user's long-term and short-term preferences for the personalized sequential recommendation. Our model builds an annular-graph attention network and also incorporates LFM for comprehensive user preferences modeling.
- AGSR builds an annular-graph on the sequence of user behavior and proposes an annular-graph attention. The annular-graph attention applys on the annular-graph in two aspects for exploring user's short-term preferences. (1) Applying annular-graph attention on sub annular graph explores local features. (2) Applying annular-graph attention on entire annular-graph explores both the global features and skip features.
- We generate a final high-level representation of the user through the hybrid structure of annular-graph attention model and LFM. We explore LFM to model the long-term preferences. Then we merge short-term preference generated by annular-graph attention model and long-term preferences into the model. AGSR achieves state-of-the-art performance on two real-world challenging datasets for the sequential recommendation.

This paper is organized as follows. In Section II, a brief overview of related works is given. Section III introduces the proposed annular-graph attention recommendation model in detail. Experiments and discussions are reported in Section IV. Conclusions are drawn in Section V.

II. Related Work

In this section, we briefly introduce the related work in traditional sequential recommender systems, deep learning based sequential recommendation and attention network.

A. Traditional Sequential Recommender Systems

The traditional recommender systems are mainly based on collaborative filtering [1], [2], [11]. [4], [12], [13] solve the problem of cold start by integrating users' personal interests, interpersonal influence of friends and other factors into matrix factorization. Cheng *et al.* [14], [15] adapt collaborative filtering for music recommendations by representing songs and venue types in the shared latent space. Yang *et al.* [16] propose a data preprocessing framework to generate the rating data as input to the collaborative system. Hidayati *et al.* [17] recommend users about what to wear better based collaborative filtering. However, these algorithms do not take the sequential features of user-item interaction into account. Rendle *et al.* [3] combine matrix factorization with personalized Markov chain to model both the long-term intents of users and the sequence effects. Following this work, Liang *et al.* [18] propose a co-factor model, combing matrix factorization with item embedding to improve the performance of standard matrix factorization and to model the

sequence pattern. Koren [19] introduces a specific mechanism to model the time to improve the performance of collaborative filtering algorithm.

Nevertheless, all these algorithms pay more attention to rating prediction task, which are not specifically design for top-*N* recommendation. To generate a ranking list for the top-*N* recommender system, Pan *et al.* [20] utilize weight low rank approximation [21] and negative example sampling to solve the problem of negative sample missing based on collaborative filtering system. Cheng *et al.* [22] merge word embeddings with matrix factorization into music recommendations. Feng *et al.* [23] propose a metric embedding model for the POI recommender system, combining personality information, geographic location information and order information to avoid the disadvantages of matrix factorization. Apart from that work, metric learning also perform well on sequential recommendation in [24]. Tay *et al.* [25] propose a model to learn latent relations that describe each user item interaction. Zhang *et al.* [26] apply metric learning into matrix factorization. A unified model to explore all the complex interactions together is proposed by [27]. However, the traditional model are not suitable to large-scale data training due to the time-consuming.

B. Deep Learning Based Sequential Recommendation

The rise of deep learning has made the recommender system change dramatically. The recommender systems based on deep learning break through the limitation of the original method and greatly improve performance. Salakhutdinov *et al.* [28] propose the model to apply neural networks to recommender systems. Later, deep learning is widely used in recommender systems. Gated recurrent unit (GRU) based recurrent neural network is applied to recommender system in [6].

The recurrent neural network models the sequential pattern well. Donkers *et al.* [8] model sequential recommendations by morphing the GRU and explicitly introducing personalized user features. Wu *et al.* [29] and [30] apply a Long Short-Term Memory (LSTM) to recurrent recommender system for capturing dynamic trajectories.

Moreover, recurrent neural network also performs well in explainable recommender system. Bansal *et al.* [31] introduce a recurrent neural network to encode text content of the item in collaborative filtering. Bharadhwaj and Joshi [32] introduce a neighborhood based scheme to an LSTM for generating explainable recommendation. Zhang *et al.* [33] apply deep features to social images by constructing a user interest tree. Bai *et al.* [34] propose a long-short demands-aware model, considering that repetitive purchasing action represents the long-term persistent interest of users.

Besides RNN, CNN is also used in recommendation system. Chu and Tsai [35] combine the visual features of the uploaded photos extracted by CNN with the collaborative filtering model to recommend personalized restaurant for users. A CNN-based framework modeling user perception of the image for dress matching recommender system is proposed in [32]. Kim *et al.* [36] combine CNN and probabilistic matrix factorization to explore the context information and Gaussian noise. CNN can also extract sequential features via continuous filter size variation. Inspired by TextCNN [37], Tang and Wang [7] use multiple horizontal and vertical filters to model the item sequence to capture sequential information. A hybrid mCNN-SVM approach to boost attribute extraction in recommender systems is introduced in [38].

Beyond RNN and CNN, reinforcement learning is introduced in sequential recommendation by [39].

C. Neural Attention Models

In recent years, the attention mechanism is an awkward new star in the field of recommender systems. It starts based on deep learning network and has been able to build a stable network with only the attention mechanism. The neural attention mechanism imitates the attention changes of human vision attention. The neural attention mechanism in deep learning is essentially similar to the human selective visual attention mechanism, and the core goal is to select more critical information from numerous information.

Neural
attention is widely used in computer vision [40]–[42] and natural language processing. It is also combined with RNN or CNN to improve the capacity to capture long distance dependence. Lu *et al.* [43] pair adaptive attention with LSTM in image caption to pay more attention to core areas. Bahdnau *et al.* [44] introduce the attention mechanism with RNN into encoder-decoder for machine translation. Vaswani *et al.* [45] propose a multi-attention model only based on self-attention without any other neural network such as CNN or RNN.

Neural attention is not only applied in these aspects. Recently, some studies introduce neural attention into recommendation [26]. Wu *et al.* [46] introduce dual graph attention networks to collaboratively learn representations for social recommendation. Ying *et al.* [47] introduce hierarchical attention mechanism to model the long-term features of users and the generated short-term features of users by using the attention mechanism. They finally get the hybrid representation for the top$-N$ recommendation. Chen *et al.* [48] introduce attention into collaborative filtering to solve the challenging of item-level and component-level implicit feedback in multimedia recommendations. He *et al.* [49] propose item collaborative filtering based on attention mechanism and neural network to improve the expressiveness of features. Tay *et al.* [50] propose a pointer mechanism based on gumbel-softmax, which allows the merging of discrete vectors into a differentiable neural structure. [5], [51], [52] use a hybrid encoder with an attention mechanism to model the session to capture the user's interest. Gong *et al.* [53] utilize CNN and attention mechanisms in conjunction with the label recommender system.

The mainly difference with exiting methods is that we build an annular-graph on the sequence of user behavior and applying an annular-graph attention on the annular-graph in two aspects for exploring user's short-term preferences. The first is that we apply annular-graph attention on sub annular-graph to

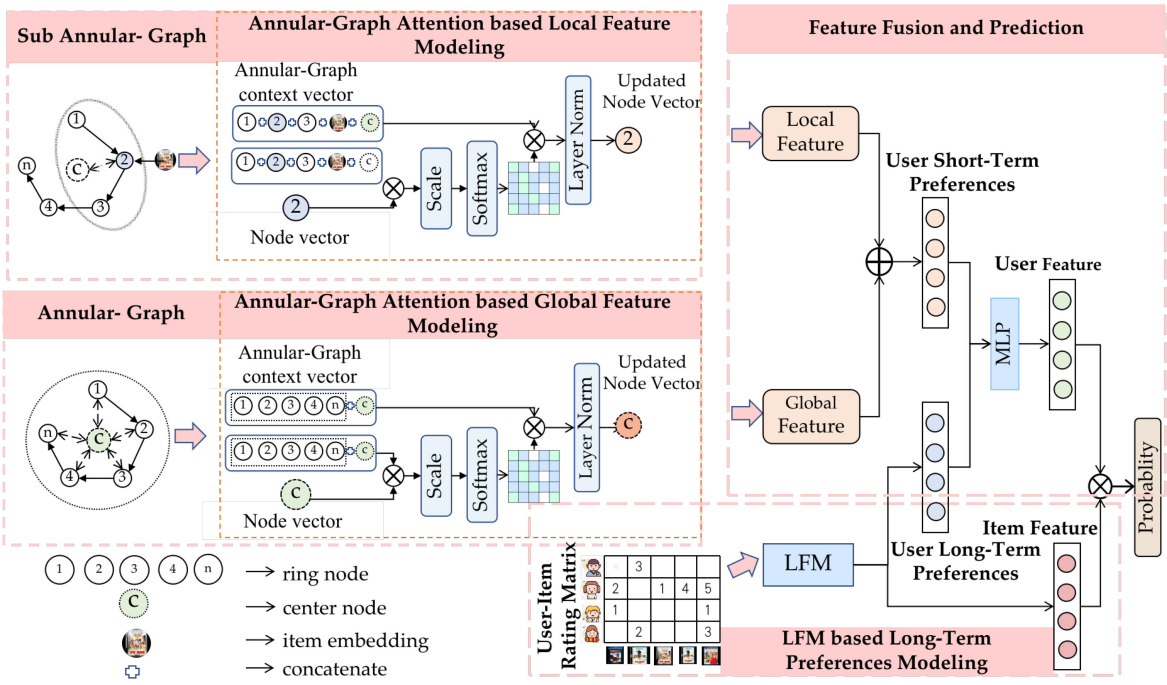


Fig. 2. The framework architecture of our Annular-Graph Attention Recommendation model. In the "Annular-Graph Attention based Local features Modeling" part, we just take the updating of the second node as example, the updating of the other nodes is as the same as the second node.

explore local features and the second is that we apply annular-graph attention on entire annular-graph to explore the global features.

III. The Proposed Annular-Graph Recommendation Model

This section presents the details of the proposed annular-graph attention recommendation (AGSR) model. Fig. 2 shows the main framework of the proposed model. Our framework is first divided into two parts to get the user's short-term preferences and long-term preferences respectively, and then the two parts are combined to get the final recommendation list.

The former part is annular-graph attention based short-term preferences modeling. For each user, we first feed the embeddings of user behaviors sequence into the model as the input to construct the annular-graph. The user behavior sequence represents a sequence of user comments, clicks, purchases, and so on. Secondly, for each annular-graph of successive items, we extract its local and global features. Thirdly, we combine the local features and global features to represent the user's short-term preferences. The other part is using the rating matrix as the input of the latent factor model to obtain user long-term preferences and item features.

Then, we fuse it with the user long-term preferences generated by the latent factor model into a multi-layer perception to get the final user features. Next, we multiply the user features and the item features generated by the latent factor model to predict the probability. Finaly, according to the rank of the predicted probability, we recommend the items to users.

A. Problem Formulation

The details of the problem formulation are as follows. Let $u \in U$ represent a set of users and $i \in I$ represent a set of items, where $|U| = M$ and $|I| = K$ denote the total number of users and items. Given the user's L behaviors sequence (usually $L \ll$ K), L is the size of the sliding window on the user's historical interactions), the goal of sequential recommendation is to predict the items that the user will interact within the next steps. In this model, 1) each item can be represented with a d-dimension embedding vector. For each user, $X \in \mathbb{R}^{L*d}$ ($\mathbb{R}$ represents the real number set) represents the embeddings of his latest L interactions (behaviors sequence) and $X = [x_1, x_2, \ldots, x_L]$. $X \in \mathbb{R}^{L*d}$ is fed into the annular-graph attention model to explore the short-term preferences. 2) We use the latent factor model to mine the long-term preferences. The latent factor model generates the user features and the item features of d-dimension as $P \in \mathbb{R}^{d*M}$, $Q \in \mathbb{R}^{d*K}$ from user-item rating matrix R, where $P \in \mathbb{R}^{d*M}$, $Q \in \mathbb{R}^{d*K}$ are two reduced dimensional matrices. We randomly initialize the $P \in \mathbb{R}^{d*M}$, $Q \in \mathbb{R}^{d*K}$ matrix. The variables u, i, L, d, R are not trainable. They are fixed before the model training. In addition, the other variables mentioned above are trainable. Notations are summarized in Table I.

B. Annular-Graph Attention Based Short-Term Preferences Modeling

We builds an annular-graph on the sequence of user behavior and proposes an annular-graph attention in our model to explore the short-term preferences. Then we introduce the details about the annular-graph, annular-graph attention and how to utilize

TABLE I
NOTATIONS

Notation	Description
u	a user
i	an item
U	the set of users
I	the set of items
M	the number of users
K	the number of items
L	the size of the sliding window on the user's historical interactions
X	represents the embeddings of his latest L interactions
R	the rating matrix
P	the matrix of user long-term preference
Q	the matrix of item feature
r^t	the ring nodes at step t
c^t	the center node at step t
r^{*t}	local feature
z	the user comprehensive preferences
y	the matching score
A	the number of actions

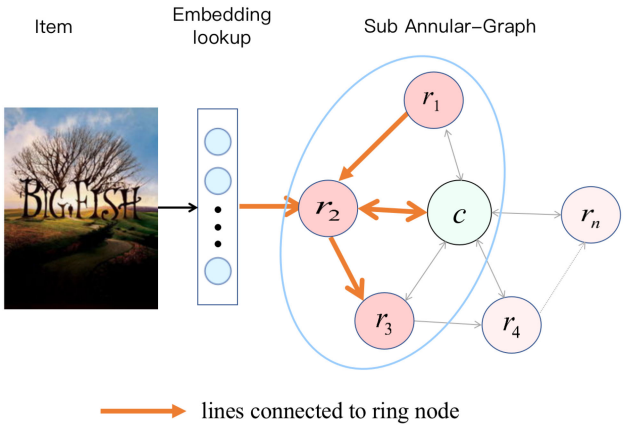


Fig. 3. When updating ring nodes, we construct a sub annular-graph to represent the context vector.

them to capture the local features and global features for the short-term preferences.

1) Annular-Graph and Annular-Graph Attention: This subsection introduces the main idea of annular-graph and annular-graph attention. An annular-graph is a representation of a graph of item sequences. As shown in Fig. 3, an annular-graph network is a combination of ring nodes and a central node. Each item in the behaviors sequence corresponds to a ring node in the annular-graph. The central node acts as a virtual hub node in this annular-graph, so any two ring nodes can be connected through the central node. Let $r^t \in \mathbb{R}^{L*d}$ denote all the ring nodes in each sequence and let $c^t \in \mathbb{R}^{1*d}$ denote the central node at step t.

What's more, the annular-graph attention has the ring connections and the radical connections, where the ring nodes reflect the local information of the sequence and the central node reflects the global information. When encoding the user interaction sequence, we first look up the embeddings to get the sequence embedding $X \in \mathbb{R}^{L*d}$. We initialize the ring nodes vector as $r^0 = X = [x_1, x_2, \ldots, x_L]$ and initialize the central node vector as $c^0 = \frac{1}{L}\sum_i x_i$. For a user, the ring nodes in the annular-graph represent the successive L items' token interacted. The central node c^t can be regarded as a hub that links all ring nodes r_i^t, $i \in (1, 2, \ldots, L)$ together. Adjacent ring nodes are connected to each other, and non-adjacent ring nodes are connected to each other through the central node (such as r_1^t and r_4^t in Fig. 3). Thus, this annular-graph attention can model the skip features of the sequential items, which means that the past behaviors may skip a few steps and impact their non-adjacent behaviors. What's more, the annular-graph attention has the ring connections and the radical connections, where the ring connections reflect the radical connections of the behaviors sequence and the central node summarizes the information of ring nodes to reflect the global features.

The block of annular-graph attention is scaled dot product attention. The input of scaled dot product attention consists of three parts like in Fig. 2. In our model, given the vectors $k \in \mathbb{R}^{l*d}$ and $v \in \mathbb{R}^{l*d}$ (l is the length of the annular-graph context vector), we can use a vector $q \in \mathbb{R}^{1*d}$ to select the values which are more important with the attention. We first map the node vector and annular-graph context vector to the same space and then put the three parts through the transformation of a nonlinear activation function to get the final parts. Then the annular-graph attention map (AGAtt) is calculated as follows:

$$AGAtt(q,k,v) = softmax(\frac{Relu(qW^q) * Relu(kW^k)^T}{\sqrt{d}}) * Relu(vW^v) \quad (1)$$

where the W^q, W^k, W^v are the weight matrices through the linear projection and trainable.

2) Annular-Graph Attention Based Global Features: Global features are the features modeling summarized information of sequential items. The central node can convey the global features of the sequence in our model. Thus, We first apply annular-graph attention to the central node to explore the graph for the global features. We splice the ring nodes vector and the central node vector to denote the annular-graph context vector of the central node. Then the following is the update function of the central node at the step t:

$$c^{t+1} = AGAtt(c^t, [r^t; c^t], [r^t; c^t]) \quad (2)$$

where the ";" represents the splicing of the vectors. Then after updating the central node at one step, we add a layer normalization function to prevent Internal Covariate Shift.

3) Annular-Graph Attention Based Local Features: Local features are the features modeling the adjacent items. We utilize annular-graph attention on the ring nodes to explore sub annular-graph for the local features. When updating the ring node, we construct a sub annular-graph to represent the annular-graph context vector of the ring node. For example, if we update r_2^t in the graph, we splice the connected nodes and itself as its graph context such as nodes connected by the orange lines to it in Fig. 3.

The i-th node r_i^t is updated by its annular-graph context vector as shown in Fig. 3. Then the following is the context vector h_i^t and the updating function of the i-th ring node at the step t:

$$h_i^t = [r_{i-1}^t; r_i^t; r_{i+1}^t; x_t^i; c^{t+1}] \quad (3)$$

$$r_i^{t+1} = AGAtt(r_i^t, h_i^t, h_i^t) \tag{4}$$

As the same as the updating of the central node, we also have a layer normalization here.

4) The Short-Term Preferences: After the update of all the ring nodes and the central node, we finally integrate them through a method to calculate the short-term preferences. To facilitate the integration of the local features and the global features, we first use a max pooling on the all ring nodes states $r^t = [r_1^t, r_2^t, \ldots, r_L^t]$ to get the r^{*t}. In addition, max pooling function can extract both significant information and compression reduction for the local features.

$$r^{*t} = MaxPooling(r^t) \tag{5}$$

Then we concatenate r^{*t} with the central node state to get the representation of the short-term preferences.

C. LFM Based Long-Term Preferences Modeling

Latent Factor Model (LFM) is a popular method of matrix factorization, which has been proved to be successful and effective in many recommender systems. The core idea of LFM is to connect users and items through latent features, and automatically cluster according to user behavior statistics. The LFM model can be divided into multi-dimensional classifications.

Thus, we utilize LFM to generate long-term preferences of users and items of d-dimension as $P \in \mathbb{R}^{d*M}$, $Q \in \mathbb{R}^{d*K}$ from user-item rating matrix R in our model. $R_{u,i}$ is a user u's rating to the item i.

The rating matrix is decomposed into two low-dimensional matrices by LFM. Then, the predicted value $\hat{R}_{u,i}$ of user u's rating of item i can be calculated by the following formula:

$$\hat{R}_{u,i} = \sum_d P_u^T Q_i \tag{6}$$

Because the rating matrix covers all the interactive information of the user's history, so the P_u generated by it can express long-term preferences very well.

D. Features Fusion and Prediction

In order to obtain user comprehensive preferences z, we fuse user short-term and long-term preferences through a multi-layer perception (MLP). z can be calculated by the following formula:

$$z = MLP([r^{*t}; c^t; P_u]) \tag{7}$$

Then the high-level user features z multiplies all the candidate item embeddings to get the scores y. When recommending, the items with the higher scores will be recommended to the user.

$$y = z * Q^T + b \tag{8}$$

where $Q \in \mathbb{R}^{d*K}$ denotes the all candidate items embeddings, $b \in \mathbb{R}^K$ denotes the bias. We project the score to the probabilities by:

$$P(X_t^u | X_{(t-1)}^u, X_{(t-2)}^u, \ldots, X_{(t-L)}^u) = \sigma(y) \tag{9}$$

where the σ is a sigmoid activation function.

E. The Model Training

The training procedure of AGSR includes two-stage strategy. The first stage is the latent factor model training and the second stage is the annular-graph attention network training.

1) The Latent Factor Model Training: The first stage is LFM training to get the users long-term preferences P and items features Q from user-item rating matrix R in our model. The rating matrix is decomposed into two low-dimensional matrices by LFM. The predicted value $\hat{R}_{u,i}$ of user u's rating of item i can be calculated by the following formula:

$$\hat{R}_{u,i} = P_u^T Q_i \tag{10}$$

In order to find appropriate matrices P and Q, the objective function is shown as follows :

$$\begin{aligned} \Psi(P,Q) &= \sum_{(u,i)\in Train} \left(R_{u,i} - \hat{R}_{u,i}\right)^2 \\ &= \sum_{(u,i)\in Train} \left(R_{u,i} - \sum_d P_u^T Q_i\right)^2 \end{aligned} \tag{11}$$

So we can learn the P and Q matrices directly by minimizing the objective function using the rating matrix. We choose stochastic gradient descent as the optimizer. Finally, the learned P is representation of user long-term preferences and the learned Q is item long-term preferences.

2) The Annular-Graph Attention Network Training: The second stage is the annular-graph attention network training. In this stage, the user long-term preferences and item features generated from the first stage is fixed.

In the second-stage training, given a positive sample and negative samples, we regard the ranking problem as a binary classification. Following previous works [7], for each user, a random sample of items he has not interacted with is taken as a negative sample. And for each target item i, three items j are randomly selected as negative samples according to the above rules in our experiments. Taking the negative logarithm of likelihood, we get the binary cross-entropy loss as the objective function:

$$loss = \sum_u \sum_i -\log\left(\sigma\left(y_i^u\right)\right) + \sum_{j\neq i} -log\left(1 - \sigma\left(y_j^u\right)\right) \tag{12}$$

In the above equation, the first term denotes the target items' negative logarithm of likelihood, and the second item denotes the negative samples' negative logarithm of likelihood. We train our model to learn the parameters by minimizing the above equation on the training set. We find the optimal hyperparameters (eg., d, N, L, learning rate) by doing a grid search on the validation set. Then, we optimize the proposed framework with adaptive moment estimation [54] (ADAM), which is an extension of the stochastic gradient descent (SGD) algorithm. ADAM enables efficient computing and requires less memory. In order to avoid the phenomenon of over-fitting, we introduce weight decay to make the weight shrink proportionally and we also introduce dropout. Instead of changing the network itself, dropout randomly sets some neurons to be 0, reducing the number of parameters.

TABLE II
STATISTICS OF THE DATASETS USED IN EXPERIMENTS

Datasets	Density	Users	Items	Avg. actions per user	Actions
Movielens	5.84%	6.0k	3.4k	165.50	993,000
Gowalla	0.29%	13.1k	14.0k	40.74	533,694

IV. EXPERIMENTS

We first introduce the datasets and metrics used in the experiment, then we give a brief overview of the comparison methods. Thirdly, we describe the details of the implementation and report the performance of our model. We end up with discussions on the influence of different parts of the model and the influence of the hyperparameters.

A. Dataset Description

We measure the performance of the proposed model and other baseline models on two benchmark datasets. Both of them have timestamps of the user-item interactions, which are indispensable for the sequential recommendation.

- MovieLens:[1] It is a commonly used dataset for the recommender system. We use user-movie rating data from this dataset. It contains approximately 100 million ratings from 6000 users who joined MovieLens in 2000 on nearly 4000 movies. The timestamp of this dataset is represented in seconds.
- Gowalla:[2] Gowalla is constructed by [55] and it is a location check-in data set of 6442890 users. We delete the latitude and longitude information and arrange the check-in locations id of each user in chronological order. Then we set the ratings of the interaction as 1.

As the previous work [7], [9], items that have been interacted by less than n users are deleted. n is 5 and 15 for MovieLens and Gowalla respectively. We sort the user behaviors in the dataset in chronological order. Then, we hold the 70% actions in each user's sequence as the training set, the next 10% actions to search the optimal settings of hyperparameters as the validation set. Then, to evaluate a model's performance, we hold the rest as the test set. To identify such datasets, we use density as the criterion. Data density refers to the ratio of elements with rating data to the whole matrix space in the user item matrix. We compute their density as follows:

$$Density = \frac{A}{M * K} \tag{13}$$

where A represents the number of actions in the dataset, M represents the user number, and K represents the number of items.

The detailed statistics of the datasets are shown in Table II. It shows that the MovieLens dataset is denser than the Gowalla dataset.

[1][Online]. Available: https://grouplens.org/datasets/movielens/1 m
[2][Online]. Available: https://snap.stanford.edu/data/loc-gowalla.html

B. Evaluation Metrics

We use Precision@N, Recall@N, and Mean Average Precision (MAP) to evaluate the performance. We recommend N items to a user u (denoted as $R(u)$) and denote the test set of items that the user u as $T(u)$: Precision describes the percentage of recommended items that are recommended correctly. Recall describes the percentage of the user's actual behaviors that recommends the correct items. The mean average precision for a set of users is the mean of the average precision scores for each user. The higher the MAP, the higher the correct items ranking in the recommendation, and the better the performance of the recommender system. As the same as MAP, other indicators are also better when higher. Precision@N and Recall@N are calculated by:

$$Precision@N = \frac{|R(u) \cap T(u)|}{N} \tag{14}$$

$$Recall@N = \frac{|R(u) \cap T(u)|}{|T(u)|} \tag{15}$$

We define $N \in (1, 5, 10)$ and then compute the average of these values of all users. The Average Precision (AP) is computed by:

$$AP = \frac{\sum_{N=1}^{|R(u)|} Precision@N * rel(N)}{|R(u)|} \tag{16}$$

where $rel(N) = 1$ if the N-th item in ***R(u)*** in ***T(u)***. Then the MAP is the mean of AP for a set of users.

C. Compared Methods

We compare the model AGSR with the following seven methods.

- POP. This is a non-personalized algorithm. It is designed to recommend the most popular items in the system for users without utilizing other information. It is the simplest, so its effect is not good.
- BPR [56]. Bayesian Personalized Ranking is a recommendation algorithm commonly used in the recommender system. The algorithm ranks items by the maximum posterior probability obtained from Bayesian analysis, so as to generate recommendations.
- FMC [3] and FPMC [3]. These models introduce a personalized transfer matrix based on the Markov chain, which enables the model to capture both short-term and long-term user preferences. In order to solve the sparse problem of transfer matrix, the matrix factorization model is introduced to reduce parameters and improve performance. FPMC is a personalized FMC, meaning that each user learns their own transformation matrix via this method.
- Fossil [57]. The model combines the similarity-based method with a high-order Markov chain to solve the challenge of the sparse datasets instead of LFM for modeling general user preferences.

TABLE III
PERFORMANCE COMPARISON ON THE TWO DATASETS

Dataset	Metric	POP	BPR	FMC	FPMC	Fossil	GRU4Rec	Caser	AGSR	Improve
MovieLens	Prec@1	0.128	0.1478	0.1748	0.2022	0.2306	0.2515	0.2502	**0.3015**	**19.89%**
	Prec@5	0.1113	0.1288	0.1505	0.1659	0.2	0.2146	0.2175	**0.2564**	**19.47%**
	Prec@10	0.1011	0.1193	0.1317	0.146	0.1806	0.1916	0.1991	**0.2295**	**15.27%**
	Recall@1	0.005	0.007	0.0104	0.0118	0.0144	0.0153	0.0148	**0.0197**	**28.76%**
	Recall@5	0.0213	0.0312	0.0432	0.0468	0.0602	0.0629	0.0632	**0.0784**	**24.05%**
	Recall@10	0.0375	0.056	0.0722	0.0777	0.1061	0.1093	0.1121	**0.136**	**21.30%**
	MAP	0.0687	0.0913	0.0949	0.1053	0.1354	0.144	0.1507	**0.1775**	**17.78%**
Gowalla	Prec@1	0.0517	0.164	0.1532	0.1555	0.1736	0.105	0.1961	**0.2105**	**7.30%**
	Prec@5	0.0362	0.0983	0.0876	0.0936	0.1045	0.0721	0.1129	**0.1201**	**6.37%**
	Prec@10	0.0281	0.0726	0.0656	0.0698	0.0782	0.0571	0.0833	**0.0872**	**4.47%**
	Recall@1	0.0064	0.025	0.0234	0.0256	0.0277	0.0155	0.031	**0.0372**	**2.00%**
	Recall@5	0.0257	0.0743	0.0648	0.0722	0.0793	0.0529	0.0845	**0.0915**	**8.28%**
	Recall@10	0.0402	0.1077	0.095	0.1059	0.1166	0.0826	0.1223	**0.1341**	**9.64%**
	MAP	0.0229	0.0767	0.0711	0.0764	0.0848	0.058	0.0928	**0.1036**	**11.63%**

- GRU4REC [6]. This is a GRU-based model for session-based recommendations. Two loss functions based on pair-wise ranking are used in this model. It performs well on the dense datasets.
- Caser [7]. This is a Convolution Sequence Embedding Recommendation Model, which uses lots of convolution filters to capture sequential features. It also captures user general preferences. It performs well on both dense and sparse datasets.
- AGSR. This model is proposed by us. It explores user sequential preferences via annular-graph attention. Our model not only captures the connection between adjacent items of a sequence, but also the interrelationship between one or even several items.

Among all of these baselines, FMC, FPMC, Fossil are based on Markov chain with matrix factorization. GRU4REC and Caser are neural network based approaches in the sequential recommendation.

D. Implementation Setting

We implement our model with Pytorch. All experiments are conducted on a Nvidia GTX 1080. We employ the grid search method to find the optimal hyperparameters in the validation set for all algorithms. These include hidden dimension d from 10, 20, 50, 100, and the learning rate from 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}. For AGSR itself, we set the target number from 1, 2, 3 and the input sequence length from 1, 2, 3, 4, 5, 6, 7, 8. We introduce weight decay into loss function, and it is set as 10^{-6}. Weight decay makes the weight decay to a smaller value, which reduces the problem of over fitting to a certain extent. For a fair comparison, when discussing, we utilize the control variable method. Only the single variable in the discussion is changed, and the other variables are set as optimal parameters.

E. Performance Comparison

Table III shows the best results under the optimal hyperparameters settings of the seven baselines and AGSR on the MovieLens and Gowalla datasets. The last column is the improvement of AGSR compared to the best baseline. From Table III, we can see the AGSR always outperforms other baselines in all evaluation metrics. It proves the effectiveness of our proposed model. We can observe that our model has significant improvement in terms of both recall rate and ranking quality (MAP). Especially with the MovieLens dataset, our precision, recall rates and MAP rates improve significantly. Precision@1 improves 19.89% compared with baseline models, Recall@1 improves 28.76% and MAP improves by 17.78% compared with the best baseline models. They all show that our model is very effective in improving performance. The Gowalla is more sparse than the Movilens1M dataset, so the performance is not improved as much on the Gowalla dataset as on the MovieLens dataset. However, even if the sequential intensity of sparse datasets is much lower, AGSR still improves the performance on the sparse dataset relative to other methods by over 7% on the average in all metrics.

By looking at other methods of comparison, we can also discover FPMC and Fossil are over FMC under all metrics on the two datasets. Besides, the performance under the POP model is always weaker than other models. It reveals that personalized information is indeed in need in the recommender system. And among the seven baselines, the performance of sequential models (eg., FPMC and Fossil) is always better than that of non-sequential models (eg., BPR, POP), revealing the effectiveness of considering the sequential pattern. The performance of GRU4REC is an approach to the performance of Caser on the Movilens1M dataset but much lower on the Gowalla dataset since Movilens1M dataset is more sequential than Gowalla. What's more, the GRU4REC is not a personalized recommender system, which is only a session-based recommender system. When comparing AGSR and Caser, we can see that the effectiveness of incorporating global features in an simple manner. It means with the help of annular-graph attention, AGSR provides a powerful capability to capture the sequential signal underlying users' behavior sequences.

In the next subsections, we set up some experiments to mine the effect of the hyperparameters deeply.

F. Ablation Study

We evaluate the contribution of each of AGSR's components, the long-term preference modeling part, the local features modeling part, the global features modeling part, to the overall

TABLE IV
MAP ABOUT ANALYSIS OF AGSR COMPONENTS

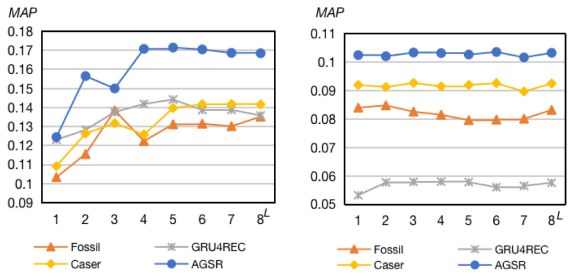
	MovieLens	Gowalla
AGSR-p	0.0864	0.0422
AGSR-l	0.1653	0.0838
AGSR-g	0.1470	0.0795
AGSR-lg	0.1703	0.1021
AGSR-pg	0.1692	0.0921
AGSR-pl	0.1718	0.0979
AGSR-plg	0.1775	0.1036

performance while keeping all hyperparameters as their optimal settings. The result is shown in Table IV for MovieLens and Gowalla. For $x \in \{p, l, g, lp, lg, pg, plg\}$, AGSR-$x$ denotes AGSR with the components x enabled. p denotes personalization, which uses LFM only, l denotes the local features modeling part, g denotes the global features modeling part, lp denotes both LFM and the local features modeling, lg denotes both the local features modeling and the global features modeling, and pg denotes both LFM and the global features modeling. Any missing component is represented by setting its corresponding to zero.

From both Table IV and Table III, we observe that the AGSR-p improves performance of recommendation compared with POP. POP is a method without any user features. When long-term features of users are added, more personalized features are considered. Thus personalized portrait of each user is fully carried out, so as to improve the accuracy of the recommender system.

However, AGSR-p performs the worst whereas AGSR-l and AGSR-g improve the performance significantly. This shows that treating sequential recommendations as the conventional recommendation will lose useful information. What's more, AGSR-l and AGSR-g both present the short-term preferences. Thus, it is also shown that the short-term preference has a stronger influence on the recommendation effect than the long-term preference.

To model the short-term preference in sequential recommendation, we propose the annular-graph attention network, which explores both local features and global features. From both Table IV and Table III, we observe that the performance of AGSR-l, AGSR-g and AGSR-lg surpass most of the compared methods. The local features can reflect the effect of the adjacent items and the global features can reflect the effect of the summarized information of sequential items. Thus, this proves the effectiveness of our proposed annular-graph network. We also find the AGSR-l significantly improves the effect of short-term preference modeling. Nevertheless, the AGSR-g doesn't achieve good performance alone, whose modeling ability is slightly inferior to Caser in MovieLens. However, it can extract skip features and summarized features which are not extracted by local features. Thus, when AGSR-g combine with AGSR-l to get the AGSR-lg, the AGSR-lg performs well on both datasets. Furthermore, the AGSR-p also can't achieve good performance alone as mentioned before, but it can extra explore the long-term preference. AGSR-plg achieves the best results by combining short-term preference and long-term preference. For both datasets, the best performance is achieved by jointly using all parts of AGSR, i.e., AGSR-plg.



(a) MAP for MovieLens (b) MAP for Gowalla

Fig. 4. Effects of sequence length L on MovieLens and Gowalla dataset. The y-axis is MAP and the x-axis is sequence length. (a) MAP for MovieLens and (b) MAP for Gowalla.

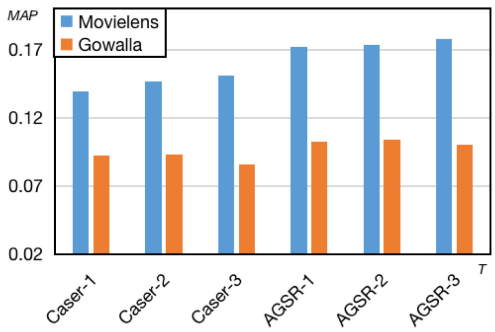
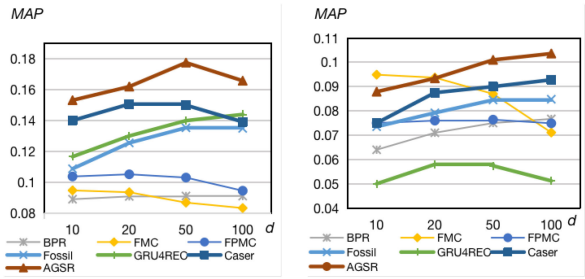


Fig. 5. Effects of target length on both MovieLens and Gowalla dataset. The y-axis is MAP and the x-axis is target length.

G. Discussions

The sequence length that we discuss in this part is the input length of the user's successive interactions. Fig. 4 shows the influence of the sequence length L. We only change the sequence length, from 1 to 8, and the other parameters are not changed. In this case, we set the length of target to 1. As can be seen from the figure, the selection of sequence length is related to datasets, and different datasets have different dependencies on sequences. We observe that in the MovieLens dataset, the larger the sequence length, the better the performance, whereas in the other dataset, the increasing sequence length does not improve performance. This is reasonable, since the sparse dataset has a weaker dependence on the sequence and the denser dataset has a stronger dependence. If the sequence length of the sparse dataset is too long, it will bring additional noise, which causes that the performance cannot improve. From the Fig. 4, we can observe that the performance of our model is better than other compared models in all sequence length. Besides, when the sequence length is set as 5, the MAP of AGSR is highest.

Target length can reflect the effect of skip features. skip features mean that users' interaction maybe influenced by the second to last item or the third to last item instead of the last item. Fig. 5 illustrates the performance of AGSR and Caser on two datasets when the length of the target is varied from 1 to 3 and other parameters remain unchangeable. It can be found from the



(a) MAP for MovieLens (b) MAP for Gowalla

Fig. 6. Effects of dim on MovieLens and Gowalla dataset. The y-axis is MAP and the x-axis is the dim of the embedding. (a) MAP for MovieLens and (b) MAP for Gowalla.

Fig. 5 that the overall performance of AGSR is better than that of Caser. Caser-1, Caser-2, and Caser-3 represent target number T at 1, 2, and 3 respectively. And AGSR-1, AGSR-2, AGSR-3 respectively represent target number T at 1, 2 and 3. On the MovieLens dataset, the performance improves with the increase of the target number. However, on the Gowalla dataset, the target number at 2 is the best hyperparameter on both models.

Then, we only change the latent dimension of the embedding, the other parameters are fixed. Fig. 6 shows the experimental results of the different latent dimensions on the two datasets. We observe from the figures that the performance of our model is better than other models in all dimensions. In the MovieLens dataset, increasing the dimensions does not improve the performance of the system. On the contrary, the larger latent dimension may lead to overfitting. However, on the Gowalla dataset, we get better results as the dimensions grow. The reason is that the Gowalla dataset is too sparse so that it needs a larger hidden vector space to represent the information. Thus, we choose the optimal parameter when the dimension value is 50 on the MovieLens dataset and 100 on the Gowalla dataset.

V. Conclusion

In this paper, we propose a novel top-N sequential recommendation based on annular-graph attention and latent factor model. We introduce an annular-graph network to model sequence patterns. Then apply annular-graph attention to this network. AGSR not only captures the connection between adjacent items of a behaviors sequence via the sub annular-graph of a sequence, but also the interrelationship among nonadjacent sequential items of a behaviors sequence and the summarized features of the sequential items via the integral graph. From the experiments, we observe that our model outperforms the state-of-the-art methods on two real-world datasets. Particularly, our precision, recall rates, and MAP rates improve significantly on the MovieLens dataset. Precision@1 improves 19.89% compared with baseline models, Recall@1 improves 28.76% and MAP improves by 17.78% compared with the best baseline models.

In the future, we will make better use of the graph information of users and items to solve the problem of sparse datasets and reduce complexity. In addition, we believe that our model is robust and can be used for other recommendation tasks.

References

[1] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. WWW, ACM*, 2001, pp. 285–295.
[2] Y. Koren, R. M. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Comput.*, vol. 42, no. 8, pp. 30–37, 2009.
[3] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proc. WWW,* ACM, 2010, pp. 811–820.
[4] X. Qian, H. Feng, G. Zhao, and T. Mei, "Personalized recommendation combining user interest and social circle," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1763–1777, Jul. 2014.
[5] X. Gao *et al.*, "Hierarchical attention network for visually-aware food recommendation," *IEEE Trans. Multim.*, vol. 22, no. 6, pp. 1647–1659, Jun. 2020.
[6] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *ICLR, Conference Track Proceedings*, San Juan, Puerto Rico, May 2-4, 2016.
[7] J. Tang and K. Wang, "Personalized top-N sequential recommendation via convolutional sequence embedding," in *WSDM*, Y. Chang, C. Zhai, Y. Liu, and Y. Maarek, Eds. ACM, Marina Del Rey, CA, USA, pp. 565–573, Feb. 5-9, 2018.
[8] T. Donkers, B. Loepp, and J. Ziegler, "Sequential user-based recurrent neural network recommendations," in *Proc. RecSys, ACM*, 2017, pp. 152–160.
[9] H. Jing and A. J. Smola, "Neural survival recommender," in *Proc. WSDM, ACM*, 2017, pp. 515–524.
[10] D. Agarwal and B. Chen, "Regression-based latent factor models," in *Proc. SIGKDD*ACM, 2009, pp. 19–28.
[11] G. Zhao, Z. Liu, Y. Chao, and X. Qian, "CAPER: Context-aware personalized emoji recommendation," *IEEE Trans. Knowl. Data Eng.*, doi: 10.1109/TKDE.2020.2966971.
[12] S. Jiang, X. Qian, T. Mei, and Y. Fu, "Personalized travel sequence recommendation on multi-source big social media," *IEEE Trans. Big Data*, vol. 2, no. 1, pp. 43–56, Mar. 2016.
[13] G. Zhao, P. Lou, X. Qian, and X. Hou, "Personalized location recommendation by fusing sentimental and spatial context," *Knowl. Based Syst.*, vol. 196, p. 105849, 2020.
[14] Z. Cheng and J. Shen, "On effective location-aware music recommendation," *ACM Trans. Inf. Syst.*, vol. 34, no. 2, pp. 13:1–13:32, 2016.
[15] Z. Cheng, J. Shen, and T. Mei, "Just-for-me: An adaptive personalization system for location-aware social music recommendation," in *Proc. SIGIR,* ACM, 2014, pp. 1267–1268.
[16] C. Yang, S. Hsu, K. Hua, and W. Cheng, "Fuzzy personalized scoring model for recommendation system," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 1577–1581.
[17] S. C. Hidayati, C. Hsu, Y. Chang, K. Hua, J. Fu, and W. Cheng, "What dress fits me best?: Fashion recommendation on the clothing style for personal body shape," in *Proc. MM ACM*, 2018, pp. 438–446.
[18] D. Liang, J. Altosaar, L. Charlin, and D. M. Blei, "Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence," in *Proc. 10th ACM Conf. Recommender Syst.*, ACM, 2016, pp. 59–66.
[19] Y. Koren, "Collaborative filtering with temporal dynamics," in *Proc. SIGKDD*, ACM, 2009, pp. 447–456.
[20] R. Pan *et al.*, "One-class collaborative filtering," in *Proc. IEEE Comput. Soc. ICDM*, 2008, pp. 502–511.
[21] N. Srebro and T. S. Jaakkola, "Weighted low-rank approximations," in *Proc. Int. Conf. Mach. Learn.* AAAI Press, 2003, pp. 720–727.
[22] Z. Cheng, J. Shen, L. Zhu, M. S. Kankanhalli, and L. Nie, "Exploiting music play sequence for music recommendation," in *Proc. IJCAI*, 2017, pp. 3654–3660.
[23] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, "Personalized ranking metric embedding for next new POI recommendation," in *Proc. IJCAI AAAI Press*, 2015, pp. 2069–2075.
[24] C. Hsieh, L. Yang, Y. Cui, T. Lin, S. J. Belongie, and D. Estrin, "Collaborative metric learning," in *Proc. WWW ACM*, 2017, pp. 193–201.
[25] Y. Tay, L. A. Tuan, and S. C. Hui, "Latent relational metric learning via memory-based attention for collaborative ranking," in *Proc. WWW ACM*, 2018, pp. 729–739.
[26] S. Zhang, L. Yao, C. Huang, X. Xu, and L. Zhu, "Position and distance: Recommendation beyond matrix factorization," CoRR, vol. abs/1802.04606, 2018.
[27] R. He, W. Kang, and J. J. McAuley, "Translation-based recommendation," in *Proc. RecSys ACM*, 2017, pp. 161–169.

[28] R. Salakhutdinov, A. Mnih, and G. E. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proc. Int. Conf. Mach. Learn.*, ACM, ACM, 2007, vol. 227, pp. 791–798.
[29] C. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks," in *Proc. WSDM, ACM*, 2017, pp. 495–503.
[30] Y. Wu, K. Li, G. Zhao, and X. QIAN, "Personalized long- and short-term preference learning for next POI recommendation," *IEEE Trans. Knowl. Data Eng.*, doi: 10.1109/TKDE.2020.300253.
[31] T. Bansal, D. Belanger, and A. McCallum, "Ask the GRU: Multi-task learning for deep text recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, Boston, MA, ACM, 2016, pp. 107–114.
[32] H. Bharadhwaj and S. Joshi, "Explanations for temporal recommendations," *Künstliche Intell.*, vol. 32, no. 4, pp. 267–272, 2018.
[33] J. Zhang, Y. Yang, L. Zhuo, Q. Tian, and X. Liang, "Personalized recommendation of social images by constructing a user interest tree with deep features and tag trees," *IEEE Trans. Multim.*, vol. 21, no. 11, pp. 2762–2775, Nov. 2019.
[34] T. Bai, P. Du, W. X. Zhao, J. Wen, and J. Nie, "A long-short demands-aware model for next-item recommendation," *arXiv preprint:1903.00066*, 2019.
[35] W. Chu and Y. Tsai, "A hybrid recommendation system considering visual information for predicting favorite restaurants," *World Wide Web*, vol. 20, no. 6, pp. 1313–1331, 2017.
[36] D. H. Kim, C. Park, J. Oh, and H. Yu, "Deep hybrid recommender systems via exploiting document context and statistics of items," *Inf. Sci.*, vol. 417, pp. 72–87, 2017.
[37] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. EMNLP ACL*, 2014, pp. 1746–1751.
[38] X. Zhang *et al.*, "Trip outfits advisor: Location-oriented clothing recommendation," *IEEE Trans. Multim.*, vol. 19, no. 11, pp. 2533–2544, Nov. 2017.
[39] P. Wang, Y. Fan, L. Xia, W. X. Zhao, S. Niu, and J. Huang, "KERL: A knowledge-guided reinforcement learning model for sequential recommendation," in *Proc. SIGIR*ACM, 2020, pp. 209–218.
[40] P. Rodríguez, D. V. Dorta, G. Cucurull, J. M. Gonfaus, F. X. Roca, and J. Gonzàlez, "Pay attention to the activations: A modular attention mechanism for fine-grained image recognition," *IEEE Trans. Multim.*, vol. 22, no. 2, pp. 502–514, Feb. 2020.
[41] L. Li, S. Tang, Y. Zhang, L. Deng, and Q. Tian, "GLA: Global-local attention for image description," *IEEE Trans. Multim.*, vol. 20, no. 3, pp. 726–737, Mar. 2018.
[42] F. Lyu, Q. Wu, F. Hu, Q. Wu, and M. Tan, "Attend and imagine: Multi-label image classification with visual attention and recurrent neural networks," *IEEE Trans. Multim.*, vol. 21, no. 8, pp. 1971–1981, Aug. 2019.
[43] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3242–3250.
[44] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR, Conf. Track Proc.*, San Diego, CA, USA, May 7-9 2015.
[45] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. 30: Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[46] Q. Wu *et al.*, "Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems," in *Proc. WWW ACM*, 2019, pp. 2091–2102.
[47] H. Ying *et al.*, "Sequential recommender system based on hierarchical attention networks," in *Proc. IJCAI*, 2018, pp. 3926–3932.
[48] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T. Chua, "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention," in *Proc. SIGIR*, ACM, 2017, pp. 335–344.
[49] X. He, Z. He, J. Song, Z. Liu, Y. Jiang, and T. Chua, "NAIS: Neural attentive item similarity model for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2354–2366, Dec. 2018.
[50] Y. Tay, A. T. Luu, and S. C. Hui, "Multi-pointer co-attention networks for recommendation," in *Proc. KDD ACM*, 2018, pp. 2309–2318.
[51] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proc. CIKM ACM*, 2017, pp. 1419–1428.
[52] J. Wang, K. Ding, L. Hong, H. Liu, and J. Caverlee, "Next-item recommendation with sequential hypergraphs," in *Proc. SIGIR*ACM, 2020, pp. 1101–1110.
[53] Y. Gong and Q. Zhang, "Hashtag recommendation using attention-based convolutional neural network," in *Proc. IJCAI IJCAI/AAAI Press*, 2016, pp. 2782–2788.
[54] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Conf. Track ICLR*, San Diego, CA, USA, 2015.
[55] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proc. SIGKDD*, ACM, 2011, pp. 1082–1090.
[56] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, Montreal. AUAI Press, 2009, pp. 452–461.
[57] R. He and J. J. McAuley, "Fusing similarity models with markov chains for sparse sequential recommendation," in *Proc. IEEE Comput. Soc. ICDM*, 2016, pp. 191–200.

Junmei Hao received the B.S. degree from Dalian Maritime University, Dalian, China, in 2018. She is currently working toward the M.S. degree with the School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, China.

Yujie Dun (Member, IEEE) received the Ph.D. degree in information and communication engineering from Xi'an Jiaotong University, Xi'an, China, in 2016. After the Ph.D. degree, she visited as a Visiting Scholar with Washington University in St. Louis, St. Louis, MO, USA, during 2017–2018, and a Postdoctoral Researcher with Washington University in St. Louis, during 2018–2019. She is currently an Associate Professor with the School of Information and Communication, Xi'an Jiaotong University. Her research interests include audio/speech signal processing and coding, statistical signal processing and modeling, biomedical signal processing, and machine learning.

Guoshuai Zhao received the B.S. degree from Heilongjiang University, Harbin, China, in 2012, the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2015 and 2019, respectively. He is currently an Assistant Professor with the School of Software Engineering, Xi'an Jiaotong University. He is mainly engaged in the research of social media big data analysis and recommender systems.

Yuxia Wu received the B.S. degree from Zhengzhou University, Henan, China, in 2014 and the M.S. degree from the Fourth Military Medical University, Xi'an, China, in 2017. She is currently working toward the Ph.D. degree with Xi'an Jiaotong University, Xi'an, China. She is mainly engaged in the research of multimedia mining and recommender systems.

Xueming Qian (Member, IEEE) received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree from the School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, China, in 2008. From 1999 to 2001, he was an Assistant Engineer with Shannxi Daily. Since 2008, he has been an Associate Professor with the School of Electronics and Information Engineering, Xi'an Jiaotong University. He is currently an Associate Professor with the School of Electronics and Information Engineering, Xi'an Jiaotong University. He is the Director of SMILES LAB. From August 2010 to March 2011, he was a Visiting Scholar with Microsoft Research Asia. His research interests include social media big data mining and search. He is a member of the ACM, and Senior Member of CCF. He was awarded the Microsoft Fellowship in 2006.