Adaptive Attention-Aware Gated Recurrent Unit for Sequential Recommendation

Anjing Luo[1], Pengpeng Zhao[1,5](✉), Yanchi Liu[2], Jiajie Xu[1,5], Zhixu Li[1],
Lei Zhao[1], Victor S. Sheng[3], and Zhiming Cui[4]

[1] Institute of Artificial Intelligence, School of Computer Science and Technology,
Soochow University, Suzhou, China
ajluo@stu.suda.edu.cn, {ppzhao,xujj,zhixuli,zhaol}@suda.edu.cn
[2] Rutgers University, New Brunswick, USA
yanchi.liu@rutgers.edu
[3] University of Central Arkansas, Conway, USA
ssheng@uca.edu
[4] Suzhou University of Science and Technology, Suzhou, China
zmcui@mail.usts.edu.cn
[5] Neusoft Corporation, Shenyang, China

Abstract. Due to the dynamic and evolutionary characteristics of user interests, sequential recommendation plays a significant role in recommender systems. A fundamental problem in the sequential recommendation is modeling dynamic user preference. Recurrent Neural Networks (RNNs) are widely adopted in the sequential recommendation, especially attention-based RNN becomes the state-of-the-art solution. However the existing fixed attention mechanism is insufficient to model the dynamic and evolutionary characteristics of user sequential preferences. In this work, we propose a novel solution, Adaptive Attention-Aware Gated Recurrent Unit (3AGRU), to learn adaptive user sequential representations for sequential recommendation. Specifically, we adopt an attention mechanism to adapt the representation of user sequential preference, and learn the interaction between steps and items from data. Moreover, in the first level of 3AGRU, we construct adaptive attention network to describe the relevance between input and the candidate item. In this way, a new input based on adaptive attention can reflect users' diverse interests. Then, the second level of 3AGRU applies adaptive attention network to hidden state level to learn a deep user representation which is able to express diverse interests of the user. Finally, we evaluate the proposed model using three real-world datasets from various application scenarios. Our experimental results show that our model significantly outperforms the state-of-the-art approaches on sequential recommendation.

Keywords: Adaptive attention mechanism · GRU · Recommender system

G. Li et al. (Eds.): DASFAA 2019, LNCS 11447, pp. 317–332, 2019.
https://doi.org/10.1007/978-3-030-18579-4_19

1 Introduction

With the explosion of Web information, recommender system plays a more and more significant role in online services, where capturing users' preferences is critical. Due to the intrinsically dynamic and evolving characteristics of user interests, sequential recommendation has attracted a lot of attention in recommender systems. A fundamental problem in the sequential recommendation is how to model dynamic and evolutionary user preferences to satisfy user needs better [19].

For modeling sequential patterns, Factorizing Personalized Markov Chain (FPMC) model was proposed to factorize user-specific transition matrix by the Markov Chain (MC) [19]. A significant drawback of MC-based solutions is that they adopt the static representation for user's interests. With the success of neural networks in many application domains, recurrent neural networks (RNNs) are widely adopted in the sequential recommendation, such as session-based recommendation [8], next-basket and next-item recommendations [12,13].

Besides the essential dynamic and evolutionary characteristics, the user's interests are also diverse (not singular) in the same period, and they usually involve multiple fields. For example, we may find that a user who likes reading books about deep learning also likes purchasing household appliances. Although various extensions of RNN, like LSTM and GRU, can better capture the long-term dependency of user preference, they assume temporal dependence has a monotonic change with each step. In other words, the current item is more significant than the previous one to predict the next one, which is not always true. Attention network based RNN can solve the above problem, where the attention mechanism can automatically assign different influences to previous items, and achieve the state-of-the-art performance [2,12,24].

However, the recommendation process can be too dynamic for the attention-based solutions to capture. A previous item may play a different role and exhibit different influences in choosing next items of different types due to the specialty. Nevertheless, existing attention-based RNN solutions use a fixed strategy to aggregate the influences of previous step items. As such, they are insufficient to capture the dynamic process of users' diverse sequential decision makings, resulting in a suboptimal solution. Let us illustrate the above problem with an example. Suppose a user bought three items e.g., Python book, iPad, RecommenderSystem book in a time order. Subsequently, the user purchases an iWatch, and the sequential history is finalized as Python book, iPad, RecommenderSystem book, iWatch. If we take the first three items as the context and the last one as the target to recommend, existing fixed attention-based methods may suggest books like deep learning books due to the more influence of book items. However, the choice of the target item iWatch may depend on the first item (iPad). In this case, recommender systems should pay more attention to the iPad when computing the score of candidate recommendation item iWatch, because iPad may be more related to the next choice iWatch. This example shows the influence of previous items may be more related to candidate items. Therefore, it may not be optimal and realistic to recommend next items with a fixed attention mechanism.

In this paper, to express the dynamics and diversity of users' interests, we develop an Adaptive Attention-Aware Gated Recurrent Unit model (3AGRU) for sequential recommendation. First of all, we leverage the strength of the recurrent architecture of GRU to capture complex long-term dependencies and that of the attention network to discover the local sequential pattern. More importantly, motivated by the observation of user behaviors, we facilitate GRU with a novel adaptive item-level attention mechanism to devise a deep adaptive user sequential interest representation. Unlike fixed attention-based user representations, adaptive user representations dynamically adapt to locally activated items. The first level of 3AGRU is conducted on the input level to make the hidden state stronger. We input a new input generated by current item and adaptive attention which considering the information of candidate item to GRU. The second level of 3AGRU is executed on the hidden state level to utilize adaptive attention network to learn a deep adaptive user representation. Accordingly, the hidden state strengthened by the adaptive input is further intensified, which is more in line with the users' interests. The contributions of this work can be summarized as follows:

- We introduce a novel adaptive attention-aware recurrent neural network for adaptive user representation, which adaptively combines both user's long-term and short-term preferences to generate a high-level hybrid representation of users.
- The adaptive contextual attention networks on input level and hidden state level are further integrated to improve the performance of sequential recommendation.
- We compare our model 3AGRU with state-of-the-art methods and verify the superiority of 3AGRU through quantitative analysis on three large real-world datasets. The experimental results reveal that our method is capable of leveraging user historical records effectively.

In the following part of the paper, we first introduce the related work in Sect. 2, and define the problem in Sect. 3. Then, we illustrate our framework in Sect. 4. In Sect. 5, we verify the effectiveness of our method with experimental results. Finally, the conclusions and outlooks of this work are presented in Sect. 6.

2 Related Work

Our Adaptive Attention-Aware Gated Recurrent Unit (3AGRU) is proposed for sequential recommendation. Therefore, in this section, we discuss related work from two aspects, i.e., general recommendation and sequential recommendation.

2.1 General Recommendation

General recommendation recommends items through modeling the users' general tastes from their historical interactions. The key idea is collaborative filtering (CF), which can be further categorized into memory-based CF and model-based

CF [21]. The memory-based CF provides recommendations by finding k-nearest-neighbours of users or items based on similarity [15], while the model-based CF tries to factorize the user-item correlation matrix for recommendation [9,11]. [17] introduces weights to user-item pairs, and optimizes the factorization with both least-square and hinge-loss criteria. [18] optimizes the latent factor model with a pairwise ranking loss in a Bayesian framework. [26] optimizes cross-entropy loss between the true pairwise preference ranking and predicted pairwise preference ranking for each user. General recommendation can capture users' general taste, but can hardly adapt its recommendations directly to users' recent interactions without modeling sequential behaviors.

2.2 Sequential Recommendation

Sequential recommendation views the interactions of a user as a sequence and aims to predict which item the user will interact with next. A typical solution to this setting is to compute an item-to-item relational matrix, whereby the most similar (the nearest) items to the last interacted one are recommended to users. For example, Markov Chain based methods estimate an item-to-item transition probability matrix and use it to predict the probability of the next item given the last interaction of a user [19,20]. [20] presents a recommender based on Markov decision processes and shows that a predictive Markov Chain model is effective for next basket prediction. [19] combines a factorization method and Markov Chains, using the factorization method to model the user general taste and Markov Chains to mine user sequential patterns. Hierarchical representation model combines the last action information with the general user interest to model user representations for next basket recommendation [22]. For these Markov Models, it is difficult to model the long-range dependence. Prod2Vec, inspired by word embedding technique [16], learns distributed item representations from the interaction sequences and uses them to compute a cosine similarity matrix [5]. [6] assumes that items are embedded into a "transition space" where each user is modeled by a translation vector.

Recently, Recurrent Neural Network (RNN), a state-of-the-art deep learning method for sequence modeling, is shown to be effective in capturing sequential user behavioral patterns [3,8,27]. Different from previous methods, applying RNN to sequential recommender introduces the capability of modeling the whole historical interactions. [8] implements an improved version of the GRU network for session-based recommendation, which utilizes a session-parallel mini-batch training process and employs ranking-based loss functions for learning the model. DREAM [25] utilizes pooling to summarize the basket of embedded items and then feeds into vanilla RNN to solve next basket recommendation. [10] integrates the RNN-based networks with knowledge base enhanced Key-Value Memory Network (KV-MN) to capture sequential user preference and attribute-level user preference. [14] explains a K-plet Recurrent Neural Network for accommodating multiple sequences jointly to capture global structure and localized relationships at the same time.

However, our proposed 3AGRU model differs from existing attention-based RNN solutions. These solutions use a fixed attention mechanism to aggregate influence of previous items while 3AGRU facilitates relevance between previous items and candidate items to devise a deep adaptive user sequential interest representation. Thus, our model 3AGRU can express the dynamics and diversity of users' interests.

3 Problem Statement

In this section, we first introduce basic notations that will be used in this paper. Let $\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$ denote a set of users and $\mathcal{I} = \{i_1, i_2, ..., i_{|\mathcal{I}|}\}$ denote a set of items, where $|\mathcal{U}|$ and $|\mathcal{I}|$ are the total numbers of users and items, respectively. Each user u is associated with a sequence of some items from $\mathcal{I}$, $\mathcal{I}^u = \{i_1^u, \ldots, i_t^u, \ldots, i_n^u\}$, where $i_t^u \in \mathcal{I}$ and n is the number of items interacted with user u. The index t for i_t^u denotes the relative time index, not the absolute timestamp.

With the above notations, we define the sequential recommendation task as follows. In this work, we focus on the case of implicit action feedback. Given a user $u's$ history transaction sequence I^u, we aim to predict a list of items that the user would probably interested in the near future.

4 Adaptive Attention-Aware Gated Recurrent Unit

In this section, we will display the process of our proposed Adaptive Attention-Aware Gated Recurrent Unit model (3AGRU) for sequential recommendation in details. Firstly, we introduce the basic GRU model. Secondly, we present the overall architecture of 3AGRU and then introduce the details of each layer of it. Finally, we offer the optimization procedures.

4.1 Gated Recurrent Unit

As a variant of LSTM, GRU solves the problem of long-term dependence of RNN well and simplifies the structure of LSTM. It contains a reset gate r_t and an update gate z_t. Besides, it has the candidate state $\tilde{h}_t$ which uses r_t to control the inflow of the last hidden state containing previous information. If the reset gate is approximately zero, the last hidden state will be discarded. r_t determines how much information was forgotten in the past. h_t is the hidden state which uses z_t to update the last hidden state h_{t-1} and the $\tilde{h}_t$. The update gate z_t controls the importance of the previous hidden state at the current moment. The formulas are as follows.

$$r_t = \sigma(x_t W_{xr} + h_{t-1} W_{hr} + b_r) \tag{1}$$

$$z_t = \sigma(x_t W_{xz} + h_{t-1} W_{hz} + b_z) \tag{2}$$

$$\tilde{h}_t = \tanh(x_t W_{xh} + r_t \odot h_{t-1} W_{hh} + b_h) \tag{3}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \tag{4}$$

In these formulas, x_t is the input vector while t is the time step, W_{xr}, W_{hr}, W_{xz}, W_{hz}, W_{xh}, W_{hh}, b_r, b_z, b_h represent the transition matrices and bias of the input and hidden levels in r_t and z_t respectively, $\odot$ is the element-wise product between two vectors, and the sigmoid function $\sigma(x)$ is used to do nonlinear projection. The last h_t is the final representation of the sequence.

4.2 3AGRU

The 3AGRU model is a hierarchical structure that consists of input level, GRU, hidden state level and output level. What we will do is to apply adaptive attention network to the input level and the hidden state level of GRU. The modeling architecture is shown in Fig. 1. In the input level, the sparse inputs are embedded into dense representations to get memory component C which contains information of all items of user u and v_j which is the representation of a candidate item i_j^u. Then the adaptive attention network is employed on C as the input of GRU. In this way, the input of GRU at each time step has different weights, and the larger the weight value is, the more similar i_j^u and the corresponding item are. In the hidden state level, the adaptive attention network is also applied to the set of the hidden state at each time step to get the new hidden state with different weights. The different weights denote the relevance between the candidate item and the hidden state of GRU at each time step. With the information passed to the final hidden state, the final hidden state is used to be the adaptive user representation. Finally, we use the final hidden state with attention to measure the user's preference score over item i_j^u. Then, we will introduce each part of our model in detail.

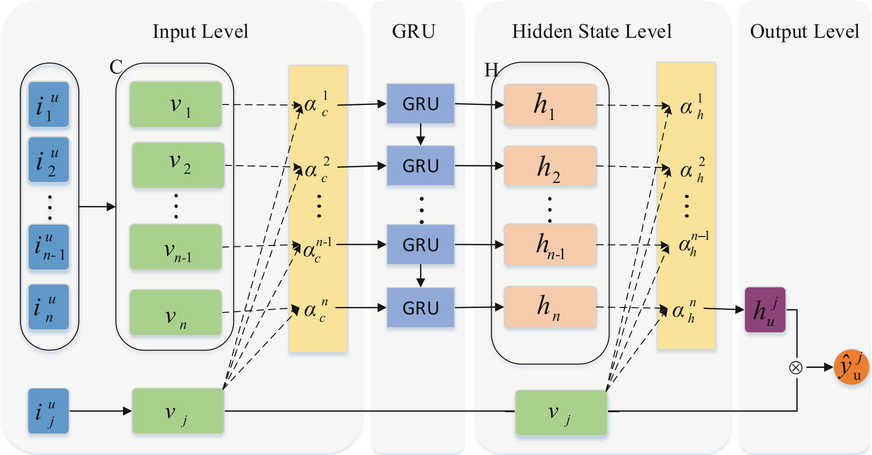


Fig. 1. The architecture of 3AGRU. The weight in Adaptive Attention Network indicates the correlation between the corresponding input item and the candidate item.

Adaptive Attention Network in Input Level. The input level is composed of embedding layer and adaptive attention network. As we all known, both original input and candidate item have limited representation ability that is similar to discrete words symbols in natural language processing. Therefore, our model 3AGRU will use item embedding to project sparse inputs to dense representations. Formally, let $C \in \mathbb{R}^{k \times n}$ represent dense input whose column corresponds to a representation of an item in $\mathcal{I}^u$ and $v_j \in \mathbb{R}^{k \times 1}$ represent a candidate item i_j^u. Hence the matrix C characterizes the user's interests and k is the dimensionality of the latent factor. Although C stores all item information of user u, our goal is to learn a deep representation of user u through the adaptive attention. Since items in $\mathcal{I}^u$ are diverse and only a subset of $\mathcal{I}^u$ is relevant to candidate item i_j^u, we introduce an item-level attention mechanism named as adaptive attention network on the memory component C to capture items in $\mathcal{I}^u$ relevant to item i_j^u. More items in $\mathcal{I}^u$ with high relevance scores to the item i_j^u denotes that the user u is more likely interested in item i_j^u. The adaptive attention network in the input level which is used to measure the relevance scores between i_j^u and each item representation in C is defined as:

$$\alpha_c^u = \frac{exp(C^T v_j)}{\sum_{i_m \in \mathcal{I}/\mathcal{I}^u} exp(C^T v_m)} \tag{5}$$

Defined in this way, $\alpha_c^u \in \mathbb{R}^{n \times 1}$ is the adaptive attention column vector of user u for item i_j^u in the input level of GRU. The larger value in α_c^u, the higher relevance between item i_j^u and a corresponding item in $\mathcal{I}^u$ and a higher weight for deriving interests of user u for item i_j^u. Therefore, the input of GRU is formed as:

$$V^u = \alpha_c^u \odot C \tag{6}$$

where $\odot$ is the element-wise product and $V^u = \{v_1^u, v_2^u, \ldots, v_n^u\}$. The formulas of GRU will be rewritten as:

$$r_t = \sigma(v_t^u W_{xr} + h_{t-1} W_{hr} + b_r) \tag{7}$$

$$z_t = \sigma(v_t^u W_{xz} + h_{t-1} W_{hz} + b_z) \tag{8}$$

$$\tilde{h}_t = \tanh(v_t^u W_{xh} + r_t \odot h_{t-1} W_{hh} + b_h) \tag{9}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \tag{10}$$

Let $H = \{h_1, h_2, \ldots, h_n\}$ represent the set of intermediate output of GRU and v_t^u denote the item in V^u at time step t. In this way, the h_t contains the information of relevance of candidate item i_j^u and the previous input items.

Adaptive Attention Network in Hidden State Level. The hidden state level is to further relieve the monotonic assumption problem and give different weights to current hidden state, especially the final hidden state. Since the RNN-based methods mostly employ the final hidden state as the user's representation then making recommendations to users, we use the adaptive attention again to get a deep adaptive user representation.

Similar to the adaptive attention in the input level, H characterizes the user's representation at each time step. With the H and v_j, we measure the relevance between each hidden state in H and item i_j^u. Thus, the adaptive attention network in hidden state level is formed as:

$$\alpha_h^u = \frac{exp(H^T v_j)}{\sum_{i_m \in \mathcal{I}/\mathcal{I}^u} exp(H^T v_m)} \tag{11}$$

The larger value in α_h^u, the higher relevance between item i_j^u and corresponding hidden state in H. Consider that GRU can transmit previous information to the current state, it can enable the final hidden state to focus on the diversified interests of users through giving different weights to the mediate hidden state. The formula of the hidden state set which contains different weights is as follows:

$$H^u = \alpha_h^u \odot H \tag{12}$$

where $\odot$ is the element-wise product and the final hidden state with attention is the adaptive user representation which is adaptively focusing on items in $\mathcal{I}^u$ activated by item i_j^u.

Output Level. In the output level, let h_u^j denote the final hidden state with attention of GRU. We use h_u^j to represent the deep adaptive user representation. Recall that the adaptive user representation h_u^j is obtained by placing the adaptive attention on the memory component C and the set of hidden state H. Given the representation of a candidate item i_j^u, we can compute the user's preference score over item i_j^u as:

$$\hat{y}_u^j = (h_u^j)^T v_j \tag{13}$$

$\hat{y}_u^j$ is the preference score which reflects the preference of user u for item i_j^u.

4.3 Network Learning

The goal of our model is to recommend a ranked list of items that satisfy user's interests. To train 3AGRU optimized for ranking inspired by BPR, we formalize the training data $\mathcal{D}$ by (u, p, q) triples as:

$$\mathcal{D} = \{(u, p, q) | u \in \mathcal{U} \wedge p \in \mathcal{I}^u \wedge q \in \mathcal{I}/\mathcal{I}^u\} \tag{14}$$

where u denotes the target user, p and q represent the positive and negative items respectively. Item p is from user's history I^u while item q is randomly chosen from the rest items. Similar to BPR, instead of scoring single items, we use item pairs to calculate user's preference for positive and negative items. The objective function minimizes the following formula:

$$\mathcal{L} = \arg\min_{\theta} \sum_{(u,p,q) \in D} - \ln(\sigma(\hat{y}_u^p - \hat{y}_u^q)) + \frac{\lambda}{2} \|\theta\|^2 \tag{15}$$

where $\hat{y}_u^p$ and $\hat{y}_u^q$ represent the user u's preference score over item p and q. θ represents all of the model parameters that are learned while λ is the regularization terms. The sigmoid function σ maps user u's preference score of item p and q into probabilities.

5 Experiment

In this section, we first explain the setup of experiments, and then analyze the results of experiments. After that, we explore the advantages of the adaptive attention network over the fixed attention mechanism. Finally, we present the influence of hyper-parameters. We aim to answer these questions as follows:

Q1: What's the performance of 3ARGU, comparing to other state-of-the-art methods?
Q2: What's the advantage of adaptive attention, comparing to fixed attention?
Q3: How do the parameters affect model performance, such as the dimension of latent vectors and the regularization terms?

5.1 Experimental Setup

Datasets. We conduct the experiments on the three datasets with different kinds of items. The basic statistics of datasets are listed in Table 1. Specifically, CA is a Foursquare[1] dataset where users are in California. It was collected from January 2010 to February 2011, and used in [4]. Gowalla[2] and Brightkite[3] are two widely used LSBN datasets, which contain massive implicit feedbacks through user-POI check-ins. To remove rare cases, we eliminated users' interactions with fewer than 15 items and items interacted by fewer than 10 users in the three datasets.

Table 1. Statistics of datasets

Dataset	Users	Items	Feedbacks	Avg.seq.len	Density
CA	2031	3112	106,229	52.30	1.68%
Gowalla	5073	7021	252,944	49.87	0.71%
Brightkite	1850	1672	257,369	139.12	8.12%

Baseline. To evaluate the performance of 3ARGU, we compare it with following comparative methods.

[1] https://sites.google.com/site/yangdingqi.
[2] http://snap.stanford.edu/data/loc-gowalla.html.
[3] http://snap.stanford.edu/data/loc-brightkite.html.

- **BPR-MF**: Bayesian Personalized Ranking based Matrix Factorization (BPR-MF) is a popular method for top-N recommendation. It is based on users' pair-wise preferences and neglects the usage of item content.
- **FPMC**[4] [19]: Factorizing Personalized Markov Chain (FPMC) was designed to predict the items in the next basket. It can not only learn the general taste of a user by factorizing the matrix over observed user-item preferences, but also model sequential behaviors by learning a transition graph over items, which is used to predict the next action based on the recent actions of a user.
- **GRU**: Gated Recurrent Unit based RNN (GRU) is the most advanced method for sequential recommendation, which can capture the long-term dependency and compact vanishing gradients of RNN to recommend following items.
- **GRU-ATT** [2]: Similar to our model 3AGRU, we apply the fixed attention mechanism on the input level and hidden state level of GRU, since the fixed attention mechanism [23] has been used in text classification tasks and has a good preference. Here, we name this method GRU-ATT.
- **RUM**[5] [1]: RUM utilizes external memories to improve sequential recommendation, which contains two variants, item-level (RUM^I) and feature-level (RUM^F).

Evaluation Metrics. For next-future recommendation whose aim is to recommend next item collection that user would probably interact with in the future, we hold out the first 70% of each user's interaction sequence in the time order as the training set and the remaining 30% for testing. Besides, we remove the duplicate items in each user's test sequence. Following previous work [7,10], we randomly sample 50 items that are not interacted by the user, while other 50 items are samples according to the popularity. We expect that the recommendation system can not only retrieve relevant items out of all available items, but also retrieve the results as accurately as possible. Besides, we hope it can provide a ranking where items of user's interests are ranked in the top. Therefore, we use the Precision, Recall and Mean Average Precision (MAP) to evaluate the preference of our model.

- **Precision, Recall.** The precision is widely used to measure the predictive accuracy in sequential recommendation system area. $\mathcal{P}@\mathcal{K}$ represents the proportion of test cases that recommend items correctly in a top $\mathcal{K}$ position in generated recommendation list for a user. The recall is used to measure how well a model can recommend relevant items out of all available items. $\mathcal{R}@\mathcal{K}$ represents the proportion of test cases that recommend items correctly in a top $\mathcal{K}$ position in the set of user's interacted items in the test data. The definitions of $\mathcal{P}@\mathcal{K}$ and $\mathcal{R}@\mathcal{K}$ are given as:

$$\mathcal{P}@\mathcal{K} = \frac{\#Hits}{\text{length of generated recommendation list}} \tag{16}$$

[4] https://github.com/khesui/FPMC.

[5] https://github.com/ch-xu/RUM.

$$\mathcal{R}@\mathcal{K} = \frac{\#Hits}{\text{total number of items the user likes}} \tag{17}$$

- **MAP.** Mean Average Precision is the average of AP to measure the ranking performances.

$$MAP = \frac{1}{|\mathcal{U}|}\sum_{u \in U} AveP(u), \tag{18}$$

$$AveP(u) = \frac{1}{|K'|}\sum_{k=1}^{K^{'}} p_u(k) rel_u(k) \tag{19}$$

where $p_u(k)$ represents the precision of the top k products recommendation to user u; $rel_u(k)$ denotes whether the k_{th} item has interacted with user u in the test data; $K^{'}$ is the cut-off point.

5.2 Comparison of Performance

Our experimental results of 3ARGU and the several comparative methods on the three real-world datasets are shown in Table 2. From Table 2, we can observe following phenomena.

First, in terms of the three evaluation metrics (i.e., precision, recall and MAP), 3AGRU consistently outperforms other methods with a large margin on all three real-world datasets. This indicates that 3AGRU is capable to model complicated process of sequential decision through adaptive attention network.

Second, the performance of BPR-MF, which contains no sequential information, is the worst among that of all competitive methods under most cases. This shows that sequential information is important for improving recommendation performance, which confirms that user's interests are dynamic. FPMC, which considers sequential information, has a certain improvement, comparing with BPR-MF. However, since the sequential information considered is based on users' recent actions in FPMC, the effectiveness is not as good as other methods utilizing long-term and short-term information such as GRU.

Third, RUM models use historical records with external memories, and aim to solve the representation of fixed hidden layers and make recommendation more explanatory. From Table 2, we can see that in terms of MAP, RUM on both item-level and feature-level does not perform as well as GRU on the Gowalla dataset, but it performs well on the other two datasets. This is possibly because the data distribution of the Gowalla dataset is relatively sparse. This suggests that the use of attention mechanisms in the hidden state level can describe the user's dynamic preferences.

Fourth, Table 2 shows that GRU-ATT performs much better than GRU. This is because GRU-ATT uses the fixed attention network. It is known that the attention mechanism can improve the performance of recommendation to a certain extent. This also proves that it is effectiveness of the attention mechanism to capture users' evolving appetites for items.

Table 2. Experimental results of different methods on three real-world datasets. Note that the larger the number in the table, the better the performance is. And the boldface results are the best while the second best are underlined.

Dataset	Method	@5		@10		@15		@20		MAP
		Precision	Recall	Precsion	Recall	Precsion	Recall	Precsion	Recall	
CA	BPR-MF	0.0282	0.0108	0.0207	0.0164	0.0170	0.0200	0.0153	0.0233	0.0136
	FPMC	0.0430	0.0191	0.0267	0.0235	0.0203	0.0266	0.0170	0.0296	0.0230
	RUM^I	0.0633	0.0253	0.0470	0.0372	0.0387	0.0454	0.0348	0.0538	0.0371
	RUM^F	0.0500	0.0206	0.0419	0.0247	0.0370	0.0279	0.0337	0.0308	0.0360
	GRU	0.0746	0.0266	0.0525	0.0363	0.0415	0.0429	0.0349	0.0476	0.0260
	GRU-ATT	0.0845	0.0293	0.0576	0.0398	0.0460	0.0481	0.0382	0.0533	0.0287
	3AGRU	**0.1074**	**0.0502**	**0.0782**	**0.0731**	**0.0591**	**0.0828**	**0.0527**	**0.0986**	**0.0477**
Gowalla	BPR-MF	0.0105	0.0234	0.0150	0.0172	0.0183	0.0144	0.0208	0.0127	0.0111
	FPMC	0.0510	0.0261	0.0329	0.0317	0.0247	0.0364	0.0202	0.0394	0.0243
	RUM^I	0.0406	0.0218	0.0299	0.0307	0.0245	0.0367	0.0216	0.0425	0.0158
	RUM^F	0.0236	0.0117	0.0188	0.0184	0.0159	0.0228	0.0146	0.0269	0.0106
	GRU	0.0634	0.0271	0.0412	0.0347	0.0321	0.0397	0.0270	0.0438	0.0252
	GRU-ATT	0.0718	0.0314	0.0445	0.0377	0.0334	0.0418	0.0273	0.0450	0.0267
	3AGRU	**0.0756**	**0.0418**	**0.0488**	**0.0512**	**0.0382**	**0.0583**	**0.0338**	**0.0669**	**0.0344**
Brightkite	BPR-MF	0.0222	0.0044	0.0198	0.0079	0.0173	0.0105	0.0160	0.0128	0.0068
	FPMC	0.0101	0.0066	0.0088	0.0109	0.0080	0.0151	0.0079	0.0191	0.0122
	RUM^I	0.0728	0.0199	0.0556	0.0320	0.0476	0.0404	0.0423	0.0483	0.0156
	RUM^F	0.0356	0.0092	0.0309	0.0166	0.0263	0.0202	0.0216	0.0211	0.0240
	GRU	0.0685	0.0120	0.0478	0.0181	0.0383	0.0221	0.0329	0.0256	0.0136
	GRU-ATT	0.0786	0.0139	0.0531	0.0200	0.0418	0.0246	0.0354	0.0285	0.0150
	3AGRU	**0.1129**	**0.0767**	**0.0769**	**0.1031**	**0.0568**	**0.1137**	**0.0467**	**0.1229**	**0.0680**

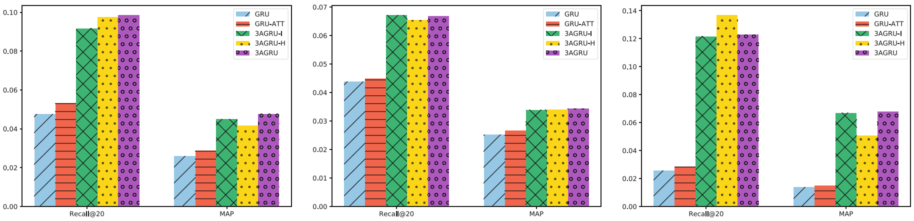
At last, we can make comparisons between our model 3AGRU and GRU-ATT, since both have adopted the attention mechanism. Table 2 shows the advantage of 3AGRU, comparing with GRU-ATT in terms of all three evaluation metrics. This indicates that the adaptive attention network is better than the fixed attention mechanism. We will further explore the advantage of the adaptive attention network over the fixed attention mechanism in the next subsection.

In Summary, our experimental results prove that our proposed model 3AGRU does reflect the user's interest effectively and performs better than other state-of-the-art methods.

5.3 Influence of Components

In order to judge the advantage of the adaptive attention network, comparing to the fixed attention mechanism, we conduct experiments on cases where the adaptive attention network is applied only to the input level or only to the hidden level of GRU. We use 3AGRU-I to present that the adaptive attention network is only applied in the input level, while 3AGRU-H is used to indicate the adaptive attention network is only applied to the hidden state level.

We will observe the results of experiments on 3AGRU-I, 3AGRU-H, 3AGRU and two competitive methods (i.e., GRU and GRU-ATT). Our experimental results are shown in Fig. 2. Due to space limitation, we only show their performance in terms of Recall@20 and MAP.



(a) CA (b) Gowalla (c) Brightkite

Fig. 2. Exploration of the role of fixed attention mechanism and adaptive attention mechanism in terms of Recall@20 and MAP.

Figure 2 shows that the fixed attention mechanism indeed improves the performance of sequential recommendation to a certain extent. However, we can see that the approaches (i.e., 3AGRU-I, 3AGRU-H, and 3AGRU) based the adaptive attention network perform better than GRU and GRU-ATT. This shows that the adaptive attention network is much better than the fixed attention mechanism. This can prove that the adaptive attention network can better depict users' dynamic preferences.

From Fig. 2, we can also find that the relationship between 3AGRU and its variants (i.e., 3AGRU-I and 3AGRU-H) is different in different datasets. In terms of performance on CA and Gowalla, 3AGRU performs more or less better than both 3AGRU-I and 3AGRU-H, this indicates that the adaptive attention mechanism is better applied to both the input level and the hidden state level, instead of only one level. This is because the diversity and dynamics of user preferences are further depicted in the hidden state level according to the items to be recommended. In addition, the effect of 3AGRU is only slightly improved compared with that of 3AGRU-I and 3AGRU-H, possibly because users' preferences for a given item via an adaptive attention network are so accurate that adding another adaptive attention network doesn't make much difference. When comparing 3AGRU-I and 3AGRU-H, we note that sometimes 3AGRU-I works better and sometimes 3AGRU-H works better, supporting that both the input level and the hidden state level play the important role in GRU. We further investigate the performance on Brightkite. We can observe that 3AGRU-H performs best in terms of Recall@20, this may be operations of GRU for forgetting and updating information play a larger role in dense datasets, making input based on adaptive attention network to some extent forgotten. And the original input to do the same operation, the disadvantages are relatively small.

The above analysis shows that the core part of our model 3AGRU, adaptive attention network, can embody dynamic and diversity characteristics of user sequential preferences. It is indeed more effective than fixed attention mechanism.

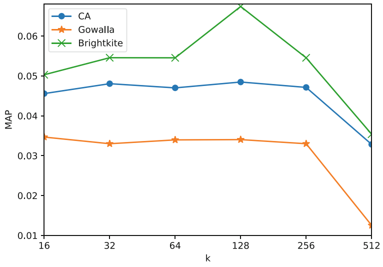


Fig. 3. Influence of dimension size

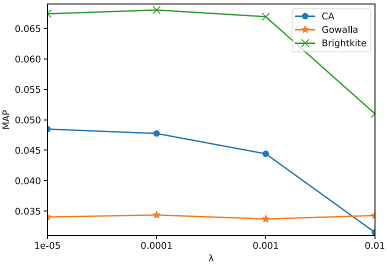


Fig. 4. Influence of regularization terms

5.4 Influence of Hyper-parameter

In this subsection, we explore the influences of the dimension size and different regularization terms in our model 3AGRU. Due to space limitation, we just show the results under the metric of MAP.

Influence of Dimension Size k. Dimension size in our model is relevant not only item embedding sizes but also hidden unit size in GRU network which represents the number of features in the hidden state. The size of hidden unit represents the number of nodes that used to remember and store previous states and shows the capability of GRU while dimension size reflects the ability of latent vector representation. Therefore, we choose k in $\{16, 32, 64, 128, 256, 512\}$ to find the size which makes the performance of our model 3AGRU better. It can be seen in Fig. 3 that, the performance of the model is improved at the beginning and reaches the best at $k = 128$ with the increase of k, and then the performance starts to deteriorate. As Brightkite dataset is relatively dense, this trend is particularly evident in it. In terms of dimension size, it indicates that low-dimension vector has a limitation of modeling complex interactions while the high-dimension vector may affect the generalization of the model and increase the number of parameters. In terms of hidden unit size, proper hidden unit size can help achieve best performance.

Influence of Different Regularization Terms. We further investigate the influence of different regularization terms λ. In our model, we utilize $\mathcal{L}_2$ regularization terms (λ) mainly on the representation vector of items to avoid overfitting problem. We search the λ from $\{0.00001, 0.0001, 0.001, 0.01\}$ to optimize performance of our model. Figure 4 shows the influence of regularization at MAP. As shown in the figure, small λ can improve our model in terms of MAP and the 3AGRU reaches its best preference when λ is set to 0.0001. When the value of λ continues to decrease, the performance hardly changes.

6 Conclusion

In this paper, we proposed a novel model named Adaptive Attention-Aware Gated Recurrent Unit (3AGRU) to learn adaptive user sequential representations and capture users' short-term and long-term interests to embody the diversity and the dynamics of user interests. With the items to be recommended and users' history records, we constructed a novel attention mechanism called adaptive attention mechanism to reflect the diverse interests of users. First, we embedded the sequence of inputs and the targets into low-rank dimension spaces, and then generated the adaptive attention network in the input level and the hidden state level to adapt the representation of user sequential preferences, and to learn the interactions between steps and items from data. Experimental results demonstrated that 3AGRU achieved a good performance in terms of precision, recall, and MAP, comparing with several competitive methods on the three real-world datasets.

Acknowledgement. This research was partially supported by the NSFC (61876117, 61876217, 61872258, 61728205), the Suzhou Science and Technology Development Program (SYG2 01803) and the Open Program of Neusoft Corporation (SKLSAOP1801).

References

1. Chen, X., et al.: Sequential recommendation with user memory networks. In: WSDM, pp. 108–116. ACM (2018)
2. Cui, Q., Wu, S., Huang, Y., Wang, L.: A hierarchical contextual attention-based GRU network for sequential recommendation. arXiv preprint arXiv:1711.05114 (2017)
3. Devooght, R., Bersini, H.: Long and short-term recommendations with recurrent neural networks. In: Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, pp. 13–21. ACM (2017)
4. Gao, H., Tang, J., Liu, H.: gSCorr: modeling geo-social correlations for new check-ins on location-based social networks. In: CIKM, pp. 1582–1586. ACM (2012)
5. Grbovic, M., et al.: E-commerce in your inbox: product recommendations at scale. In: SIGKDD, pp. 1809–1818. ACM (2015)
6. He, R., Kang, W.C., McAuley, J.: Translation-based recommendation. In: RecSys, pp. 161–169. ACM (2017)
7. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: WWW, pp. 173–182. IW3C2 (2017)
8. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. In: ICLR (2016)
9. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: 2008 Eighth IEEE International Conference on Data Mining, ICDM 2008, pp. 263–272. IEEE (2008)
10. Huang, J., Zhao, W.X., Dou, H., Wen, J.R., Chang, E.Y.: Improving sequential recommendation with knowledge-enhanced memory networks. In: SIGIR, pp. 505–514. ACM (2018)
11. Lee, J.S., Jun, C.H., Lee, J., Kim, S.: Classification-based collaborative filtering using market basket data. Expert Syst. Appl. **29**(3), 700–704 (2005)

12. Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J.: Neural attentive session-based recommendation. In: CIKM, pp. 1419–1428. ACM (2017)
13. Li, Z., Zhao, H., Liu, Q., Huang, Z., Mei, T., Chen, E.: Learning from history and present: next-item recommendation via discriminatively exploiting user behaviors. In: KDD, pp. 1734–1743. ACM (2018)
14. Lin, X., Niu, S., Wang, Y., Li, Y.: K-plet recurrent neural networks for sequential recommendation. In: SIGIR, pp. 1057–1060. ACM (2018)
15. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. IEEE Internet comput. **1**, 76–80 (2003)
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
17. Pan, R., Scholz, M.: Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering. In: SIGKDD, pp. 667–676. ACM (2009)
18. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-fifth Conference on Uncertainty in Artificial Intelligence, pp. 452–461. AUAI Press (2009)
19. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized Markov chains for next-basket recommendation. In: WWW, pp. 811–820. ACM (2010)
20. Shani, G., Heckerman, D., Brafman, R.I.: An MDP-based recommender system. J. Mach. Learn. Res. **6**(Sep), 1265–1295 (2005)
21. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv. Artif. Intell. **2009**, 19 (2009)
22. Wang, P., Guo, J., Lan, Y., Xu, J., Wan, S., Cheng, X.: Learning hierarchical representation model for nextbasket recommendation. In: SIGIR, pp. 403–412. ACM (2015)
23. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. NAACL-HLT **2016**, 1480–1489 (2016)
24. Ying, H., et al.: Sequential recommender system based on hierarchical attention networks. In: IJCAI (2018)
25. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T.: A dynamic recurrent model for next basket recommendation. In: SIGIR, pp. 729–732. ACM (2016)
26. Zhang, Y., Wang, H., Lian, D., Tsang, I.W., Yin, H., Yang, G.: Discrete ranking-based matrix factorization with self-paced learning. In: SIGKDD, pp. 2758–2767. ACM (2018)
27. Zhang, Y., et al.: Sequential click prediction for sponsored search with recurrent neural networks. In: AAAI, vol. 14, pp. 1369–1375 (2014)