Information Sciences 510 (2020) 122–134

Contents lists available at ScienceDirect

Information Sciences

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

Attention-based context-aware sequential recommendation model

Weihua Yuan[a,b], Hong Wang[a,*], Xiaomei Yu[a], Nan Liu[b], Zhenghao Li[a]

[a] School of Information Science and Technology, Shandong Normal University, Jinan, SD, 250014, China
[b] School of Computer Science and Technology, Shandong Jianzhu University, Jinan, SD, 250101, China

ARTICLE INFO

Article history:
Received 21 May 2018
Revised 5 September 2019
Accepted 9 September 2019
Available online 9 September 2019

Keywords:
Deep learning
Attention mechanism
Collaborative filtering
Sequential recommendation
Context-aware recommendation

ABSTRACT

Recurrent neural networks (RNN) based recommendation algorithms have been introduced recently as sequence information plays an increasingly important role when modeling user preferences. However, these methods have numerous limitations: they usually give undue importance to sequential changes and place insufficient emphasis on the correlation between adjacent items; additionally, they typically ignore the impacts of context information. To address these issues, we propose an attention-based context-aware sequential recommendation model using Gated Recurrent Unit (GRU), abbreviated as ACA-GRU. First, we consider the impact of context information on recommendations and classify them into four categories, including input context, correlation context, static interest context, and transition context. Then, by redefining the update and reset gate of the GRU unit, we calculate the global sequential state transition of the RNN determined by these contexts, to model the dynamics of user interest. Finally, by leveraging the attention mechanism in the correlation context, the model is able to distinguish the importance of each item in the rating sequence. The impact of outliers that are less informative or less predictive decreases or is ignored. Experimental results indicate that ACA-GRU outperforms state-of-the-art context-aware models as well as sequence recommendation algorithms, demonstrating the effectiveness of the proposed model.

1. Introduction

Collaborative filtering is the most popular technique in recommender systems; it can be divided into rating prediction and ranking prediction based on output form [37]. Rating prediction makes recommendations by filling in the missing entries in the rating matrix based on explicit feedback. In addition, the objective of ranking prediction, namely, Top-N recommendation, is to generate a sorted recommendation list of length N based on implicit feedback. As a recent research hotspot in recommender systems, implicit feedback is more attainable than explicit feedback, which mainly reflects user preferences indirectly by analyzing various elements such as the videos being played, products being purchased, and items being clicked. In this paper, we focus on implicit feedback and study context-aware sequential recommendation based on the attention mechanism.

In collaborative filtering recommendations, algorithms based on matrix factorization (abbreviated as MF) are the target of many studies by virtue of their high precision. These models generally assume that both user interest and item attributes

[*] Corresponding author.
E-mail address: wanghong106@163.com (H. Wang).

https://doi.org/10.1016/j.ins.2019.09.007
0020-0255/© 2019 Elsevier Inc. All rights reserved.

are static, ignoring the time variability of user and item factors. However, in the real world, these may vary with time, and different historical behaviors may have a different impact on users. Therefore, researchers are increasingly concerned with the importance of sequential information in recommender systems. Moreover, to further improve performance, we need to take the context information, including time, location, and nearby people, into consideration, in addition to the interactions between users and items. The efficiency of targeting user preferences with contextual information has been proven in [1].

As the exploration of deep learning [2] in computer vision and natural language processing, studies on deep learning-based recommender systems have gathered an increasing amount of attention [37]. A recurrent neural network (RNN) [37] is an artificial neural network characterized by its orientated and recurrently connected nodes. It can effectively model the dynamics of both user preferences and item attributes and make predictions based on present trends. Current sequential recommendations [40] tend to attach undue importance to sequence changes and the way they affect recommendation results. The recommending list focuses on the shifts in user interest evidenced by his recent behavior and it ignores the continuity of his interest. For example, a user who typically is a fan of romantic movies may try a thriller by accident ; this might be the result of the increasing popularity of the film, or he may be watching the movie in the company of friends. Hence, the movie loses its discriminative role in modeling user interest, which could lead to suboptimal performance for the recommendation.

In addition, if we consider recommendation as a sequential prediction problem, each item in the sequence has a different contribution in the final results. We should not only focus on the impact of sequential variances but also on the correlation of adjacent items and its impact on the transition in user interest. As a recent advance in neural representation learning, the attention mechanism [33] allows different parts to contribute differently to the final predictions, and therefore, we apply it to address the issues of sequential recommendation mentioned above. To summarize, we propose an attention-based context-aware sequential recommendation using GRU (a gating mechanism in RNNs [6]), abbreviated as ACA-GRU, and the main contributions are as follows:

(1) In ACA-GRU, we consider the impact of various types of context information on recommendations, and divide the context into four types, including input context, transition context, static interest context, and correlation context.
(2) By redefining the update gate and reset gate of the GRU unit, we calculate the global sequential state transition of the RNN determined by the defined context of input, transition, static user interest and correlation, to model the dynamics of user interest.
(3) We leverage the attention mechanism in the calculation of the correlation context, to distinguish the importance of various items in the rating sequence. Hence, the impact of outliers that are less informative or less predictive is reduced or ignored. In addition, we use BPR (Bayesian personalized ranking) [29] and BPTT (backpropagation through time) [32] to conduct the study of ACA-GRU.

To verify the effectiveness of the proposed method and model, we conduct extensive experiments on three widely used real datasets, and it empirically demonstrates the performance improvement brought by the attention mechanism in context-aware sequential recommendations, indicating that our model is more expressive and superior to state-of-the-art methods based on matrix factorization as well as sequence recommendation algorithms.

The remainder of the paper is organized as follows: Section 2 discusses related work; we elaborate on our proposed method and model in Section 3, including the problem definition, the detail of the ACA-GRU model, optimization and the summarization of the algorithm. We present the results of our experiments and analysis in Section 4. Finally, in Section 5, we conclude the paper and discuss future work.

2. Related work

With the advances in information acquisition capability, a substantial amount of contextual information including locations as well as time and weather can be readily accessed now. Adomavicis et al. [1] pointed out that beyond the user-item interaction matrix, user preferences are affected by different types of contextual information, and it is advantageous for recommender systems to improve their performance when incorporating contextual information. Thus, they proposed the concept of context-aware recommender systems, dubbed CARs, extending the traditional two-dimensional utility model, namely $user \times item \rightarrow R$, to a multidimensional utility model consisting of various contextual information, represented as $user \times item \times context \rightarrow R$. Other related work on context-aware recommendation includes time SVD++ [16] as well as contextual operating tensor (COT) [21] etc.

In collaborative recommendations, algorithms based on matrix factorization usually decompose the user-item rating matrix into the product of low-rank user factors and item factors, to represent user interest and item attributes in a common latent space, respectively. Popular methods based on matrix factorization include probability matrix factorization [34], non-negative matrix factorization [19], factorization machine [28], SVD++ [17], and FISM [15]. In the absence of any time variability of user or item factors, the description of user interest or item attributes in the form of implicit factors can be viewed as a static user interest expression and static item feature expression. For example, SVD++ [17] describes the user preference by implicit feedback, whereas FISM [15] represents the static user interest by aggregating low-dimensional dense embeddings of user-item interactions. Other works based on matrix factorization [9,23] are also applied to model static user preferences.

Collaborative filtering techniques tend to neglect sequential patterns in user behaviors, which nevertheless play an increasingly important role in recommender systems. Traditional sequential recommendation methods are primarily based on the Markov model, including FPMC [30], HRM for next-basket recommendation [38], and playlist prediction [4] based on latent Markov embedding. These Markov based methods fail to capture the features of long-term behaviors in the recommender systems, and suffer from high computational complexity and low prediction accuracy.

With advances in deep learning in the area of computer vision and natural language processing, deep learning-based recommender systems have gathered the attention of many researchers. For instance, FNN [44] and PNN [27] use deep learning models to extract higher-order and non-linear features between user-item interactions. Wide and deep model [5], proposed by Google, is capable of extracting lower-order and higher-order combinatorial features. Although empirical verification shows that additional performance improvement can be yielded, its wide part requires feature engineering. To handle it, Guo et al. introduced DeepFM [10], combining FM [28] under a deep learning framework, which conducts end-to-end learning by focusing on the combination of both high-rank and low-rank features. The idea in the work above is to concatenate various low-dimensional feature embeddings in the underlying layers, and then stack hidden layers above to learn higher-order feature interactions. However, there are limitations as they rarely account for feature combinations in the lower layers, and they do not consider the impact of time variation on user-item interactions.

RNN models the dynamics of user preference and item features effectively and has been successfully applied to various sequential modeling tasks, such as sentence translation [24] and click prediction [45]. Long Short Term Memory network (LSTM) [39] is a popular RNN variant, while GRU [6] is a simplified version of LSTM that preserves the properties of the LSTM while simplying its structure. Hidasi et al. [12] first introduced the RNN to sequential recommendations based on short-term session data. In their work, the actual session status is used as the input and the next transaction as the output. As RNNs and other sequential recommendation models have limitations in terms of modeling context information, Liu et al. [20] proposed CA-RNN, a context-aware sequential recommendation model based on vanilla RNN, adopting an adaptive input and transition matrix to represent various specific contexts. However, the focus on the influence of sequences is excessive in this model and the correlation between adjacenct items is ignored.

The attention mechanism [33] in neural networks allows various parts to contribute in different ways to the final predictions when compressed into a single expression. It is inspired by the attention mechanism in human vision [31]. When processing signals, the human cognition focuses selectively on specific content within the whole cognitive space and continuously adjusts the focal point to different times or places. The attention mechanism has been proven to be effective in various types of machine learning tasks, ranging from image captioning [42] and abstractive sentence summarization [33] to machine translation [7].

He et al. first introduced the attention mechanism to deep learning-based recommendations and proposed the ACF [3] and AFM [41] models. The key idea for AFM is to have an attention network to determine the relative importance of each interaction in the user rating list. In contrast to current mainstream approaches, AFM is more powerful and expressive as it merely adds a small number of attention parameters to the model; in addition, it can deliver an incremental performance improvement. However, both ACF and AFM are built on a multi-layer perceptron framework, which lack a study and application of the attention mechanism in context-aware sequential recommendations. Recently, Zhao et al. [46] introduced a movie recommender system LSRC fusing MF with RNN. They adopted Generative Adversarial Networks (GAN) to capture both long-term user preferences and short-term session information. In contrast with our work, LSRC does not consider the impact of contextual information on recommendations. In addition, the attention weights, which are treated as global variables, neglect the impact of target items on historical ratings and limit the expressiveness and extensibility of the model.

3. Model and methods of ACA-GRU

This section introduces our attention-based context-aware sequential recommendation model using GRUs, dubbed ACA-GRU. First, we present the problem statement as well as a summary of the symbols and notations used in the paper; we then offer the description of an attentional GRU (aGRU) defined by our ACA-GRU model, as well as the model optimization, parameter learning and, the summation of the algorithm.

3.1. Notations and problem definition

We provide here a summary of the symbols and notations used in the paper.

u, v: users, i, j: items.

p_i: The embedding of item i.

q_j: The embedding of item j, to describe the static interest context C_S^u.

r_{ui}: The rating of user u on item i. Here, we only focus on implicit feedback. 1 denotes that there is an interaction between user u and item i, and 0 represents no interactions.

R^u: The behavioral history of user u, expressed as $\{i_1^u, i_2^u, i_3^u, \ldots\}$.

T^u: The timestamp corresponding to the behavioral history of u, expressed as $\{t_1^u, t_2^u, t_3^u, \ldots\}$.

C^u: Different contextual information related to user u. In ACA-GRU, we classify C^u into the following four categories:

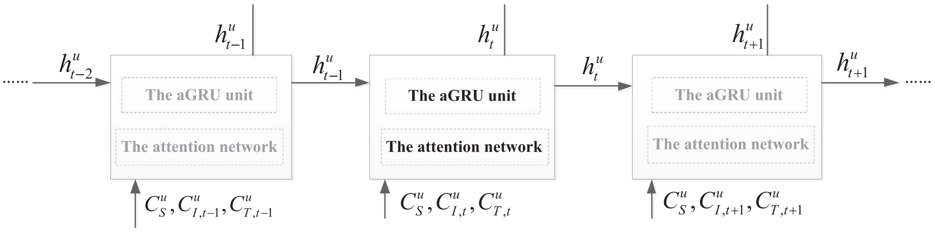
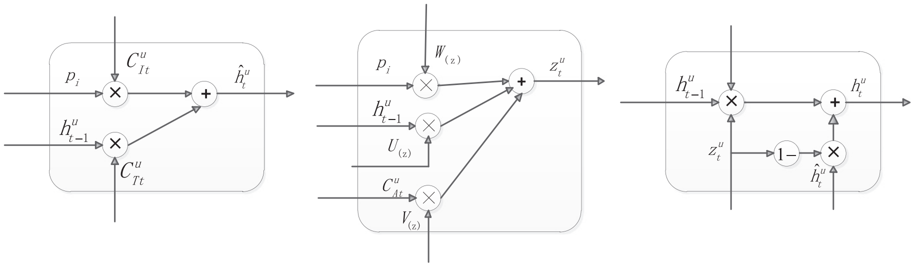


Fig. 1. the framework of ACA-GRU.



(a) candidate knowledge (b) update gate (c) hidden state of time t

Fig. 2. the framework of aGRU units in ACA-GRU.

- C^u_I: Input context, denoting contextual information related to the input of RNN, that is to say, it is the time, weather or places that a user conduct his behaviors. C^u_{It} represents the input context at time t. For example, we can extract two kinds of input context from T^u, day of the week and hour of the day.
- C^u_T: Transition context, denoting contextual information relevant to the transition of RNN, for instance, the time intervals between a user's two adjacent actions. C^u_{Tt} denotes the transition context of time t. For the transition context, we usually discrete the time bin as the time of the day from T^u.
- C^u_S: Static interest context, representing the set of static user interest expressions.
- C^u_A: Correlation context, representing the correlation between candidate knowledge $\hat{h}_t$ and static interest context C^u_S based on the attention mechanism. Here, C^u_{At} indicates the correlation context at time t.

Accordingly, the problem we need to address is described as follows:

Given the behavioral history R^u, timestamp T^u and the context C^u of user u, we need to predict the Top-N recommendation list at time $t+1$ under the new context C^u_{t+1}, which can be formalized as (1):

$$P_{C^u_{t+1}}(Y = 1|R^u, C^u, T^u) \tag{1}$$

The neural framework of ACA-GRU is illustrated in Fig. 1, which follows the flow chart of the traditional RNN. Its basic unit is an aGRU cell, which we redefine based on the GRU unit and the attention network, using all the contexts defined above, as we found that the traditional GRU unit does not perfectly fit the proposed model. In the sequential transition of ACA-GRU, the input of time t mainly includes the hidden state of the previous state, the static interest context, input context, and transition context. The aGRU cell captures the dynamics of user interest based on both sequential and contextual information, and we use it as the building block for the model. The details of the aGRU design are illustrated in Fig. 2. The attention network is defined to determine the relative importance of each item using the calculation of the correlation context. An overview of the attention network is provided in Fig. 3.

3.2. The ACA-GRU model

User interest has been changing over time. Aside from the user-item interaction matrix, user preferences are also affected by both sequence and context information, including time and location, among others. Traditionally, the relationship between the dynamic interest of a user and his behavioral history is described as follows: earlier behaviors have a lower impact on his preference modeling, while recent behaviors better reflect the drift in his interest. When modeling user preferences, his recent actions are endowed with higher weights, and hence, items that are highly similar to these interactions are prioritized for recommendation. For example, the weights of the early behaviors of a user are lowered by adding time attenuation factors [16,18,20].

The abnormal points or outliers hidden in user's behaviors. In addition, popular items do not necessarily represent real user interest since they are incapable of discerning personalized user interest. For instance, a hit movie with a high box-office receipt would fail to indicate personalized user interest if it was almost universally loved. Therefore, if the items

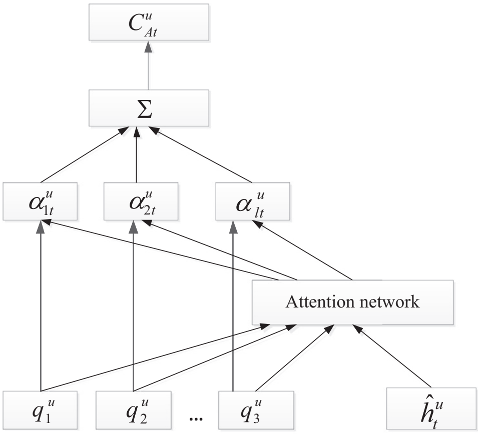


Fig. 3. Correlation context by static interest context and candidate knowledge based on attention network.

that were most recently interacted with belong to a popular catagory, such as a hit movie, it is not advisalbe to assign them with higher weights. In this paper, the items that are incapable of discriminating personalized user interest or have weak predictability are called abnormal points or outliers. In short, in sequential recommendations, we should decrease the impacts of these outliers when modeling user preferences.

3.2.1. Static interest context C^u_S

The static interest context C^u_S can be expressed as the set of latent factor vectors. As the user behavior history R^u can represent his preference or interest, we first obtain the hidden factor vectors of each item in R^u by MF. FISM [15] learns the user and item hidden submatrix through structural equation models, and the predicting score $\hat{y}_{ui}$ is parameterized as the inner product of the embedding vector u and i:

$$\hat{y}_{ui} = p_i^T \left(\frac{1}{|R_u^+|^\alpha} \sum_{j \in R_u^+ \setminus \{i\}} q_j \right) \tag{2}$$

Among which α is a hyperparameter that controls the effect of regularization, and p_i and q_j indicate the embedding expressions of item i and j, respectively. Here, the embedding [13] refers to the low-dimensional and dense expressions of these sparse vectors. To enhance model expressiveness, each item in FISM corresponds to two embeddings, namely p_i and q_j, to separate a predicting object from a historical interaction. $\setminus\{i\}$ is used to avoid the self-similarity modeling of items.

According to Koren [17], the list of interacted items for user u can be treated as his hidden interest expression. In other words, the formula within the brackets can be expressed as (3):

$$\frac{1}{|R_u^+|^\alpha} \sum_{j \in R_u^+ \setminus \{i\}} q_j \tag{3}$$

that is, user $u's$ interest is expressed as the aggregation of the low-dimensional embeddings in R^u. It is apparent from (3) that FISM considers the interactive items to have equal contributions when reflecting user interest. However, in ACA-GRU, we insist that the interactions of a user possess distinct predicting abilities under different contexts, and therefore, we view the collection of embedding vectors in R^u as a user static interest, also called the static interest context, formularized as (4):

$$C^u_S = \{q^u_1, q^u_2, q^u_3, \ldots\} \tag{4}$$

among which q^u_j is an embedding vector of length k. The static interest context C^u_S is essential in calculating the correlation context C^u_{At}.

3.2.2. Definition of the aGRU unit

Given the current input x_t and state h_{t-1} of RNN, the probability distribution of the next output is expressed as:

$$h_t = g(Wx_t + Uh_{t-1}) \tag{5}$$

In our view, a vanilla RNN (5) is insufficient to address the problem described here, since C^u, the context of ACA-GRU, is composed of C^u_I, C^u_T, C^u_A and C^u_S. GRU is a more complex RNN unit, with its new hidden state being a linear interpolation between the former hidden state and its current candidate knowledge:

$$h_t = (1 - z_t)h_{t-1} + z_t\hat{h}_t \tag{6}$$

and its candidate knowledge, update gate and reset gate are defined as (7)–(9), respectively:

$$\hat{h}_t = \tanh\left(Wx_t + U(r_t \odot h_{t-1})\right) \tag{7}$$

$$z_t = \sigma\left(W_z x_t + U_z h_{t-1}\right) \tag{8}$$

$$r_t = \sigma\left(W_r x_t + U_r h_{t-1}\right) \tag{9}$$

Vectors from the update gate and reset gate determine the final output of the gated loop unit. By analysis, the GRU unit expressed according to (6)–(9) does not perfectly fit our proposed model ACA-GRU, and we therefore redefine the GRU unit and refer to it as an aGRU unit. The framework of the aGRU unit is shown in Fig. 2.

Candidate knowledge $\hat{h}_t^u$ of aGRU. Candidate knowledge is a status determined by current input and the previous hidden state, denoted as $\hat{h}_t^u$. Its framework is depicted in Fig. 2(a), and its formal definition is given below:

$$\hat{h}_t^u = \sigma\left(x_t, h_{t-1}\right) = \sigma\left(p_i I_{C_{It}^u} + h_{t-1}^u T_{C_{Tt}^u}\right) \tag{10}$$

where p_i is a k dimensional embedding vector related to item i, C_{It}^u represents the input context at time t, and $I_{C_{It}^u}$ is an embedding matrix concerned with C_{It}^u. h_{t-1}^u denotes the previous hidden state, C_{Tt}^u the transition context of time t, and $T_{C_{Tt}^u}$ an embedding matrix related to C_{Tt}^u. σ is the nonlinear activation function of ReLu, sigmoid and tanh, among others. In CA-RNN, Liu et al. [20] treat the expression (10) as the output of time t and ignore the impact of correlation between adjacent items on sequence predictions. In other words, the relevance between all adjacent items in CA-RNN is treated equally, which is inadequate when trying to reduce or eliminate the negative impact of these outliers. However, in ACA-GRU, the expression obtained in (10) is regarded as candidate knowledge $\hat{h}_t^u$. It is the update gate that determines the proportion of $\hat{h}_t^u$ to be exported as the hidden state of time t.

Definition of update gate, reset gate and hidden state of aGRU. The structure of the update gate and the output of the hidden state are shown in Fig. 2(b) and (c), respectively, among which the update gate is defined as (11):

$$z_t^u = \sigma\left(p_i W_{(z)} + h_{t-1}^u U_{(z)} + C_{At}^u V_{(z)}\right) \tag{11}$$

The update gate of the aGRU is jointly determined by item embedding p_i, previous hidden state h_{t-1}^u and correlation context C_{At}^u, among which, $W_{(z)}$, $U_{(z)}$ and $V_{(z)}$ are the corresponding weight matrices, and C_{At}^u indicates the attention-based correlation context at time t, which will be described in the subsection below. In contrast to traditional GRU units, the update gate of an aGRU depends not only on the current input and previous implicit status but also on the correlation context vector.

In addition, we define the reset gate as $r_t = I$. Accordingly, the hidden state h_t^u as the output of time t for user u can be expressed as:

$$h_t^u = (1 - z_t^u) \odot \hat{h}_t^u + z_t^u \odot h_{t-1}^u \tag{12}$$

where $\odot$ denotes the element-wise product between vectors. If the candidate knowledge $\hat{h}_t^u$ has a lower interrelation with the user static interest context C_S^u, h_{t-1}^uwill be a major source of h_t^u; otherwise, there will be more $\hat{h}_t^u$ retained in h_t^u.

3.2.3. Correlation context C_{At}^u

Fig. 3 demonstrates the structure of correlation context C_{At}^u by static interest context C_S^u and candidate knowledge $\hat{h}_t^u$ based on the attention network.

In ACA-GRU, the correlation context C_{At}^u refers to the correlation between candidate knowledge $\hat{h}_t^u$ and static interest context C_S^u. In addition, the correlation context at time t, namely C_{At}^u, is learned automatically from the data with no human domain knowledge. As each item in R^u has a varied predictive power in various contexts, we should lower the impact of abnormal points on the global sequential transition, as they make little or no contribution to the final predictions. The current node is more likely to be an abnormal point when the candidate knowledge $\hat{h}_t^u$ has lower or little correlation with C_S^u. We therefore calculate the weight between C_S^u and $\hat{h}_t^u$ based on the attention mechanism, represented as $f(\hat{h}_t^u, q_j^u)$:

$$f\left(\hat{h}_t^u, q_j^u\right) = h^T \mathrm{Relu}\left(W\left(\hat{h}_t^u \odot q_j^u\right) + b\right) \tag{13}$$

In (13), $f(\hat{h}_t^u, q_j^u)$ is formalized as a feed-forward neural network and is jointly trained with other parts of the system, where W and b denote the weight matrix and bias vector, respectively, to project the input to hidden layers ; h^T indicates the map from the hidden layers to the attention weights. The correlation context C_{At}^u is computed according to (14):

$$C_{At}^u = \sum_{l=1}^{|R_u|} \alpha_{lt} \odot q_l^u, \alpha_{jt} = \frac{\exp\left(f\left(\hat{h}_t^u, q_j^u\right)\right)}{\sum_{l=1}^{|R_u|} \exp\left(f\left(\hat{h}_t^u, q_l^u\right)\right)} \tag{14}$$

We use softmax to transform the weight α_{jt} into a probability expression so that it has a decent probability interpretation. Thus, C_{At}^u can be quantified as the expected correlation on all items in the current user's static interest context C_S^u with probability α_{jt}.

3.2.4. Making predictions

The predicting process of ACA-GRU is expressed as:

$$\hat{y}^{u}_{t+1,j'} = \sigma\left(h^{u}_{t} T_{C^{u}_{Tt+1}} \left(p_{j'} I_{C^{u}_{It+1}}\right)^{T}\right) \tag{15}$$

where $\hat{y}^{u}_{t+1,j'}$ is the predicted rating of user u on item j' at time $t+1$ or the probability that u will interact with j'. $I_{C^{u}_{It+1}}$ and $T_{C^{u}_{Tt+1}}$ are the low-dimensional dense embedding matrices related to the current input context C^{u}_{It+1} and transition context $T_{C^{u}_{Tt+1}}$, respectively. σ denotes the sigmoid function, which is used to convert the prediction value $\hat{y}^{u}_{t+1,j'}$ into a probability to represent the possibility of interaction between u and j'.

When making predictions, the cold-start problem [36] remains one of the most challenging ones in many recommender systems. Traditionally, the cold-start issue can be better handled if the user and item attributes can be accessed and combined when making recommendations, e.g., the content-based methods [26], WDMMA [43] and NCF [11].

Note that the proposed ACA-GRU can conveniently handle the cold-start problem if we obtain the initial embeddings of the cold-start items or users.

First, let us consider cold-start items or a user's lack of available attributes. Taking cold-start items j' as an example, the embedding $p_{j'}$ can be initiated randomly, and h^{u}_{t}, $T_{C^{u}_{Tt+1}}$ and $I_{C^{u}_{It+1}}$ calculated according to (4) and (10)–(14), respectively. As a result, the initial predicted rating of user u on item j' can be obtained based on (15) and its embedding expression and the predicted rating will be updated along with the learning and optimization of the model.

The second consideration revolves around when user and item attributes can be accessed in the recommendation model. Under this circumstance, a clustering method may be implemented beforehand on the users and the items based on their attributes, e.g., some similarity based or neighborhood based clustering algorithms [22]. For example, when making predictions for a cold-start item j', a primary and necessary step is to determine which cluster it belongs to. Its initial embedding can be denoted as the aggregate of all the items' embeddings from the same cluster. Accordingly, the predicted rating of user u on item j' can be computed as in the first situation.

3.3. Model optimization

In implicit feedback, a user's rating on an item is either 0 or 1, and therefore, we can treat the recommendation model as a binary classification problem. Since implicit feedback only provides a noise signal on the user's preference, 1 implies that the user likes the item, and 0 indicates that the user does not like it or does not know it at all [14]. Therefore, there is a natural loss of negative feedback in the implicit datasets. To address the issue, we can either treat all unobserved items as negative feedback or conduct negative sampling from the unobserved datasets [25].

When optimizing ACA-GRU, we utilize BPR [29], a pairwise ranking loss function, in which the observed items are assumed to be ranked ahead of unobserved ones. The objective function is expressed as maximizing the margin between the predictions of observed items and those of unobserved ones:

$$L = \sum_{(u,k,i,j)\in O} \ln\left(1 + e^{-\left(\hat{y}_{u,k,i} - \hat{y}_{u,k,j}\right)}\right) + \lambda/2\|\Theta\|^{2} \tag{16}$$

In (16), O is the set of training instances and quadruple (u, k, i, j) denotes the kth interaction between user u and item i in his behavioral history, and j is the negative sample for (u, k, i) by negative sampling. Θ is the set of parameters, and λ is the regulation coefficient preventing over-fitting.

We sample negative elements that the user has not interacted with for each item in R^{u} using a ratio of 1:1. When sampling we consider the popularity of negative items so that the model tends to extract items with a higher prevalence ; as such negative items are more representative. In addition, we use BPTT [32] to conduct the study of ACA-GRU. We use SGD to iteratively update all parameters until convergence, where a user's behavioral history is randomly picked out from training instances, and all parameters are updated along the directions of the negative gradient.

3.4. Algorithm of ACA-GRU

Given the parameters required by ACA-GRU, we first obtain the static interest context C^{u}_{S} according to (4). Each item in R^{u} is computed through (10)–(12) combined with the various contexts, as well as the correlation context based on (13)–(14). In short, our proposed attention-based context-aware sequential recommendation model using GRU is summarized as Algorithm 1 (in the last page of the paper).

4. Experimental results and analysis

In this section, we verify the effectiveness of ACA-GRU empirically in context-aware sequential recommendations. We first introduce the experimental setting and datasets and then compare the proposed algorithm with the state-of-the-art.

Because of the nonlinearity of neural network models and the non-convexity of the objective functions, during optimization SGD is prone to falling into local optimality, and hence, the initialization of parameters is of great importance to the

Algorithm 1 attention-based context-aware sequential recommendation model using GRU (ACA-GRU).

```
Input: Step size of stochastic sub-gradient decent $\eta$, regulation parameter $\lambda$, max iterations $T$, input context $C_{It}^u$, transition context $C_{Tt}^u$, embedding matrix $I_{C_{It}^u}$ related to $C_{It}^u$, embedding matrix $T_{C_{Tt}^u}$ concerned with $C_{Tt}^u$, user behavioral history set $\{R\}$.
Output: Top-N recommendation list.
Initialization of the following parameters:embedding matrices $I_{C_{It}^u}$ and $T_{C_{Tt}^u}$, variables of (10), including item embedding $p_i$, previous hidden state $h_{t-1}^u$, correlation context $C_{At}^u$ as well as the corresponding weight matrices $W_{(z)}$, $U_{(z)}$ and $V_{(z)}$. The variables in formula (13) containing weight matrix $W$ and bias vector $b$, as well as $h^T$ in (12).
Obtain static interest context $C_S^u$ according to (4) by FISM;
for $i = 1$ to $T$ do
  for each $R^u$ in $\{R\}$ do
    if $r^u \in \{R^u\}$ then
      Generate candidate knowledge $\hat{h}_t^u$ according to (10)
      Set update gate $z_t$ according to (11)
      Obtain correlation context $C_{At}^u$ based on (13) and (14)
      Obtain hidden state $h_t^u$ of user $u$ at time $t$ according to (12)
      Calculate predictions and loss based on (15) and (16), respectively
    end if
    Optimize parameters through BPTT
  end for
end for
return Top-N recommendation list
```

ultimate performance of the model [8]. In our experiments, the values of the hyperparameters involved in the algorithm are shown below:

(1) The range of the vector embedding size m is set to [16,32,64,128], and the dimension of the attention factor α is also set to [16,32,64,128], with $m = \alpha$.
(2) The range of the regularization coefficient λ is set to $\left[10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\right]$.
(3) The range of the step size η of SGD is set to $\left[10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\right]$. The default values of the parameters are set to $m = \alpha = 32$ and $\lambda = \eta = 0.01$, respectively.

4.1. Datasets

In the experiments, our datasets mainly include MovieLens-100K[1], MovieLens-1M[2], and Netflix.[3] MovieLens-100K contains 100,000 ratings accessed by 943 users on 1682 movies as well as appropriate timestamps from the online movie recommendation service of MovieLens. For Movielens-1M, it contains 1,000,209 ratings contributed by 6040 users on 3900 movies, as well as the corresponding timestamps. Each user in either MovieLens-100K or MovieLens-1M is required to rate no less than 20 movies. In addition, we also explore the effectiveness of our algorithm on the Netflix dataset, which contains approximately 100,000,000 ratings contributed by 480,189 anonymous users on 17,770 movies, with approximately 99% unknown items. The datasets mentioned here are frequently employed in many state-of-the-art methods and related work. Hence, we also use them in our experiments to demonstrate the effectiveness of the proposed ACA-GRU, and to facilitate comparative analysis.

The ratings are integers ranging from 1 to 5 in all datasets, which we convert into implicit feedback with value 1 indicating an interaction between users and items. For each user's behavior sequence R^u, we pick 80% for training set and the remaining 20% for testing. As for the input context C_I^u, we extracted two types of context from T^u, that is, day of the week and hour of the day, and therefore, we obtained 168 types of input context for C_I^u. For the transition context, we discretize the time bin as the time of a day, and there are 32 types of transition context C_T^u. In addition, we obtain the static interest context C_S^u and initial values of the item embedding p_i by running FISM on Netflix and MovieLens, respectively. Therefore, the input of the proposed ACA-GRU principally involves R^u, C_I^u, C_T^u and C_S^u.

4.2. Evaluation indices

We evaluate the performance of the algorithms using several frequently used ranking based indices, such as *Recall@k*, *Precision@k*, $F1 - score@k$, and mean average precision (MAP), with the value in a [0,1] range. *Recall@k* and *Precision@k* are

[1] https://grouplens.org/datasets/movielens/100k/.
[2] https://grouplens.org/datasets/movielens/1m/.
[3] https://www.netflix.com/.

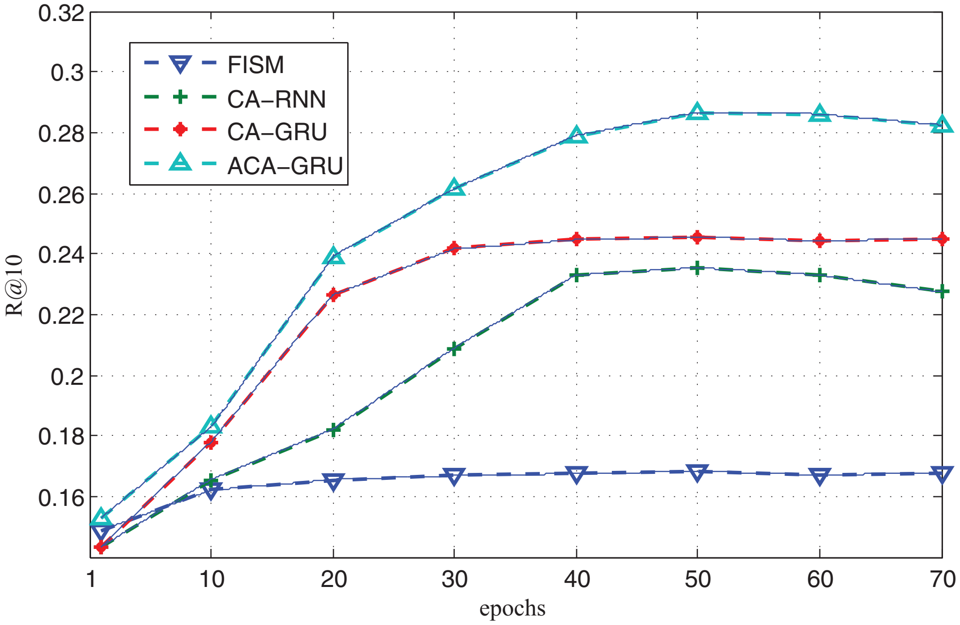


Fig. 4. Effect of attention mechanism: the trend of R@10 with the number of epochs.

used to test and verify the effectiveness of the recommendation methods, while $F1 - score@k$ is the harmonic average of *Recall@k* and *Precision@k*. MAP is an indicator related to the location of items in the recommendation list, while the others measure whether the target item appears in the list.

For all the metrics, the larger the value, the better the performance. *Recall@k*, *Precision@k*, $F1 - score@k$ are abbreviated as *R@k*, *P@k* and *F1@k*, respectively, and k is the length of the recommendation list where we set $k = 3, 5, 10$.

4.3. Comparison algorithms

In the experiments, we evaluate and compare our models with several state-of-the-art methods. In particular, amidst an abundance of available related work, we picked the most representatives for comparison, to indicate the effectiveness and novelty of the proposed ACA-GRU.

(1) POP and BPR [29]
POP is a non-personalized method used to test the performance of Top-N recommendations, in which a recommendation list is generated based on the popularity of the items. BPR is an algorithm designed for implicit feedback, adopting a pairwise ranking loss to optimize the latent factor models.

(2) ItemKNN [35] and FISM [15]
ItemKNN is an item-based collaborative filtering algorithm, and here, we calculate item similarity using Cosine similarity. FISM is an item-based Top-N recommendation model defined in (2). Both itemKNN and FISM are the representatives of item-based collaborative filtering, used to verify the effectiveness of our recommendation model.

(3) CARs [1] and CA-RNN [20]
CARs is a context-aware recommendation system proposed by Adomavicis et al. CA-RNN is a context-aware sequential recommendation algorithm based on vanilla RNN.

(4) Recurrent recommender networks (RRN) [40]
In this model, an RNN is used to supplement the results of matrix factorization through a hard-mixing mechanism.

(5) CA-GRU
CA-GRU is a context-aware sequential recommendation using GRUs, which is a simplified version of our proposed ACA-GRU with no attention mechanism.

4.4. Effect of the attention mechanism and the convergence analysis

To confirm the effectiveness of the attention mechanism in ACA-GRU, we run FISM, CA-RNN, CA-GRN, and ACA-GRU on MovieLens-100K, where we set the embedding size $m = 32$. The variations in $R@10$ with the number of epochs are plotted in Fig. 4.

It is apparent in Fig. 4 that the $R@10$ of all the algorithms shows consistent improvements with the number of iterations. During the first five iterations, the difference in the performance of ACA-GRU and the other three algorithms is not large;

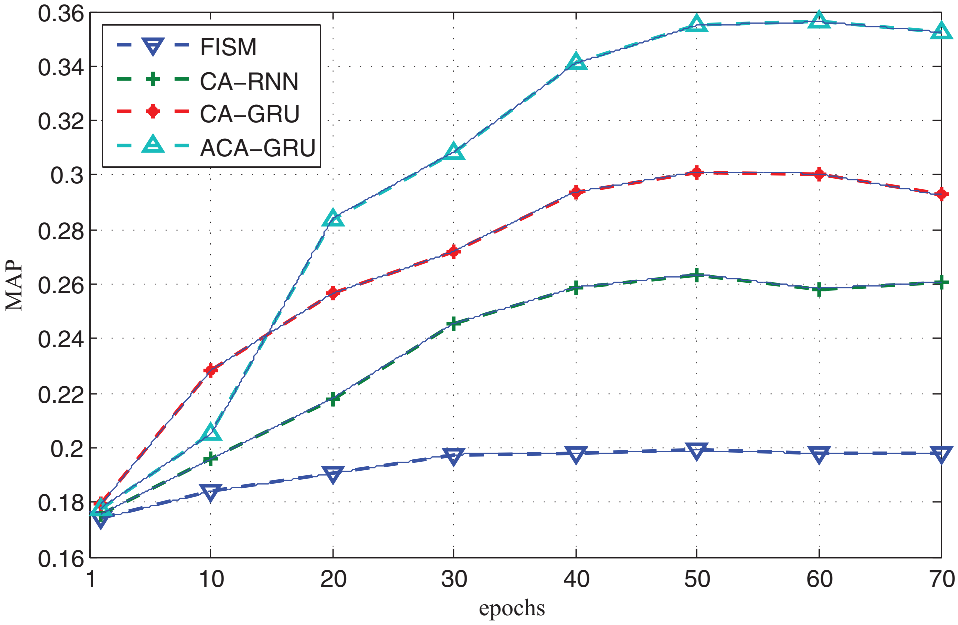


Fig. 5. Convergence analysis: variations of MAP with the number of epochs.

Table 1
Performance comparison of ACA-GRU with other algorithms on MovieLens-1M.

Method	R@3	R@5	R@10	P@3	P@5	P@10	F1@3	F1@5	F1@10	MAP
PoP	0.0070	0.0098	0.0134	0.0017	0.0026	0.0020	0.0027	0.0041	0.0035	0.0120
itemKNN	0.0305	0.0421	0.0504	0.0291	0.0288	0.0243	0.0298	0.0342	0.0328	0.0979
BPR	0.0333	0.0572	0.1142	0.0317	0.0327	0.0326	0.0325	0.0416	0.0507	0.1151
FISM	0.0377	0.0690	0.1324	0.0359	0.0394	0.0378	0.0368	0.0501	0.0588	0.1375
CARs	0.0462	0.0864	0.1535	0.0439	0.0494	0.0439	0.0450	0.0628	0.0682	0.1610
RRN	0.0552	0.0943	0.1683	0.0525	0.0539	0.0481	0.0538	0.0686	0.0748	0.1884
CA-RNN	0.0530	0.1057	0.1931	0.0504	0.0604	0.0551	0.0517	0.0769	0.0858	0.2056
CA-GRU	0.0704	0.1124	0.2025	0.0671	0.0642	0.0578	0.0687	0.0817	0.0900	0.2228
ACA-GRU	**0.0754**	**0.1221**	**0.2207**	**0.0718**	**0.0697**	**0.0630**	**0.0735**	**0.0888**	**0.0980**	**0.2432**

however, ACA-GRU significantly outperforms the others with the fastest enhancement in *R*@10. The gap between ACA-GRU and CA-GRU widens after approximately ten iterations, indicating that the attention-based correlation context is starting to work then. The purpose of the attention mechanism is to find information from C_S^u that is most relevant to current candidate information $\hat{h}_t^u$, to determine the impact on the hidden state h_t^u. As depicted in Fig. 4, CA-GRU performs slightly better than CA-RNN, which might be attributed to the effect of the aGRU threshold.

Fig. 5 illustrates the MAP variations of FISM, ACA-RNN, CA-GRU, and ACA-GRU with an increasing number of epochs. We can conclude from the figure that, after approximately 40 to 55 iterations, the MAP of ACA-RNN, CA-GRU, and ACA-GRU gradually becomes stable. With further increase in the number of iterations, the MAP for each algorithm decreases slightly, at which point the algorithms exhibit overfitting.

4.5. Performance analysis of ACA-GRU on Top-N recommendations

First, we compare the performance of Top-N recommendations between ACA-GRU and other state-of-the-art algorithms on MovieLens-1M, with parameter m set to 32. The results are summarized in Table 1. As shown in Table 1, our proposed ACA-GRU achieves the best performance on all evaluation indices.

From Table 1, we find that FISM outperforms POP, itemKNN, and BPR, which indicates the superiority of learning-based CF methods over heuristic-based methods, while CARs and RRN perform better than the algorithms aforementioned, which further illustrates the positive effect of contextual and sequential information on the performance of Top-N recommendations, respectively. CA-RNN and CA-GRU achieve consistent improvements over the six algorithms mentioned above, which highlights the significance and effectiveness of fusing context information into sequential recommendations.

Meanwhile, the proposed ACA-GRU achieves consistent improvements over all other algorithms on MovieLens-1M, and we owe its performance enhancements to the influence of the attention-based correlation context on the transition of user dynamic interest. This result is highly encouraging, indicating that the application of the attention mechanism can capture

Table 2
Performance comparison of the algorithms on Netflix dataset.

Method	R@3	R@5	R@10	P@3	P@5	P@10	F1@3	F1@5	F1@10	MAP
PoP	0.0080	0.0107	0.0138	0.0024	0.0030	0.0023	0.0036	0.0047	0.0039	0.0131
itemKNN	0.0305	0.0447	0.0537	0.0291	0.0307	0.0252	0.0298	0.0364	0.0343	0.1028
BPR	0.0322	0.0570	0.1146	0.0306	0.0326	0.0327	0.0314	0.0414	0.0509	0.1179
FISM	0.0428	0.0715	0.1335	0.0408	0.0409	0.0381	0.0418	0.0520	0.0593	0.1401
CARs	0.0482	0.0896	0.1817	0.0459	0.0512	0.0519	0.0470	0.0651	0.0807	0.1767
RRN	0.0497	0.0943	0.1839	0.0473	0.0539	0.0525	0.0484	0.0686	0.0817	0.1881
CA-RNN	0.0695	0.1076	0.2203	0.0662	0.0614	0.0629	0.0678	0.0782	0.0979	0.2334
CA-GRU	0.0664	0.1247	0.2218	0.0632	0.0712	0.0633	0.0648	0.0907	0.0985	0.2414
ACA-GRU	**0.0739**	**0.1296**	**0.2308**	**0.0704**	**0.0741**	**0.0659**	**0.0721**	**0.0943**	**0.1025**	**0.2620**

Table 3
The influence of embedding size m on recommendation precision.

	$m = 16$		$m = 32$		$m = 64$		$m = 128$	
	R@10	MAP	R@10	MAP	R@10	MAP	R@10	MAP
FISM	0.1285	0.1324	0.1344	0.1372	0.1295	0.1375	0.1358	0.1416
RRN	0.1473	0.1683	0.1694	0.1710	0.1575	0.1884	0.1851	0.1905
CA-RNN	0.1798	0.1931	0.1907	0.2003	0.1918	0.2056	0.2077	0.2079
CA-GRU	0.1811	0.2025	0.2050	0.2021	0.1944	0.2228	0.2221	0.2225
ACA-GRU	0.2122	0.2207	0.2201	0.2310	0.2291	0.2432	0.2419	0.2501

the influence of historical interactions on the transition of user interest. Consequently, it indicates indirectly that ACA-GRU can reduce or eliminate the effect of outliers on prediction results.

To be more explicit on the influence of the attention mechanism in ACA-GRU, we also conduct the same comparison test on the Netflix dataset, and the result is shown in Table 2. From the table, we can see that the proposed ACA-GRU algorithm still outperforms substantially.

From Tables 1 and 2, it has been shown that ACA-GRU achieves the best performance on both datasets on all evaluation indices, which illustrates that: 1) considering both sequential and contextual information can improve the performance of Top-N recommendations; 2) the attention-based correlation context C^u_{At} has a positive and effective impact on each hidden state of the RNN, which can reduce the impact of outliers when modeling user dynamic interest.

4.6. *Influence analysis of parameter* m

To analyze the influence of the embedding size m on RNN-based recommendation methods, Table 3 shows a comparison of R@10 and MAP among FISM, RRN, CA-RNN, CA-GRU, and ACA-GRU with different values of m.

As shown in Table 3, with the increase in embedding size m, the performance of each algorithm improves gradually, which indicates that a higher value of m can further strengthen the expressiveness of the models. As for ACA-GRU with no attention mechanism, it achieves excellent performance in most cases; moreover, its performance is robust to the variations in m. In addition, from the table we can see that, ACA-GRU substantially outperforms both CA-RNN and CA-GRU for different values of m. It indicates the robustness and flexibility of the proposed method on various datasets and embedding dimensions. This further illustrates the positive effects of the attention mechanism on sequence recommendations.

5. Conclusions and future work

In this paper, we presented a novel RNN based recommendation model ACA-GRU that focuses on the design of the attention mechanism in sequential recommendations combining various types of contexts. We considered the influence of the sequential changes and the correlation between adjacent items under various contexts, as each item in the sequence makes a different contribution to the final recommendations. In summary, we have provided the following contributions:

First, we depicted the dynamics of user interest by calculating the global state transition based on the context defined as the input, transition, static interest, as well as correlation context. Based on the definition of the aGRU unit, we were able to calculate the global sequential state transition determined by the contexts, to model the dynamics of user interest. We then quantified the importance of each item by calculating the attention-based correlation context, to reduce or ignore the impact of the abnormal points that are less informative or less predictive. Finally, we conducted empirical studies to validate the effectiveness of the ACA-GRU methods. The experimental results indicated the stronger expressiveness and performance superiority of the proposed model.

Compared with the state-of-the-art, the proposed ACA-GRU is more powerful in its representation ability as it can model user dynamics effectively with the aGRU unit, which considers various contexts in sequential recommendations. Another strength of the ACA-GRU is the design of the attention-based correlation context, which is capable of reducing the impact of the abnormal points in user behavioral history.

For further increases in recommendation accuracy, it is natural to handle the issue of recommending with/for cold-start items/users, as well as enhancing the interpretability of the proposed deep-learning based recommendation models. Another promising direction for further study will be to explore the modeling of other types of data, such as recommendation tasks based on comments and multimedia data with ACA-GRU.

Declaration of Competing Interest

None.

Acknowledgment

This work is supported by the National Natural Science Fund(No. 61672329, 61373149, 61472233), and the Technology Program of Shandong Province under Grant(No.2014GGB01617), Science and Technology Program Project of Shandong Colleges and Universities(No.J17KA070), Undergraduate Education Reform Project of Shandong Province(No.Z2016M016, M2018X197), Excellent Course Project of Shandong Province(No.2012BK294, 2013BK399, 2013BK402), Educational Science Planning Project of Shandong Province(No.ZK1437B010), Natural Science Foundation of Shandong Province(ZR2018MF012), Key Research and Development Project of Shandong Province(2018GX106006) and Soft Science Project of Shandong Province(201704065), PhD Funding Project of Shandong Jianzhu University(X19044Z).

References

[1] G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, in: ACM Conference on Recommender Systems, 2008, pp. 335–336.
[2] Y. Bengio, Learning deep architecture for ai, Found. Trends Mach. Learn. 2 (1) (2009) 1–127.
[3] J. Chen, H. Zhang, X. He, W. Liu, W. Liu, T.S. Chua, Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention, in: International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp. 335–344.
[4] S. Chen, J.L. Moore, D. Turnbull, T. Joachims, Playlist prediction via metric embedding, in: ACM Knowledge Discovery and Data Mining, 2012, pp. 714–722.
[5] H.T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, Wide and deep learning for recommender systems, 2016, pp. 7–10.
[6] K. Cho, B. van Merriënboer, C. Gulcehre, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, Comput. Sci. (2014) 1–11, doi:10.3115/v1/D14-1179.
[7] S. Chopra, M. Auli, A.M. Rush, Abstractive sentence summarization with attentive recurrent neural networks, in: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 93–98.
[8] D. Erhan, Y. Bengio, A. Courville, P.A. Manzagol, P. Vincent, S. Bengio, Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. 11 (3) (2010) 625–660.
[9] X. Geng, Z. Song, Z. Song, Y. Yang, H. Luan, T.S. Chua, One of a kind: User profiling by social curation, in: ACM International Conference on Multimedia, 2014, pp. 567–576.
[10] H. Guo, R. Tang, Y. Ye, Z. Li, X. He, DeepFM: a factorization-machine based neural network for CTR prediction, 2017, pp. 1725–1731.
[11] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.S. Chua, Neural collaborative filtering, 2017.
[12] B. Hidasi, A. Karatzoglou, L. Baltrunas, D. Tikk, Session-based recommendations with recurrent neural networks, Comput. Sci. (2015).
[13] B. Hu, W. Hong, X. Yu, W. Yuan, T. He, Sparse network embedding for community detection and sign prediction in signed social networks, J. Ambient Intell. Humaniz. Comput. (1) (2017) 1–12.
[14] Y. Hu, Y. Koren, C. Volinsky, Collaborative filtering for implicit feedback datasets, in: Eighth IEEE International Conference on Data Mining, 2009, pp. 263–272.
[15] S. Kabbur, X. Ning, G. Karypis, FISM: factored item similarity models for top-n recommender systems, in: ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, 2013, pp. 659–667.
[16] Y. Koren, Collaborative filtering with temporal dynamics, in: Proc Kdd, 53, 2009, pp. 447–456.
[17] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 426–434.
[18] N. Lathia, S. Hailes, L. Capra, X. Amatriain, Temporal diversity in recommender systems, in: International ACM SIGIR Conference on Research and Development in Information Retrieval, 2010, pp. 210–217.
[19] D. Lee, Learning the parts of objects with nonnegative matrix factorization, Nature 401 (6755) (1999) 788–789.
[20] Q. Liu, S. Wu, D. Wang, Z. Li, L. Wang, Context-aware sequential recommendation, 2016.
[21] Q. Liu, S. Wu, L. Wang, Cot: contextual operating tensor for context-aware recommender systems, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 203–209.
[22] R. Liu, H. Wang, X. Yu, Shared-nearest-neighbor-based clustering by fast search and find of density peaks, Inf. Sci. 450 (2018) 200–226.
[23] T. Mei, B. Yang, X.S. Hua, L. Yang, S.Q. Yang, S. Li, VideoReach: an online video recommendation system, in: International ACM SIGIR Conference on Research and Development in Information Retrieval, 2007, pp. 767–768.
[24] T. Mikolov, M. Karafit, L. Burget, J. Cernock, S. Khudanpur, Recurrent neural network based language model, in: INTERSPEECH 2010, Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September, 2010, pp. 1045–1048.
[25] R. Pan, Y. Zhou, B. Cao, N.N. Liu, R. Lukose, M. Scholz, Q. Yang, One-class collaborative filtering, in: Eighth IEEE International Conference on Data Mining, 2008, pp. 502–511.
[26] M.J. Pazzani, D. Billsus, Content-Based Recommendation Systems, 2007.
[27] Y. Qu, H. Cai, K. Ren, W. Zhang, Y. Yu, Y. Wen, J. Wang, Product-based neural networks for user response prediction, 2016, pp. 1149–1154.
[28] S. Rendle, Factorization machines, in: IEEE International Conference on Data Mining, 2011, pp. 995–1000.
[29] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, 2012, pp. 452–461.
[30] S. Rendle, C. Freudenthaler, L. Schmidt-Thieme, Factorizing personalized Markov chains for next-basket recommendation, in: International Conference on World Wide Web, 2010, pp. 811–820.
[31] R.A. Rensink, The dynamic representation of scenes, Vis. Cogn. 7 (1–3) (2000) 17–42.
[32] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors., Read. Cognit. Sci. 323 (6088) (1988) 399–421.
[33] A.M. Rush, S. Chopra, J. Weston, A neural attention model for abstractive sentence summarization, Comput. Sci. (2015).
[34] R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization, in: International Conference on Neural Information Processing Systems, 2007, pp. 1257–1264.

[35] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: International Conference on World Wide Web, 2001, pp. 285–295.

[36] A.I. Schein, A. Popescul, L.H. Ungar, D.M. Pennock, Methods and metrics for cold-start recommendations, in: ACM, 2002, pp. 253–260.

[37] Z. Shuai, L. Yao, A. Sun, T. Yi, Deep learning based recommender system: a survey and new perspectives, ACM Comput. Surv. 1 (1) (2017) 1–35.

[38] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, X. Cheng, Learning hierarchical representation model for nextbasket recommendation, 2015, pp. 403–412.

[39] X. Wei, From recurrent neural network to long short term memory architecture, Tech. Rep. (2013), https://hal.archives-ouvertes.fr/hal-00861063.

[40] C.Y. Wu, A. Ahmed, A. Beutel, A.J. Smola, H. Jing, Recurrent recommender networks, 2017, pp. 495–503.

[41] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, T.S. Chua, Attentional factorization machines: learning the weight of feature interactions via attention networks, 2017, pp. 3119–3125.

[42] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, Y. Bengio, Show, attend and tell: neural image caption generation with visual attention, Comput. Sci. (2015) 2048–2057.

[43] W. Yuan, H. Wang, B. Hu, L. Wang, Q. Wang, Wide and deep model of multi-source information-aware recommender system, IEEE Access 6 (2018) 49385–49398.

[44] W. Zhang, T. Du, J. Wang, Deep learning over multi-field categorical data: A case study on user response prediction, 2016.

[45] Y. Zhang, H. Dai, C. Xu, J. Feng, T. Wang, J. Bian, B. Wang, T.Y. Liu, Sequential click prediction for sponsored search with recurrent neural networks, in: Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, pp. 1369–1375.

[46] W. Zhao, B. Wang, J. Ye, Y. Gao, M. Yang, Z. Zhao, X. Chen, Leveraging long and short-term information in content-aware movie recommendation, 2017.