

# Context-aware Sequential Recommendation

Anonymous CIKM submission

Paper ID 22

## ABSTRACT

Sequential information is becoming a significant factor in modeling user behavior, and various sequential recommendation methods are proposed. As a widely used approach, Markov chain based models are constructed on a strong independence assumption. And recurrent neural networks based methods have achieved state-of-the-art performances in sequential modeling. However, recurrent neural networks and other conventional sequential methods are unable to modeling variety of contextual information in complex real-world application, which has been proved to be very important for modeling user behaviors. Meanwhile, existing context-aware recommendation methods also can not take sequential information and temporal dependency into consideration. Thus, it is vital to address and investigate the problem of context-aware sequential recommendation. In this paper, considering the properties of sequential user behaviors, we conclude two types of contexts: input contexts and transition contexts. Input contexts are the situations that user behaviors happen, e.g., time, location and weather. And transition contexts denote time intervals between adjacent behaviors in historical sequences. Based on these information, we propose a novel Context-Aware Recurrent Neural Networks (CA-RNN) model. Instead of the constant input matrix and transition matrix in conventional recurrent neural networks, the corresponding matrices in CA-RNN change according to input contexts and transition contexts respectively. Then, context-specific input matrices and context-specific transition matrices are incorporated and applied. Experimental results show that the proposed CA-RNN model yields significant improvements over the state-of-the-art sequential recommendation methods and context-aware recommendation methods on three typical datasets, i.e., Tmall dataset and Fashion dataset.

## Keywords

Sequential recommendation, context-awareness, recurrent neural networks

## 1. INTRODUCTION

Nowadays, people are overwhelmed by huge amount of information, the exposure to information made people tired of extracting useful and valuable information that they are interested in. This phenomenon turns out to facilitate the development of recommender systems in social networking, e-commerce, online movie and reading websites. Recommender system has now been an important tool for helping people to filter information and locate their preference. Conventional recommendation methods focus on modeling users' preference based on their historical choices of items and always ignore the sequential information. However, user preferences always change with time. Historical behaviors in different time periods have different effects on users' next choice. Accordingly, sequential recommendation is a crucial task for predicting users' next behaviors in recommender systems.

Nowadays, the importance of sequential information in recommender system has been gradually recognized by researchers in many disciplines, and some efforts have been put into developing CF methods with sequential information [4]. Markov Chain (MC) based models [35, 28, 25, 5] have been widely used for sequential prediction. MC based models aim to predict the users' next behavior based on the past behaviors in sequential data. A transition matrix is estimated, which can give the probability of an action based on the previous ones. For personalized recommendation, Factorization Personalized Markov Chain (FPMC) [28] provide more accurate prediction by factorizing a personalized transition tensor. However, a major problem of MC based models is that all the components are independently combined, indicating that it makes strong independence assumption among multiple factors. Furthermore, MC based methods have been extended via representation learning, which have been applied to next basket recommendation [33]. Recently, as another typical representation learning method, Recurrent Neural Networks (RNN) have been employed to model temporal dependency for different applications successfully. RNN consists of an input layer, an output unit and multiple hidden layers. Hidden representation of RNN can change dynamically along with a sequential history. Each layer of RNN contains an input element and recurrent transition from the previous status, which are captured by an input matrix and a transition matrix respectively. RNN is famous for its being successfully applied in sentence modeling tasks [21, 22, 23]. It also achieves state-of-the-art performances in sequential click prediction [37], location prediction [19] and next basket recommendation [36]. Considering its great per-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$15.00.

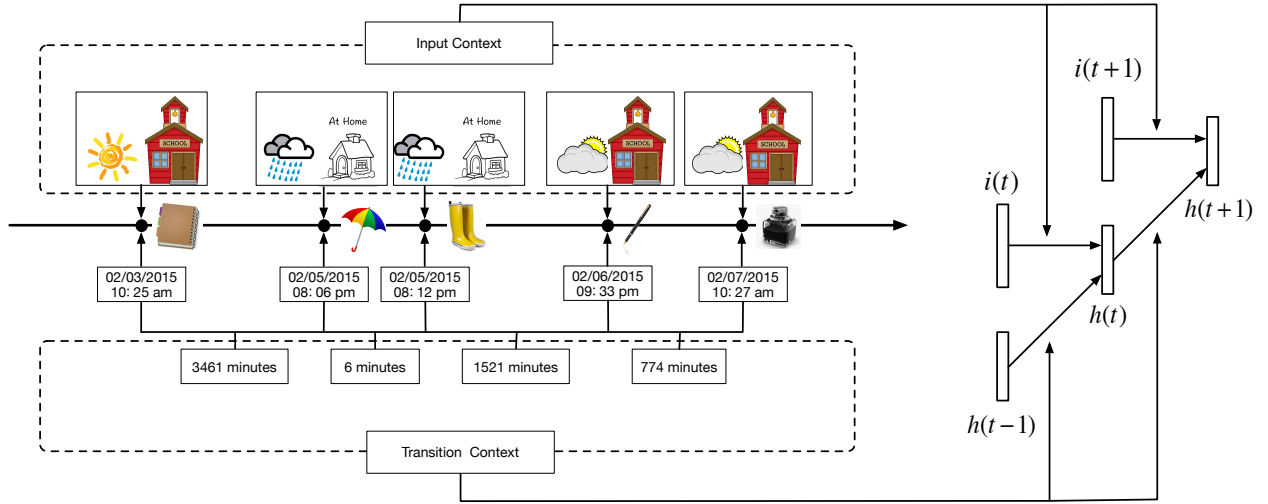


Figure 1: The scenario of online purchase as an example of CA-RNN sequential prediction. This example shows how the input context and transition context are used in recurrent neural networks to predict user’s next preferred item.

performances in sequential modeling tasks, RNN is a suitable tool for sequential recommendation.

Though RNN has achieved satisfactory performances, it still has its drawbacks in sequential recommendation situations. Nowadays, with enhanced ability of systems in collecting information, a great amount of contextual information in recommender systems has been collected such as location, time, weather and so on. These kinds of contextual information have significant effect on user behaviors. For instance, a man may like to watch cartoon with his children while may like to read novels during weekend while may tend to read professional books during weekdays. Contextual information has been proved to be useful in determining users’ preferences in recommender systems [26, 1]. Context-aware recommendation has been extensively studied and several methods have been proposed to achieve state-of-the-art performances [29, 32, 10, 31, 18]. Without considering the rich contextual information in real world, RNN and other sequential recommendation methods can only model historical sequences and are hard to further improve the performances of recommendation in complex real applications. In contrast, there are also no context-aware recommendation methods that take sequential information into consideration. Unfortunately, as shown in the example in Figure 1, complex real-world applications usually have both sequential and contextual information. Thus, context-aware recommendation becomes an emerging task, and adapting RNN for variety of contextual information is a good methodology.

To incorporate contextual information in RNN and other sequential models, we investigate the properties of sequential behavioral histories and conclude two types of contexts: **input contexts** and **transition contexts**. These two types of contexts are shown in the demonstration in Figure 1. And we will explain them one by one.

Input contexts denote the contexts of input elements in behavioral sequences, that is to say, input contexts are situations that users conduct behaviors, e.g., shopping, visiting or reading. Such contexts usually include location (home

or working place), time (weekdays or weekends, morning or evening), weather (sunny or rainy) and so on. As we mentioned above, input contexts have significant effects on predicting present behaviors of users. Similarly, input contexts in the history are also useful for predicting the future. For instance, if a man usually takes exercise in the morning, then we can predict he will go to the fitting room in the morning rather than evening, even if there are also lots of people going to fitting rooms in the evening.

Transition contexts are the contexts for the transition from previous behaviors in historical sequences. They capture how the previous behaviors affect the future. Specifically, transition contexts denote time intervals between adjacent behaviors in sequences in this paper. Generally speaking, longer time intervals and shorter time intervals means different for the transition from the past. If a user’s last behavior on an online shopping website happens half a year ago, then his or her past behaviors have limited effects on the user’s future purchasing and a recommendation with popular items may be a better choice. If the user’s last behavior happens yesterday, then his or her next purchasing will be significantly affected by the previous ones and a personalized recommendation should be made.

Thus, to model sequential information and contextual information in one framework, we propose a novel model called Context-Aware Recurrent Neural Networks (**CA-RNN**). Instead of a constant input matrix for capturing input elements in each layer of RNN, we use context-specific input matrices for each specific input contexts. Similarly, we use context-specific transition matrices for modeling transition effects from previous behaviors in historical sequences under specific transition contexts, i.e., time intervals between adjacent behaviors. Then, we implement our CA-RNN model in a Bayesian Personalized Ranking (BPR) [27] framework to make personalized ranking of recommended items, and Back Propagation Through Time (BPTT) [30] is applied for learning parameters of CA-RNN. In summary, the main contributions of this work are listed as follows:

- We address the problem of context-aware sequential

recommendation, which presents a novel perspective for enhanced recommendation. And we conclude two types of contexts in this problem: input contexts and transition contexts.

- We apply context-specific input matrices to model input contexts, which can capture the properties of contextual information in behavior histories and can make context-aware recommendations for users.
- We incorporate context-specific transition matrices for modeling transition contexts, i.e., time intervals between adjacent behaviors in historical sequences. This process captures transition effects from the past behaviors to the future behaviors within different time intervals.
- Experiments conducted on two real-world datasets show that CA-RNN is effective and clearly outperforms the state-of-the-art methods.

The rest of the paper is organized as follows. In section 2, we review some related work on sequential recommendation and context-aware recommendation. Then we give the formulation of context-aware sequential recommendation in section 3. Section 4 details our CA-RNN model. In section 5, we introduce the learning methods of our proposed model. In section 6, we conduct experiments in two real-world datasets and compare with several state-of-the-art methods. Section 7 concludes our work and discusses future research.

## 2. RELATED WORKS

In this section, we review some related works on sequential recommendation and context-aware recommendation.

### 2.1 Sequential Recommendation

Time-aware neighbourhood models [7, 15, 16] may be the most natural methods for modeling sequences, which employ neighbourhood based algorithms to capture temporal effects via giving more relevance to recent observations and less to past observations. Using frequent pattern mining, sequential pattern based methods [24, 9] seek sequential patterns that occur most frequently to predict the future. However, both neighbourhood based and sequential pattern based methods are unable to reveal the underlying properties in users' historical sequences. And sequential pattern based methods are time consuming in large scale datasets.

Matrix factorization based methods [14] have become the state-of-the-art approach to conventional recommendation, which have been extended to time-aware factorization based models recently. Tensor Factorization (TF) [34, 2] treats time intervals as another dimension and generate latent vectors of time intervals via factorization to capture the underlying properties in historical sequences. TimeSVD++ [13] learns time-aware representations for users and items. However, factorization based models have difficulties in generating latent representations for time intervals which has never or seldom appeared in the training data.

The MC based methods are widely used models for sequential applications [35]. Via factorization of the probability transition matrix, Factorizing Personalized Markov Chain (FPMC) [28] can provide more accurate prediction for each sequence. FPMC is also extended by using user

group [25] or incorporating location constraint [6]. Recently, some factors of human brain have been added into MC based methods, including interest-forgetting curve [5] and dynamics of boredom [11]. However, the main drawback of MC based models is the independent combination of the past components, which lies in a strong independence assumption and confines the prediction accuracy. MC based methods are then extended by representation learning. Hierarchical Representation Model (HRM) [33] learns the representation of behaviors in the last transaction and predicts behaviors for the next transaction. And Personalized Ranking Metric Embedding (PRME) [8] learns embeddings of users according to the location distance.

Recently, a few prediction models, especially language models, are proposed based on neural networks. The most classical language model is proposed via a single layer neural network [3]. And models based on RNN have been successfully used in modeling sentences [21, 22, 23]. RNN also brings satisfying results for sequential click prediction for sponsored search [37], location prediction [19] and next basket recommendation [36]. RNN is showing its great performances in sequential modeling.

### 2.2 Context-aware Recommendation

Context-aware recommender system (CARS) [1] generates more relevant recommendations by adapting them to the specific contextual situation of the user, and it built based on the knowledge of contextual user preferences and typically deal with data records of user, item and context. Recent works on contextual modeling approaches extend the user-item preference relations with contextual information and incorporate factorization models to compute recommendations. Multi-verse recommendation [12] uses tensor factorization to model n-dimensional contextual information. The social network aided context-aware recommender system [20] splits contexts and performs general matrix factorization on each leaf of decision trees. As a extended model of tensor factorization, Factorization Machine (FM) [29] can model a wide variety of contextual information by specifying contextual information as the input dimensions and provide context-aware predictions. The work on Heterogeneous Matrix Factorization (HeteroMF) [10] generates context-specific latent vectors of entities using a context-dependent transfer matrix and the original latent vectors of entities. The Tensor Factorization for MAP maximization (TFMAP) model [32] uses tensor factorization and Mean Average Precision (MAP) objective to model implicit feedback data with contextual information. The newly CARS2 [31] and Contextual Operating Tensor (COT) [18] models represent the common semantic effects of contexts as contextual operating tensor and represents contexts as latent vectors. And the Hierarchical Interaction Representation (HIR) model [17] generates the interaction representation via tensor multiplication which can be applied in context-aware recommender system.

## 3. PROPOSED MODEL

### 3.1 Problem Formulation

We have a set of users denoted as  $U = \{u_1, u_2, \dots\}$ , a set of items denoted as  $V = \{v_1, v_2, \dots\}$ . There are multiple input contexts denoted as  $C^I = \{c_{i1}, c_{i2}, \dots\}$ , such as weather, location, mood and so on. Meanwhile, each context con-

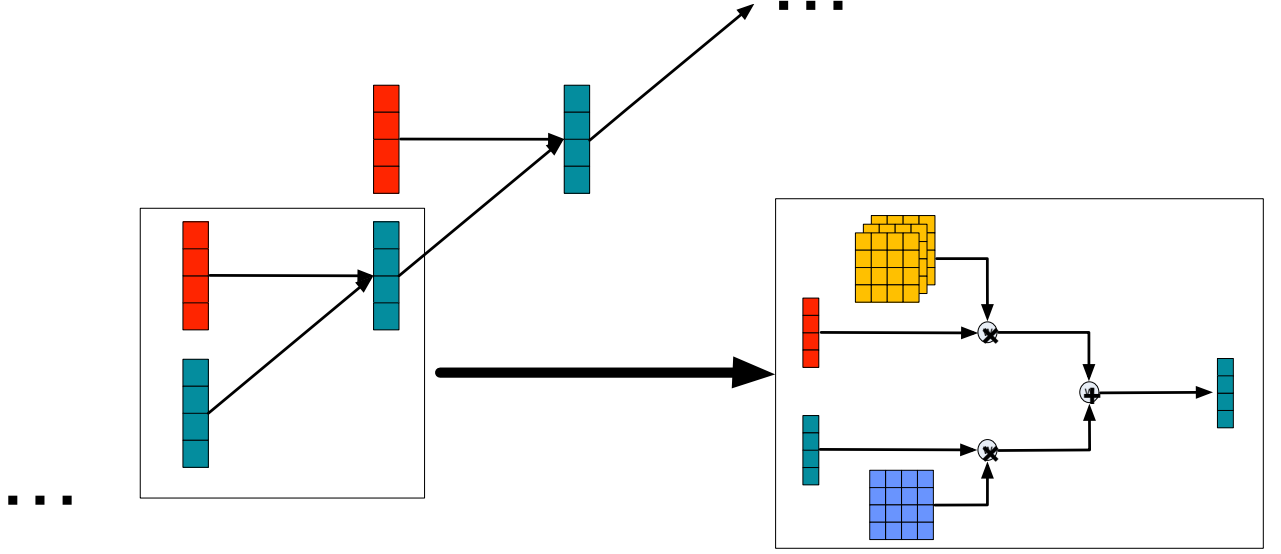


Figure 2: The model

tains several conditions. For example, mood context may consist of happy, sad and flat. The transition context  $C_t$  accounts for the different time interval between two records of user. For each user  $u$ , the behavioral history is given as  $V^u = \{v_1^u, v_2^u, \dots\}$ , where  $v_i^u$  denotes the  $i$  th selected item of user. And the history of all users is denoted as  $V^U = \{V_1^u, V_2^u, \dots\}$ . The behavioral history is associated with corresponding timestamps  $T^u = \{t_1^u, t_2^u, \dots\}$  where  $t_k$  denotes the  $k$  th timestamp in one sequence. For each user  $u$ , at specific timestamp  $t_k$ , a combination of input contexts is denoted as  $\{c_{i1,k}^u, c_{i2,k}^u, \dots\}$ .

### 3.2 Recurrent Neural Networks

The architecture of a recurrent neural network consists of an input layer  $i$ , a hidden layer  $h$ , an output unit, as well as inner weight matrices [37]. RNN model the current output as the function of the previous output and a hidden layer. At each time steps, we can predict the output unit given the hidden layer, and then feed the new output back into the next hidden state. The vector representation of the hidden layer is:

$$\mathbf{h}_k^u = f(\mathbf{r}_{v_k^u} \mathbf{M} + \mathbf{h}_{k-1}^u \mathbf{W}), \quad (1)$$

where  $\mathbf{h}_k^u$  is the  $d$  dimensional representation of user  $u$  at time  $k$  in a behavior sequence,  $\mathbf{r}_{v_k^u}$  denotes the  $d$  dimensional latent vector of the corresponding item the user selects,  $\mathbf{W}$  is the  $d \times d$  dimensional recurrent connection of the previous status propagating sequential signals and  $\mathbf{M}$  denotes the  $d \times d$  dimensional transition matrix for input elements to capture the current behavior of the user. The activation function  $f(x)$  is chosen as a *sigmoid* function  $f(x) = \exp(1/(1 + e^{-x}))$ .

### 3.3 Incorporating Transition Context

Since the length of time interval between two statues has different impacts on prediction and short time interval seems

to have significant effect in predicting next behavior compared with long ones. Therefore, the length of time interval is essential for predicting future behaviors and we incorporate Transition Context for time interval in the formula. We replace the transition matrix  $\mathbf{W}$  with type-specific transition matrices. Thus, Equation should be rewritten as:

$$\mathbf{h}_k^u = f(\mathbf{r}_{v_k^u} \mathbf{M} + \mathbf{h}_{k-1}^u \mathbf{W}_{t_k^u - t_{k-1}^u}) \quad (2)$$

where  $t_k^u$  and  $t_{k-1}^u$  denotes the present and the previous statuses respectively,  $t_k^u - t_{k-1}^u$  is the time interval. All kinds of time intervals are continuously classified into several types according to their length. We suppose that each type should contain near the same quantity of different time intervals. Then, Equation should be rewritten as:

$$\mathbf{h}_k^u = f(\mathbf{r}_{v_k^u} \mathbf{M} + \mathbf{h}_{k-1}^u \mathbf{W}_{\lfloor t_k^u - t_{k-1}^u \rfloor}) \quad (3)$$

where  $\lfloor t_k^u - t_{k-1}^u \rfloor$  denotes for the type of the time interval between  $t_k^u$  and  $t_{k-1}^u$ .

### 3.4 Context-aware Recurrent Neural Networks

Since context information is important in future predictions and conventional RNN cannot model it, it is essential to involved context information in our model. Thus, we add the input context  $c_{I,k}^u$  to the transition matrix  $\mathbf{M}$  in traditional RNN model. In CA-RNN model, given a user  $u$ , his or her vector representation at time  $k$  can be calculated by using an integrated contextual operating matrix for a specific context combination. This matrix describe how a context combination affects the properties of entities and how context-specific latent vectors of users and items can be generated from their original ones. Here, we can calculate context-specific latent vectors as:

$$\mathbf{h}_k^u = f(\mathbf{r}_{v_k^u} \mathbf{M}_{c_{I,k}^u} + \mathbf{h}_{k-1}^u \mathbf{W}_{\lfloor t_k^u - t_{k-1}^u \rfloor}) \quad (4)$$

where  $\mathbf{W}$  is a  $d * d$  dimensional transition matrix from the last status,  $\mathbf{r}_{v_k^u}$  is a  $d$  dimensional vector representation of the input item at time  $k$  of user  $u$ , and  $\mathbf{M}_{c_{I,k}^u}$  is a  $d * d$  transition matrix of correspondence context, denoting the aggregation of all contextual operating matrices for users and items under the context combination.

Linear computation is another common method to calculate the aggregation of all contextual operating matrices, It treats the effect of each context separately, the  $\mathbf{M}_{c_{I,k}^u}$  can be calculated as:

$$\mathbf{M}_{c_{I,k}^u} = \sum_{c_{i,k}^u \in C^I} \mathbf{M}_{c_{i,k}^u} \quad (5)$$

Nonlinear method incorporates sigmoid function to the computation, such as tanh or logistic function. Accordingly, the equation could be rewritten as :

$$\mathbf{M}_{c_{I,k}^u} = \left( 1 + \exp \left( - \sum_{c_{i,k}^u \in C^I} \mathbf{M}_{c_{i,k}^u} \right) \right)^{-1} \quad (6)$$

In STRNN [19], the author incorporates temporal and spatial contexts into conventional RNN by multiplying time-specific operating matrix and distance-specific operating matrix. Analogously, we can treat input contextual operating matrices in the same way :

$$\mathbf{M}_{c_{I,k}^u} = \prod_{c_{i,k}^u \in C^I} \mathbf{M}_{c_{i,k}^u} \quad (7)$$

Moreover, the  $\mathbf{M}_{c_{I,k}^u}$  can be a combination-specific matrix. For example, if there are two contexts: weather and mood, then each context combination has its own matrix as shown in table 1. From a set of matrices for all combinations, the  $\mathbf{M}_{c_{I,k}^u}$  will be chosen as the representation for specific context combination at time  $k$ . If the weather is sunny and mood is happy, then the  $\mathbf{M}_{c_{I,k}^u}$  will equal to  $\mathbf{M}_{11}$ .

**Table 1:**

	Mood			
Weather	happy	sad	plain	
sunny	$\mathbf{M}_{11}$	$\mathbf{M}_{12}$	$\mathbf{M}_{13}$	
rainy	$\mathbf{M}_{21}$	$\mathbf{M}_{22}$	$\mathbf{M}_{23}$	
cloudy	$\mathbf{M}_{31}$	$\mathbf{M}_{32}$	$\mathbf{M}_{33}$	

### 3.5 Context-aware Prediction

Finally, the prediction of CA-RNN can be yielded via calculating inner product of user and item representations. The prediction of whether user  $u$  purchase item  $v$  under context  $c$  at status  $k+1$  can be computed as:

$$y_{u,k+1,c,v} = \mathbf{h}_k^u \mathbf{P}_{c,k+1} (\mathbf{r}_v)^T \quad (8)$$

where  $\mathbf{P}_{c,k+1}$  is the representation of input context and transition context at status  $k+1$ . And it can be calculated by the addition or multiplication of  $\mathbf{M}'$  and  $\mathbf{W}'$ , the corresponding matrix of the two kinds context. Unlike  $\mathbf{M}$  and  $\mathbf{W}$  in equation 4, which are transition matrices,  $\mathbf{M}'$  and  $\mathbf{W}'$  are designed for incorporating context information to improve

the predicting process. According to our experiments, addition performances better, so  $\mathbf{P}_{c,k+1}$  can be calculated as:

$$\mathbf{P}_{c,k+1} = \mathbf{M}_{c_{I,k+1}}' + \mathbf{W}_{[t_{k+1}^u - t_k^u]}' \quad (9)$$

where  $\mathbf{M}_{c_{I,k+1}}'$  is the representation of specific input contexts combination for user  $u$  at time  $k$ , and  $\mathbf{W}_{[t_{k+1}^u - t_k^u]}'$  is the representation of transition context for interval between the current and the next status. These matrices indicate the effect of current context to the following behavior.

### 3.6 Parameter Learning

In this subsection, we introduce the learning process of CA-RNN with Bayesian Personalized Ranking (BPR) [27] and Back Propagation Through Time (BPTT) [30]. BPR is a state-of-the-art pairwise ranking framework for the implicit feedback data. The basic assumption of BPR is that a user prefers a selected item than a negative one. Thus, the training objective of CA-RNN in BPR framework is to maximize the following probability:

$$p(u, k+1, c, v \succ v') = g(y_{u,k+1,c,v} - y_{u,k+1,c,v'}), \quad (10)$$

where  $v'$  denotes a negative item sample, and  $g(x)$  is a non-linear function which is selected as  $g(x) = 1/(1 + e^{-x})$ . Incorporating the negative log likelihood, we can solve the following objective function equivalently:

$$J = \sum \ln(1 + e^{-(y_{u,k+1,c,v} - y_{u,k+1,c,v'})}) + \frac{\lambda}{2} \|\Theta\|^2, \quad (11)$$

where  $\lambda = R, M, W$  denotes all the parameters to be estimated,  $\lambda$  is a parameter to control the power of regularization. And the derivation of  $J$  with respect to the parameters can be calculated as:

$$\frac{\partial J}{\partial \mathbf{r}_v} = - \sum \frac{\mathbf{h}_k^u \mathbf{P}_{c,k+1} e^{-(y_{u,k+1,c,v} - y_{u,k+1,c,v'})}}{1 + e^{-(y_{u,k+1,c,v} - y_{u,k+1,c,v'})}} + \lambda \mathbf{r}_v,$$

$$\frac{\partial J}{\partial \mathbf{r}_{v'}} = \sum \frac{\mathbf{h}_k^u \mathbf{P}_{c,k+1} e^{-(y_{u,k+1,c,v} - y_{u,k+1,c,v'})}}{1 + e^{-(y_{u,k+1,c,v} - y_{u,k+1,c,v'})}} + \lambda \mathbf{r}_{v'},$$

$$\frac{\partial J}{\partial \mathbf{h}_k^u} = - \sum \frac{\mathbf{r}_v (\mathbf{P}_{c,k+1})^T e^{-(y_{u,k+1,c,v} - y_{u,k+1,c,v'})}}{1 + e^{-(y_{u,k+1,c,v} - y_{u,k+1,c,v'})}},$$

$$\frac{\partial J}{\partial \mathbf{P}_{c,k+1}} = - \sum \frac{(\mathbf{h}_k^u)^T \mathbf{r}_v e^{-(y_{u,k+1,c,v} - y_{u,k+1,c,v'})}}{e^{-(y_{u,k+1,c,v} - y_{u,k+1,c,v'})}}.$$

Moreover, parameters in CA-RNN can be further learnt with the BPTT algorithm. According to Equation (2), given the derivation  $\partial J / \partial \mathbf{h}_k^u$ , the corresponding gradients of all parameters in the hidden layer can be calculated as:

$$\frac{\partial J}{\partial \mathbf{h}_{k-1}^u} = \left( f'(\cdot) \otimes \frac{\partial J}{\partial \mathbf{h}_k^u} \right) (\mathbf{W}_{[t_k^u - t_{k-1}^u]})^T,$$

$$\frac{\partial J}{\partial \mathbf{W}_{[t_k^u - t_{k-1}^u]}} = (\mathbf{h}_{k-1}^u)^T \left( f'(\cdot) \otimes \frac{\partial J}{\partial \mathbf{h}_k^u} \right),$$

$$\frac{\partial J}{\partial \mathbf{r}_{v_k^u}} = \left( f'(\cdot) \otimes \frac{\partial J}{\partial \mathbf{h}_k^u} \right) (\mathbf{M}_{c_{I,k+1}}^u)^T,$$

$$\frac{\partial J}{\partial \mathbf{M}_{c_{I,k+1}}^u} = (\mathbf{r}_{v_k^u})^T \left( f'(\cdot) \otimes \frac{\partial J}{\partial \mathbf{h}_k^u} \right).$$

This process can be repeated iteratively in the whole behavioral sequence. Then, according to Equation (5) and (9), given  $\partial J / \partial \mathbf{P}_{c,k+1}$ , the gradients of  $\mathbf{M}'_{c_{i,k+1}^u}$ ,  $\mathbf{M}_{c_{i,k+1}^u}$  and  $\mathbf{W}'_{[t_{k+1}^u - t_k^u]}$  can be computed as:

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{M}'_{c_{i,k+1}^u}} &= \frac{\partial J}{\partial \mathbf{P}_{c,k+1}}, \\ \frac{\partial J}{\partial \mathbf{W}'_{[t_{k+1}^u - t_k^u]}} &= \frac{\partial J}{\partial \mathbf{P}_{c,k+1}}, \\ \frac{\partial J}{\partial \mathbf{M}'_{c_{i,k+1}^u}} &= \frac{\partial J}{\partial \mathbf{M}'_{c_{i,k+1}^u}}, \\ \frac{\partial J}{\partial \mathbf{M}_{c_{i,k+1}^u}} &= \frac{\partial J}{\partial \mathbf{M}_{c_{i,k+1}^u}}.\end{aligned}$$

Now, after calculating all the gradients, we can employ stochastic gradient descent to estimate the model parameters.

## 4. EXPERIMENTS

In this section, we empirically analysis the performance of CA-RNN. First we introduce the settings of our experiments. Then we compared our CA-RNN to the state-of-the-art methods. Finally, we analysis the impact of parameters, especially the length of hidden layer.

### 4.1 Experiment Datasets

### 4.2 Evaluation Metrics

### 4.3 Compared Methods

### 4.4 Input Context Aggregation Computation

As mentioned above, there are several methods, such as Linear, Nonlinear, Multiplied and combination-specific, to compute the aggregation matrix for all contextual operating matrices. Here in this part, we will discuss the effect of each method on Tianchi dataset.

**Table 2: Performance of CA-RNN with different method of computing the  $\mathbf{M}_{c_{i,k}^u}$  evaluated by reall**

	Method	recall@1	recall@5	recall@10
Tianchi	Linear	0.1366	0.3819	0.5427
	Nonlinear	0	0	0
	Mutiplied	0.0761	0.2052	0.3220
	Combination	0.1302	0.3659	0.5220
Tamll	Linear	0	0	0
	Nonlinear	0	0	0
	Mutiplied	0	0	0
	Combination	0	0	0

As indicated in Table 2, on all evalution matrices, the Linear method is clearly the best one of integrating input contexts. The performance of combination-specific method seems to be close to that of the Linear method. However, the amount of parameters of combination-specific method is clearly larger than Linear method, especially when the number of conditions in each context is large. Thus, we suppose

to utilize Linear method to compute the synthesized matrix for all contextual operating matrices in CA-RNN.

## 4.5 Performance Comparison

## 4.6 Impact of Parameters

## 5. CONCLUSIONS AND FUTURE WORK

In this paper, we have address the problem of context-aware sequential recommendation and proposed a novel method, i.e., context-aware recurrent neural networks. In CA-RNN, the constant input matrix of conventional RNN is replaced with context-specific input matrices to modeling complex real-world contexts, e.g., time, location and weather. Meanwhile, to model transition contexts, i.e., time intervals between adjacent behaviors in historical sequences, context-specific transition matrices are incorporated, instead of the constant one in conventional RNN. Then, to make personalized ranking of recommended items, we implement CA-RNN in the BPR framework and apply BPTT to learn parameters in CA-RNN. The experimental results on two real datasets show that CA-RNN outperforms the state-of-the-art sequential and context-aware recommendation models.

In the future, we can further investigate the following direction. In CA-RNN, parameters are the same for different users, which does not confirm to practical situations. So, to make better personalized recommendation, we need to find a method to determine different parameters for different users or different user groups. Moreover, we didn't take items' features, e.g., categories, descriptions and images of items, into consideration. Thus, incorporating our CA-RNN with features of items may also be our next step.

## 6. REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer, 2011.
- [2] M. T. Bahadori, Q. R. Yu, and Y. Liu. Fast multivariate spatio-temporal analysis via low rank tensor learning. In *NIPS*, pages 3491–3499, 2014.
- [3] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *JMLR*, 3:1137–1155, 2003.
- [4] P. G. Campos, F. Díez, and I. Cantador. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *UMUAI*, 24(1-2):67–119, 2014.
- [5] J. Chen, C. Wang, and J. Wang. A personalized interest-forgetting markov model for recommendations. In *AAAI*, pages 16–22, 2015.
- [6] C. Cheng, H. Yang, M. R. Lyu, and I. King. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, pages 2605–2611, 2013.
- [7] Y. Ding and X. Li. Time weight collaborative filtering. In *CIKM*, pages 485–492, 2005.
- [8] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan. Personalized ranking metric embedding for next new poi recommendation. In *IJCAI*, pages 2069–2075, 2015.
- [9] N. Hariri, B. Mobasher, and R. Burke. Context-aware music recommendation based on latent topic sequential patterns. In *RecSys*, pages 131–138, 2012.
- [10] M. Jamali and L. Lakshmanan. Heteromf: recommendation in heterogeneous information networks using context dependent factor models. In *WWW*, pages 643–654, 2013.
- [11] K. Kapoor, K. Subbian, J. Srivastava, and P. Schrater. Just in time recommendations: Modeling the dynamics of

**Table 3: Performance comparison on two datasets evaluated by recall, F1-score, MAP and AUC.**

		recall@1	recall@5	recall@10	F1-score@1	F1-score@5	F1-score@10	MAP	AUC
Tianchi	TOP	0	0	0	0	0	0	0	0
	MF	0	0	0	0	0	0	0	0
	RNN	0	0	0	0	0	0	0	0
	CA-RNN	<b>0.0304</b>	<b>0.1524</b>	<b>0.2714</b>	<b>0.0304</b>	<b>0.0508</b>	<b>0.0493</b>	<b>0.1038</b>	<b>0.8115</b>
Tmall	TOP	0	0	0	0	0	0	0	0
	MF	0	0	0	0	0	0	0	0
	RNN	0	0	0	0	0	0	0	0
	CA-RNN	<b>0.1654</b>	<b>0.4986</b>	<b>0.6812</b>	<b>0.1654</b>	<b>0.1662</b>	<b>0.1239</b>	<b>0.3238</b>	<b>0.9042</b>

- boredom in activity streams. In *WSDM*, pages 233–242, 2015.
- [12] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *RecSys*, pages 79–86, 2010.
- [13] Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.
- [14] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.
- [15] N. Lathia, S. Hailes, and L. Capra. Temporal collaborative filtering with adaptive neighbourhoods. In *SIGIR*, pages 796–797, 2009.
- [16] N. N. Liu, M. Zhao, E. Xiang, and Q. Yang. Online evolutionary collaborative filtering. In *RecSys*, pages 95–102, 2010.
- [17] Q. Liu, S. Wu, and L. Wang. Collaborative prediction for multi-entity interaction with hierarchical representation. In *CIKM*, pages 613–622, 2015.
- [18] Q. Liu, S. Wu, and L. Wang. Cot: Contextual operating tensor for context-aware recommender systems. In *AAAI*, pages 203–209, 2015.
- [19] Q. Liu, S. Wu, L. Wang, and T. Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*, 2016.
- [20] X. Liu and K. Aberer. Soco: a social network aided context-aware recommender system. In *WWW*, pages 781–802, 2013.
- [21] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, pages 1045–1048, 2010.
- [22] T. Mikolov, S. Kombrink, L. Burget, J. H. Cernocký, and S. Khudanpur. Extensions of recurrent neural network language model. In *ICASSP*, pages 5528–5531, 2011.
- [23] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Cernocký. Rnnlm-recurrent neural network language modeling toolkit. In *ASRU Workshop*, pages 196–201, 2011.
- [24] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Using sequential and non-sequential patterns in predictive web usage mining tasks. In *ICDM*, pages 669–672, 2002.
- [25] N. Natarajan, D. Shin, and I. S. Dhillon. Which app will you use next?: Collaborative filtering with interactional context. In *RecSys*, pages 201–208, 2013.
- [26] C. Palmisano, A. Tuzhilin, and M. Gorgoglione. Using context to improve predictive modeling of customers in personalization applications. *Knowledge and Data Engineering, IEEE Transactions on*, 20(11):1535–1549, 2008.
- [27] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.
- [28] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, pages 811–820, 2010.
- [29] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *SIGIR*, pages 635–644, 2011.
- [30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5:3, 1988.
- [31] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, and A. Hanjalic. Cars2: Learning context-aware representations for context-aware recommendations. In *CIKM*, pages 291–300, 2014.
- [32] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, A. Hanjalic, and N. Oliver. Tfmap: Optimizing map for top-n context-aware recommendation. In *SIGIR*, pages 155–164, 2012.
- [33] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng. Learning hierarchical representation model for next basket recommendation. In *SIGIR*, pages 403–412, 2015.
- [34] L. Xiong, X. Chen, T.-K. Huang, J. G. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SDM*, pages 211–222, 2010.
- [35] Q. Yang, J. Fan, J. Wang, and L. Zhou. Personalizing web page recommendation via collaborative filtering and topic-aware markov model. In *ICDM*, pages 1145–1150, 2010.
- [36] F. Yu, L. Qiang, S. Wu, L. Wang, and T. Tan. A dynamic recurrent model for next basket recommendation. In *SIGIR*, 2016.
- [37] Y. Zhang, H. Dai, C. Xu, J. Feng, T. Wang, J. Bian, B. Wang, and T.-Y. Liu. Sequential click prediction for sponsored search with recurrent neural networks. In *AAAI*, pages 1369–1376, 2014.