

PAPER • OPEN ACCESS

## Weighted sequence loss based recurrent model for repurchase recommendation

To cite this article: Pengda Chen and Jian Li 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **490** 062062

View the [article online](#) for updates and enhancements.

# Weighted sequence loss based recurrent model for repurchase recommendation

Pengda Chen<sup>1,\*</sup>, Jian Li<sup>1</sup>

<sup>1</sup>School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China

\*Corresponding author e-mail: pengpeng@bupt.edu.cn, lijian@bupt.edu.cn

**Abstract.** Next basket recommendation becomes an increasing concern. Repurchase recommendation, i.e., predicting which products a user will buy again in a user's next order, is a key subproblem. However, most conventional models are not able to extract the whole important features to describe the customer's repurchase process: context information and sequential information. In our work, we firstly utilize the causal dilated convolutions and recurrent neural network to capture context information and sequential information in different ways. Furthermore, the information extracted by causal dilated convolutions and recurrent neural network is combined at each time step for recommendation. More importantly, to effectively adapt the repurchase recommendation, we introduce a weighted sequence loss, which is able to ignore invalid logloss at special time steps to guide *the RNN combined with causal dilated convolutions*(RCCNN) training. A deep experimentation shows that RCCNN is able to explain the customer repurchase behaviors, and provide reasonable recommendation.

## 1. Introduction

Repurchase recommendation is a key subproblem of next basket recommendation[1]. Next basket recommendation is aimed to predict items that are more likely to be purchased in her next transaction, which focuses on a combination of long-term and short-term interests. However, repurchase recommendation predicts which products a user will buy again in a user's next order, which mainly models users' periodic consumption pattern. Compared with next basket recommendation, repurchase recommendation need to utilize context information fully, such as the specific time interval.

With the development of e-commerce, people's daily shopping and consumption, such as fruits and vegetables, are beginning to be done more by app rather than going to the supermarket in person. As a result, E-commerce supermarket such as *Instacart* began to rise. Because people will periodically repeat the same commodity in the daily shopping consumption, for e-commerce supermarkets, the accurate prediction of the user's repeated purchases in the next shopping will affect the earnings of the e-commerce supermarket to a great extent. It can be seen that repurchase recommendation as a subproblem will play an more important role in the product recommendation.

Most conventional models[2] for repurchase recommendation firstly obtain the characteristics that can describe the periodic consumption rules of the user via extensive feature engineering, and then predict whether the user will buy the specific goods at the next shopping time by the classifiers. In this paper, a recurrent neural network model combined with causal dilated convolutions is proposed, which can not only better describe the interaction between user behaviors and context information, but also better describe the periodic consumption rules of the user. At the same time, in order to better train the



model to complete the repurchase recommendation, we put forward a weighted sequence loss, thus avoiding the effect of the invalid loss of certain time steps.

## 2. Related work

Repurchase recommendation is mainly related to using context information for recommendation and next basket recommendation. We will introduce the related research on these two aspects separately as follows.

Rational use of context information is an important topic in the field of recommendation. For example, Pedro et al.[3] have explored timing information for recommendation widely. To win the Netflix Prize [4], Koren [5] combined temporal features with Collaborative Filtering (CF) model to capture the significant long-term dynamics in the Netflix dataset. To explore how preferences evolve in shorter time-scales, session-based models[6,7] have been presented. Moreover, geographical data has been widely used in recommendation such as probabilistic models [8, 9], matrix factorization [10], and tensor factorization [11]. Although using context information for collaborative filtering has been a great success, there are few studies about how to introduce these information in neural recommender systems. In the previous researches, deep neural network (DNN) just incorporate the context information as direct features for recommendation systems[12]. For recurrent neural network, [13, 14] include timing information as features, but these features are just concatenated with input, which provides limited benefit.

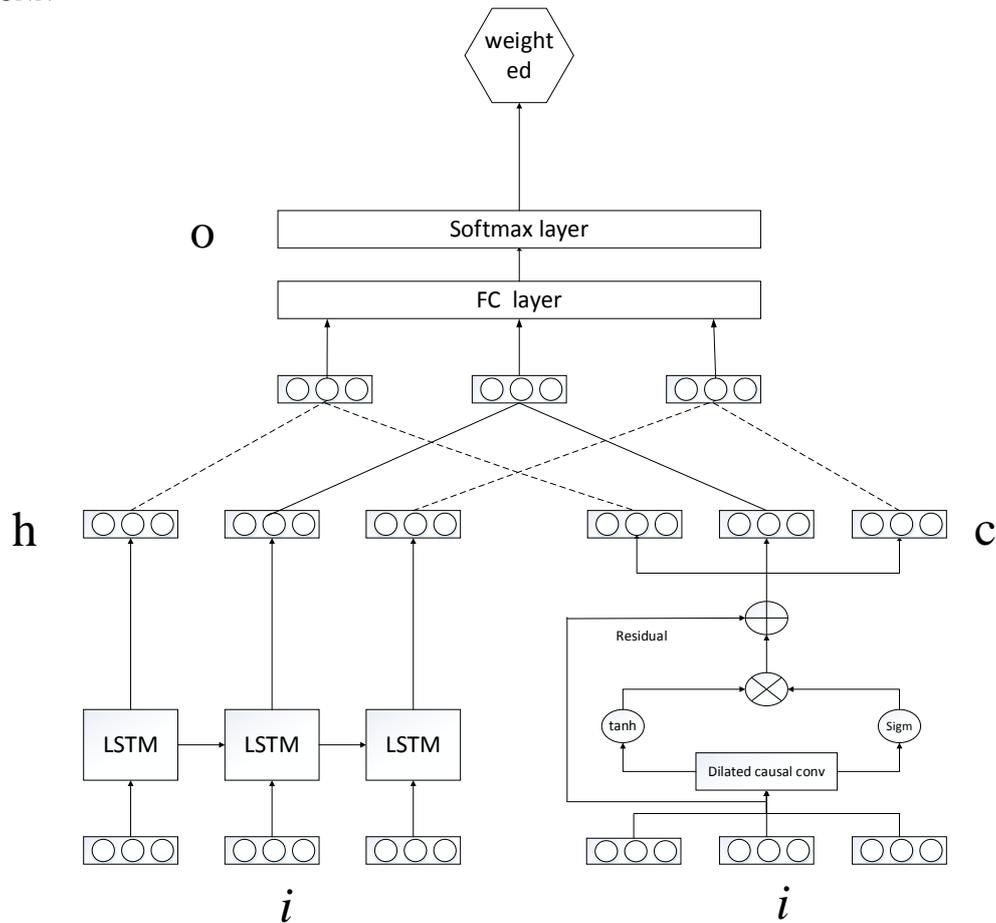
Next basket recommendation is the task of predicting items that are more likely to be purchased in her next transaction. Collaborative filtering (CF) models are the first to be used for next basket recommendation. Matrix factorization (MF)[15] could capture users' general interests through the vectors of users and items, but it ignores the sequential information of historical orders. Furthermore, the classical sequential model for recommendation is MC[16], which can extract sequence information from transaction history and then recommend items for customers based on these sequence information. Factorizing Personalized Markov Chains(FPMC)[17] can capture the sequence information and user general interests at the same time, and Hierarchical Representation Model(HRM)[18] is a hybrid model which can more effectively introduce sequence information to model user general interests. However, all the MC based methods are just able to extract local sequence information between two adjacent transaction records. To extract global sequential information through all baskets of customers, a dynamic recurrent model(DREAM)[1] was proposed for next basket recommendation. As a subproblem of next basket recommendation, the traditional methods[2] for repurchase recommendation mainly extract features from the transaction history of all customers while ignoring sequential features.

## 3. The proposed approach

### 3.1. Problem definition

For repurchase recommendation, each user has his own historical records of products and we should model his periodic repurchase behaviors. For a user  $u$ ,  $S^u$  is the set of all products that  $u$  has purchased in previous baskets. For each product  $v \in S^u$  of a user  $u$ , we calculate the probability  $p_v^u$  that  $u$  will repurchase  $v$  in the next basket according the historical transaction data. And then, we recommend a ranking list of products  $v \in S^u$  for the user  $u$ . For repurchase recommendation with historical record data, problem is defined as recommending a ranking list of products  $v \in S^u$  for each customer  $u$  at the next basket.

3.2. RCCNN



**Figure 1.** The general architecture of the RCCNN.

The general architecture of RCCNN is showed in Figure 1. Basically, for each user  $u$  and for each product  $v \in S^u$ , we generate a training sequence, where the RCCNN predicts 0 or 1 at each time step for "wouldn't order" or "would order". At each time step, the RCCNN gets some input describing the current user, the current product, time information on the current order and statistical information on the prior order. And the detail input information at each time step is shown in Table 1. Specially, we also train embedding layers for products and users as parts of the RCCNN. The embedding for product  $v$  and user  $u$  are repeated as input to the RSCNN at each time step. According to the output of RCCNN at the last time step, We can get the repurchase probability of each product  $p_v^u$  for each user  $u$ , and then we can recommend a ranking list products  $v \in S^u$ .

**Table 1.** Description of input information at each time step.

Input at each time step	Description
embedding vector	the current user vector
	the current product vector
time information on the current order	time interval since the prior order
statistical information on prior order	whether product is ordered in prior order
	number of products in prior order
	number of reorder products in prior order

As is shown in Figure 1, the  $O_t$  represents the probability that the user would purchase the product at time  $t$  and  $y_t$  represents whether the user would purchase the product at time  $t$ . To make the possibility more accurate, we should capture contextual and sequential information in different ways at the same time, so we concatenate the RNN's hidden layer  $h_t$  and CNN's output layer  $c_t$ .  $W$  is a linear transformation between the output of RCCNN and the purchase probability. Then, the purchase probability can be computed as:

$$O_t = \sigma(W * [h_t, c_t]) \quad (1)$$

The score  $O_t$  represents the possibility that the user will purchase the product at time  $t$ .

**3.2.1. Causal dilated convolutions.** To fully model the input information at each time step, we introduce causal dilated convolutions in the model RCCNN. To make sure there is no information leakage from future, we incorporate the causal dilated convolutions[19] in the convolution neural network and we are able to extract the contextual information and sequential information again in a different way from the recurrent neural network. As a result, we can extract contextual and sequential information in a more comprehensive way and provide more reasonable recommendation.

We also use the gated structure as shown in the WaveNet[19]:

$$z = \tanh(W_{f,k} * x) \otimes \sigma(W_{g,k} * x) \quad (2)$$

In our model,  $k$  is 6, which means the causal dilated convolutions have 6 layers. And for the sake of convenience, we just show the situation of one layer causal dilated convolution in Figure 1.

### 3.3. Weighted sequence loss

$T^u$  is the total number of time steps for each user  $u$ . Then, the repurchase recommendation task for a user  $u$  is to predict whether each product  $v \in S^u$  will be ordered at  $t_r^u$  time step. Obviously, for a user  $u$  and for each product  $v \in S^u$ , we can easily know whether the product  $v$  was ordered from  $t_1^u$  time step to  $t_{r-1}^u$  time step. And we should be able to predict whether the product  $v$  will be ordered at  $t_r^u$  time step according to the previous time steps. To model users' periodic purchase of products exactly, we calculate the mean softmax loss from  $t_1^u$  time step to  $t_{r-1}^u$  time step instead of just calculating the softmax loss at  $t_{r-1}^u$  time step. Because of that, we introduce a basic average sequence loss function, as shown in Equation 3.

$$L_{avg} = -\frac{1}{T-1} \sum_{i=t_1^u}^{t_{r-1}^u} y_i \log o_i + (1-y_i) \log(1-o_i) \quad (3)$$

where the  $O_i$  is calculated by the Equation 1.

The basic average sequence loss function acts on the whole sequence input, so it is able to calculate a general loss, but the distinct importance of time steps is still awfully neglected. As a matter of fact, only parts of the time steps' softmax loss could play an active role in network training for repurchase recommendation assignment. As a result, the average sequence loss is not proper loss value for the repurchase recommendation. To overcome this shortcoming, we introduce a weighted vector  $W$  to make sure that each time step has its own special weight value, as shown in Equation 4. The RCCNN with weighted sequence loss can get a more useful sequence loss by ignoring invalid softmax loss at special time steps. In the repurchase recommendation problem, we try to model the regular pattern of repeated purchase of each product by users. So the model is not able to predict whether the user  $u$

will purchase the product  $v$  until the user first buys the product at the time step  $t_v^u$ . Obviously, the time step  $t_v^u$  is related to the product  $v$  and the user  $u$ . And the softmax loss from time step  $t_1^u$  to time step  $t_v^u$  is invalid, which we should ignore. For the weighted sequence loss, we let the  $W$  between time step  $t_1^u$  and time step  $t_v^u$  to zero. So we can make the model to learn regular pattern of repeated purchase of each product by users more effectively.

$$\begin{aligned} L_{\text{wei}} &= -\frac{1}{T-1} \sum_{i=t_1^u}^{t_{T-1}^u} W_i [y_i \log o_i + (1-y_i) \log(1-o_i)] \\ &= -\frac{1}{T-t_v^u-1} \sum_{i=t_v^u+1}^{t_{T-1}^u} y_i \log o_i + (1-y_i) \log(1-o_i) \end{aligned} \quad (4)$$

## 4. Experiments

### 4.1. Datasets and Baselines

For the task of repurchase recommendation, our experiments are made on a real-world dataset from *Instacart*, which is a grocery ordering and delivery app. The *Instacart* dataset contains a sample of 589,928 grocery orders from 10,000 *Instacart* users. The dataset provides the week and hour of day the each basket was purchased, and the relative time interval between two adjacent baskets. The train and test sets are split based on time. Specially, for each user, the last order belongs to the test set and the rest orders except the last order belong to the train set. To generate the training sequences, for each user and each product, we combine the current order's time information with the prior order's statistical information at each time step.

We compare RCCNN to the following methods including traditional machine learning methods and the original LSTM network. For traditional machine learning methods, we have manually extracted more than 100 features from profiles for users, products, time and their interactions, and then, we use LR and XGBOOST separately to predict whether a user will repurchase a product in the next order. And for original LSTM network, we don't use the causal dilated convolutions and the weighted sequence loss to show the performance of RCCNN clearly.

### 4.2. Metrics

We recommend a ranking list of  $K$  products for each user  $u$  in the task of repurchase recommendation. To evaluate the performance of repurchase recommendation, we adapt F1-score and AUC-score. F1-score can both measure the precision and recall measurements. AUC-score is the probability that a model will rank a randomly chosen positive one higher than a randomly chosen negative one. For both metrics, the larger the value, the better the performance.

In addition, the number of orders per user is different. To assess the influence of order quantity on recommendation results, we divide the users into four groups according to the order quantity and also use the above metrics to evaluate repurchase recommendation.

### 4.3. Results and analyses

**Table 2.** The performance of repurchase recommendation.

Methods	F1		AUC
	K=3	K=5	
LR	0.3195	0.3589	0.7661
XGBOOST	0.3270	0.3649	0.7759
LSTM	0.3285	0.3677	0.7775
RCCNN	<b>0.3313</b>	<b>0.3715</b>	<b>0.7812</b>

**Table 3.** The F1-score of repurchase recommendation with different order quantity.

Methods	F1(K=3)				F1(K=5)			
	<=25	26~50	51~75	76~100	<=25	26~50	51~75	76~100
LR	0.3302	0.2770	0.2775	0.2532	0.3674	0.3251	0.3286	0.2963
XGBOOST	0.3370	0.2867	0.2858	0.2659	0.2858	0.3393	0.3440	0.3192
LSTM	0.3364	<b>0.2962</b>	0.2950	0.2850	0.2950	0.3432	0.3492	<b>0.3287</b>
RCCNN	<b>0.3396</b>	0.2960	<b>0.2990</b>	<b>0.2908</b>	<b>0.2990</b>	<b>0.3468</b>	<b>0.3575</b>	0.3284

**Table 4.** The AUC-score of repurchase recommendation with different order quantity.

Methods	AUC			
	<=25	26~50	51~75	76~100
LR	0.7496	0.8297	0.8513	0.8450
XGBOOST	0.7599	0.8367	0.8600	0.8569
LSTM	0.7611	<b>0.8399</b>	0.8633	0.8600
RCCNN	<b>0.7657</b>	0.8397	<b>0.8634</b>	<b>0.8627</b>

As shown in Table 2, the RCCNN model can outperform all the base models in both two metrics on the real dataset. For the baseline XGBOOST, we just manually extract features by experience, so XGBOOST can't fully model the user's consuming patterns about repurchase. For the original RNN, user vectors, product vectors and context information are just concatenated as the input at each time step. As a result, it ignores the interaction between users, products and contextual information at each time step. Specially, these results show that the LSTM with causal dilated convolutions is effective in capturing sequential features and the input information at each time step. The causal dilated convolutions help the RCCNN to model the interaction between users, products and contextual information, which is benefit to repurchase recommendation. And compared with the basic sequence loss, the weighted sequence loss is more fit for the repurchase recommendation assignment. Moreover, the Table 3, 4 demonstrate the RCCNN always provides the best repurchase recommendation compared with LR, XGBOOST and the original LSTM, no matter how many orders the users have.

## 5. Conclusion

This paper proposes a RNN with causal dilated convolutions(RCCNN) model for repurchase recommendation. The causal dilated convolutions help the RCCNN to model the interaction between users, products and contextual information at each time step and extract sequential information, which is benefit to repurchase recommendation. Moreover, we introduce a weighted sequence loss function, which could ignore invalid logloss at special time steps to guide RCCNN training well. Detailed experiments on a real-world dataset illustrated the effectiveness of our model.

## References

- [1] Yu, Feng, et al. "A Dynamic Recurrent Model for Next Basket Recommendation." International ACM SIGIR Conference on Research and Development in Information Retrieval ACM, 2016:729-732.
- [2] Liu, Guimei, et al. "Repeat Buyer Prediction for E-Commerce." ACM SIGKDD International Conference on Knowledge Discovery and Data Mining ACM, 2016:155-164.
- [3] Pedro G Campos, Fernando Díez, and Ivón Cantador. 2014. Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction* 24, 1-2 (2014), 67 - 119.

- [4] James Bennett, Stan Lanning, and others. 2007. The netflix prize. In Proceedings of KDD cup and workshop, Vol. 2007. New York, NY, USA, 35.
- [5] Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Commun.ACM* 53, 4 (2010), 89 – 97.
- [6] Jannach, Dietmar, and M. Ludewig. "When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation." Eleventh ACM Conference on Recommender Systems ACM, 2017:306-310.
- [7] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. ACM, 17 – 22.
- [8] Amr Ahmed, Liangjie Hong, and Alexander J Smola. 2013. Hierarchical geo-graphical modeling of user locations from social media posts. In Proceedings of the 22nd international conference on World Wide Web (WWW). ACM, 25 – 36.
- [9] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. 2010. You are where you tweet: a content-based approach to geo-locating twitter users. In Proceedings of the 19th ACM international conference on Information and knowledge management. ACM, 759 – 768.
- [10] Haokai Lu and James Caverlee. 2015. Exploiting geo-spatial preference for personalized expert recommendation. In Proceedings of the 9th ACM Conference on Recommender Systems (RecSys). ACM, 67 – 74.
- [11] Hancheng Ge, James Caverlee, and Haokai Lu. 2016. TAPER: A contextual tensor-based approach for personalized expert recommendation. (2016).
- [12] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems (RecSys). ACM, 191 – 198.
- [13] How Jing and Alexander J. Smola. 2017. Neural Survival Recommender. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM). 515 – 524.
- [14] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J. Smola, and How Jing. 2017. Recurrent Recommender Networks. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM). 495 – 503.
- [15] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, (8):30 – 37, 2009.
- [16] S. Chen, J. L. Moore, D. Turnbull, and T. Joachims. Playlist prediction via metric embedding. In SIGKDD, pages 714 – 722, 2012.
- [17] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In WWW, pages 811 – 820, 2010.
- [18] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng. Learning hierarchical representation model for nextbasket recommendation. In SIGIR, pages 403 – 412, 2015.
- [19] Hsu, Charles. "WaveNet processing brassboards for live video via radio." *Journal of Electronic Imaging* 7.4(1998):755-769.