Modeling the sequential behaviors of online users in recommender systems

Tuc Nguyen[a], Linh Ngo Van[a], and Khoat Than[a]

[a]Hanoi University of Science and Technology, No. 1, Dai Co Viet road, Hanoi, Vietnam

ABSTRACT

Analyzing sequential user behaviors plays an important role to build an effective recommender system and it has been paid a great deal of attention by researchers. Previous work exploits two types of sequential user behaviors: Item sequence (each user interacts with items in order) and sequential interactions on an item (e.g. clicking an item, then adding it to cart, finally purchasing it). While a vast number of studies focus on modeling item sequence, a few works exploit sequential interactions on an item in recent years. However, there is no work that focuses on both of them. In our work, we propose a novel model which directly models both the types to capture user behaviors completely. Our model can combine multiple types of behaviors as a sequence of actions, moreover, it can model users' preferences through time with the sequence items which they have interacted in the past. The intensively experimental results show that our method significantly outperforms the effective baselines which are designed to learn from either item sequence or sequential user interactions.

Keywords: Recommendation systems, Implicit Feedback, Explicit Feedback, Deep Learning, Collaborative Filtering.

1. INTRODUCTION

In the age of information technology, e-commerce is growing in popularity. One of main challenges is to help customers find out suitable items for their preferences. Recommendation Systems (RS) is attracting a lot of attention to address this challenge. RS aims to predict the "rating" or "preference" which the user would give to an item. User preferences can be learned from analyzing the past interactions of the user, such as the item sequence which the user interacted or the sequential interactions (e.g: clicking, adding to cart, purchasing) which the user does on an item.

Regarding the sequential interactions, traditional methods for RS usually classify these interactions into two groups: implicit feedback (e.g. clicking, viewing)[1,2] and explicit feedback (e.g. rating, purchasing)[3] and manipulate either of them to find out the relationship between users and items. Recently, numerous studies[4–7] exploited both implicit and explicit feedback in the recommender system. Liu et al.[5] and Shi et al.[6] developed models based on matrix factorization method for implicit and explicit feedback. However, they ignored the order of user interactions on an item. Neural Multi-Task Recommendation (NMTR)[7] directly modeled multiple types of behavior as multi-task learning, where each task learned a specific type of interaction. NTMR attempted to keep the sequential relation by using the output of this action as the input to predict the next action. However, propagating information with only one scalar between two consecutive tasks is too simple, thus NMTR cannot capture the more complex association between implicit and explicit interactions. Recently, Tuan et al.[8] proposed a novel Implicit-to-Explicit model (ITE) that models the sequential behavior of users in two consecutive phases in a single task. ITE based on the sequential influences: The preferences of user influence implicit actions and then the implicit actions influence the explicit actions. The experimental results showed that ITE outperforms the baselines. However, ITE ignores the order of item sequence that a user interacts.

In terms of the item sequence, various studies[9–17] showed that exploiting the sequential data in a session obtain impressive results. In particular, deep learning emerges as an effective solution to model the item sequence. Hidasi et al.[18] applied Gated Recurrent Unit (GRU)-based RNN for session-based recommendation and used the

Further author information: (Send correspondence to Khoat Than)
Khoat Than: E-mail: khoattq@soict.hust.edu.vn

ranking loss as the objective function. It achieved a significant improvement over traditional methods. Tan et al.[16] improved GRURec[18] by utilizing data augmentation, embedding dropout and pre-training techniques. Ren et al.[15] proposed the repeated consumption phenomenon as the context of session information and introduced a repeat-explore mechanism to train the model. Li et al.[17] explored an attention mechanism to model the user item sequence and capture the main user purpose in the current session. Liu et al.[19] used the embedding of the last-click to represent the current user interests and built an attention model to capture the short-term user intention. These models can learn the user preferences through the sequence items which the user has interacted. However, they ignored the order of interactions which the user does on an item. Therefore, they cannot model many types of behaviors, that a user makes on an item, to learn sufficient representation of user preferences in the real-world.

In this paper, we propose a novel model that directly models both the types (item sequence and sequential user interactions on an item) to capture user behaviors namely *Sequential Implicit To Explicit (SITE)*. SITE uses continuous two-phase in a single task to model the multiple types of behavior as a sequence of actions. Therefore, it is possible to model complex relationships between user interactions on an item. Moreover, our model can capture the evolution of user preferences through time by exploiting sequence items they have interacted in the past.

We conduct experiments on two large datasets and the results show that directly modeling sequential user interactions on an item and the item sequence actually achieves many impressive results. Our method outperforms the state-of-the-art baselines by significant margin.

The rest of the paper is organized as follows. Section 2 briefly reviews some related works. Section 3 presents the details of SITE model and discusses several benefits. In Section 4, we conduct extensive experiments and describe the evaluation results of the proposed model. Finally, we have some conclusions in Section 5.

2. RELATED WORK

2.1 Matrix Factorization

Matrix factorization (MF)[20] is one of the most popular methods that aims to learn the latent representations of each user and each item. It decomposes the user-item interaction matrix into the product of two matrices with lower dimensions. More specifically, let K dimensional vectors p_u and q_v be the latent vectors of user u and item v respectively, the rating of user u for item v is approximated to the inner product of p_u and q_v:

$$\hat{y}_{uv} = p_u^T q_v = \sum_{k=1}^{K} p_{uk} q_{vk} \tag{1}$$

The target of the MF is to learn those latent vectors. Each latent vector can reveal some intuition of user's preferences or item's features.

The original MF assumes the interaction of user u for the item v is a linear combination of two vectors p_u, q_v with the same dimensions' weights. Xiangnan He et al.[13] mentioned the MF's limitation, they presented a neural network architecture to model latent features of users and items for collaborative filtering. However, the original MF only use implicit feedback, therefore, it cannot exploit the wide variety of interactions. The recent models, which are investigated in the next subsection, are proposed to simultaneously use many kinds of interactions.

2.2 Exploiting implicit and explicit feedback in recommender system

Combining multiple types of behavior (e.g: implicit, explicit) has been paid a lot of attention by several researchers recently.[1,4,5] These methods combined the kinds of feedback in an unified objective function, however, they ignored the order of sequential behaviors. Therefore, it cannot naturally model the behavior of the user. Shi et al.[6] and Gao et al.[7] attempted to find out the latent relationship between multiple behaviors by using the neural network architecture to model user-item interactions. However, the way that these models associate implicit and explicit behaviors is too simple, thus it cannot represent the complex relationship between user and item.

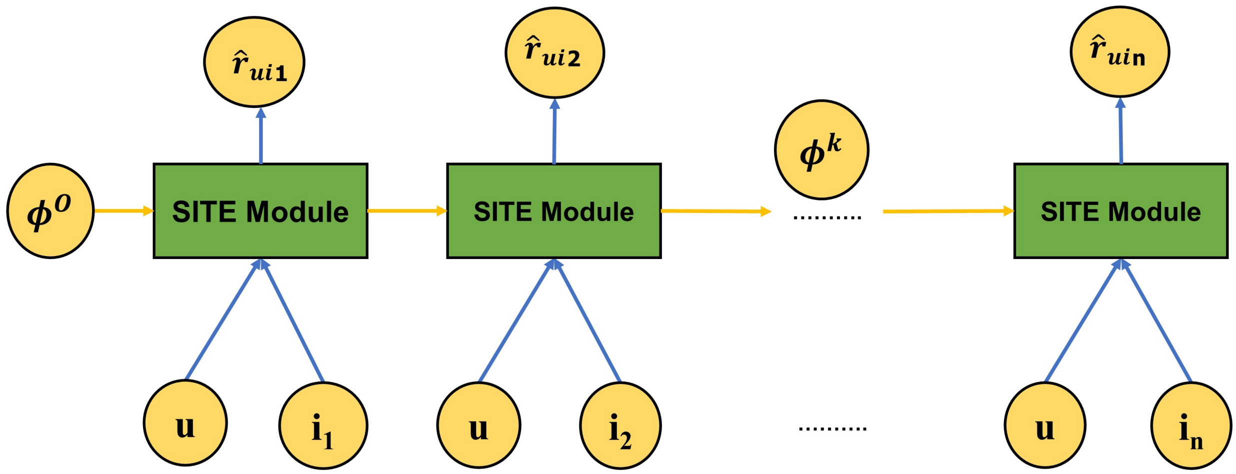

Figure 1: **The architecture of SITE model**

2.3 Session-based Recommendation

Session-based recommender systems (SRS) is a subject in the recommendation and aims to recommend the next item based on the item sequence which a user interacts in history. Accordingly, it can be grouped into two main approaches: next-item recommendations[9,15,16,18,19] that recommend the next item of the current session and next-session recommendations (e.g: next-basket)[21,22] that recommend partial or whole of future sessions. These models learned the user preferences through the item sequence which a user has interacted in the past but they cannot model the sequential user interactions on an item.

3. PROPOSED MODEL

In this section, we propose a novel model, namely Sequential Implicit To Explicit, which simultaneously models two types of user behaviors: the item sequence (which a user interacts in order) and the sequential interactions on an item (clicking, then adding to cart, finally purchasing). In this paper, we limit the sequential interactions in scope of two kinds: implicit and explicit behaviors in which a user first does a implicit behavior, then decides a explicit one based on the result from the implicit.

Throughout this paper, we use the following notations:

- M, N: The number of users and items respectively.
- $X = (x_{ui})_{M \times N}$: Data of implicit behaviors where $x_{ui} = 1$ when user u has implicitly interacted with the item i; $x_{ui} = 0$ otherwise.
- $Y = (y_{ui})_{M \times N}$: Data of explicit behaviors where $y_{ui} = 1$ when user u has explicitly interacted with the item i; $y_{ui} = 0$ otherwise.
- $u = (\alpha_1, \alpha_2, \alpha_3, \dots \alpha_V)$ and $i = (\beta_1, \beta_2, \beta_3, \dots, \beta_W)$: The representations vectors of user u and item i (where V and W are the dimensions) respectively.
- Session $\mathcal{S} = \{s_1, s_2, ..., s_{|\mathcal{S}|}\}$ with $s = \{i_1, i_2, ..., i_{|s|}\}$ is a set of items that a user have interacted in session s.

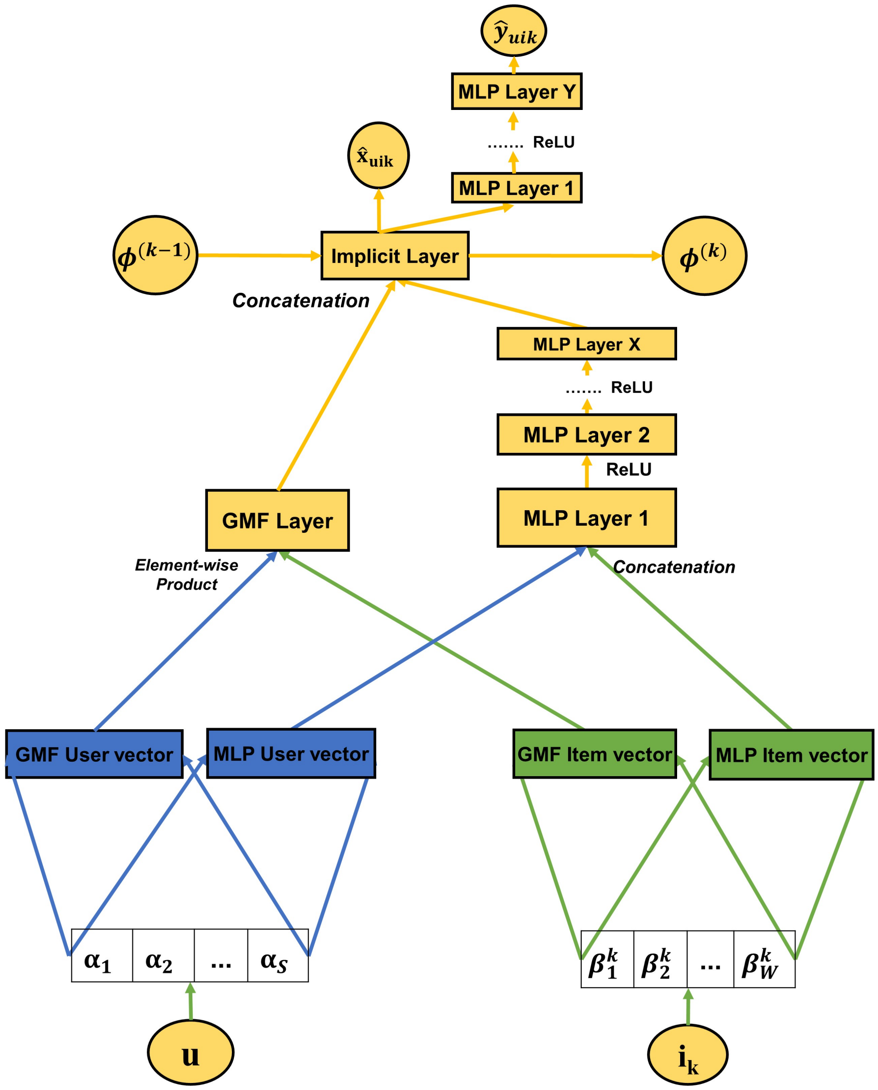

Figure 2: **Detail architecture of SITE module**

3.1 Sequential Implicit to Explicit

In this work, we will exploit both the item sequence and the sequential interactions to model the deep relations between user and item. We present the architecture of the **Sequential Implicit To Explicit (SITE)** model in Fig 1. The input consists of a user u and list of items $(i_1, i_2, ..., i_n$ where n is the number of item in the user session) which are interacted by u. We use the one-hot encoding of user and item for representing the input. SITE will model multiple types of behavior as a sequence of actions, moreover, it represents the evolution of user preferences through time with the sequence items they have interacted in the past.

For the sequential interactions on an item, we inherit the architecture of ITE model[8] which can well capture the relations between implict and explicit behaviours. In details, we use an implicit module which includes both General Matrix Factorization (GMF) and MultiLayer Perceptron (MLP) to learn non-linear combination of a pair of use and item. The output of this module is implicit feedback. Regading explicit feedback, we utilize an explicit module which consists of a MultiLayer Perceptron (MLP) network. In order to capture the order of user interactions, the last hidden layer of the implicit module, which is named as **Implicit Layer**, is used as the input of the explicit module (Fig 2). The output of the explicit module is explicit feedback.

In terms of the item sequence in a session, we use an recurrent neural networks (RNN) to propagate the influences of the user interactions on the previous items upon the user behaviours on the current item. In particular, each state of RNN is an ITE model to model the user behaviours on an item. Then, we connect the Implicit Layers of ITE models between two consecutive states. This way helps us remain the orders of both the item sequence and the sequential user actions on an item.

In details, let p_u^G, p_u^M, q_i^{kG}, q_i^{kM} be the embedding vectors of user u and item i^k (which is the k^{th} item in session of user u) for GMF and MLP Layer respecitively.

$$\phi^{kGMF} = p_u^G \odot q_i^{kG} \tag{2}$$

where $\odot$ denotes the pair-wise product. Meanwhile, p_u^M, q_i^{kM} are concatenated and then are fed into a Multi-Layer Perceptron network including L hidden layers. Particularly, each layer performs the following computation:

$$\phi_{(l+1)}^{MLP} = f(W_{(l)}\phi_{(l)}^{MLP} + b_{(l)}) \tag{3}$$

where l is the layer index, f is the activation function (usually ReLU function). At the Implicit Layer of state k^{th} (Fig. 2), the concatenation of GMF Layer and MLP Layer is combined with the Implicit Layer of the previous state.

$$\phi^{kI} = f(W.concat(\phi^{kGMF}, \phi_{(L)}^{kMLP}) + \tilde{W}.\phi^{(k-1)I}) \tag{4}$$

where W and $\tilde{W}$ are weight matrices, f is the activation function and ϕ^{kI} is the representation of a pair of user and item k^{th} at the implicit layer. $\phi^{(k-1)I}$ is a zero vector when $k = 1$. The output of the implicit module $\hat{x}_{ui}^k$ denotes the likelihood that u will perform an implicit behavior with item k^{th}:

$$\hat{x}_{ui}^k = \sigma(\mathbf{h}_I^T \phi^{kI}) \tag{5}$$

Then, the **explicit module** receives the representation ϕ^{kI} of the Implicit Layer as a input to propagate through several hidden layers of MLP. At the hidden layer l of the explicit module performs the following computation:

$$\phi_{(l)}^{kE} = f(U_{(l-1)}\phi_{(l-1)}^{kE} + b_{(l-1)}) \tag{6}$$

where $\phi_{(l)}^{kE}$ is the representation at the layer l, $\phi_{(0)}^{kE} = \phi^{kI}$, and U denotes the weight of the hidden layer. The last layer of explicit module $\phi_{(Y)}^{kE}$ is used to compute the likelihood that u will perform an explicit behavior with the item i:

$$\hat{y}_{ui}^k = \sigma(\mathbf{h}_E^T \phi_{(Y)}^{kE}) \tag{7}$$

where $\mathbf{h}$ is the edge weight of the output layer.

$\hat{x}_{ui}^k$ and $\hat{y}_{ui}^k$ are the predictive score of implicit and explicit behaviors of user u on item k^{th} in the session of the user u. After learning the model, the probability that user u will relevant with the item i can be computed by:

$$\hat{r}_{ui} = \hat{x}_{ui}^k \hat{y}_{ui}^k \tag{8}$$

It is obvious that $\hat{r}_{ui}$ can achieve high value when both $\hat{x}_{ui}$, $\hat{y}_{ui}$ are high.

3.2 Learning the model

The loss function is defined through the session of user as below:

$$\mathcal{L} = \sum_{k=1}^{n} \eta \mathcal{L}_I^k(\hat{x}, x) + \mathcal{L}_E^k(\hat{y}, y) \tag{9}$$

Dataset	RetailRocket	Recobell
# Implicit	396923	2285261
# Explicit	18450	52786
# Users	36751	206203
# Items	83274	118293
# Sparsity	99.987%	99.999%

Table 1: Statistics of the Retail Rocket and Recobell datasets

where $\mathcal{L}_I^k$ and $\mathcal{L}_E^k$ are the objective functions of the implicit and explicit modules respectively, η is a hyperparameter to balance between the terms. The loss function of the implicit and explicit modules are calculated as follows:

$$\mathcal{L}_I = \sum_{(u,i)\in\{\mathcal{S}_I^+ \bigcup \mathcal{S}_I^-\}} x_{ui} \log \hat{x}_{ui} + (1 - x_{ui}) \log(1 - \hat{x}_{ui}) \tag{10}$$

$$\mathcal{L}_E = \sum_{(u,i)\in\{\mathcal{S}_E^+ \bigcup \mathcal{S}_E^-\}} y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log(1 - \hat{y}_{ui}) \tag{11}$$

where $\mathcal{S}_I^+$ and $\mathcal{S}_E^+$ denote the sets of observed implicit and explicit interactions in the session $\mathcal{S}$ respectively. $\mathcal{S}_I^-$ and $\mathcal{S}_E^-$ are the sets of negative samples for implicit and explicit behaviors in the session $\mathcal{S}$. We use the Adam method for optimizing this objective function.

4. EXPERIMENTS

In this section, we conduct experiments to examine the performances of the SITE model.

4.1 Datasets

We perform experiments on the two datasets: Retail rocket and Recobell.

- **Retailrocket:** is collected from a e-commerce website*. It contains information about the history of user interactions collected in 4.5 months with the timestamp. Multiple types of interactions include view, click, add-to-cart and transaction. In our experiments, view and click actions are considered as implicit behavior, while add-to-cart and transaction are explicit behavior. Users which have more than 5 interactions are used to build experimental data.
- **Recobell:** is an e-commerce dataset† in which data is collected over a period of 2 months from August 2016 to October 2016. User interactions contain view (implicit behavior) and order (explicit behavior). Similar to Retail rocket dataset, we only use users which have more than 5 interactions to conduct experiments.

Some statistics of these data are presented in Table 1.

4.2 Evaluated scenario and Measures

We use the leave-one-out[23] strategy to evaluate the performances of SITE. For each user, the last item, which the user does explicit action, is used to test, while we train SITE on the remaining interacted items. In the evaluation phase, we create a set of item candidates to recommend for this user. The set consists of the test item and 999 random items with which the user has not interacted. We examine how SITE can find out the test item from candidates.

In experiments, we use two common measures *Hit Ratio (HR)*[24] and *Normalized Discounted Cumulative Gain (NDCG)*[25] to examine the predictive ability.

*https://www.kaggle.com/retailrocket/ecommerce-dataset
†http://www.recobell.co.kr/rb/main.php?menu=pakdd2017

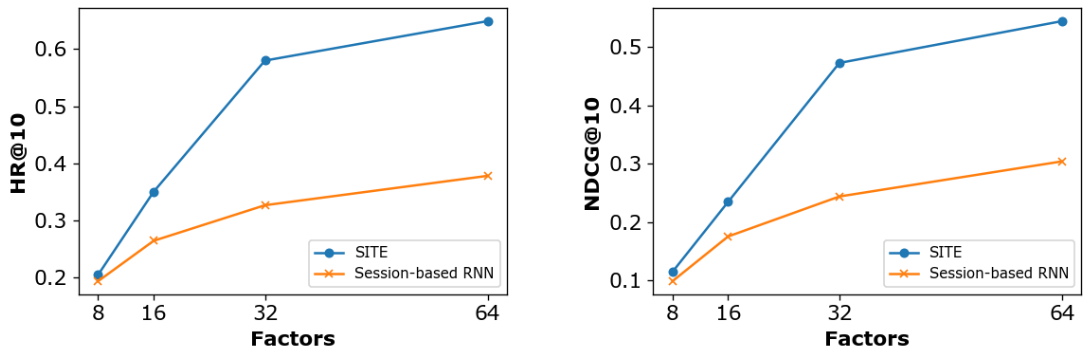

Figure 3: Comparison of the various models in **Retail Rocket** dataset with K varied

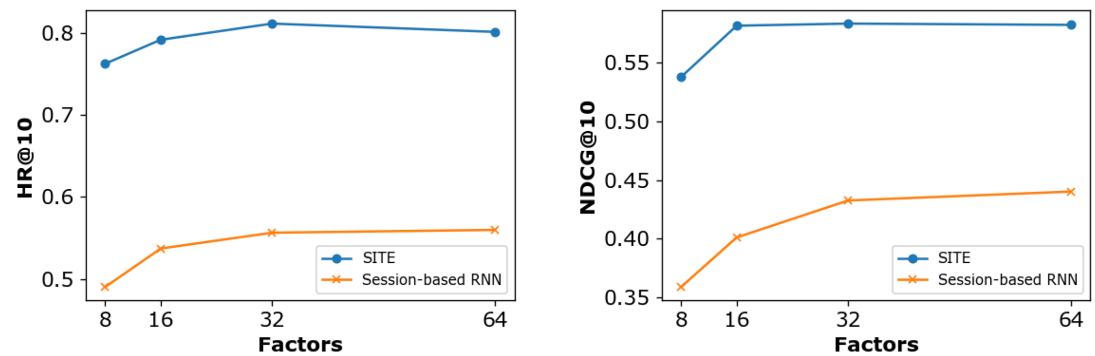

Figure 4: Comparison of the various models in **Recobell** dataset with K varied

Hit Ratio (HR): For each user, HR@K corresponds to whether the test item belongs to the top K items that SITE recommends to the user. HR@K can be formulated as follows:

$$HR@K = \begin{cases} 1, \text{if test item is in top K.} \\ 0, \text{otherwise.} \end{cases} \tag{12}$$

Normalized Discounted Cumulative Gain (NDCG): Instead of checking whether the test item is in top K recommended items as Hit Ratio, NDCG@K considers the order of the test item in the top K items. Therefore, the value of NDCG is always smaller than the value of HIT. NDCG@K can be formulated as follows:

$$NDCG@K = \begin{cases} \dfrac{log(2)}{log(i+1)}, \text{if test item is ranked at position i.} \\ 0, \text{otherwise.} \end{cases} \tag{13}$$

The final HR@K and NDCG@K are averaged on the HR@K and NDCG@K of all users. We compute HR@K and NDCG@K with K = 10 to evaluate the model.

4.3 Performance comparison

We use RNN session-based model[18] as the baseline.

Parameter Settings: Some hyper-parameters having major impacts to the model are: lr (learning rate), η (the parameter determine the influence of implicit and explicit module into final loss), K (number of factors using for the user, item representation). We tune the hyper-parameters with $lr \in \{0.0001, 0.0005, 0.001, 0.005\}$, $\eta \in \{0.1, 0.2, 0.5, 1.0, 2.0\}$. After the tuning process, we select $lr = 0.001$, $\eta = 0.5$ that achieve the best result. In the experiments, we also consider the impact of **number factors** into these models with $K \in \{8, 16, 32, 64\}$.

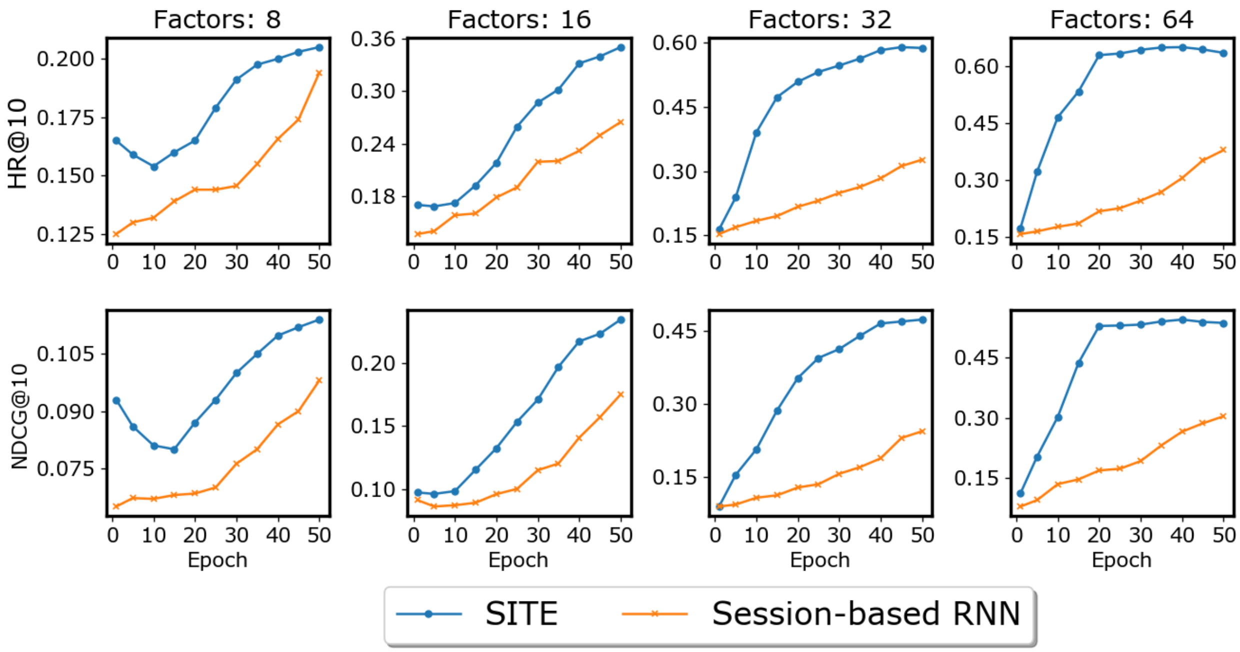

Figure 5: Comparison of the various models in **Retail Rocket** when increasing the number of epochs gradually. From left to right: $K = \{8; 16; 32; 64\}$

Performance comparison: Fig. 3 and Fig. 4 show the HR@10 and NDCG@10 with respect to the number of factors $K = \{8, 16, 32, 64\}$. It is obvious that SITE achieves higher results of both *HR@10* and *NDCG@10* than RNN session-based. Using the deep network to exploit the complex behavior of the user is a main reason why SITE is better than the RNN session-based model. For the result of Recobell in Fig. 4, the best *HIT@10* and *NDCG@10* of SITE is 0.8113 and 0.583 (K = 32), while the highest of RNN session-based are 0.559 and 0.442 ($K = 64$) respectively. In Retail rocket dataset, the ability prediction of SITE increases when the number of hidden factors increases (Fig. 3), but seemly different from Recobell. Specifically, the *HR@10* and *NDCG@10* of SITE reaches its maximum value with number factors $K = 32$ and decreases when the number of factors $K = 64$ (Fig. 4).

Fig. 5 and Fig. 6 reveal the performance of SITE on Retail Rocket and Recobell when the number of epochs increases gradually from 1 to 50. In each epoch, the model learns through the whole training data. As we can see, SITE still achieves better results than the RNN session-based model in most of the cases. In Recobell dataset, the SITE model gains high performance in first several epochs, which indicates that we can save training time for our model.

5. CONCLUSION

We presented a model based on deep learning network architecture to model simultaneously both kinds of sequential behaviours: The item sequence and the sequential user interactions on an item. Deep learning architecture is completely effective in modeling the complex user-item relationships and capturing the evolution of user preferences over time. The experiments showed that the model performed well on the two Retail Rocket and Recobell datasets.

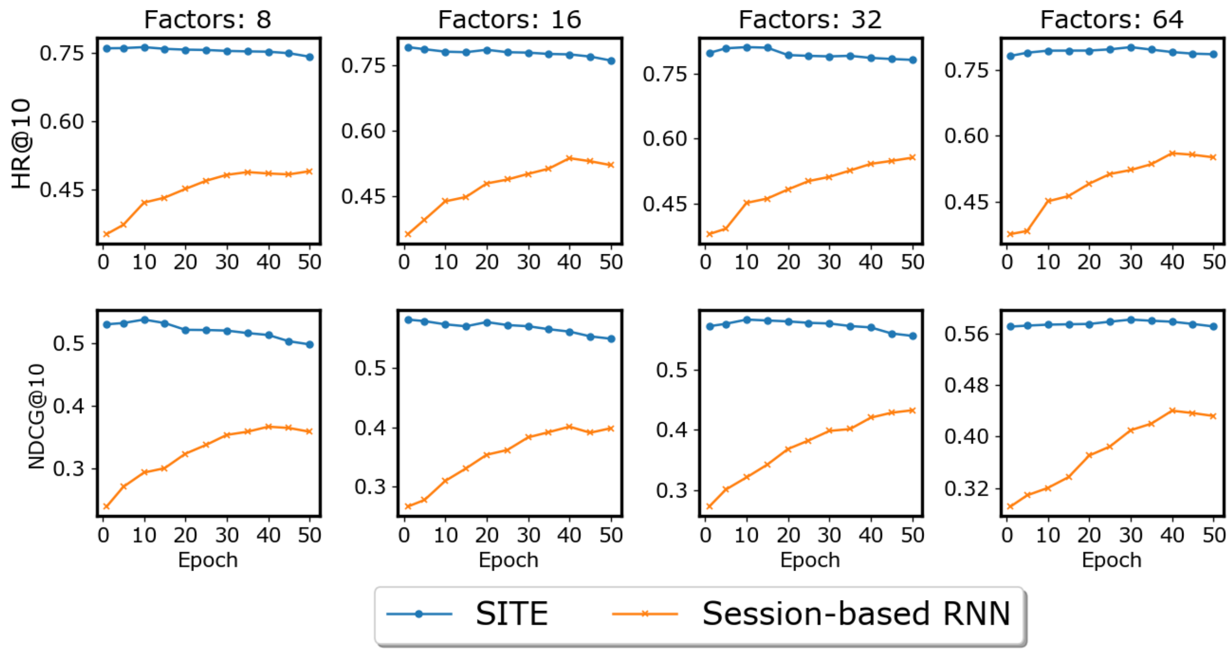

Figure 6: Comparison of the various models in **Recobell** when increasing the number of epochs gradually. From left to right: $K = \{8; 16; 32; 64\ \}$

ACKNOWLEDGMENTS

This unnumbered section is used to identify those who have aided the authors in understanding or accomplishing the work presented and to acknowledge sources of funding.

REFERENCES

[1] Hu, Y., Koren, Y., and Volinsky, C., "Collaborative filtering for implicit feedback datasets," in [*2008 Eighth IEEE International Conference on Data Mining*], 263–272, Ieee (2008).

[2] Ding, J., Yu, G., He, X., Quan, Y., Li, Y., Chua, T.-S., Jin, D., and Yu, J., "Improving implicit recommender systems with view data," in [*Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*], 3343–3349, International Joint Conferences on Artificial Intelligence Organization (7 2018).

[3] Mnih, A. and Salakhutdinov, R. R., "Probabilistic matrix factorization," in [*Advances in neural information processing systems*], 1257–1264 (2008).

[4] Koren, Y., Bell, R., and Volinsky, C., "Matrix factorization techniques for recommender systems," *Computer* **42**(8), 30–37 (2009).

[5] Liu, N. N., Xiang, E. W., Zhao, M., and Yang, Q., "Unifying explicit and implicit feedback for collaborative filtering," in [*Proceedings of the 19th ACM international conference on Information and knowledge management*], 1445–1448 (2010).

[6] Shi, W., Lu, T., Li, D., Zhang, P., and Gu, N., "Multi-task matrix factorization for collaborative filtering," in [*2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD)*], 343–348, IEEE (2017).

[7] Gao, C., He, X., Gan, D., Chen, X., Feng, F., Li, Y., Chua, T.-S., and Jin, D., "Neural multi-task recommendation from multi-behavior data," in [*2019 IEEE 35th International Conference on Data Engineering (ICDE)*], 1554–1557, IEEE (2019).

[8] Tuan, A. P., Trong, N. N., Trong, D. B., Van, L. N., and Than, K., "From implicit to explicit feedback: A deep neural network for modeling the sequential behavior of online users," in [*Asian Conference on Machine Learning*], 1188–1203 (2019).

[9] Hu, L., Cao, L., Wang, S., Xu, G., Cao, J., and Gu, Z., "Diversifying personalized recommendation with user-session context," in [*Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*], 1858–1864 (2017).

[10] Zhang, F., Yuan, N. J., Lian, D., Xie, X., and Ma, W.-Y., "Collaborative knowledge base embedding for recommender systems," in [*Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*], 353–362 (2016).

[11] Mei, L., Ren, P., Chen, Z., Nie, L., Ma, J., and Nie, J.-Y., "An attentive interaction network for context-aware recommendations," in [*Proceedings of the 27th ACM International Conference on Information and Knowledge Management*], 157–166 (2018).

[12] Wang, H., Wang, N., and Yeung, D.-Y., "Collaborative deep learning for recommender systems," in [*Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*], 1235–1244 (2015).

[13] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S., "Neural collaborative filtering," in [*Proceedings of the 26th international conference on world wide web*], 173–182 (2017).

[14] Loyola, P., Liu, C., and Hirate, Y., "Modeling user session and intent with an attention-based encoder-decoder architecture," in [*Proceedings of the Eleventh ACM Conference on Recommender Systems*], 147–151 (2017).

[15] Ren, P., Chen, Z., Li, J., Ren, Z., Ma, J., and de Rijke, M., "Repeatnet: A repeat aware neural recommendation machine for session-based recommendation," in [*Proceedings of the AAAI Conference on Artificial Intelligence*], **33**, 4806–4813 (2019).

[16] Tan, Y. K., Xu, X., and Liu, Y., "Improved recurrent neural networks for session-based recommendations," in [*Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*], 17–22 (2016).

[17] Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., and Ma, J., "Neural attentive session-based recommendation," in [*Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*], 1419–1428 (2017).

[18] Hidasi, B., Karatzoglou, A., Baltrunas, L., and Tikk, D., "Session-based recommendations with recurrent neural networks," in [*4th International Conference on Learning Representations (ICLR)*], (2016).

[19] Liu, Q., Zeng, Y., Mokhosi, R., and Zhang, H., "Stamp: short-term attention/memory priority model for session-based recommendation," in [*Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*], 1831–1839 (2018).

[20] Koren, Y., "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in [*Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*], 426–434 (2008).

[21] Le, D.-T., Lauw, H. W., and Fang, Y., "Correlation-sensitive next-basket recommendation," in [*Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*], 2808–2814, International Joint Conferences on Artificial Intelligence Organization (7 2019).

[22] Yu, F., Liu, Q., Wu, S., Wang, L., and Tan, T., "A dynamic recurrent model for next basket recommendation," in [*Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*], 729–732 (2016).

[23] He, X., Zhang, H., Kan, M.-Y., and Chua, T.-S., "Fast matrix factorization for online recommendation with implicit feedback," in [*Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*], 549–558 (2016).

[24] Deshpande, M. and Karypis, G., "Item-based top-n recommendation algorithms," *ACM Transactions on Information Systems (TOIS)* **22**(1), 143–177 (2004).

[25] He, X., Chen, T., Kan, M.-Y., and Chen, X., "Trirank: Review-aware explainable recommendation by modeling aspects," in [*Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*], 1661–1670 (2015).