Collaborative Sequence Prediction for Sequential Recommender

Shuzi Niu[1], Rongzhi Zhang[2]
Institute of Software, Chinese Academy of Sciences,China[1], University of Chinese Academy of Sciences,China[2]
shuzi@iscas.ac.cn,zhangrzpanda@gmail.com

ABSTRACT

With the surge of deep learning, more and more attention has been put on the sequential recommender. It can be casted as sequence prediction problem, where we will predict the next item given the previous items. RNN approaches are able to capture the global sequential features from the data compared with the local features derived in Markov Chain methods. However, both approaches rely on the independence of users' sequences, which are not true in practice. We propose to formulate the sequential recommendation problem as collaborative sequence prediction problem to take the dependency of users' sequences into account. In order to solve the collaborative sequence prediction problem, we define the dynamic neighborhood relationship between users and introduce manifold regularization to RNN on the basis of the multi-facets of collaborative filtering, referred to as MrRNN. Experimental results on benchmark datasets show that our approach outperforms the state-of-the-art baselines.

KEYWORDS

sequential recommender; collaborative sequence prediction; recurrent networks; manifold regularization

1 INTRODUCTION

In recommender systems, matching users with appropriate items is critical and difficult, as they deal with comparing two totally different objects: users and items [6]. With the surge of sequence modeling promoted by deep learning, sequential recommender is attracting more and more attention. As we know, sequential recommenders are introduced to learn what an arbitrary user buys next when he has bought certain items in the recent past [9]. Sequential recommenders attempt to model the sequence of user's actions, which are folded into a set of actions in general recommenders.

Sequential recommendation problem is usually casted as sequence prediction problem. Existing approaches focus on Markov Chain methods. These methods obtain local features from such sequential data in a fixed window. For instance, Markov Chain [2] and Factorized Personalized MC [8, 9] transform the sequential data into a transition matrix, which only capture the relationship between two adjacent items. The beauty of RNN lies in modeling sequences with variable length, which helps to capture the global sequential feature. Therefore, many RNN based sequential recommenders [1, 2, 5, 13] have been proposed. However, both Markov Chain based and RNN based models assume that users' consuming or rating sequences are independent, which does not hold actually [3]. Thus it is challenging to model the correlation of sequences to improve the sequence prediction problem. We borrow the basic idea in collaborative filtering that two users who have rated a set of items in a similar way in the past are supposed to rate new items likewise as well, and refer to this new problem as Collaborative Sequence Prediction.

RNN based model is a kind of a latent factor model in the following sense. The output of the recurrent layer is supposed to be user representations, while the weight matrix between the recurrent layer and the output layer is obtained as the item representations. As we know, latent factor models are effective at estimating overall structure that relates simultaneously to most or all items, while neighborhood based collaborative filtering models are effective at detecting strong local associations among a small set of closely related items. None of them is optimal on its own [6].

In light of this multi-facets of collaborative filtering, we propose to introduce the neighborhood information into the latent factor model RNN. Specifically, we define the neighborhood relationship between dynamic user representations, derive a manifold based on this relationship, then introduce this manifold regularization into RNN and make the manifold assumption hold that, referred to as MrRNN. Due to the multi-layers of RNN, there are various ways to regularize the output of the layer [12]. In this sense, our approach is flexible and can be extended easily. We conduct comprehensive experiments on two benchmark datasets: MovieLens and Netflix. Experimental results show that our approach outperforms baselines. Our main contribution lies in the following three aspects:

- Taking the correlation between users' behavior sequences, we frame the sequential recommendation as the collaborative sequence prediction problem.
- Instead of the static neighborhood, we introduce a dynamic neighborhood relationship between users at different timestamps, which is represented as manifold.
- For the collaborative sequence prediction problem, we propose to introduce manifold regularization to RNN.

2 RELATED WORK

According to how to capturing sequential features, existing sequential recommenders mainly fall into two categories: Markov Chains and Recurrent Neural Networks.

Markov Chain. Collaborative filtering was first reduced to a univariate time series problem in [14] and then Markov models was employed for webpage recommendation. Higher order of Markov Chain was adopted in [7] to mining web navigation pattern. Most Markov Chain models transform the sequential data into a transition graph. To handle sparsity and generalize to unobserved data,

CIKM'17 , November 6–10, 2017, Singapore, Singapore

ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3133079

factorized Personalized Markov Chain [9] was proposed to factorize the transition graphs and it constructed a transition cube and got the factors through tensor decomposition instead. To enhance the MLE of the transition graphs, a MDP based recommender [10] utilized several heuristic approaches such as clustering and skipping in a Markov Decision Processes. Hierarchical Representation Model combined the last action information with general user interest to model user representations for next basket recommendation [11]. All the MC based methods have the same deficiency that these recommenders only obtain the local sequential behaviors between every two adjacent items.

Recurrent Neural Network. Recently, RNN approaches have achieved much success in sequence modeling. The early attempt was to use vanilla RNN to predict next tracks users wanted to play in the blog [1] based on sequential prediction. Later, a comprehensive study [2] views collaborative filtering problem equal to sequential recommendation and was conducted to show how to utilize various RNNs in sequential recommender. GRU based RNN was used for session-based recommendation with two types of ranking loss functions: BPR [8] and TOP1 [4], which was first devised. DREAM [13] was proposed to utilize pooling first to summarize the basket and then fed into vanilla RNN to solve a special case of sequential recommendation, namely next basket recommendation. C-RNN [5] assumed network architecture should be personalized. Thus RNN was trained for each user and only weight parameters of input and output layers were shared among users. Although also motivated by collaborative recommendation, C-RNN has serious data sparsity problem. Instead, our approach assumes that all the network parameters are shared among users. The collaborative information lies in users' neighborhood relationship to regularize the network parameter learning.

3 COLLABORATIVE SEQUENCE PREDICTION

We first formalize traditional sequence prediction problems, then the collaborative sequence prediction is proposed.

3.1 Formalization

Let $U = \{u_1, \ldots, u_{|U|}\}$ be a set of users and $I = \{i_1, \ldots, i_{|I|}\}$ be a set of items. For each user u, a purchase or rating history is denoted by a sequence $S^u_{<t_u-1} = (s^u_1, \ldots, s^u_{t_u-1})$ with $s^u_t \subseteq I$. For next basket recommendation, there are usually more than one items in s^u_t for each timestamp, i.e. $|s^u_t| > 1$; for sequential recommendation in MovieLens and Netflix, there is usually one item per timestamp such as $|s^u_t| = 1$. The history for all users is $S = \{S^{u_i}\}_{i=1}^{|U|}$.

For sequential recommendation with history S, we formalize the problem as predicting a ranking list of items for each user u at time t_u. Existing approaches solve this sequence prediction problem by optimizing the following objective function.

$$\mathcal{L} = \sum_u \sum_{t_u} L_e(f(S^u_{t_u-1}), s^u_{t_u}) \quad (1)$$

where $f(\cdot)$ is the output of Recurrent Neural Network, and $L_e(\cdot)$ is a loss function between the network prediction output and the target, such as likelihood , cross entropy and margin.

3.2 Incorporating the Collaborative information

The latent factor model encodes users and items in the same latent space, and build the overall structure between users and all items. Neighborhood model emphasizes strong relationship on the local structure between users and a small subset of items. In light of this multi-facets of collaborative filtering, we introduce the neighborhood relationship as the collaborative information to the sequence prediction problem.

Given behavior sequences of all users S, the collaborative sequence prediction problem is to optimize the following problem.

$$\mathcal{L} = \sum_u \sum_{t_u} L(\{S^u_{<t_u-1}, \mathcal{N}^u_{t_u-1}\}, s^u_{t_u}) \quad (2)$$

where $\mathcal{N}^u_{t_u-1}$ denote the neighborhood of user u at time t_u. Flexibly, neighbors can play a role in the prediction of the recurrent neural network or not. Thus the loss function can be rewritten as $L(f(S^u_{<t_u-1}, \mathcal{N}^u_{t_u-1}), s^u_{t_u})$ or $L(f(S^u_{<t_u-1}), \mathcal{N}^u_{t_u-1}, s^u_{t_u})$ respectively. Here we only consider the latter form.

4 MANIFOLD REGULARIZED RECURRENT NEURAL NETWORK

Recurrent neural networks model the global feature of sequential data. Now the question is how to incorporate the local neighborhood information to solve the collaborative sequence prediction problem. Assume users with similar representations at this time are more likely to consume or rate the same item next time. First, we treat RNN as latent factor model. Then, we define dynamic neighborhood and encode this user neighborhood information in a manifold. Finally, we impose this manifold based locality preserving constraints on the network outputs.

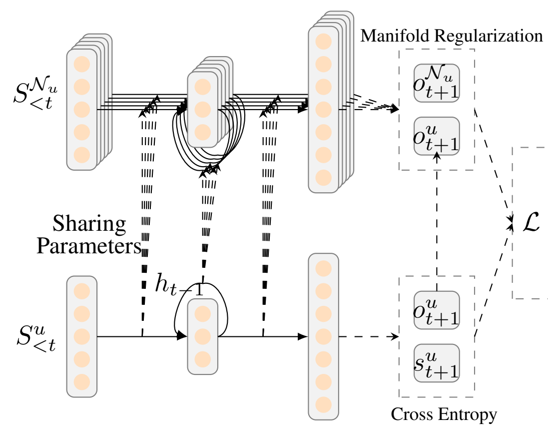


Figure 1: Architecture of MrRNN

4.1 RNN as Latent Factor Model

In conventional sequential recommenders, an input instance of RNN is a sequence of items. Suppose each item is represented as a one-hot vector $x \in \mathbb{R}^{|I|}$. As shown in the bottom network of Fig. 1, the output of the hidden layer at time t is dynamic representation of user u at time t denoted as $h^u_t \in \mathbb{R}^d$. We use LSTM block to show how to derive h^u_t from the input given item representation

x_t^u through the input gate Eq.(3), forget gate Eq.(4), cell Eq.(5) and output gate Eq.(6).

$$i_t^u = \sigma(W_i \cdot [h_{t-1}^u, x_t^u] + b_i) \tag{3}$$

$$f_t^u = \sigma(W_f \cdot [h_{t-1}^u, x_t^u] + b_f) \tag{4}$$

$$C_t^u = f_t^u * C_{t-1}^u + i_t^u * tanh(W_C \cdot [h_{t-1}, x_t^u] + b_C) \tag{5}$$

$$h_t^u = \sigma(W_o \cdot [h_{t-1}^u, x_t^u] + b_o) * tanh(C_t^u) \tag{6}$$

Finally, RNN outputs the probability over user u buying or rating all items at time $t+1$ by $f(S_{<t}^u) = \text{softmax}(W_m' \cdot h_t^u)$, where the item matrix W_m and a user's dynamic representation h_t^u share a latent space. Therefore RNN is kind of latent factor model.

4.2 Dynamic Neighborhood Construction

In sequential recommendation, user u's representation at timestamp t is derived from subsequence $S_{<t}^u$. Thus neighbors of a certain user u are varied over time. As illustrated in Fig. 2, neighborhood of u_3 at timestamp t_3 denoted as $\mathcal{N}_{t_3}^{u_3}$ is $\{S_{<t_2}^{u_1}, S_{<t_1}^{u_5}\}$ and $\mathcal{N}_{t_6}^{u_3} = \{S_{<t_4}^{u_4}, S_{<t_5}^{u_2}\}$. In this sense, the neighborhood is dynamic.

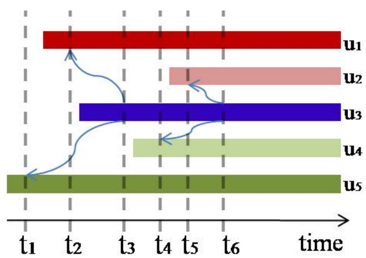


Figure 2: Illustration of Dynamic Neighborhood

Note that we can only utilize the history of the other users before time t to predict which items one user u will buy or rate at time $t+1$. So u's neighbors are constrained to users before time t. According to the assumption mentioned above, user representations from subsequences $S_{<t_i}^{u_i}$ and $S_{<t}^u$ are supposed to be similar if the next item of $S_{<t_i}^{u_i}$ ($s_{t_i+1}^{u_i}$) is the next item of $S_{<t}^u$ (s_{t+1}^u). Formally we define the neighbors of user u at timestamp t denoted as $\mathcal{N}_t^u = \{S_{<t_i}^{u_i} | t_i \leq t,\ \ s_{t_i+1}^{u_i} = s_{t+1}^u\}$. We construct the dynamic neighborhood by randomly choosing k-Nearest Neighbor in $\mathcal{N}_t^u$ with time complexity $O(|I| + \log l)$ per query, where l is the averaged length of user sequences. We compare other sequence similarity measures, like dynamics time warping. They have a similar effect with our work, but they cost more time, i.e. $O(|U|l^2)$, and space.

4.3 Manifold Regularized Loss Function

Specifically, we introduce the manifold regularization term in Eq.(7) based on the dynamic neighborhood $\mathcal{N}_t^u$. $\omega_{u,i}^{t_u}$ denotes the ground truth similarity between two users's subsequences. In this paper, we set it to 1 if the next item in u_i's sequence is the same as that in u's sequence; 0 otherwise. $f(\cdot)$ is RNN function, where parameters are shared among all users. So the overall loss function can be written in Eq.(8), referred to as MrRNN. In our experiments, cross entropy is used to measure the prediction error L_e.

$$L_{\mathcal{G}}(f(S_{<t_u-1}^u), \{f(S_{<t_i}^{u_i}) | S_{<t_i}^{u_i} \in \mathcal{N}_{t_u-1}^u\}) = \sum_{S_{<t_i}^{u_i} \in \mathcal{N}_{t_u-1}^u} \omega_{u,i}^{t_u} \|f(S_{<t_i}^{u_i}) - f(S_{<t_u-1}^u)\|^2 \tag{7}$$

$$L(\{S_{<t_u-1}^u, \mathcal{N}_{t_u-1}^u\}, s_{t_u}^u) = L_e(S_{<t_u-1}^u, s_{t_u}^u) + \lambda L_{\mathcal{G}} \tag{8}$$

MrRNN takes one item sequence along with its neighbor sequences as an input instance. Compared with RNN, it needs additional query time and k times gradient computational complexity for each training sequence. The specific implementation is illustrated in Fig. 1. Different from manifold regularized single-layer neural networks, manifold error will have effect on the multi-layered architecture through back propagation. Finally, we optimize this regularized loss function through Adagrad.

5 EXPERIMENTAL RESULTS

We first utilize the experimental setting, then compare MrRNN with baselines on two benchmark datasets, and finally explore parameters' effect in our proposed method.

5.1 Experimental Setting

Datasets. We conduct comprehensive experiments on two public benchmark datasets: MovieLens 1M and Netflix 1K. MovieLens 1M contains 1, 000, 209 ratings from 6, 040 users rated 3900 movies. Netflix is a much larger dataset, with 100 million ratings rated by 480,000 customers over 17, 000 movies. For time limitation, we randomly sample 1, 000 users in our experiment denoted as Netflix 1K. Each dataset is split into train, validation, and test set by a proportion 8:1:1 according to the number of users.

Baselines We use three kinds of baselines: (1) General recommenders: POP, UKNN and BPRMF [8]. (2) Sequential Recommenders based on Markov Chain: MC [2] and FPMC [9]. (3) Sequential recommenders based on RNN: CRNN [5] and RNN [2].We use one LSTM layer in RNN based methods. Adagrad is adopted for optimization with learning rate 0.001 in RNN and 0.1 in MrRNN. The number of latent features is 20 in BPRMF, 32 in FPMC. The neighborhood size is 80 in UKNN and 7 in MrRNN with $\lambda = 0.5$. All the performances are on the test set.

Evaluation. Top-k recommendation is used to give user k potential items with ranking. We set $k = 10$ and use *sps@k*, *ndcg@k*, *rec@k* (short for recall@k) and ucov@k (short for user_coverage@k) for evaluation. *sps@k* represents successful prediction in short period. Both *rec@k* and *ndcg@k* all focus on recommendation capacity in long period prediction. Finally We use *ucov@k* to testify the generalization of recommender.

5.2 Performance Analysis

General v.s. Sequential. For short term prediction, sequential models outperforms the general recommenders shown in Tab. 1. This is more obvious on MovieLens 1M, where the highest sps@10 achieved by MrRNN is 91.24% higher than the best general method UKNN.

MC based v.s. RNN based. For short term prediction, MC based methods performs as well as RNN based methods in most cases on MovieLens 1M, but underperform on Netflix 1K. In Tab. 1, sps@10 of MrRNN is 3 times higher than the best Markov Chain methods on Netflix 1K. For long term prediction, RNN methods has overwhelming superiority to Markov Chain methods on both datasets. For instance, rec@10 on MovieLens 1M of MrRNN is 36.84% higher than the best MC based methods, ndcg@10 on Netflix 1K of MrRNN

Table 1: Performance Comparison (%) on Movielens 1M and Netflix 1K. All the performance differences between MrRNN and baselines are statistically significant with p-value< 0.01 for paired t-test.

(a) Movielens 1M

Methods	rec@10	sps@10	ucov@10
POP	3.73	5.10	70.95
UKNN	5.94	13.02	80.28
BPRMF	1.50	2.64	47.18
MC	4.75	24.47	78.16
FPMC	2.09	5.63	50.88
RNN	6.23	23.42	87.66
CRNN	3.88	15.82	64.60
MrRNN	6.50	24.90	91.30

(b) Netflix 1K

Methods	rec@10	sps@10	ndcg@10	ucov@10
POP	2.02	1.06	46.06	87.23
UKNN	3.36	2.12	53.21	95.74
BPRMF	0.48	0.00	10.46	54.32
MC	1.20	2.12	23.09	87.23
FPMC	0.11	0.00	3.55	25.53
RNN	1.92	5.31	34.37	93.61
CRNN	2.21	8.49	2.95	17.58
MrRNN	2.36	8.51	41.38	94.68

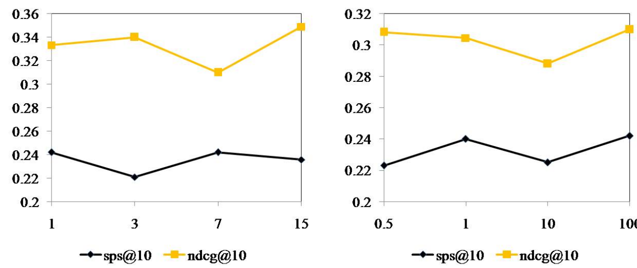
is 79.21% better than the best MC based methods. For generalization performance, RNN based methods achieve better ucov@10 on both datasets. Therefore RNN based methods are the better way to model the global sequential feature than MC methods.

Non-Collaborative v.s. Collaborative. For long and short term prediction, MrRNN outperforms non-collaborative RNN method on both datasets. For generalization performance in terms of ucov@10, MrRNN achieves 4.15% better than RNN on Movielens 1M. In addition, MrRNN has better collaborative performance than CRNN. Collaborative information plays an important role on RNN based sequential recommender. MrRNN combines RNN's modeling global sequential feature with the manifold regularization's strong focus on local relationship.

5.3 Influence of Regularization Parameter

We explore whether manifold regularization parameters will have effect on MrRNN's performance: the number of neighbors in the kNN graph (neighborhood size) and the tradeoff parameter between the prediction loss and manifold error denoted as λ.

We observe the performance variation on MovieLens 1M along with the increase of neighborhood size when $\lambda = 100$ shown in Fig. 3(a). ndcg@10 increases by 4.73% along with neighborhood size from 1 to 15. The performance variation on MovieLens 1M with the increase of λ is shown in Fig. 3(b) when the neighborhood size is fixed to 7. The short term prediction performance (sps@10) is increased by 8.57% when λ varies from 0.5 to 100. Though these parameters are not so sensitive to variation, appropriate ranges are needed to be tuned for them.



(a) neighborhood size **(b) λ**

Figure 3: Performance variation along with Manifold Regularizer Parameters

6 CONCLUSION

In this paper, we introduce a collaborative sequence prediction problem to take the sequence dependence into account in sequential prediction. In sequential recommendation, we propose the dynamic neighborhood of users to emphasize on the strong local relationship. Then we encode these information in manifold and incorporate manifold regularization into Recurrent Neural Networks which capture the global sequential features. We optimize it through multi-layered architectures namely MrRNN. Experimental results show that our proposed approach MrRNN outperforms the state-of-the-art baselines on benchmark datasets.

ACKNOWLEDGEMENT

The research was supported by National Natural Science Foundation of China under Grant No.61602451 and No.61603372.

REFERENCES

[1] Erik Bernhardsson. 2014. *Recurrent Neural Networks for Collaborative Filtering.* https://erikbern.com/2014/06/28/recurrent-neural-networks-for-collaborative-filtering.html.
[2] Robin Devooght and Hugues Bersini. 2016. Collaborative Filtering with Recurrent Neural Networks. *CoRR* abs/1608.07400 (2016).
[3] Kostadin Georgiev and Preslav Nakov. 2013. A non-IID Framework for Collaborative Filtering with Restricted Boltzmann Machines. In *Proceedings of ICML 2013*, Vol. 28. 1148–1156.
[4] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based Recommendations with Recurrent Neural Networks. *CoRR* abs/1511.06939 (2015).
[5] Young Jun Ko, Lucas Maystre, and Matthias Grossglauser. 2016. Collaborative Recurrent Neural Networks for Dynamic Recommender Systems. In *Journal of Machine Learning Research: Workshop and Conference Proceedings*, Vol. 63.
[6] Yehuda Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *Proceedings of KDD 2008*. 426–434.
[7] Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa. 2002. Using Sequential and Non-Sequential Patterns in Predictive Web Usage Mining Tasks. In *Proceedings of ICDM 2002*. 669–.
[8] Steffen Rendle and Christoph Freudenthaler. 2014. Improving Pairwise Learning for Item Recommendation from Implicit Feedback. In *Proceedings of WSDM 2014*.
[9] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing Personalized Markov Chains for Next-basket Recommendation. In *Proceedings of WWW2010*. 811–820.
[10] Guy Shani, David Heckerman, and Ronen I. Brafman. 2005. An MDP-Based Recommender System. *J. Mach. Learn. Res.* 6 (Dec. 2005), 1265–1295.
[11] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning Hierarchical Representation Model for NextBasket Recommendation. In *Proceedings of SIGIR 2015*. 403–412.
[12] Jason Weston, Frédéric Ratle, and Ronan Collobert. 2008. Deep Learning via Semi-supervised Embedding. In *Proceedings of ICML 2008*. 1168–1175.
[13] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A Dynamic Recurrent Model for Next Basket Recommendation. In *Proceedings of SIGIR 2016*. 729–732.
[14] Andrew Zimdars, David Maxwell Chickering, and Christopher Meek. 2001. Using Temporal Data for Making Recommendations. In *Proceedings of UAI 2001*.