This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2021.3084513, IEEE Signal Processing Letters

Next-item Recommendation with Deep Adaptable Co-Embedding Neural Networks

Daochang Chen, Wenzheng Hu, Bo Yuan, *Senior Member, IEEE*, Rui Zhang, *Senior Member, IEEE*, and Jianqiang Wang

***Abstract*—The next-item recommendation has been in the central of interest in real-world applications such as e-commerce. However, it is challenging to infer what a user may purchase next due the complex interactions in the historical sessions and the changing semantics of an item over time. Most existing methods employ separate models to generate the general preference and the sequential patterns for the next-item recommendation without considering the interactions between the two factors or use a simple linear combination of the two factors. In this paper, we propose a deep adaptable co-embedding neural network (ACENet) to address these limitations. ACENet not only adaptably balances the combination of general preference and sequential patterns but also introduces dynamic attention for each factor in hybrid representations. Extensive experiments on two real-world datasets show the superiority of ACENet compared with other state-of-the-art methods.**

***Index Terms*—Sequential Behavior, Evolving Preferences, Co-Embedding, Dynamic Integration**

I. Introduction

NEXT-ITEM recommendation systems [1] have become an important part of many real-world applications and the core business of leading companies, such as e-commerce in Alibaba, online news in ByteDance, and social media in Facebook. They aim to accurately predict a user's next actions based on the historical sequential interactions and connect users with interesting items. For example, an e-commerce next-item recommendation system is to recommend the potentially most favorable products to the user in the next session based on a series of historical sessions, where each session contains a set of products purchased over some time. Generally, a user's activities in the next session are influenced by both the general preference (long-term preference, e.g., the user prefers Apple's products to Microsoft's products) and sequential patterns (short-term preference, e.g., the products purchased recently). Meanwhile, there are mutual interactions between the two factors as well as among the different attributes of a factor. For example, after the user who generally prefers Apple's products to Microsoft's products purchased a MacBook Pro, he/she may prefer Lumias to iPhones due to the total cost, and the importance of the price attribute increases compared to brand.

To recommend the next item, many methods have been proposed. One way is based on the general preference, which assumes that the profiles of a user's preference and items are relatively stable. Another way is to learn static representations of users from their sequential interactions [2]–[4]. Some other works [5], [6] extend to generate user representations using individual linear combinations of these two factors. However, few methods except [7], [8] attempt to dynamically combine the user's long-term and short-term preferences, which is important in real-world applications [7]. LSAMN [7] attempts to introduce an attention mechanism to adapt the general preference and sequential patterns, and HyperRec [8] adopts the hypergraph to infer the dynamic user preferences.

In this paper, we propose a novel adaptable co-embedding neural network (ACENet) to achieve the desirable dynamic combination. Different from LSAMN and HyperRec, ACENet not only adaptably balances the combination of general preference and sequential patterns but also introduces dynamic attention for each factor in hybrid representations. Furthermore, the item attention mechanism in LSAMN requires fixed-length items, which may inevitably result in information loss for sessions of different lengths and is not very suitable for sessions with very few items. By contrast, ACENet implements pooling operations and employs a co-embedding module. It captures the user's dynamic and evolving interests through the high-level affinity of short-term patterns and long-term patterns at each time step. In summary, we make the following key contributions:

- We introduce a novel co-embedding mechanism and an affinity operation to model a user's dynamic and evolving interests for the next-item recommendation.
- Through an adaptable co-embedding deep network, ACENet strategically combines a user's general preference and sequential patterns to generate a high-level hybrid representation of a user's preference.
- Compared with state-of-the-art approaches, ACENet demonstrates consistent superiority based on commonly used evaluation metrics.

II. Related Work

Our work focuses on generating dynamic and evolving user preferences from sequential user interactions. Most previous methods such as Collaborative Filtering (CF) [9]–[11] take

This work was supported in part by the China Postdoctoral Science Foundation (No. 2020M680563). The authors thank the support of Tsinghua-DiDi Joint Research Center for Future Mobility.

Daochang Chen, Department of Automation, Tsinghua University, Beijing, China (e-mail: cdc16@mail.tsinghua.edu.cn).

Bo Yuan, Shenzhen International Graduate School, Tsinghua University, Shenzhen, China (e-mail: boyuan@ieee.org).

Wenzheng Hu and Jianqiang Wang are with The State Key Laboratory of Automotive Satefy and Energy, Tsinghua University, Beijing, China (e-mail: hwz@mail.tsinghua.edu.cn, wjqlws@tsinghua.edu.cn).

Rui Zhang, The University of Melbourne, Melbourne, Australia (e-mail: rui.zhang@unimelb.edu.au).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2021.3084513, IEEE Signal Processing Letters

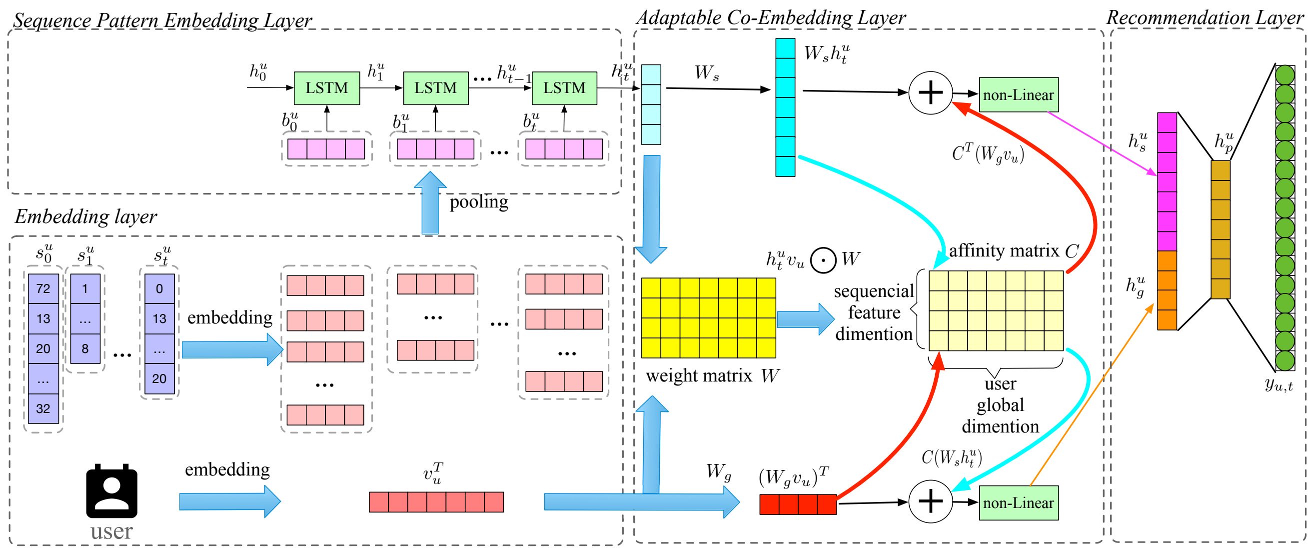

Fig. 1: The ACENet model

advantage of implicit feedbacks [12]–[14] including check-in history, search patterns, browsing history, etc. CF based methods usually assume that a user's preference is static. However, in real-world scenarios, there may be duplicate items in a session, and the popularity of items may frequently drift over time [15]–[17]. Similarly, a user's preference for items can be evolving and constantly changing [18]–[20].

Most previous sequential recommendation systems focus on sequential pattern mining to predict the next item. For example, Markov Chain based methods make recommendations based on the L-previous actions [6], [21]–[24]. Recurrent Neural Network (RNN) [4], [25]–[30] based methods learn user the representations of long-term user interests from sequential sessions. However, items in a session may not strictly follow a sequential order and a session may contain duplicate items in various application scenarios [31]–[33]. Consequently, we propose to use a co-embedding attention network to make more effective recommendations.

III. Methodology

A. Problem Formulation

In this paper, we focus on the sequential next-item recommendation based on implicit feedbacks (e.g., check-in, purchases, clicks). Let $U = \{u_1, u_2, \cdots, u_m\}$ and $I = \{i_1, i_2, \cdots, i_n\}$ represent the user set and the item set, respectively. Without loss of generality, let $i_j = j, j \in \{1, 2, \cdots, n\}$. The user u's historical record consists of a sequence of sessions denoted by $S_t^u = \{s_0^u, s_1^u, \cdots, s_t^u\}$. Each session contains a subset of items, namely $\forall k, s_k^u \subset I$. We can simply regard items as the purchased good IDs for convenience. Note that different sessions may have different numbers of items. The next-item recommendation is to recommend the top-K items for user u in the next session:

$$TopK(\{y_{t+1,c}^u = Preference(; S_t^u, I, u) | c \in I\}).$$

B. The Proposed ACENet

ACENet models the hybrid representation of a user's preference using the current context, including general preference and all historical sequential patterns at the current time step. Figure 1 illustrates the framework of ACENet.

*1) **The Embedding Layer**:* In a recommender system, users and items are usually represented by sparse one-hot vectors. Given a user u and his/her historical record S_t^u, embedding layers $\Phi_u(\cdot)$ and $\Phi_i(\cdot)$ map users and items into two continuous spaces, respectively:

$$\boldsymbol{v}_u = \Phi_u(u) \in R^l, u \in U, \quad (1)$$

$$\boldsymbol{x}_c = \Phi_i(c) \in R^d, c \in I. \quad (2)$$

Then, $\Phi_i(s_k^u) = [\boldsymbol{x}_c \ for \ c \in s_k^u]^T$ is a matrix of size $|s_k^u| \times d$, and $\Phi_i(S_t^u)$ is a session matrix.

*2) **The Sequence Pattern Embedding Layer**:* Since the number of items may be different across sessions, the historical record mentioned before is not suitable for certain sequential methods, for example, RNN-based models. Therefore, we introduce a *pooling* operation to embed the session matrix $\Phi_i(s_t^u)$ into a d-dimension vector:

$$\boldsymbol{b_t^u} = \mathcal{P}([\boldsymbol{x}_{s_t^u(1)}, \boldsymbol{x}_{s_t^u(2)}, \cdots, \boldsymbol{x}_{s_t^u(|s_t^u|)}]) \quad (3)$$

where $s_t^u(i)$ is the i-th element in session s_t^u, and $\mathcal{P}(\cdot)$ is one of the pooling operations (e.g., max pooling, average pooling). Such an embedding contributes to the extraction of joint high level representations for sessions with different lengths.

To capture sequential patterns and to learn dynamic user interest representations, the sequence pattern embedding layer leverages a long short-term memory network (LSTM) [34] to update sequence patterns at each time step. The hidden state $\boldsymbol{h_t^u}$ at time step t is calculated based on the input session's latent factor $\boldsymbol{b_t^u}$ and the last hidden state $\boldsymbol{h_{t-1}^u}$:

$$\boldsymbol{h_t^u} = LSTM(\boldsymbol{h_{t-1}^u}, \boldsymbol{b_t^u}) \quad (4)$$

where $\boldsymbol{h_t^u} \in R^g$ denotes the hidden state of the user u at time step t, and $\boldsymbol{h_{t-1}^u} \in R^g$ denotes the previous state.

*3) **The Adaptable Co-Embedding Layer**:* Users' preferences and behaviors are likely to change dynamically. To capture a user's dynamic changes, we introduce an adaptable co-embedding layer to learn the collective impact between the user's general preference and sequential patterns. Unlike existing methods, the co-embedding layer dynamically updates the user's general preference and sequential patterns with a covariance affinity matrix and obtains a hybrid high-level representation of the user's preference.

The core of the adaptable co-embedding layer is the covariance affinity matrix C, which is used as a feature connection between the user's general preference and the sequential patterns. The matrix C consists of two parts, the learnable parameter matrix $\boldsymbol{W} \in R^{g\times l}$ and the covariance matrix at the current time step t:

$$\boldsymbol{C} = \boldsymbol{h_t^u} \boldsymbol{v}_u^T \bigodot \boldsymbol{W} \quad (5)$$

where $\boldsymbol{C}$ is a matrix of size $g \times l$.

To update the user's general perference and the sequential patterns, we define the new general perference representation and the new sequential patterns as follows:

$$\boldsymbol{h_g^u} = tanh(\boldsymbol{W_g v_u} + \boldsymbol{C}(\boldsymbol{W_s h_t^u})) \quad (6)$$

$$\boldsymbol{h_s^u} = tanh(\boldsymbol{W_s h_t^u} + \boldsymbol{C^T}(\boldsymbol{W_g v_u})), \quad (7)$$

where $\boldsymbol{W_g} \in R^{g\times l}$ and $\boldsymbol{W_s} \in R^{l\times g}$ are weight parameters, while $\boldsymbol{h_g^u} \in R^g$ and $\boldsymbol{h_s^u} \in R^l$ represent user u's new high-level general preference and high-level squential pattern, respectively.

*4) **The Recommendation Layer**:* We concatenate the outputs of the co-embedding layer and feed them into a fully-connected neural network to predict the preferences of each item in the next session:

$$\boldsymbol{h_p^u} = tanh(\boldsymbol{W_p} \begin{bmatrix} \boldsymbol{h_g^u} \\ \boldsymbol{h_s^u} \end{bmatrix} + \boldsymbol{\beta_p}) \quad (8)$$

$$\boldsymbol{y_{t+1}^u} = \boldsymbol{W_f^T h_p^u} + \boldsymbol{\beta_f} \quad (9)$$

where $\boldsymbol{W_p} \in R^{d_0\times(g+l)}$ is a weight matrix that projects the concatenation layer to a d_0-dimention hidden layer and reduces the redundancy; $\boldsymbol{\beta_p} \in R^{d_0}$ is a bias vector; $\boldsymbol{W_f} \in R^{d_0\times|I|}$ is the prediction matrix, and $\boldsymbol{\beta_f} \in R^{|I|}$ is the bias for the final prediction of preferences.

C. Training Loss

The network is trained under the same framework as the Factorized Personalized Markov Chain (FPMC) model [22], which uses the Bayesian Personalized Ranking (BPR) approach [35] to optimize the personalized ranking. The ACEN model is estimated using the maximum a posterior (MAP) estimation:

$$\begin{aligned} &\arg\max_{\Theta} \\ &= \ln \prod_{u\in U} \prod_{s_t^u \in S_t^u} \prod_{i\in s_t^u} \prod_{j\in I\setminus s_t^u} p(\boldsymbol{y_{t,i}^u} > \boldsymbol{y_{t,j}^u}|\Theta) p(\Theta) \\ &= \sum_{u\in U} \sum_{s_t^u \in S_t^u} \sum_{i\in s_t^u} \sum_{j\in I\setminus s_t^u} \ln p(\boldsymbol{y_{t,i}^u} > \boldsymbol{y_{t,j}^u}|\Theta) + \ln p(\Theta) \end{aligned} \quad (10)$$

where $p(\cdot)$ is the preference function, and Θ denotes all learnable parameters in the model. Given user u's sequential feedback history S_{t-1}^u, the probability of preference is defined as:

$$p(\boldsymbol{y_{t,i}^u} > \boldsymbol{y_{t,j}^u}|\Theta) = \frac{1}{1 + e^{-(\boldsymbol{y_{t,i}^u} - \boldsymbol{y_{t,j}^u})}} \quad (11)$$

However, Eq.(10) is difficult for training in practice, as the user and session data are often large and sparse. To tackle this issue, we uniformly sample a user's pair-wise item (i, j) from the dataset, and follow the negative-item selection method used in FPMC. The final loss function is:

$$\begin{aligned} &\arg\min_{\Theta} \sum_{D} - \ln p(\boldsymbol{y_{t,i}^u} > \boldsymbol{y_{t,j}^u}|\Theta) + \lambda||\Theta|| \\ &(u, S_{t-1}^u, i \in s_t^u, j \in s_t^u, i \neq j) \end{aligned} \quad (12)$$

where set D consists of a large amount of elements generated from the training set, and λ is a regularization coefficient. The model is subject to end-to-end training with the SGD optimizer [36] with the loss function in Eq.(12).

IV. Experiments

A. Datasets and Preprocessing

We conducted experiments on two real-world datasets, *Foursquare* [37], [38] and *Gowalla* [39]. These two datasets contain user's check-in sessions, and each session contains the items that a user checked in within a day. Following the previous study [40], [41], we preprocessed the datasets by removing long-tail users and items: removed all items bought by less than 20 users and users who bought less than 20 items; removed users with less than two sessions. The basic statistics of datasets are summarized in Table I.

TABLE I: Statistics of the Datasets

Dataset	#user $\|U\|$	#item $\|I\|$	#sequence	#avg.session length	#train session	#test session
Gowalla	17.9k	14.1k	164.5k	5.32	134.7k	29.8k
Foursquare	2.1k	1.3k	62.2k	2.05	54.3k	7.8k

B. Experimental Settings

*1) **Baseline Methods**:* We compared our method, ACENet[1], with the following baselines.

- **POP** recommends the most popular items in the training set to users.
- **BPR-MF[2009]** [35] is a matrix factorization method only focusing on users' long-term preferences.
- **FMC[2010]** [22] is a Markov chain method based on the last session items.
- **FPMC[2010]** [22] mines a user's long-term preferences and sequential behaviors for next-item recommendation based on first-order Markov chains.
- **HRM[2015]** [40] aggregates users' general preference and their last sessions using aggregation operations such as max-pooling and avg-pooling.
- **DREAM[2016]** [3] is an RNN-based approach taking advantage of the sequential feature of a user.
- **Caser[2][2018]** [5] incorporates the convolutional neural network to capture sequential patterns and linearly combine general preference for the user's hybrid represent.
- **SHAN[3][2018]** [41] employs attention networks to obtain the representation and uses matrix factorization to compute each item's score.
- **LSAMN[2019]** [7] jointly models a user's long-term and short-term preferences in two embedding spaces, which use two-level attention to combine user's preferences.

[1] https://github.com/jackenmm/ACENet
[2] https://github.com/graytowne/caser_pytorch
[3] https://github.com/uctoronto/SHAN

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2021.3084513, IEEE Signal Processing Letters

TABLE II: Performance Comparison on the Two Real-World Datasets

Models	Foursquare							Gowalla						
	Rec@5	Rec@10	Rec@20	Prec@5	Prec@10	Prec@20	MRR	Rec@5	Rec@10	Rec@20	Prec@5	Prec@10	Prec@20	MRR
POP	0.0007	0.0045	0.0078	0.0006	0.0017	0.0018	0.0046	0.00004	0.0001	0.0003	0.00006	0.00005	0.00007	0.0006
BPR-MF	0.1491	0.2067	0.2461	0.1150	0.0817	0.0934	0.1001	0.0304	0.0555	0.0854	0.0216	0.0189	0.0145	0.0245
FMC	0.1531	0.1937	0.2315	0.1252	0.0823	0.0505	0.0981	0.0319	0.0515	0.0804	0.0218	0.0175	0.0133	0.0229
FPMC	0.1551	0.2022	0.2397	0.1249	0.0815	0.0528	0.1063	0.0287	0.0528	0.0800	0.0197	0.0175	0.0132	0.0224
HRM	0.1536	0.2017	0.2385	0.1257	0.0813	0.0517	0.1038	0.0320	0.0559	0.0916	0.0220	0.0199	0.0157	0.0251
DREAM	0.1409	0.1964	0.2369	0.1148	0.0804	0.0493	0.0958	0.0309	0.0537	0.0809	0.0213	0.0176	0.0133	0.0229
Caser	0.1473	0.2108	0.2163	0.1032	0.0786	0.0479	0.1045	0.0297	0.0513	0.0791	0.0207	0.0182	0.0128	0.0213
SHAN	0.1531	0.2057	0.2506	0.1252	0.0818	0.0497	0.0903	0.0344	0.0607	0.0892	0.0221	0.0189	0.0155	0.0247
LSAMN	0.1457	0.2081	0.2521	0.1272	0.0818	0.0502	0.1047	0.0321	0.0596	0.0876	0.0227	0.0183	0.0147	0.0233
ACENet	**0.1631**	**0.2189**	**0.2675**	**0.1287**	**0.0827**	**0.0534**	**0.1181**	**0.0357**	**0.0646**	**0.0970**	**0.0237**	**0.0194**	**0.0163**	**0.0287**

TABLE III: The Impact of ACENet Components

Model	Pooling	Co-embeding	Foursquare			Gowalla		
			Recall@10	Recall@20	MRR	Recall@10	Recall@20	MRR
ACENet	avg	×	0.2007	0.2486	0.0980	0.0628	0.0880	0.0236
	avg	√	0.2146	0.2646	0.1123	0.0632	0.0917	0.0266
	max	×	0.2046	0.2493	0.0997	0.0608	0.0895	0.0247
	max	√	**0.2189**	**0.2675**	**0.1181**	**0.0646**	**0.0970**	**0.0287**

*2) **Evaluation Metrics**:* We employed three commonly used metrics [42]–[45] (Rec@K, Prec@K and MRR).

- **Rec@K:** This metric evaluates the recall of the top-K ranked items over the test sessions.
- **Prec@K:** This metric evaluates the precision of the top-K ranked items over the test sessions.
- **MRR:** This metric evaluates the mean reciprocal rank of the predictive position of the ground-truth items over the test sessions.

*3) **Implementation Details**:* In our experiments, we randomly selected 30% of sessions in the last month for testing and used all others as the training set. Following the convention of previous studies [1], [46]–[48], we selected the top K items in the test session for each user u, where $K = \{5, 10, 20\}$, and compared these recommended items with the ground truth to evaluate the recommendation performance. The embedding size was varied within the range of $\{5, 10, 20, 40, 60, 80, 100, 120\}$ with $g = l$. Other parameters were set as: the regularization parameter $\lambda \in \{0.0001, 0.001, 0.01, 0.1\}$; the learning rate $\eta \in \{0.0002, 0.002, 0.02, 0.2\}$; to accelerate training, each batch consisted of 5 negative samples for each positive label [46]. The hyper-parameters of baseline models are selected via the same search strategy and the same space, and all models run 15 times and are present with the average value.

C. Results and Analysis.

According to Table II. Under all evaluation metrics (Rec@K, Prec@K, and MRR), ACENet significantly outperformed all baseline algorithms on both real-world datasets, demonstrating its effectiveness in next-item recommendation. Compared to LSAMN, ACENet also achieved noticeable improvement, confirming the value of adaptabe co-embedding in balancing both the between-factor and within-factor relationships of the general preference and sequential patterns.

D. Ablation Study.

We also conducted experiments to evaluate the influence of *co-embedding* and the *pooling* operation. In Table III, "avg" means using the *average pooling* operation, and "max" means using the *max pooling* operation. In Table III, *max pooling* is slightly better than *average pooling*. We believe that this is due to the fact that some important items may be accidentally dropped when the entire item set was divided into sessions. By contrast, *max pooling* maintains important items and ignores some unimportant items. Note that, under the same pooling operation, the adaptable co-embedding layer always helped ACENet obtain better results, showing the efficacy of adaptable co-embedding layer.

E. Influence of Hyper-Parameter

We investigated the influence of embedding dimensions on our ACENet model. Figure 2 shows the performance of ACENet under Recall@20 and MRR for various dimensions without changing the values of other hyper-parameters. For all datasets, higher embedding dimensions did not necessarily result in better performance, and a reasonable dimension range was between 20 and 60.

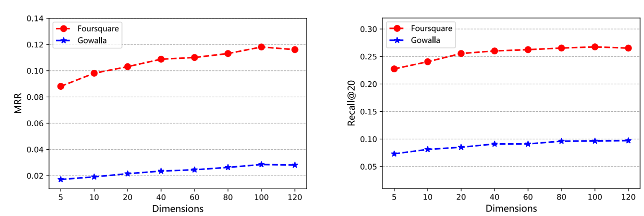

Fig. 2: The impact of embedding dimensions.

V. Conclusion

To effectively recommend the next item within a series of sessions, this paper proposed a novel Adaptable Co-Embedding Deep Neural Network (ACENet). ACENet is a hybrid co-embedding neural network, which can generate users' general tastes and sequential dynamics for next-item recommendation in the current context. Moreover, it shows that ACENet can effectively capture the user's high-level hybrid preferences and focus on the most relevant items over all observed sessions in the current time. Experiments show that ACENet can outperform several state-of-the-art models according to three popular metrics (Rec@K, Prec@K, MRR) on two real-world datasets.

References

[1] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," in *TOIS*, 2004, pp. 143–177.

[2] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *ICDM*, 2009, pp. 263–272.

[3] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A dynamic recurrent model for next basket recommendation," in *SIGIR*, 2016, pp. 729–732.

[4] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *ICLR*, 2016.

[5] J. Tang and K. Wang, "Personalized top-n sequential recommendation via convolutional sequence embedding," in *WSDM*, 2018, pp. 565–573.

[6] R. He and J. McAuley, "Fusing similarity models with markov chains for sparse sequential recommendation," in *ICDM*, 2016, pp. 191–200.

[7] D. Chen, R. Zhang, J. Qi, and B. Yuan, "Sequence-aware recommendation with long-term and short-term attention memory networks," in *MDM*. IEEE, 2019, pp. 437–442.

[8] J. Wang, K. Ding, L. Hong, H. Liu, and J. Caverlee, "Next-item recommendation with sequential hypergraphs," in *SIGIR*. New York, NY, USA: Association for Computing Machinery, 2020, p. 1101–1110.

[9] C. Chen, M. Zhang, Y. Zhang, W. Ma, Y. Liu, and S. Ma, "Efficient heterogeneous collaborative filtering without negative sampling for recommendation," in *AAAI*, vol. 34, no. 01, 2020, pp. 19–26.

[10] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *WWW*, 2017, pp. 173–182.

[11] Y. Koren, "Collaborative filtering with temporal dynamics," in *SIGKDD*, 2009, pp. 447–456.

[12] F. Zhang, N. J. Yuan, K. Zheng, D. Lian, X. Xie, and Y. Rui, "Exploiting dining preference for restaurant recommendation," in *WWW*, 2016, pp. 725–735.

[13] N. N. Liu, E. W. Xiang, M. Zhao, and Q. Yang, "Unifying explicit and implicit feedback for collaborative filtering," in *CIKM*, 2010, pp. 1445–1448.

[14] W. Yu and Z. Qin, "Sampler design for implicit feedback data by noisy-label robust learning," in *SIGIR*, 2020, pp. 861–870.

[15] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.

[16] L. Wu, L. Chen, R. Hong, Y. Fu, X. Xie, and M. Wang, "A hierarchical attention model for social contextual image recommendation," *TKDE*, 2019.

[17] T. Zhang, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, D. Wang, G. Liu, and X. Zhou, "Feature-level deeper self-attention network for sequential recommendation." in *IJCAI*, 2019, pp. 4320–4326.

[18] X. Wang, L. Yu, K. Ren, G. Tao, W. Zhang, Y. Yu, and J. Wang, "Dynamic attention deep model for article recommendation by learning human editors' demonstration," in *KDD*, 2017, pp. 2051–2059.

[19] M. Wang, M. Gong, X. Zheng, and K. Zhang, "Modeling dynamic missingness of implicit feedback for recommendation," in *NIPS*, 2018, pp. 6669–6678.

[20] Z. Yu, J. Lian, A. Mahmoody, G. Liu, and X. Xie, "Adaptive user modeling with long and short-term preferences for personalized recommendation." in *IJCAI*, 2019, pp. 4213–4219.

[21] Y. Zhang, Z. Shi, W. Zuo, L. Yue, S. Liang, and X. Li, "Joint personalized markov chains with social network embedding for cold-start recommendation," *Neurocomputing*, vol. 386, pp. 208–220, 2020.

[22] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *WWW*, 2010, pp. 811–820.

[23] Q. Yang, J. Fan, J. Wang, and L. Zhou, "Personalizing web page recommendation via collaborative filtering and topic-aware markov model," in *ICDM*. IEEE, 2010, pp. 1145–1150.

[24] F. Peng, J. Lu, Y. Wang, R. Yi-Da Xu, C. Ma, and J. Yang, "N-dimensional markov random field prior for cold-start recommendation," *Neurocomputing*, vol. 191, pp. 187–199, 2016.

[25] X. Chen, H. Xu, Y. Zhang, J. Tang, Y. Cao, Z. Qin, and H. Zha, "Sequential recommendation with user memory networks," in *WWW*, 2018, pp. 108–116.

[26] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, "Stamp: short-term attention/memory priority model for session-based recommendation," in *SIGKDD*, 2018, pp. 1831–1839.

[27] C. Zhou, J. Bai, J. Song, X. Liu, Z. Zhao, X. Chen, and J. Gao, "Atrank: An attention-based user behavior modeling framework for recommendation," in *AAAI*, vol. 32, no. 1, 2018.

[28] S. Zhang, L. Yao, and A. Sun, "Deep learning based recommender system: A survey and new perspectives," *arXiv preprint arXiv:1707.07435*, 2017.

[29] J. Gao, Y. Lin, Y. Wang, X. Wang, Z. Yang, Y. He, and X. Chu, "Set-sequence-graph: A multi-view approach towards exploiting reviews for recommendation," in *CIKM*, 2020, pp. 395–404.

[30] Q. Guo, Z. Sun, J. Zhang, and Y.-L. Theng, "An attentional recurrent neural network for personalized next location recommendation," in *AAAI*, vol. 34, no. 01, 2020, pp. 83–90.

[31] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *CIKM*, 2017, pp. 1419–1428.

[32] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *AAAI*, vol. 33, 2019, pp. 346–353.

[33] S. Wang, L. Hu, L. Cao, X. Huang, D. Lian, and W. Liu, "Attention-based transactional context embedding for next-item recommendation," in *AAAI*, vol. 32, no. 1, 2018.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*, 2009, pp. 452–461.

[36] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," *NIPS*, vol. 26, pp. 315–323, 2013.

[37] D. Yang, B. Qu, and P. Cudré-Mauroux, "Privacy-preserving social media data publishing for personalized ranking-based recommendation," *TKDE*, vol. 31, no. 3, pp. 507–520, 2018.

[38] D. Yang, D. Zhang, B. Qu, and P. Cudré-Mauroux, "Privcheck: Privacy-preserving check-in data publishing for personalized location based services," in *UbiComp*, 2016, pp. 545–556.

[39] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *SIGKDD*, 2011, pp. 1082–1090.

[40] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for nextbasket recommendation," in *SIGIR*, 2015, pp. 403–412.

[41] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, and J. Wu, "Sequential recommender system based on hierarchical attention networks," in *IJCAI*, 2018, pp. 3926–3932.

[42] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *TOIS*, pp. 5–53, 2004.

[43] P. Ren, Z. Chen, J. Li, Z. Ren, J. Ma, and M. de Rijke, "Repeatnet: A repeat aware neural recommendation machine for session-based recommendation," in *AAAI*, vol. 33, 2019, pp. 4806–4813.

[44] M. Wang, P. Ren, L. Mei, Z. Chen, J. Ma, and M. de Rijke, "A collaborative session-based recommendation approach with parallel memory modules," in *SIGIR*, 2019, pp. 345–354.

[45] K. Zhao, Y. Zhang, H. Yin, J. Wang, K. Zheng, X. Zhou, and C. Xing, "Discovering subsequence patterns for next poi recommendation," in *IJCAI*, 2020, pp. 3216–3222.

[46] L. Hu, L. Cao, S. Wang, G. Xu, J. Cao, and Z. Gu, "Diversifying personalized recommendation with user-session context," in *AAAI*, 2017, pp. 1858–1864.

[47] M. Jamali and M. Ester, "Using a trust network to improve top-n recommendation," in *RecSys*, 2009, pp. 181–188.

[48] D. Li, R. Jin, J. Gao, and Z. Liu, "On sampling top-k recommendation evaluation," in *SIGKDD*, 2020, pp. 2114–2124.