CA2Rec: Context-Aware Hotel Recommender System

Kwei-Herng Lai
Texas A&M University
College Station, Texas
khlai037@tamu.edu

Tsung-Lin Yang
Texas A&M University
College Station, Texas
lin.yang@tamu.edu

Chao Fan
Texas A&M University
College Station, Texas
chfan@tamu.edu

Qing Cao
Texas A&M University
College Station, Texas
caoqingmvp@me.com

ABSTRACT

The session-based and context-aware recommender system developed in this project using GRU4REC, auto-encoder, network embedding, KNN, and re-ranking function, allows us to reap the benefits of the consideration of dynamic interactions between users and items. For example, the developed system factorizes the user profiles, observable click behaviors, and the location contexts for learning session-based preferences in the domain of hotel recommendation. Doing so enables tracking of the changes in users' preferences over time, so that the system can adapts to these changes. Furthermore, mining the sequential patterns of user-item interactions is also beneficial to capture the characteristics of items which may attract attentions from broad costumers in a given context. The re-ranking function we proposed is the key to enhance the performance of our concatenated models and final outputs. The evaluation metric, Mean Reciprocal Rank (MRR), demonstrates the high performance of our model, which allows us to be the top 20% of the teams in the 2019 RecSys competition.

CCS CONCEPTS

• **Information system** → **Recommender systems**.

KEYWORDS

Recommendation, Neural Networks, Embedding

ACM Reference Format:

Kwei-Herng Lai, Tsung-Lin Yang, Chao Fan, and Qing Cao. 2019. CA2Rec: Context-Aware Hotel Recommender System. In *Proceedings of April 29–30, 2019 (CSCE670 '19)*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Hotel recommendation is a daunting task due to the wealth of attributed information of hotels and dynamic preferences of online users. Massive studies have studied the hotel recommender system via different perspectives, such as classifying reviews of users for rating hotels [9, 11] and ranking the relevance of hotel attributes for identifying user preferences [2, 7]. However, due to the evolving context of users for hotel selection and the confidentiality of users' identities, existing studies fail to be aware of the context of the users for hotel searching and recommendation. Hence, a context-aware recommender system is significantly essential for hotel recommendation.

Trivago, a global hotel search platform, specializes in comparison and recommendation in the hotel, lodging, and meta search fields [22]. They offer a product that separates the search sessions of their costumers from regular search behaviors, enabling a capability of context-aware hotel searching and recommendation. Specifically, Trivago first identify a preliminary list in accordance with the filtering rules provided by the users in a search session. Capitalizing on the user behaviors such as interaction with images and rating, the website can learn the preferences and provide a list of accommodations that can fit the preferences of the users. The iteration of the learning and recommendation process allow the website to capture the latent context of the users and narrow down the selection of hotels for their costumers.

Despite the novel idea and platform for context-aware hotel recommendation, effective hotel recommender still remain challenging due to algorithmic and data limitations. Much of the literature in conventional recommender systems focus on the models that work when user profile is available [18]. For example, matrix factorization, a representative model for collaborative filtering recommender system, bunch together users' transactions and interactions into one row of a matrix, which fails to embed the intrinsic nature of the shift of users' transactional behavior [19]. In addition, due to some privacy issues, the conventional recommender system are not able to get access to users' behavioral information [16]. The restrictions on data availability also block the effectiveness of conventional recommender system.

To address this methodological gap, we propose a context-aware hotel recommender system integrating session-based user preferences and item features (see Figure 1). The proposed method advances conventional recommender systems with a proxy of item similarities and user similarities, learning the weights for both user preferences and item features through neural networks. Doing so allows us to straightforwardly model the needs of the costumers and precisely recommend relevant items.

CSCE670 '19, College Station, TX,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

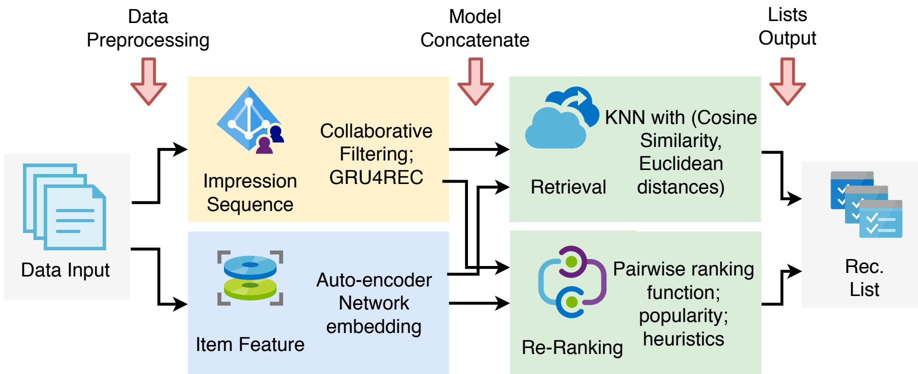

Figure 1: Context-aware System Framework for Hotel Recommender System

Specifically, we first adopt single layer RNN to extract the sequential items that the users interact for collaborative filtering. Meanwhile, a two-layer Auto-Encoder is used to reduce the dimentionality of the item features based on the provided meta-data for network embedding. Finally, we concatenate the models and outputs to re-rank the recommendation lists using K-nearest-neighbors (KNN) to optimize the results. We intend to use Mean Reciprocal Rank(MRR) as our metric to evaluate the performance of our models. MRR only cares about where the first relevant document is. In Trivago's scenario, user often links from one relevant item to another. So when a relevant item presents, others might not be as important as search engine.

2 RELATED WORK

Initially, most of the works solved this problem by rule-based, sequential pattern-based [25], and Markov Chain (MC) based recommender system [17]. For example, some algorithms replace inner products with low-dimensional embeddings and squared Euclidean distance, and then use Factorization Machines (FM) to incorporate content-based features [12]. The FM can specify the feature vectors and allow the addition of other content information to change and optimize the performance of the model [14]. Besides, MC methods are adopted to model sequential behavior of the users by learning the transition graph over items [15]. The methods benefit the prediction of next actions of the users based on their recent actions. However, these models cannot perform well to recommend items with session-based datasets.

Starting from 2013, derived from factorization-based model, there are several related works aiming to solve this problem by neural network. Recurrent neural network (RNN) has been employed in the domain of session-based recommendar systems by introducing different ranking loss functions [5] and different architectures [13]. For example, some studies considered the first click item as the initial input of the RNN and then query the model based on the initial input for a recommendation [5]. They further used the click item streams to train and refine the model. In addition, a study proposed a long-term memory model that model long sessions that usually contain user interests drift caused by unintended clicks [10]. This approach explicitly take the effects of users' current actions on their next moves into account. Another studies on considering user long-term preferences in evolving situation was conducted in a sequential recommender system which was built upon Hierarchical Attention Network. However, these studies can only perform on the sessions that contain enough user interactions with items [24]. The two-layer hierarchical attention network learns the users preferences on the historical purchased item representation and outputs final user representation through coupling user long-term and short-term preferences [6].

Despite the massive studies working on the session-based recommender systems, effective and robust recommender system still remain unknown. Most of the existing work, which is built upon recurrent neural network, would be problematic when dealing with long-length sequences such as users' behavior sequences. The reason to this limitation are twofold: first, gradient explosion/vanishing would occur in the process of training; and, a loss of information may be induced by using tricks such as batch normalization to overcome the gradient explosion/vanishing. Furthermore, there still exists open questions such as incorporating user's explicit feedback, considering users' experiences without explicit feedback for recommendation and transfer learning for cross-domain/session recommendation.

To bridge the methodological gap, this study integrates the item-based and user-based models, and concatenate the models to re-rank the output lists for recommendation. The testbed is the Trivago data set which has rich features about the items and a large number of session-based users behaviors that were recorded for every users interact with the items on their website. We develop models to incorporate existing models with RNN and also advance them to consider all enriched features in the dataset. The proposed methods and evaluation results are presented in the following sections.

3 PROPOSED METHOD

3.1 Time Series Feature Extraction

We utilize Recurrent Neural Network to capture the time series features. We use a single layer GRU with 100 hidden unit and a 100 dimension embedding layer. Like GRU4REC, we predict the next item user interacted with using the current item. The whole network was trained with 5 epochs. We extracts the output of GRU hidden units to get the final feature vector.

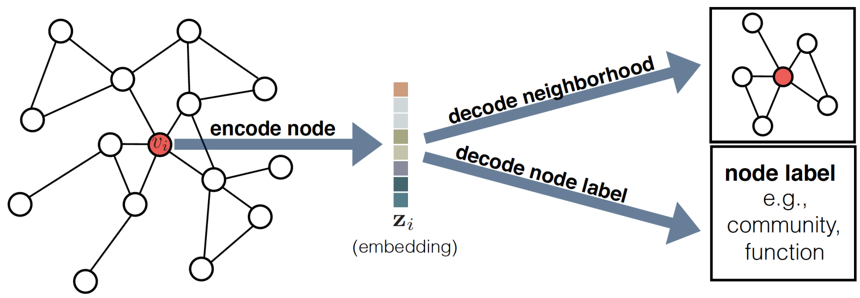

Figure 2: The encoder-decoder approach for network embedding [3]

The goal of GRU4REC is to predict next event according to the current event. The performance after adding GRU4REC embedding is not as promising as we expected. The reason maybe be twofold, too many length 1 session in test data and time feature importance in this problem is weak. In order to prevent popular item dominate the gradient, we utilize BPR-Max loss function described in [4] to regularize. Also, to speed up the training process, we adopted session mini-batch.

3.2 Dimensionality Reduction for Item Features

Trivago provides a large set of metadata for accommodations (items). It contains item ids used to be identifiers of the accommodation for item related action types, and item properties which is a pipe-separated list of filters that are applicable for the given items. To reduce the dimensionality of the metadata for items, we adopt a two-layer Auto-encoder to project the data to a low-dimension space. In this study, we use the gradient-based learning method to approximate the latent repsentation vectors for the input matrix[1]. The set of parameter W can be adjusted in an iterative manner [8]:

$$W_k = W_{k-1} - \epsilon \frac{\partial E(W)}{\partial W} \tag{1}$$

where ϵ is a scalar constant to adjust the learning rate for the encoding process.

3.3 Network Embedding

In this section, we aim to embed the user-item interaction matrix into a low-dimension space and remain the proximity attribute of the users and items. As such, we can effectively recommend the items in accordance with the preferences of users based on their behaviors in a search session. The ubiquitous network embedding methods are based on the architecture of encoder and decoder (see Figure 2). The method will maps the nodes into a low dimensional vector based on the nodes’ position in the graph. Then, the decoder can extract the user-specified information from the low-dimensional embedding and enrich the performance of node classification, which is associated with the community label for a node [3].

There are two widely used network embedding algorithms: DeepWalk and HopRec, which are applicable to our project. The first model we adopt is DeepWalk which enables unsupervised feature learning from truncated random walks.The key elements of the DeepWalk algorithm lie on twofold: SkipGram model and Hierarchical Softmax model. The SkipGram model maximizes the co-occurrence probability among the nodes that appear within a window w. The formulae to approximate the conditional probability is shown as below:

$$Pr(\{v_{i-w}, ..., v_{i+w}\} \backslash v_i | \Phi(v_i)) = \prod_{\substack{j=i-w \\ j \neq i}}^{i+w} Pr(v_j | \Phi(v_i)) \tag{2}$$

where $\Phi(v_j)$ is the current representation of v_j. In addition, the hierarchical softmax model is to factorize the conditional probability to save the computational power as:

$$Pr(u_k | \Phi(v_j)) = \prod_{l=1}^{\lceil log|V| \rceil} Pr(b_l | \Phi(v_j)) \tag{3}$$

where b_l is a node in a sequence u_k.

Another network embedding approach is called HOP-Rec (i.e., High-Order Proximity for Implicit Recommendation), which enables the incorporation of both factorization approach and graph-based models to capture the observed direct interactions between users and items and the indirect preferences from the graphs constructed by user-item interactions [23]. The objective of HOP-Rec is shown as follows:

$$\mathbb{L}_{HOP} = \sum_{\substack{1 \leq k \leq K \\ u,(i,i')}} C(k) \mathbb{E}_{\substack{i \sim P_u^k \\ i' \sim P_N^u}} [\mathbb{F}(\theta_u^T \theta_i', \theta_u^T \theta_i)] + \lambda_\Theta \|\Theta\|_2^2 \tag{4}$$

where $S_u$is the walk sequence and θ_u and θ_i are the set of data for users and items respectively. The main idea of using the HOP-Rec approach for network embedding is the approximation of high-order probabilistic matrix factorization by conducting random walk (RW) with a decay factor for confidence weighting $C(k)$ [23]. By doing so, not only the strict boundary between observed and unobserved items can be smoothed, but also it can be scalable to our large scale datasets.

3.4 Model Concatenation and Re-ranking

To integrate both item features and user behaviors into the recommender systems, we concatenate the models and outputs to re-rank the recommendation lists based on the proximity distances of the items. Specifically, we concatenate the learned representations for each item, then we normalized the concatenated features to prevent from imbalanced contributions to further operations. Then, we apply K-nearest-neighbors retrieval for each accessed items in each session to generate a list of recommendations. During the data processing stage, we observed that users in this platform tend to click on the items with more references (e.g reviews, images...etc), which drove us to re-rank the generated list by the times of being accessed of each item in the end.

4 EVALUATION AND ANALYSIS OF THE RESULTS

4.1 Data Preparation

The dataset is obtained from Trivago 2019 RecSys Challenge. In training set, there are in total 582544 users, 927143 items and 15734730 events. In addition, there is a problem of cold start because a great number of users in the testing set do not appear in the training set. In addition, every event can have one of 10 possible action types such as interaction with item info, interaction with item image, and interaction with item rate. Not every event associate with an item, some event may contain user input(e.g., search for poi). To mitigate the affect of the cold start, we filter out sessions that only have less than 5 events, which are considered as outliers. As a result, every event belong to one of 586363 sessions. After preparing the data, we can apply the feature extraction algorithms, embedding models, and re-ranking to the datasets. The final goal is to predict the item id when user performs clickout action.

4.2 Evaluation

To evaluate the performance of the proposed methods and compare the results with existing studies, we adopt the mean reciprocal rank (MRR) as the metric to measure the effectiveness of the models. MRR is a statistic measure for evaluating the list of possible responses to a sample of queries, ordered by probability of correctness. In this project, our model is supposed to be able to predict which items have been clicked by the users in a session of search result. The MRR is the average of the reciprocal ranks of results for a sample of queries [20]. The higher the actually clicked item appears on the list the higher the score. The formulae for MRR is shown as follows:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{5}$$

where $rank_i$ refers to the rank position of the first relevant document for the i-th query.

In addition, we also apply precision@25 and recall@25 to measure the performance of our proposed models. That is because, for Trivago, it requires the output to be a list of maximum 25 items for each click-out ordered by preferences for a specific user. As such, due to the diversity of the click out actions by users, the precision in such long recommendation list in this project is quite low. But, it is still a good measurement to compare the performance of the models. Recall is the fraction of the relevant documents that are successfully retrieved [21]. Applying recall@25 can benefit us to select our best model when there is a high cost associated with False Negative. The formulae for obtaining precision@25 and Recall@25 are shown as follows:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{6}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{7}$$

4.3 Results Analysis

The performance of our models are presented in Figure 3 - 5. We conduct the training and testing tasks in six models and measure their performance using MRR@25, precision@25 and recall@25. As shown in the figures, we observe that, apparently, the model with item feature and HOP-Rec serve as the best method to predict the click out items in session-based datasets, because the model achieves the highest scores in all three measurements.

In addition, observed from figure 3 and 5, the models considering item features, such as the model integrating item feature and HOP-Rec, and the model integrating impression sequence and item feature, and the model integrating impression sequence and item KNN, can perform better than other models. This results suggest that the item features play the critical role in session-based recommender systems. Meanwhile, the model with impression sequence only reach to the worse performance among the six models. It implies that impression sequence is not a very effective feature to identify the users' preferences. That is because the session-based dataset is limited to the information about users' behavioral patterns, which may lead to difficulties for the model to learn the patterns. As such, the precision of the model with impression sequence only is the lowest one that needs to be improved by item features.

To further examine the effects of the models on the effectiveness of the proposed model using impression sequence as the feature in recommendations, we compare the performance of the model incorporating impression sequence, HOP-Rec and KNN to the performance of the model incorporating impression sequence, DeepWalk, and KNN. We find that the HOP-Rec performs better than DeepWalk. That is because HOP-Rec can capture the high-order preference information in a given user-item interaction matrix.

5 CONCLUDING REMARKS

We present a CA4Rec, a unified and efficient method considering both user preferences and item feature for hotel recommendation using session-based data. The proposed method integrates Recurrent Neural Networks to extract time-series impression sequence, Auto-encoder for dimensionality reduction of item features, network embedding approaches to learn the proximity of items, and KNN for re-ranking recommendation lists. The effectiveness of the proposed method is attested by experiments on the large Trivago datasets, the results of which suggest:

- The model with item feature and HOP-Rec perform best among the six models in accordance with the MRR@25, precision@25, and recall@25.
- The impression only model cannot effectively recommend items for session-based data because the impressive items are unavailable in plenty of cases.
- Comparing to the impression sequence, item features play the key role in effectively recommending items in session-based cases.

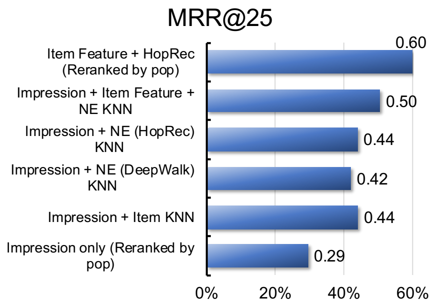

Figure 3: MRR@25 for all models

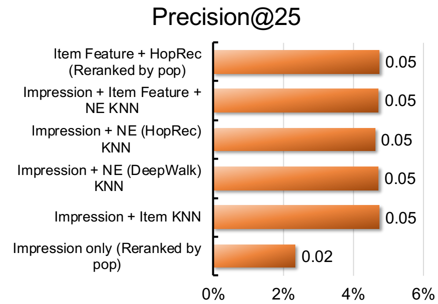

Figure 4: Precision@25 for all models

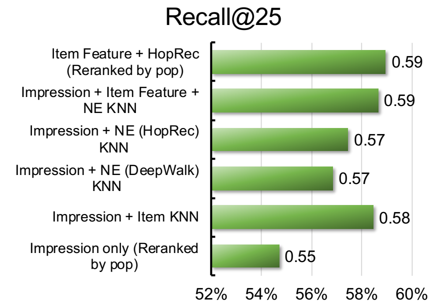

Figure 5: Recall@25 for all models

The proposed method can further be improved by employing other technical elements and incorporating more item-based features. In addition to the hotel recommender systems, the method can also be applied to other session-based datasets and platforms such as social media for disaster context.

6 CODE AVAILABILITY

The code can be found on GitHub: github.com/lhenry15/CSCE670

7 DATA AVAILABILITY

The dataset is provided by Trivago for the 2019 RecSys Challenge: https://recsys.trivago.cloud/challenge/dataset/

ACKNOWLEDGMENTS

This paper is based on the course CSCE 670: Information Retrieval and Storage.Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the instructor and teaching assistants.

REFERENCES

[1] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. 2015. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519* (2015).

[2] Astrid Dickinger and Josef Mazanec. 2008. Consumers' preferred criteria for hotel online booking. *Information and Communication Technologies in Tourism 2008* (2008), 244–254.

[3] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584* (2017).

[4] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 843–852.

[5] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[6] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 241–248.

[7] Mu-Hsing Kuo, Liang-Chu Chen, and Chien-Wen Liang. 2009. Building and evaluating a location-based service recommendation system with a preference adjustment mechanism. *Expert Systems with Applications* 36, 2 (2009), 3543–3554.

[8] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[9] Asher Levi, Osnat Mokryn, Christophe Diot, and Nina Taft. 2012. Finding a needle in a haystack of reviews: cold start context-based hotel recommender system. In *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 115–122.

[10] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. 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*. ACM, 1831–1839.

[11] Michael P O'Mahony and Barry Smyth. 2009. Learning to recommend helpful hotel reviews. In *Proceedings of the third ACM conference on Recommender systems*. ACM, 305–308.

[12] Rajiv Pasricha and Julian McAuley. 2018. Translation-based factorization machines for sequential recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 63–71.

[13] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 130–137.

[14] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International Conference on Data Mining*. IEEE, 995–1000.

[15] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 811–820.

[16] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to recommender systems handbook. In *Recommender systems handbook*. Springer, 1–35.

[17] Guy Shani, David Heckerman, and Ronen I Brafman. 2005. An MDP-based recommender system. *Journal of Machine Learning Research* 6, Sep (2005), 1265–1295.

[18] Katrien Verbert, Nikos Manouselis, Xavier Ochoa, Martin Wolpers, Hendrik Drachsler, Ivana Bosnic, and Erik Duval. 2012. Context-aware recommender systems for learning: a survey and future challenges. *IEEE Transactions on Learning Technologies* 5, 4 (2012), 318–335.

[19] Shoujin Wang, Longbing Cao, and Yan Wang. 2019. A Survey on Session-based Recommender Systems. *CoRR* abs/1902.04864 (2019). arXiv:1902.04864 http://arxiv.org/abs/1902.04864

[20] Wikipedia contributors. 2018. Mean reciprocal rank — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Mean_reciprocal_rank&oldid=872349108 [Online; accessed 28-April-2019].

[21] Wikipedia contributors. 2019. Precision and recall — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Precision_and_recall&oldid=893227571 [Online; accessed 29-April-2019].

[22] Wikipedia contributors. 2019. Trivago — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Trivago&oldid=893546521 [Online; accessed 27-April-2019].

[23] Jheng-Hong Yang, Chih-Ming Chen, Chuan-Ju Wang, and Ming-Feng Tsai. 2018. HOP-rec: high-order proximity for implicit recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. ACM, 140–144.

[24] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential recommender system based on hierarchical attention networks. In *the 27th International Joint Conference on Artificial Intelligence*.

[25] Andrew Zimdars, David Maxwell Chickering, and Christopher Meek. 2001. Using temporal data for making recommendations. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 580–588.