Time-Dependent Next-Basket Recommendations

Sergey Naumov[1,3][0000-0002-0682-5973], Marina Ananyeva[1,2][0000-0002-9885-2230], Oleg Lashinin[1][0000-0001-8894-9592], Sergey Kolesnikov[1], and Dmitry I. Ignatov[2⋆][0000-0002-6584-8534]

[1] Tinkoff, 2-Ya Khutorskaya Ulitsa, 38A, bld. 26, Moscow, 127287, Russian Federation
[2] National Research University Higher School of Economics, 20 Myasnitskaya St, Moscow, 101000, Russian Federation
[3] Moscow Institute of Physics and Technology, Institutskiy Pereulok, 9, Dolgoprudny, Moscow Oblast, 141701, Russian Federation

Abstract. There are various real-world applications for next-basket recommender systems. One of them is guiding a website user who wants to buy anything toward a collection of items. Recent works demonstrate that methods based on the frequency of prior purchases outperform other deep learning algorithms in terms of performance. These techniques, however, do not consider timestamps and time intervals between interactions. Additionally, they often miss the time period that passes between the last known basket and the prediction time. In this study, we explore whether such knowledge could improve current state-of-the-art next-basket recommender systems. Our results on three real-world datasets show how such enhancement may increase prediction quality. These findings might pave the way for important research directions in the field of next-basket recommendations.

Keywords: Recommender systems · Next-basket recommendations · Time-dependent recommendations.

1 Introduction

Next-basket recommender systems (NBR) have been actively studied in the research community [19, 24]. The developed methods may employ a variety of data sources, including past user purchases [2, 8, 11, 12], current session click history [1, 21], and other user and item attributes [3, 10, 17, 26]. However, state-of-the-art approaches [8, 12] usually do not take into account timestamps of interactions. Even though they weigh the baskets according to their order of appearance, they are still not (1) time-aware approaches nor (2) able to generate time-dependent recommendations.

Time-aware recommender models can extract additional information from historical interaction timestamps [27]. If the model does not consider them, it treats the baskets as equidistant. In practise, time gaps are very important when

⋆ The contribution of D.I. Ignatov to the paper was done within the framework of the HSE University Basic Research Program.

modelling user behaviour. Small gaps between baskets could result in greater dependence on recent interactions in subsequent baskets. According to [18], large time gaps could be a sign of weaker connections between user behaviour in the past and present.

Time-dependent recommendations can change depending on when the predictions were made [25]. In non-time-dependent approaches [2, 8, 12], users' representations are calculated at the time of the last known basket. However, the user's interests could change if some time elapses between the last known interactions and the prediction time [18]. Fortunately, we have the ability to update representations to reflect the current moment in time. We can use the times when test baskets were purchased in offline experiments. Alternatively, we can use the time period when a user sees recommendations in an online scenario. The key concept is that recommendations change over time, even when a user does not further interact with any items. As a result, these models are known as time-dependent ones [25, 27].

Recent works have emphasised the superiority of straightforward frequency-based approaches in the next-basket recommendations [12, 16, 24]. Unfortunately, the majority of cutting-edge algorithms lack time features. One of them is TIFU-KNN [12], which uses purchase frequency to make recommendations based on the purchases of the target user's neighbours. In this paper, we add time information to TIFU-KNN. Specifically, the main contribution can be listed as follows:

- We modify TIFU-KNN, a state-of-the-art approach for next-basket recommendations, to make it (1) time-aware and (2) time-dependent
- We conduct comprehensive experiments to demonstrate how such a straightforward change can enhance the quality of recommendations on three real-world datasets.

2 Related Work

Different approaches have been applied to solve next-basket recommendations. Previously published works employ Markov Chains [30], Recurrent Neural Networks [2, 11, 22, 30], Attention mechanisms [23, 31], Graph Neural Networks [31], and frequency-based approaches [8, 12]. Frequency-based methods perform better [19, 24] than other methods, despite deep neural networks' great success in other research areas. It emphasises the significance of enhancing frequency-based models.

The addition of time features to recommender models is another area of study. Time is used as additional information by time-aware models to model user interests [4, 25, 27]. For instance, the well-known SASRec [15] has been improved by the TiSASRec [18], which has a time interval-aware Self-Attention mechanism. Recent independent studies [6, 13, 20] have revealed that TiSASRec typically outperforms SASRec in terms of quality. Time-dependent models' predictions can differ depending on the current time context (hour, day of the week, or month) [7,9,29] or the time before the most recent interaction [5,14,32]. The authors of [28] introduced DRM that can dynamically change next-basket recommendations based on the user's current time context. Although it makes sense to use time as a feature for both training and predictions, there are not many time-dependent next-basket approaches.

3 Original and Modified Versions of TIFU-KNN

Original TIFU-KNN is a KNN-based non-DL method described in [12]. Among non-deep-learning models, it currently displays the best results in the next-basket recommendation task [19, 24]. The baskets are separated into nearly equal-sized groups. It allows to introduce an additional global time-decayed factor. Within the group, each basket has a unique ordinal number. Utilisation of two different weights simultaneously is the key concept; baskets are weighted within groups, and groups are weighted among themselves. The weight of each basket in the group varies depending on when it was purchased $r_b(i) = r_b^i$ (r_b power i), and $i = 0, 1, \ldots, B(g) - 1$ is the index number from the most recent basket in the group to the earliest basket, $B(g)$ is the number of baskets in the group g. Similarly, earlier groups of baskets have smaller weight $r_g(j) = r_g^j$ (r_g power j), $j = 0, 1, \ldots, G - 1$ from the most recent group to the earliest group.

We consider a I-sized multi-hot vector v_b that represents a basket b, where I is the number of items. If a basket b contains an item i then the corresponding component equals 1, and otherwise equals 0. If the group vector v_g is thus obtained as a weighted average vector of the baskets v_b, and the user vector v_u is taken into consideration as a weighted average vector of the groups v_g:

$$v_g = \sum_{i=0}^{B(g)-1} \frac{r_b(i) \cdot v_{b_i}}{B(g)}, \quad v_u = \sum_{j=0}^{G-1} \frac{r_g(j) \cdot v_{g_j}}{G}, \tag{1}$$

where r_b is the time-decayed ratio within a group, r_g is the time-decayed ratio across groups, $B(g)$ is the number of baskets in group g, and G is the number of groups, v_{b_i} is the vector of the i-th basket, v_{g_j} is the vector of the j-th group, v_u is the user's final vector representation.

The average of the vectors v_u of the k closest users is also calculated for each user's nearest neighbours vector $v_{nn}(v_u)$ (using Euclidean distance).

$$KNN(v_u) = \{v_{u_0}, v_{u_1}, v_{u_2}, \ldots, v_{u_{K-1}}\}, \quad v_{nn}(v_u) = \sum_{i=0}^{K} \frac{KNN(v_u)[i]}{K}. \tag{2}$$

The prediction vector $P(u)$ for each user is the weighted sum of the user's own vector v_u and the nn-vector $v_{nn}(v_u)$:

$$P(u) = \alpha \cdot v_u + (1 - \alpha) \cdot v_{nn}(v_u), \tag{3}$$

where α is the balance coefficient between two parts. $P(u)$ is used to calculate the final recommendations.

Time-aware TIFU-KNN (TIFU-KNN-TA) is easily attainable with a few adjustments. Each user's entire purchase history is divided into equal time segments of gs days. $ts_{last}(u)$ corresponds to the timestamp of the last train basket of user u. Then the first group's baskets are distributed between $ts_{last}(u) - gs$ and $ts_{last}(u)$. Interactions between $ts_{last}(u) - 2 \cdot gs$ and $ts_{last}(u) - gs$ form the second group. Group timestamp restrictions for user u are as follows:

$$group_m(u): \ (ts_{last}(u) - (m+1) \cdot gs, \ ts_{last}(u) - m \cdot gs), \quad m = 0, 1, 2, \dots \tag{4}$$

As a result, each user's group size in days is fixed, but the number of baskets in each group and the total number of groups can vary. For the group $r_g(j)$, the attenuation is still the same as it is in the default TIFU-KNN (Equation 1). On the other hand, the group's r_b basket coefficient has changed. Instead of the basket number, the exponent is now the number of days until the group's end (or natural logarithm of the number of days, depending on the hyperparameter use_log). Let us denote the right limit of the group g from Equation 4 as $rl(g)$, and the timestamp of the basket b as $ts(b)$:

$$\Delta ts(b,g) = rl(g) - ts(b), \tag{5}$$

$$r_b(\Delta ts) = r_b^{\Delta ts(b,g)} \quad \text{or} \quad r_b(\Delta ts) = r_b^{\ln(\Delta ts(b,g))}, \tag{6}$$

$$v_g = \sum_{i=0}^{B(g)} \frac{r_b(\Delta ts) \cdot v_{b_i}}{B(g)}. \tag{7}$$

Time-dependent TIFU-KNN (TIFU-KNN-TD) has two differences from Time-aware TIFU-KNN. During the prediction stage, a timestamp of the next basket $ts_{test}(u)$ is served to the model for each user. In offline experiments, this could be the moment when a user buys test or validation baskets. Additionally, we can use time of predictions if the experiments are online. In order to create groups of baskets for the purpose of calculating the user vector v_u^{new}, $ts_{test}(u)$ is used instead of the maximum timestamp $ts_{last}(u)$ from the train. User u now has the following group timestamp limitations:

$$group_m(u): \ (ts_{test}(u) - (m+1) \cdot gs, \ ts_{test}(u) - m \cdot gs), \tag{8}$$

Thus, on the validation and test stages, the model has new vectors v_u^{new} for each user. However, the nearest neighbour representations are determined for $ts_{last}(u)$. This is done to prevent the need to continually recalculate the vectors of all nearby users. As a result, we calculate vectors v_u^{new} for the target user u using the current moment of time. However, the vectors for nearest neighbours are only based on training stage.

$$\begin{gathered} KNN(v_u^{new}) = \{v_{u_0}, v_{u_1}, v_{u_2}, ..., v_{u_{K-1}}\}, \\ v_{nn}(v_u^{new}) = \sum_{i=0}^{K} \frac{KNN(v_u^{new})[i]}{K}, \end{gathered} \tag{9}$$

$$P(u) = \alpha \cdot v_u^{new} + (1 - \alpha) \cdot v_{nn}(v_u^{new}). \tag{10}$$

Table 1: Dataset statistics after preprocessing.

Dataset	#users	#items	#baskets	#baskets per user	#items per basket	#items per user
Dunnhumby	2471	8644	251361	101.72	7.71	381.09
Tafeng	14006	13674	94274	6.73	6.34	37.61
Instacart	19999	26677	629067	31.45	9.94	100.22

4 Experiments

We have provided experiments to answer the following research questions:

- **RQ1**: Can we increase the quality of recommendations by taking time intervals into account instead of basket numbers?
- **RQ2**: Does the consideration of the time of prediction improve the quality of recommendations?

4.1 Datasets

We make use of the three open source datasets for the Next Basket Recommendation problem to ensure the reproducibility of our research:

- **Dunnhumby**[4] includes transactions of 2,500 households at a retailer over a two-year period. A basket is a collection of all the items that were purchased in a single transaction.
- **TaFeng**[5] includes four months of Chinese grocery store transactions. Each basket contains the user's daily purchases.
- **Instacart**[6] it contains a sample of over 3 million grocery orders from over 200,000 users with an average of 4 to 100 orders from each user. Every order is considered to be one basket.

From each dataset, we remove users with fewer than three baskets and items bought by fewer than five users. We sample 20,000 Instacart users and 10,000 Dunnhumby items before filtering. Table 1 displays the statistics of the datasets after prepossessing. Every dataset was divided into a training, validation, and test set for our experiments. For each user, the training part consists of all baskets except the final one. The remaining baskets are split in half, with 50% going to the test part and 50% to the validation part.

[4] `https://www.kaggle.com/datasets/frtgnn/dunnhumby-the-complete-journey`
[5] `https://www.kaggle.com/datasets/chiranjivdas09/ta-feng-grocery-dataset`
[6] `https://www.kaggle.com/competitions/instacart-market-basket-analysis/data`

Table 2: Results of our models compared against the baselines. The best and second best performing models are indicated by boldface and underline, respectively. ▲% shows our models' improvements over the best baseline.

Dataset	Metric	Baselines G-Pop	GP-Pop	UP-CF@r	TIFU-KNN	Ours TIFU-KNN-TA (▲%)	TIFU-KNN-TD (▲%)
DHB	Recall@5	0.1379	0.2326	0.2397	0.2491	**0.2572** (3.3)	<u>0.2570</u> (3.2)
	nDCG@5	0.1229	0.2222	0.2294	0.2355	**0.2433** (3.3)	<u>0.2422</u> (2.8)
	Recall@10	0.1359	0.2473	0.2611	0.2709	**0.2760** (1.9)	<u>0.2743</u> (1.3)
	nDCG@10	0.1158	0.2188	0.2298	0.2384	**0.2439** (2.3)	<u>0.2425</u> (1.7)
TaFeng	Recall@5	0.0815	0.1026	0.1244	0.1403	<u>0.1415</u> (0.9)	**0.1448** (3.2)
	nDCG@5	0.0895	0.0979	0.1121	<u>0.1347</u>	0.1341 (-0.4)	**0.1393** (3.4)
	Recall@10	0.0841	0.1260	0.1537	0.1632	<u>0.1642</u> (0.6)	**0.1673** (2.5)
	nDCG@10	0.0877	0.1047	0.1227	<u>0.1406</u>	0.1401 (-0.4)	**0.1453** (3.3)
Instacart	Recall@5	0.1092	0.4070	0.4371	0.4524	<u>0.4541</u> (0.4)	**0.4559** (0.8)
	nDCG@5	0.1183	0.4238	0.4527	0.4668	<u>0.4691</u> (0.5)	**0.4725** (1.2)
	Recall@10	0.0969	0.4000	0.4276	<u>0.4476</u>	0.4469 (-0.2)	**0.4496** (0.4)
	nDCG@10	0.1051	0.4039	0.4320	0.4484	<u>0.4493</u> (0.2)	**0.4526** (0.9)

4.2 Baseline Methods

In order to ensure that our research is thorough, we also include the following baselines:

- **G-Pop**: this baseline just recommends the most frequent items in the dataset.
- **GP-Pop**: for each user, the most frequently purchased items are recommended first, followed by the most frequent items in the entire dataset.
- **UP-CF@r**: a hybrid of the recency-aware user-wise popularity and user-wise collaborative filtering presented in [8].

4.3 Metrics

We calculate **Recall** and **nDCG**, which have been used in previous NBR studies, to assess the effectiveness of our methods. Based on the average basket size in the datasets Table 1, we picked values of 5 and 10 for the *topk* parameter.

4.4 Experiment Settings

We search for the optimal parameters using Optuna[7] with 300 trials for each model, optimising Recall@10. The random seeds are all fixed. We make the experiment code available online, including hyperparameter search spaces[8].

[7] https://optuna.org
[8] https://github.com/sergunya17/time_dependent_nbr

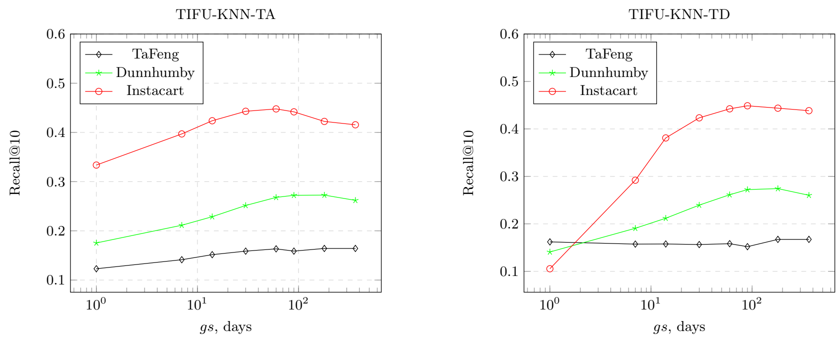

Fig. 1: Recall@10 w.r.t. different gs values across all included datasets.

4.5 Results

Table 2 answer both **RQ1** and **RQ2**. TIFU-KNN-TA outperforms all included baselines on Dunnhumby and Instacart and shows similar performance on other datasets (**RQ1**). This demonstrates the value of replacing ordinal number weighting of baskets with weighting based on the amount of time between interactions.

Moreover, TIFU-KNN-TD outperforms all included algorithms both on TaFeng and Instacart. Additionally, it has higher quality on original TIFU-KNN in all experiments (**RQ2**). Finally, TIFU-KNN is better than UP-CF@r on all metrics and datasets which is in line with [2]. As we can see, our modifications improved quality of recommendations by introducing time features both for training and prediction stages.

It is important to note the dependence on hyperparameter values. The two novel hyperparameters for the suggested methods are use_log and gs. Our experimental findings across all included datasets indicate that the quality is either unchanged or slightly improved when the logarithm is used. Additionally, the quality of recommendations can be completely affected by varying the value of parameter gs. While fixing the remaining values in the optimal configuration for each model and dataset, we varied the value of hyperparameter gs. Figure 1 shows the results.

5 Conclusion

In this study, we demonstrated the importance of providing time-dependent and time-aware next-basket recommendations. We made some minor adjustments to the state-of-the-art TIFU-KNN next-basket recommender system to show the impact of time context. On three real-world datasets, the quality of next-basket predictions was improved by merely substituting basket number for interaction weighting using timestamp-based descent. We believe that these findings will spur additional study into the creation of time-sensitive next-basket recommendation techniques for both training and prediction phases.

References

1. Ananyeva, M., Lashinin, O., Ivanova, V., Kolesnikov, S., Ignatov, D.I.: Towards interaction-based user embeddings in sequential recommender models. In: Vinagre, J., Al-Ghossein, M., Jorge, A.M., Bifet, A., Peska, L. (eds.) Proceedings of the 5th Workshop on Online Recommender Systems and User Modeling, ORSUM@RecSys 2022. CEUR Workshop Proceedings, vol. 3303. CEUR-WS.org (2022), `http://ceur-ws.org/Vol-3303/paper10.pdf`
2. Ariannezhad, M., Jullien, S., Li, M., Fang, M., Schelter, S., de Rijke, M.: ReCANet: A Repeat Consumption-Aware Neural Network for Next Basket Recommendation in Grocery Shopping. In: Proceedings of the 45th International ACM SIGIR Conference. p. 1240–1250. SIGIR '22, ACM (2022)
3. Bai, T., Nie, J.Y., Zhao, W.X., Zhu, Y., Du, P., Wen, J.R.: An attribute-aware neural attentive model for next basket recommendation. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 1201–1204 (2018)
4. Campos, P.G., Díez, F., Cantador, I.: Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. User Modeling and User-Adapted Interaction **24**(1), 67–119 (2014)
5. Cao, J., Lin, X., Cong, X., Guo, S., Tang, H., Liu, T., Wang, B.: Deep structural point process for learning temporal interaction networks. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 305–320. Springer (2021)
6. Chen, Z., Zhang, W., Yan, J., Wang, G., Wang, J.: Learning Dual Dynamic Representations on Time-Sliced User-Item Interaction Graphs for Sequential Recommendation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 231–240 (2021)
7. Cho, J., Hyun, D., Kang, S., Yu, H.: Learning heterogeneous temporal patterns of user preference for timely recommendation. In: Proceedings of the Web Conference 2021. pp. 1274–1283 (2021)
8. Faggioli, G., Polato, M., Aiolli, F.: Recency aware collaborative filtering for next basket recommendation. In: Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization. pp. 80–87 (2020)
9. Fan, Z., Liu, Z., Zhang, J., Xiong, Y., Zheng, L., Yu, P.S.: Continuous-time sequential recommendation with temporal graph collaborative transformer. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 433–442 (2021)
10. Fouad, M.A., Hussein, W., Rady, S., Philip, S.Y., Gharib, T.F.: An Efficient Approach for Rational Next-Basket Recommendation. IEEE Access **10**, 75657–75671 (2022)
11. Hu, H., He, X.: Sets2sets: Learning from sequential sets with neural networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1491–1499 (2019)
12. Hu, H., He, X., Gao, J., Zhang, Z.L.: Modeling personalized item frequency information for next-basket recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1071–1080 (2020)
13. Huang, Z., Ma, J., Dong, Y., Foutz, N.Z., Li, J.: Empowering Next POI Recommendation with Multi-Relational Modeling. arXiv preprint arXiv:2204.12288 (2022)

14. Ji, W., Wang, K., Wang, X., Chen, T., Cristea, A.: Sequential recommender via time-aware attentive memory network. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 565–574 (2020)
15. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM). pp. 197–206. IEEE (2018)
16. Kolesnikov, S., Lashinin, O., Pechatov, M., Kosov, A.: TTRS: Tinkoff Transactions Recommender System benchmark. arXiv preprint arXiv:2110.05589 (2021)
17. Lashinin, O., Ananyeva, M.: Next-basket recommendation constrained by total cost. In: Vinagre, J., et al. (eds.) Proceedings of the 5th Workshop on Online Recommender Systems and User Modeling, ORSUM@RecSys 2022. CEUR Workshop Proceedings, vol. 3303. CEUR-WS.org (2022), `http://ceur-ws.org/Vol-3303/brainstorming1.pdf`
18. Li, J., Wang, Y., McAuley, J.: Time interval aware self-attention for sequential recommendation. In: Proceedings of the 13th international conference on web search and data mining. pp. 322–330 (2020)
19. Li, M., Jullien, S., Ariannezhad, M., de Rijke, M.: A Next Basket Recommendation Reality Check. arXiv preprint arXiv:2109.14233 (2021)
20. Liu, C., Liu, X., Zheng, R., Zhang, L., Liang, X., Li, J., Wu, L., Zhang, M., Lin, L.: C^2-Rec: An Effective Consistency Constraint for Sequential Recommendation. arXiv preprint arXiv:2112.06668 (2021)
21. Moreira, G.d.S.P., Rabhi, S., Ak, R., Kabir, M.Y., Oldridge, E.: Transformers with multi-modal features and post-fusion context for e-commerce session-based recommendation. arXiv preprint arXiv:2107.05124 (2021)
22. Qin, Y., Wang, P., Li, C.: The world is binary: Contrastive learning for denoising next basket recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 859–868 (2021)
23. Ren, P., Chen, Z., Li, J., Ren, Z., Ma, J., De Rijke, M.: Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 4806–4813 (2019)
24. Shao, Z., Wang, S., Zhang, Q., Lu, W., Li, Z., Peng, X.: A Systematical Evaluation for Next-Basket Recommendation Algorithms. arXiv preprint arXiv:2209.02892 (2022)
25. Shi, Y., Larson, M., Hanjalic, A.: Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. ACM Computing Surveys (CSUR) **47**(1), 1–45 (2014)
26. Van Maasakkers, L., Fok, D., Donkers, B.: Next-basket prediction in a high-dimensional setting using gated recurrent units. Expert Systems with Applications **212**, 118795 (2023)
27. Vinagre, J., Jorge, A.M., Gama, J.: An overview on the exploitation of time in collaborative filtering. Wiley interdisciplinary reviews: Data mining and knowledge discovery **5**(5), 195–215 (2015)
28. Wang, P., Zhang, Y., Niu, S., Guo, J.: Modeling temporal dynamics of users' purchase behaviors for next basket prediction. Journal of Computer Science and Technology **34**(6), 1230–1240 (2019)
29. Wu, Y., Li, K., Zhao, G., Xueming, Q.: Personalized long-and short-term preference learning for next poi recommendation. IEEE Transactions on Knowledge and Data Engineering (2020)
30. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T.: A Dynamic Recurrent Model for Next Basket Recommendation. In: Proceedings of the 39th International ACM SIGIR

Conference. p. 729–732. SIGIR '16, Association for Computing Machinery, New York, NY, USA (2016)

31. Yu, L., Sun, L., Du, B., Liu, C., Xiong, H., Lv, W.: Predicting temporal sets with deep neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1083–1091 (2020)
32. Zhu, Y., Li, H., Liao, Y., Wang, B., Guan, Z., Liu, H., Cai, D.: What to Do Next: Modeling User Behaviors by Time-LSTM. In: IJCAI. vol. 17, pp. 3602–3608 (2017)