The Impact of Contexts in Contextual Matrix Factorization Models

Alejandro Bellogín[0000−0001−6368−2510]

Universidad Autónoma de Madrid, Madrid, Spain
alejandro.bellogin@uam.es

Abstract. While matrix factorisation and neural network techniques have become increasingly popular for incorporating contextual information in recommendation, a gap exists in understanding the impact of multidimensional contexts on the performance of these methods. Our study highlights this gap, noting the absence of analysis determining which types of multidimensional contexts yield the greatest benefits from the application of these approaches. In our experiments with real data from the tourism domain, where several contexts are available at the same time, we conclude that inherent venue features – such as their categories or whether a parking is available – impact more in the results than other contexts that can be captured with other data signals – such as venue popularity.

Keywords: Recommender Systems · Deep Learning · Context.

1 Introduction

In today's dynamic and personalized online environment, context plays an increasingly vital role in shaping user behaviour and preferences. Whether it is the time of day influencing purchasing decisions, the location affecting restaurant choices, or the presence of social companions impacting entertainment preferences, context is inextricably linked to user needs and desires [2]. Recognising this importance, Context-Aware Recommender Systems (CARS) have gained significant traction [3], since they allow to capture the complexity of real-world decision-making, where context plays a crucial role. By explicitly incorporating contextual information into the recommendation process, CARS lead to more personalised and relevant recommendations [3], where Matrix Factorisation (MF) and Neural Networks (NNs) have become dominant paradigms to achieve this goal [9].

As presented in [10] and [12], incorporating contexts in deep learning architectures allows for better contextual representations, which in turn help learning more accurately the user preferences. Indeed, those two works propose deep context-aware recommendation models supporting explicit, latent, or structured context embeddings. To do that, the authors from [10] extend the neural collaborative filtering models previously proposed in [8] (NCF and NeuMF) by adding a new component of contextual information. In this way, a generalisation of NeuMF is defined that consists of two major components which incorporate contextual information in different ways: Generalised Matrix Factorisation (GMF) and MultiLayer Perceptron (MLP). In that model, these GMF and MLP towers allows to obtain better performance, either in terms of error prediction or ranking [10].

Following that line of work, in [12] the authors extended that method by exploiting in which components the context vectors can be incorporated and how they should be learned. By doing this, authors found that there was always a variant of their approach (named NeuCMF) which obtained better results than DeepFM [7], which can take rich information into consideration and was not included in the experiments from [10]. Hence, NeuCMF seems like a promising state-of-the-art algorithm for CARS, however, no analyses have been made to understand how this approach depends on context variability.

In this work, we analyse the impact of different contexts on the performance of NeuCMF. In particular, we consider how the model is affected by being trained with or without some contexts in the tourism domain. By doing this, we conclude that inherent venue features – such as their categories or whether a parking is available – impact more in the results than other contexts that can be captured with other signals of the data – such as how popular a venue is. Our main contribution is, thus, a deeper understanding on how contextual recommendation approaches based on MF and NNs depend on the type of contexts being exploited. Even though this is an ongoing work, we aim to bring the current gap in the literature about this issue and motivate future research to explore the interplay between multidimensional context characteristics and complex recommendation techniques such as NeuCMF or others based on MF and NNs.

2 Contextual Matrix Factorization Models

NeuCMF models, as defined in [12], are a family of approaches comprising the 6 variations summarised in Table 1 and implemented in the DeepCARSKit library [11]. These models may use up to 3 GMF towers (called simply MF towers in that work) and an MLP tower. The MF towers execute the elementwise product to simulate a Matrix Factorisation method either between the user and item embeddings (UI tower), user and context embeddings (UC tower), or item and context embeddings (IC). At the same time, the MLP tower may concatenate embeddings from users (MLP$_u$), from items (MLP$_i$), or from the contexts (MLP$_c$) as the first input to several layers of the multilayer perceptron.

Moreover, [12] proposes 3 options to incorporate the context in the model, based on the components previously defined:

- Option 1, where the contexts are only considered in the MLP tower, ignoring the UC and IC towers.
- Option 2, where contexts are only incorporated in the MF towers, hence using the UC and IC towers, and discarding MLP$_c$ from the MLP tower.
- Option 3, where contexts are used in both the MF towers and MLP towers.

Furthermore, the authors propose 2 methods to represent the context as the embeddings: the w mode, where the embedding is created for the whole context situation; or the i mode, where embeddings are created for each context condition, and the representation of a context situation is obtained by concatenating each individual embedding vector for the conditions in the context situation. This method, in principle, is expected to alleviate the sparsity issue, especially when there are many context dimensions in the data set, even though it assumes an independent relationship among context conditions, which may not always hold.

Table 1. Variations of NeuCMF model as defined in [12].

Variation	Option	Towers in NeuCMF	Mode for Context Embedding
NeuCMF$_{0i}$	1	UI + MLP Towers	i mode
NeuCMF$_{0w}$		with MLP$_c$	w mode
NeuCMF$_{i0}$	2	All 4 towers	i mode
NeuCMF$_{w0}$		without MLP$_c$	w mode
NeuCMF$_{ii}$	3	All 4 towers	i mode
NeuCMF$_{ww}$		with MLP$_c$	w mode

3 Experiments and Results

In this section, we first present the settings followed in the reported experiments (Section 3.1) and how the contexts were obtained from the data (Section 3.2). Then, in Sections 3.3 and 3.4 we present the results found to answer the main research question we consider in this work: *what is the impact of contexts in NeuCMF models?* We first consider only the effect of individual contexts (Section 3.3), consequently, some preliminary tests for pairs of contexts are included and analysed (Section 3.4).

3.1 Experimental settings

We consider the Yelp dataset[1], which was not included as a tested domain in [12]. This dataset includes user reviews and ratings to different tourism venues, especially restaurants, of several cities in the United States. Because of limitations with the library, we selected a city large enough to perform the experiments while having a decent amount of items. After filtering out interactions with incomplete information, or users with few reviews (to have enough information to train and evaluate the algorithm), we selected the city of *Tampa* with 13K reviews, a size comparable to the second-largest dataset used in [12].

We performed a 5-fold cross validation split, and report Precision and Recall metrics at various cutoffs [6]. Even though in [12] the authors reported error metrics (i.e., MAE and RMSE), in this work we decided to focus on ranking metrics, as they are more appropriate for the top-N recommendation problem [6].

3.2 Considered contextual dimensions

One of the motivations to use this dataset is that several contexts become available. In this section, we present which ones we considered. In every case, they were extracted from information derived from data, not inferred from textual reviews as in recent work [4].

First of all, taking the timestamp of the review into account, we extracted two contextual dimensions: based on the day of the week, whether it was a weekday (`WD`) or weekend day (`WE`); and based on the hour, if the meal was dinner (`Din`, after 5PM) or lunch (`Lch`). These two dimensions represent the temporal context attached to the user interaction.

[1] `https://business.yelp.com/data/resources/open-dataset/`

Table 2. Precision and Recall values of NeuCMF when all the contexts are considered (row 'all'), when one context is removed (middle block), or two contexts are removed (last block). Best improvement with respect to 'all' in bold, worst result in italic. Arrows denote whether results with respect to 'all' are improved (↑) or not (↓) when removing those contexts.

Contexts		P@10	P@20	R@10	R@20
all		0.0176	0.0111	0.1760	0.2213
no categories	↓	*0.0094*	*0.0084*	*0.0646*	*0.1112*
no day	↑	0.0220	0.0149	0.2187	0.2964
no parking	↓	0.0139	0.0099	0.1386	0.1964
no meal	↓	0.0170	0.0111	0.1689	0.2208
no popularity	↑	**0.0270**	**0.0207**	**0.2686**	**0.4117**
no pop, no day	↓	0.0154	0.0095	0.1531	0.1876
no cat, no par	↓	*0.0085*	*0.0084*	*0.0520*	*0.0977*

Secondly, we understand the category of a venue can be interpreted as a contextual dimension representing the type of item a user is interested in at that moment. Since items in this dataset may have more than one category assigned, we use all the categories combined and use their value after hashing to obtain unique representations.

Finally, we also consider two more contexts related to characteristics of the items (item features). Whether the venue had its own parking lot (`Gar`) or not (`NoGar`) and whether it could be considered a popular place. For this, we determined that it was popular (`Pop`) if its number of reviews was within the top 20% of all the venues in that city (for Tampa, this meant that it had to be greater than 418), otherwise the venue was considered not popular (`NoPop`).

We understand the last three contextual dimensions (category, parking, popularity), while being strictly item features, they can be associated to contextual preferences of users at a specific time, such as whether they need a restaurant with parking or from a specific category.

3.3 Performance analysis when one contextual dimension is removed

Table 2 shows the performance obtained when the selected NeuCMF model[2] exploits all the dimensions explained in Section 3.2, and how the performance is affected by removing one context at a time.

We observe that performance improves whenever the popularity or the day contexts are removed. On the other hand, if the other dimensions (categories, parking, or meal) are removed, performance decreases. This behaviour is consistent for both metrics and cutoffs.

Based on these results, we conclude that the most impactful contexts are those that are not easy to be represented already by other signals of the data while, at the same time, embed enough information by themselves. For example, *popularity* can already be captured by rating patterns, and several methodologies have been proposed to model (and uncover) this signal [1]. The case of the contextual dimension *day* is related to the granularity of this context; whereas in

[2] For this preliminary study, we conducted some tests with our data and the best performing method was NeuCMF$_{ii}$, so this is the NeuCMF variation used here. We leave for future work an exhaustive analysis of the dependency of these contextual dimensions for the other variations described in Section 2.

some domains it might be enough to discriminate between weekends and week days (such as restaurant or movie recommendation [5]), for the overall domain of tourism, and in particular, for the Yelp dataset, this is not enough, as interactions are spread across the entire week.

3.4 Performance analysis when removing pairs of contextual dimensions

The last rows of Table 2 show the performance of NeuCMF when a pair of contextual dimensions was removed. For the sake of space, only two pairs are included in the table. The first one is a combination *(no pop, no day)* where both contexts have demonstrated to be bad for the model (i.e., their removal increased performance); the second one *(no cat, no par)* illustrates a situation where both are individually good for the model.

According to the obtained results, for the two analysed combinations performance decrease. This may be attributed, in the first case, to the fact that too much information is removed, since even though both dimensions (individually) improve the model when removed, it seems the method is not able to learn user preferences and discriminate the corresponding contextual situations well enough without this pair of dimensions. For the second case, since the performance decreases when each context was removed, it is no surprise that when removing both, performance decreases too. However, it should be noted that, in terms of obtained values, this second combination produces the worst results, evidencing that these contextual dimensions are more useful than other dimensions, and it is not only a matter of reducing too much the amount of information available for the model.

4 Discussion and Limitations

In the study presented herein, we have discovered the impact of multidimensional contexts in the performance of NeuCMF; we now dig further in this analysis. We note that the analysed contextual dimensions have a larger impact on Recall than Precision, and this impact is more positive in higher cutoffs (20), both in terms of actual improvement (from 53.4% at cutoff 10 to 86.5% for *no popularity*, for example), but also in lower losses (from -46.6% at cutoff 10 to -24.3% for *no categories*, for example).

We have also observed that removing two dimensions at the same time is not convenient, at least for the combinations tested in our study. But the fact that removing one contextual dimension may improve the performance, raises the importance of properly understanding the research question of this work (*what is the impact of contexts in NeuCMF models?*), since it could bring advantages in terms of efficiency, but also in terms of the accountability of the algorithms, as the less information an algorithm takes into account, the less exposed users will be and less privacy risks they would face. Hence, when evidencing comparable performances, models using (and requiring) less data from users should be preferred.

One important limitation of this work is that more contextual dimensions should be tested. In this work, we extracted dimensions where some contextual meaning could be attached to, but according to recent works, many others could be extracted directly from the user reviews [10], so it would be worthwhile to analyse if either types of contexts impact in the same way. Other NeuCMF models should be considered, as we focused on the one that performed the best in some preliminary experiments. By comparing other variations, it would also make it possible to analyse the impact of the model complexity on how capable they are to continue working at decent performance levels with a varying number of contexts.

5 Conclusions

Better exploiting and integrating the contexts of user interactions is key to understand the user behaviour and improve their recommendations. However, for such purpose it is necessary to know which context information is more valuable for a given contextual recommendation algorithm. The work reported in this paper presents a preliminary study to analyse this issue. We observed that some dimensions have a larger impact on the performance than others, in particular, those encoding signals that could be derived from interactions (such as popularity) are good candidates to not be considered as contextual dimensions. No conclusive answer was obtained regarding the impact of pairs of contexts, as the ones analysed always reduced the performance, so further work is deserved to fully understand how and which contexts should be exploited by contextual MF neural models.

Acknowledgments. This work was supported by Grant PID2022-139131NB-I00 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF, a way of making Europe." We thank Fernando Huidobro for running the experiments.

References

1. Abdollahpouri, H., Burke, R., Mobasher, B.: Managing popularity bias in recommender systems with personalized re-ranking. In: Barták, R., Brawner, K.W. (eds.) Proceedings of the 32nd FLAIRS. pp. 413–418. AAAI Press (2019)
2. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P.: Towards a better understanding of context and context-awareness. In: Gellersen, H. (ed.) Proceedings of 1st HUC. Lecture Notes in Computer Science, vol. 1707, pp. 304–307. Springer (1999)
3. Adomavicius, G., Bauman, K., Tuzhilin, A., Unger, M.: Context-aware recommender systems: From foundations to recent developments. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 211–250. Springer US (2022)
4. Bauman, K., Tuzhilin, A.: Know thy context: Parsing contextual information from user reviews for recommendation purposes. Inf. Syst. Res. **33**(1), 179–202 (2022)
5. Campos, P.G., Díez, F., Cantador, I.: Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. User Model. User Adapt. Interact. **24**(1-2), 67–119 (2014)
6. Gunawardana, A., Shani, G., Yogev, S.: Evaluating recommender systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 547–601. Springer US (2022)
7. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: Deepfm: A factorization-machine based neural network for CTR prediction. In: Sierra, C. (ed.) Proceedings of the 26th IJCAI. pp. 1725–1731 (2017)
8. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.: Neural collaborative filtering. In: Barrett, R., Cummings, R., Agichtein, E., Gabrilovich, E. (eds.) Proceedings of the 26th WWW. pp. 173–182. ACM (2017)
9. Mateos, P., Bellogín, A.: A systematic literature review of recent advances on context-aware recommender systems. Artif. Intell. Rev. **58**(1), 20 (2025)
10. Unger, M., Tuzhilin, A., Livne, A.: Context-aware recommendations based on deep learning frameworks. ACM Trans. Manag. Inf. Syst. **11**(2), 8:1–8:15 (2020)
11. Zheng, Y.: DeepCARSKit: A demo and user guide. In: Proceedings of 30th UMAP. pp. 18–21. ACM (2022)
12. Zheng, Y., Arias, G.F.: A family of neural contextual matrix factorization models for context-aware recommendations. In: Proceedings of 30th UMAP. pp. 1–6. ACM (2022)