Pre-compiled Recommendation Lists for Online Recommendations

Brainstorming session report

Denis Krasilnikov[1,3], Oleg Lashinin[1], Marina Ananyeva[1,2] and Sergey Kolesnikov[1]

[1]Tinkoff, 2-Ya Khutorskaya Ulitsa, 38A, bld. 26, Moscow, 117198, Russian Federation
[2]National Research University Higher School of Economics, Myasnitskaya Ulitsa, 20, Moscow, 101000, Russian Federation
[3]Lomonosov Moscow State University, Ulitsa Kolmogorova, 1, bld. 52, Moscow, 119234, Russian Federation

Abstract

Recommendations in the online scenario play a crucial role. Algorithms that are sensitive to user actions in the online environment can improve the user experience and impact on key business metrics. Although some practitioners may face problems in integrating online methods due to many challenges. One of them is very limited number of available interactions from new users. Another is the need for speed in generating recommendations. In this paper we consider a way to overcome these limitations. The proposed method pre-calculates a certain number of recommendation lists from an offline model. These lists could cover different intersets of users. Once we have optimal precomputed recommendations, we can display them in an online scenario by matching users to the lists. We briefly discuss the motivation, idea, possible research questions and challenges of such an approach.

Keywords

recommender systems, online recommender systems, memory optimisation

CEUR-WS.org/Vol-3549/brainstorming1.pdf

1. Idea

Online recommendations need to quickly take into account user interactions and adapt the recommendation to the user's context. Some users may be new to the system and no historical information is available for this group of users. One of the most leveraged, simple and effective methods is to show the most popular items to new users. This heuristic works well because popular items are popular relative to a meaningful number of users.

However, this approach does not take into account the user's behaviour online. This could potentially be improved by some other lists of pre-computed recommendations. Formally, a list of most popular items is a precomputed recommendation that is suitable for some group of users. But we can extend the number of stored suggestions and try to sample them based on user's action in online scenario. This can be realised based on multi-armed bandits or contextual multi-armed bandits.

ORSUM@ACM RecSys 2023: 6th Workshop on Online Recommender Systems and User Modeling, jointly with the 17th ACM Conference on Recommender Systems, September 19th, 2023, Singapore
d.i.krasilnikov@tinkoff.ru (D. Krasilnikov); o.a.lashinin@tinkoff.ru (O. Lashinin); m.ananyeva@tinkoff.ru (M. Ananyeva); scitator@gmail.com (S. Kolesnikov)
0009-0009-6864-392X (D. Krasilnikov); 0000-0001-8894-9592 (O. Lashinin); 0000-0002-9885-2230 (M. Ananyeva); 0000-0002-4820-987X (S. Kolesnikov)
© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

Multi-armed bandits are used for recommendations. They sample some items and estimate the potential reward for each item. We can use them not to sample items but whole lists. If we use Multi-Armed Bandits without any user context, we can try to recommend different pre-computed lists for each user click. In this case, visitors will see different items. It is more likely to show them relevant content than just a list of the most popular items.

If we use context multi-armed bandits, we can take user context into account. For example, a user might fill out a registration form with relevant information, and context-aware bandits could take advantage of this new information. Additionally, practitioners can use user clicks as a form of context and improve the approach.

2. Precomputed recommendation lists

To realise the method, we need to create a list of different recommendation lists. This is a challenging task for many reasons.

Formally, we need to define an algorithm that generates such lists. It seems reasonable to use an offline recommendation model that can use different historical data. This model could even be a deep neural network, which would be difficult to use in an online scenario due to computational speed. Such an algorithm could generate recommendations for all users in historical data. Building such a model is a well-studied topic in the research community. The main challenge is to find a subset of lists that should be treated as "precomputed lists". We would like to consider the following research questions:

1. What metric should be used to estimate generated lists of lists? How can we measure the similarity between two selected recommendation lists?
2. What is the optimal algorithm to optimise the metric chosen in the first question?
3. What are the main hyperparameters of such an algorithm? Is it scalable to many users or items? What is the time and memory complexity?
4. Is it crucial to use some side information about items? Could we build an optimal approach using only a typical interaction matrix?

These questions could potentially formalise the proposed method and lead to a possible solution.

3. Use of precomputed lists

The second part of the discussion could be about the use of such lists. As mentioned above, we can use contextual multi-armed bandits. We propose to consider pre-computed lists as weapons. Some user information could be used as context in an online environment.

In addition, we highlight an opportunity to use such lists in an offline recommendation scenario. Instead of storing all lists or relevance scores for all users, we can sample some optimal predictions and save computational and storage costs. This effect could be interesting for small companies or start-ups that do not have enough computational resources to develop recommender systems. Surprisingly, a single algorithm for selecting a few samples of predictions could be used for both online and offline scenarios. As the workshop is dedicated to online scenarios, we suggest to discuss the following questions

1. What approaches could be used for sampling precomputed lists?
2. What type of context seems valuable for adopting recommendations in the online scenario?
3. What data sets, evaluation protocols and metrics could be used in offline experiments to assess the performance of the proposed method?
4. What are the possible limitations of offline evaluation in this case?

4. Related work

We found some related work. In it, the authors [1] generate semi-personalised lists for cold users. However, this paper does not consider an optimal choice of such lists. Another work [2] considers item filtering schemes, but it does not propose and idea for online recommendations.

5. Result of the discussion

During the discussion session, questions (1), (2) and (3) of section 2 were addressed. One of the simplest and easiest to understand metrics is the Rank-Biased Overlap (RBO) [3]. This measure evaluates not only the thoroughness of the catalogue, but also the sequence of elements. Initially, the metric introduced above can be optimised using a validation dataset and simple greedy methods. The main hyperparameter of this approach is the number of selected lists, which can be chosen based on memory constraints. However, this approach does not scale effectively in terms of the number of users, but does scale well in terms of items. The implementation of side information has not been considered.

With respect to the topics covered in section 3, a potential method for precomputed listings is to use greedy sampling to maximise the coverage metric of the original recommendations introduced above. In scenarios where evaluations are conducted offline, standard ranking metrics and time splitting can be used. Overall, it was concluded that there are no new limitations in the offline scenario.

References

[1] L. Briand, G. Salha-Galvan, W. Bendada, M. Morlon, V.-A. Tran, A semi-personalized system for user cold start recommendation on music streaming apps, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 2601–2609.

[2] A. Makmal, J. Ephrath, H. Berezin, L. Allerhand, N. Nice, N. Koenigstein, Pick & merge: an efficient item filtering scheme for windows store recommendations, in: Proceedings of the 13th ACM Conference on Recommender Systems, 2019, pp. 472–476.

[3] W. Webber, A. Moffat, J. Zobel, A similarity measure for indefinite rankings, ACM Transactions on Information Systems (TOIS) 28 (2010) 1–38.