Periodic Reranking Strategy in Unsupervised Recommender Systems*

Rémi Delanghe[1][0000−0002−6621−7291], Pieterjan Criel[2][0000−0002−1038−2642], and Tom Dhaene[3][0000−0003−2899−4636]

Ghent University - imec, IDLab, Technologiepark-Zwijnaarde 15, Ghent 9052, Belgium `Remi.Delanghe@UGent.be` - www.ugent.be/ea/idlab/en

Abstract. It is challenging for recommender systems to generate accurate and responsive rankings with respect to changes in user behavior and updates in the recommended content. However, updating the system with every user interaction comes at a high economic cost, especially if the system is cloud-hosted. In this paper a reranking strategy is presented that can be used to train a recommender system after collecting a batch of user interactions while still being accurate and responsive.

Keywords: Recommender System · Ranking · Ground truth generation

1 Introduction

The goal of a recommender system (RS) is to provide a ranked list of items according to the probability of users interacting with these items. The challenge is that this probability distribution is unknown and variable over time, due to changes in user behavior and updates to the ranked items.

The proposed strategy is able to rerank the result of the RS to accommodate recent user behavior using a batch of user interactions. This reranked list can then be used in combination with an optimization technique to retrain the RS. No algorithms yielding the same result were found in litterature.

2 Reranking Strategy

Algorithm 1: Reranking Strategy

Data:
- Initial ranking **A**: List of ranked items. Contains the *ID* and *score* for every item.
- User Interactions **U**: List of user interactions. Each element contains an *ID* from **A**.
- Redistribution factor $\boldsymbol{\alpha} \in [0, 1]$ and Redistribution range $\mathbf{rr} \in 2\mathbb{N}^+$.

```
begin
    forall u ∈ U do
        r ← rank of u[ID] in A
        for i ← (r − rr/2) to (r + rr/2) do
            if i not out of bounds then
                A(r)[score] ← A(i)[score]×α
                A(i)[score] ← A(i)[score]×(1 − α)
        Order(A, descending)
    return A
```

* The research was conducted at, and supported by Showpad.

An overview of the pseudocode for the reranking strategy is provided by **Algorithm 1**. The RS produces an initial ranking based on its current perception of the user behavior. The strategy uses chronologically ordered user interactions to iteratively redistribute the score from items with a low click-rate to items with a high click-rate.

The *redistribution factor* and the *redistribution range* are parameters that influence the behaviour of the reranking strategy by respectively changing the rate at which the score is redistributed and the amount of items that are involved in this redistribution per iteration. These parameters should be optimized for each use case, based on the amount of user interactions and the desired aggressiveness of the strategy. To evaluate the strategy the NDCG [1] ranking metric is used.

3 Results

In **figure 1** an example of the strategy with 750 artificially generated user interactions is shown. At every 150th iteration the user behavior is drastically changed. The strategy consistently evolves towards the artificial ground truth. Similar results were found during the prototyping of this strategy in the Showpad data-science team.

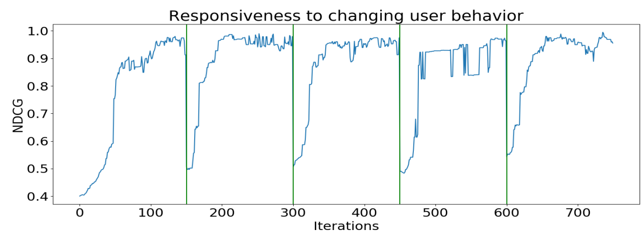

Fig. 1: Example of reranking strategy: Similarity (NDCG) per iteration between the reranked list and the artificial ground truth. *length list* $= 26$, $rr = 6$, $\alpha = 0.1$

.

4 Conclusion

The proposed strategy is able to quickly and accurately adapt to recent changes in user behavior, thus enabling their inclusion in the training of the RS. The strategy has been tested to train a RS on the Showpad platform with promising results. Future developments include differentiating between multiple types of user interactions.

References

1. Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Tie-Yan Liu, Wei Chen: A Theoretical Analysis of NDCG Type Ranking Measures. CoRR (2013)