Challengers' Solution for Debiasing

Shunyao Wu
Chuanyu Xue
wushunyao@qdu.edu.cn
chuanyu.xue@uconn.edu
Qingdao University
Qingdao, Shandong, China

Shouhua Liu
liushouhua@ime.ac.cn
University of Chinese Achemy of Sciences
Beijing, China

Mingyuan Cheng
Chuwen Huang
mundane0827@gmail.com
462719284@qq.com
Renmin University of China
Beijing, China

Chengbin Huang
51184501017@stu.ecnu.edu.cn
National Trusted Embedded Software Engineering Technology Research Center; Shanghai Key Laboratory of Trustworthy Computing, East China Normal University
Shanghai, China

Shihuan Liu
laresh007@163.com
HelloBike
Shanghai, China

Xin Chen
Qing Li
18150135990@163.com
961213106@qq.com
Netease
Guangzhou, Guangdong, China

Huan Liu
942833674@qq.com
JD Group
Beijing, China

ABSTRACT

This article describes the final solution of Team Challengers, who finished in 6^{th} place in the KDD CUP 2020 Challenge for Debiasing sponsored by KDD 2020, Alibaba Group, Duke University, Tsinghua University and University of Illinois. In this paper, we provide a framework to recommend items for target users based on historical clicks, user information and item feature vectors. Some interesting methods are attempted, such as making full use of further clicks for candidate generation, caculating second-order proximities in item similarity network, distribution-free independence test for feature selection and so on, which may be potential for recommendation system.

KEYWORDS

Recommendation System, Network Representation, Debiasing

1 INTRODUCTION

Most e-commerce and retail companies achieve their targets by leveraging the power of data and boosting sales by implementing search and recommender systems on their websites. With the above mentioned new upcoming display trends and huge increment of the online traffic volume, two fundamental challenges have to be solved. First, we need to design a model framework to facilitate an effective semantic understanding, search, and retrieval of images and videos, so that we can recall and rank products with their appropriate multimodal contents. This can help consumers make better decisions as well as delve more into the platforms with great potential. Second, machine learning systems are prone to exploitation if equipped with short-term goals, e.g., ctr, cvr or gmv, and the direct effects would cause severe Matthew's effects where only very small proportion of items from different sellers get exposed to online users. Thus, it is crucial to understand whether the systems are fair, which has deep influence on the development of the e-commerce platforms.

The KDD CUP 2020 Challenge for Debiasing focuses on the fariness of exposure, i.e., how to recommend items that are rarely exposed in the past, to combat the Matthew effect frequently encountered in a recommender system. In particular, performing bias reduction when training on the click data is crucial for the success of this task. Just like there is a gap between the logged click data and the actual online environment in a modern recommender system, there will be a gap between the training data and the test data, mostly with respect to the trends and the items' popularity. The Normalized Discounted cumulative gain (NDCG) is used to evaluate the recommendation performance in the contest. NDCG@50-full of the winning teams need to be among the top 10% while achieving the best NDCG@50-half among the qualified teams. Here, NDCG@50-full is computed on the whole test set while NDCG@50-half is computed on half of the test cases whose next items to be predicted are less explored than the other half in the past training sets.

2 SOLUTION

2.1 Framework

The framework of the current solution is shown in Figure 1. In the recalling stage, an variant of item-CF method is developed to retrieve 1,000 items for each user, which incorporates item-CF similarity, node2vec similarity [1], deepwalk similarity [2] and text similarity. Secondly, item similarity network is constructed to obtain second-order proximities and the self-attention based sequential model [3] is constructed, which are prepared for feature engineering. Then, positive samples and negative samples are selected from the candidate item list, and a number of features are extracted for the ranking stage. Finally, gradient boosting tree methods and model averaging methods are adopted to re-rank the candidate items.

2.2 Candidate Generation

Item CF is one of the most typical and effective method for candidate generation, which caculates item similarities from large-scale historical user activity logs. Therefore, we develope an variant of item CF for this track. Firstly, for phrase T, we put the historical clicks of phrase $T + 1 \sim$ phrase 9 in front of phrase 0. It can make full use of user activity logs and achieve much better performance, while make sure recent clicks still play a vital role for phrase T. Secondly, user activity, item popularity, click sequence and time difference are taken into consideration. Thirdly, node2vec similarity, deepwalk similarity and text similarity are introduced to compute item similarites. A simple multiplication form is adopted to integrate the four similarities, which makes the integrated similarities too small. Unfortunately, we truncate the similarties to save memory and hard disk storge in Track B, which decreases the performance to some extent.

2.3 Second-order Proximities in Item Similarity Network

Item similarity network is constructed with recall similarities so as to uncover high-order proximities between items. Six local similarity indicies [4, 5], including Common Neighbors, Hub Promoted Index, Hub Depressed Index, Leicht-Holme-Newman Index, Adamic-Adar and Resource Allocation are adopted to obtain second-order proximities. They can be utilized for candidate generation, which has been proven effective in Track A. However, we only use them to generate features for re-ranking in Track B so as to save computing resources. Additionally, quasi-Local Indices, such as Local Random Walk and Local Path, would be potential for recommendation. We think these proximities may be simlar with some operations of graph neural networks, but have obvious physical meaning.

As shown in Figure 2, item x and item y are unconnected but trend to be established a connection since they share many common neighbors. Take Adamic-Adar as as an example, which is defined as

$$s_{xy}^{AA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{w_{xz} * w_{zy}}{log(1 + s_z)}.$$

Here, s_z is the strength of item z, w_{xz} is the link weight bewteen x and z, and $\Gamma(x)$ is the neighbor set of x. It mainly considers that hub common neighbors have little influence on link formation in some cases. For example, opinion leaders on Facebook do not have time to help their fans become friends. Similarily, the common neighbor z_2 has greater effects on the simialrity between x and y. Additionally, Hub Promoted Index can effectively increase the importance of items with fewer clicks, which is defined as

$$s_{xy}^{HPI} = \frac{\sum_{z \in \Gamma(x) \cap \Gamma(y)} w_{xz} * w_{zy}}{min(s(x), s(y))}.$$

2.4 Self-Attentive Sequnetial Recommendation

Self-attentive sequential model is implemented mainly according to Kang et al's work [3]. Specially, item embedding matrix is initialized through dividing 256-dimensional item features with 25. Additionally, prediction layer is defined as

$$r_{i,t} = (F_t + u)M_i^t.$$

Here, u is the embedding feature of a given user; $r_{i,t}$ is the probability that the user will click the item i in timestamp t; F is the output of Feed-Forwad Network; M_i^t is the embedding feature of the item i in timestamp t. Since the recalling performance of the self-attentive sequential model is not good, we only utilize the probability as a similarity feature.

2.5 Feature Engineering

Re-ranking is transformed into a binary classifier problem. Positive samples are obtained from the candidate item list, while negative samples are randomly selected with 5 times the number of positive samples. In our solution, 6 datasets are generated for robustness, and 68 features are extracted in the feature engineering stage. They can be divided into four categories, including similarity features, item count features, time-dependent features and interactive features.

2.5.1 Similarity features. Based on different kinds of similarities generated by item CF, second-order proximities, image/text similarities, deepwalk, node2vec and the self-attentive sequential model, 23 similarity features are extracted by calculating the statistics of similarities between items in user's historical list and items in candidate list.

2.5.2 Item count features. To represent the dynamic change of item popularity, 18 item count features are generated by counting and ranking the number of clicks on items in a special phase or certain time interval.

2.5.3 Time-dependent features. The time-dependent features are generated to capture the temporal characteristics of items. 11 features are obtained by aggregating item-count features and similarity features within time frame in a specific length.

2.5.4 Interactive features. The number of clicks on items are deemed as item category, and a few of interactive features between users and the number of clicks are generated in a special phase.

Additionally, mean variance index [6, 7] is employed to eleminate useless features, and finally 58 features are selected for re-ranking. Mean variance index is computed from a distribution-free independent test, which can analyze the independence between a categorical variable and a numerical variable. It is different from feature importances obtained through gradient boosting tree methods, and can probably avoid the homogeneity between the modeling method and the feature selection method.

2.6 Re-Rank

Catboost and Lightgbm are adopted for re-ranking, and large weights are set for the samples whose item has fewer clicks so as to improve the performance of NDCG@50-half. Since each dataset only have about 20,000 samples, it costs little time to train 12 models, including 6 catboost models and 6 lightgbm models. Finally, arithmetic mean, harmonic mean and geometric mean are adopted for model averaging, and postprocess is implemented through multiplying predicted probabilities with $\frac{1}{ln(count(i)+1)}$. Here, $count(i)$ represents the click count of item i.

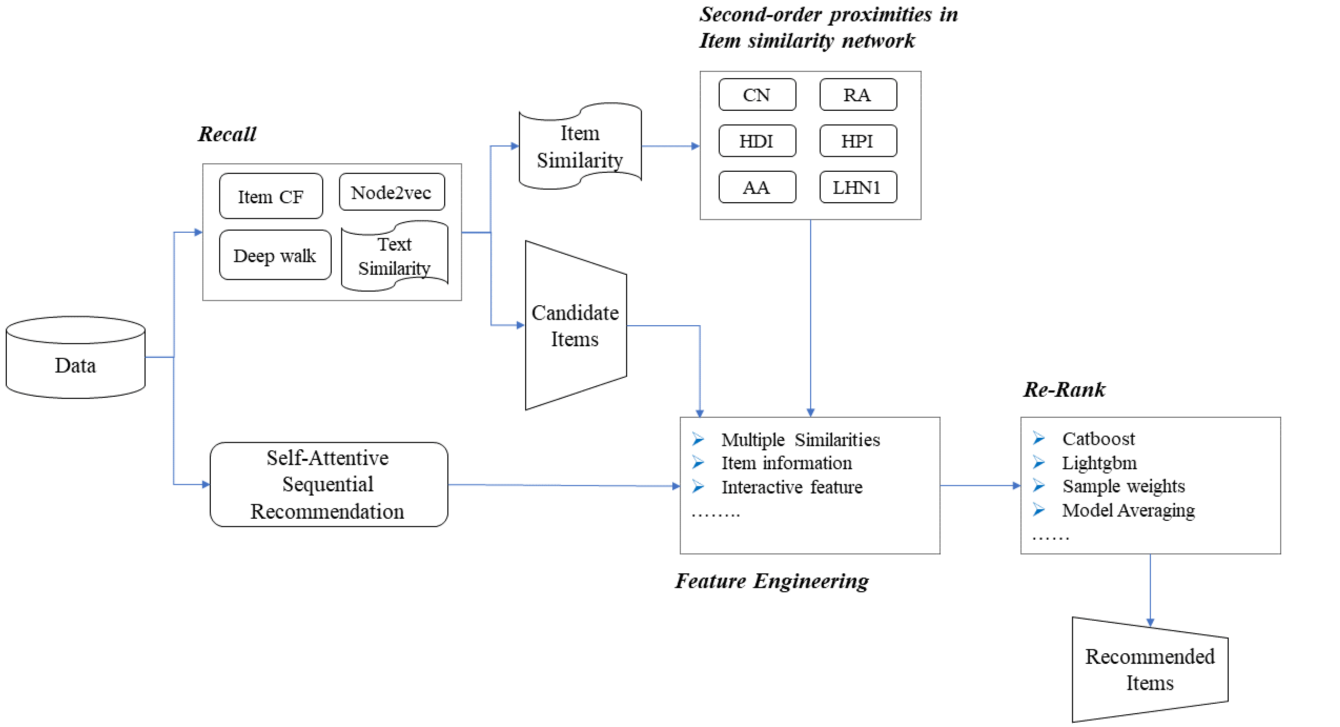

Figure 1: Flow char of the current solution

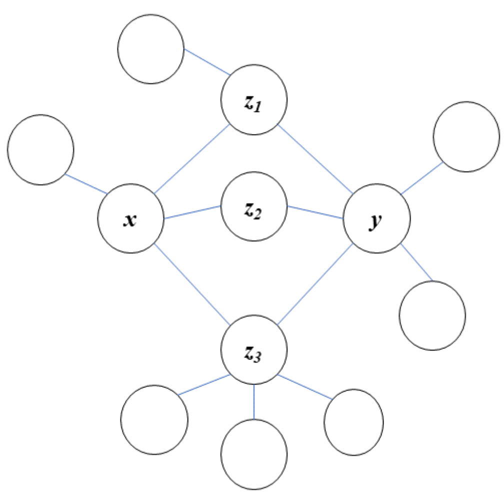

Figure 2: A demo for item similarity network

3 CONCLUSIONS

In the present study, feasible solution method for the debiasing recommendation mission is provided. First, we develop an variant of item CF to generate candidate item list. Afterwards, we caculate second-order proximities in item similarity network, and train the self-attentive sequential model with historical clicks. Then, a number of features are extracted in the feature engineering stage, and a novel feature selection method is adopted to eliminate useless features. Finally, gradient boosting tree methods and model averaging methods are employed to re-rank candidate items. Our results rank the 6^{th} place in the final evaluation of the KDD Cup Challenge for Debiasing.

ACKNOWLEDGMENTS

The current research focused on the KDD Cup 2020, supported by KDD 2020, Alibaba Group, Duke University, Tsinghua University and University of Illinois.

REFERENCES

[1] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.
[2] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.
[3] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206. IEEE, 2018.
[4] Jing Zhao, Lili Miao, Jian Yang, Haiyang Fang, Qian-Ming Zhang, Min Nie, Petter Holme, and Tao Zhou. Prediction of links and weights in networks by reliable routes. *Scientific reports*, 5:12261, 2015.
[5] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.
[6] Hengjian Cui, Runze Li, and Wei Zhong. Model-free feature screening for ultrahigh dimensional discriminant analysis. *Journal of the American Statistical Association*, 110(510):630–641, 2015.

[7] Hengjian Cui and Wei Zhong. A distribution-free test of independence based on mean variance index. *Computational Statistics & Data Analysis*, 139:117–133, 2019.