RUCIR21 at the NTCIR-16 ULTRE Task

Zechun Niu[1], Yurou Zhao[2,3], Feng Wang[4], Jiaxin Mao[2,3]
[1]Wuhan University
[2]Beijing Key Laboratory of Big Data Management and Analysis Methods
[3]Gaoling School of Artificial Intelligence, Renmin University of China
[4]University of Electronic Science and Technology of China
China
2018300003042@whu.edu.cn, maojiaxin@gmail.com

ABSTRACT

The RUCIR21 team participated in both the offline and online subtasks of the NTCIR-16 Unbiased Learning to Rank Evaluation (ULTRE) task. This paper describes our approaches and reports the results in the ULTRE task. In the offline subtask, we tried four learning to rank models based on Mobile Click Model (MCM), as well as a revived Dual Learning Algorithm (DLA) model. In the online subtask, we revived a Pairwise Differentiable Gradient Descent (PDGD) run and two online DLA runs, we also tried an online DLA model based on MCM.

KEYWORDS

unbiased learning to rank, MCM, offline ULTR, online ULTR

TEAM NAME

RUCIR21

SUBTASKS

offline, online

1 INTRODUCTION

It has been popular to train a ranking model with users' implicit relevance feedback such as their clicks. However, user clicks are noisy and heavily biased [6]. Therefore, many Unbiased Learning to Rank (ULTR) methods have been proposed to debias click data and train an unbiased ranking model. Existing researches on ULTR can be broadly categorized into two groups: offline ULTR [4][14] that trains an unbiased ranking model with logged click data, and online ULTR [15][5] which makes online interventions of the ranking lists and trains an unbiased ranking model with online user interactions.

Unbiased Learning to Rank Evaluation (ULTRE) [16] is a pilot task in NTCIR-16, which aims to evaluate and compare different Unbiased Learning to Rank (ULTR) approaches with a shared benchmark. In the task, provided with initial ranking lists and query-document features, as well as 5 different datasets of simulated user clicks, we need to train a feature-based ranking model on each dataset and use it to re-rank the ranking lists of the test queries.

In the NTCIR-16 ULTRE task, we participated in both the offline and online subtasks. This paper reports our approaches to solving the problem and discusses the official results. In the offline subtask, we tried four learning to rank models based on MCM [9], as well as a revived DLA [1] model. The results show that MCM serves as an effective propensity model for ULTR, especially when the mixed click data follows a variety of user behavior models. In the online subtask, we revived a PDGD [10] run and two online DLA runs, we also implemented an online DLA model based on MCM. We also find that instead of only using the clicks collected in each iteration, using all the click data collected so far leads to a better performance for the online DLA model.

2 OFFLINE SUBTASK

In the ULTRE offline subtask, we submitted four runs based on MCM and a revived DLA run. We will introduce the details about our offline runs and discuss the official results in this section.

2.1 MCM

The ULTRE task utilized four different user simulation models including the Position-Based Model (PBM) [11], Dependent Click Model (DCM) [3], User Browsing Model (UBM) [2], and Mobile Click Model (MCM), to generate synthetic user clicks. In addition, the synthetic click logs generated by these four models are further combined to generate the FUSION dataset.

As the most sophisticated model among these four user simulation models, MCM comprehensively considers the position bias and the influence of previous clicks on the examination probability by generalizing the assumptions of PBM and UBM. It also incorporates post-observation satisfaction and post-click satisfaction into the model, which to some extent models the cascade behavior defined by DCM. Therefore, we hypothesize that MCM can effectively capture and model the biases in all five kinds of click data in the ULTRE task.

We first directly use MCM to estimate the relevance of training documents. Note that the ULTRE task does not provide the vertical type information for the documents, so we modify MCM by setting the click necessity parameter to 1.0 and assuming that a user can only be satisfied by clicking a document. We utilize EM algorithm to learn the parameters of MCM from the click data, and obtain a relevance estimation for each query-document pair through multiplying the learned attractiveness parameter and satisfaction parameter. Then, we use the relevance estimation as a supervision signal to train two different ranking models, LambdaMART and deep neural networks (DNN),

Table 1: ULTRE offline subtask official results of RUCIR21 1-5 runs

Run	Model	PBM	DCM	UBM	MCM	FUSION	AVG
1	MCM (LambdaMART)	0.7822	0.7872	0.7803	0.7206	0.7865	0.7714
2	MCM (DNN)	0.7705	0.7765	0.7807	0.7834	0.7846	0.7791
3	MCM-IPS	0.7969	0.8006	0.7746	0.7778	**0.8102**	**0.7920**
4	Simplified MCM-IPS	**0.7997**	0.7933	0.7735	0.7724	0.8007	0.7879
5	DLA	0.7820	**0.8019**	**0.7875**	**0.7866**	0.7806	0.7877

respectively. The results are shown in Table 1 as run 1 and run 2.

2.2 MCM-IPS

Wang et al. [13] and Joachims et al. [8] introduced inverse propensity scoring (IPS) method to debias user clicks and train unbiased ranking models. However, most IPS methods are based on the assumption of PBM, that the click probability only depends on the ranking position and relevance of document. As this assumption does not hold for different kinds of simulated click data, these IPS methods may not be optimal in the ULTRE task.

Therefore, we try to use MCM as an alternative to PBM and propose a mobile click model-based inverse propensity score (MCM-IPS) method. We use a variable E_i to denote whether the user examined the document d_i, a variable C_i to denote whether the user clicked the document d_i, and a variable S_i to denote user's state of satisfaction after position i. Here, E_i, C_i, and S_i are binary variables. The click propensity can be computed as:

$$P(E_i = 1|C_{1..i-1}) = P(E_i = 1|S_{i-1} = 0) * P(S_{i-1} = 0|C_{1..i-1}) \quad (1)$$

$P(E_i = 1|S_{i-1} = 0)$ is learned through EM algorithm from the click data, and $P(S_{i-1} = 0|C_{1..i-1})$ is obtained through the forward algorithm of Hidden Markov Model (HMM).

In addition, following Vardasbi et al. [12], we propose a simplified MCM-IPS method. The click propensity is obtained by:

$$P(E_i = 1|C_{1..i-1}) = P(E_i = 1|S_{i-1} = 0) * P\left(S_{last\_click} = 0\middle|C_{last\_click} = 1\right) \quad (2)$$

We use deep neural networks as the ranking model, as we do in run 2. The results of MCM-IPS and the simplified version are shown in Table 1 as run 3 and run 4, respectively.

2.3 DLA

The revived model is based on Dual Learning Algorithm (DLA), which jointly learns an unbiased ranking model and an unbiased propensity model. Following Ai et al. [1], we utilize PBM as the propensity model, and implement the ranking model with deep neural networks. The result is shown in Table 1 as run 5.

2.4 Results

Table 1 shows the individual and average NDCG@5 performance of our five runs submitted for the offline subtask on all five click datasets. In general, the two MCM-IPS runs perform better than the two MCM runs, which shows that inverse propensity score (i.e. the examination probability estimated by MCM) is more reliable than the relevance estimation of the same click model, MCM.

The MCM-IPS run achieves the best performance on the FUSION dataset and on average over the five datasets. The revived DLA model, which is an effective benchmark for unbiased learning to rank, performs slightly worse than MCM-IPS on average, and significantly worse on FUSION. It illustrates that based on a more sophisticated and general click model, MCM, the MCM-IPS method is more effective when the collected user clicks follow different user behavior models.

3 ONLINE SUBTASK

In the ULTRE online subtask, we submitted one revived PDGD run, two revived online DLA runs, and one online DLA run based on MCM. We will introduce the details about our online runs and discuss the official results in this section.

3.1 PDGD

The revived model is based on Pairwise Differentiable Gradient Descent (PDGD) [10], which constructs a weighted differentiable pairwise loss after each interaction. We implement the ranking model with deep neural networks. The result is shown in Table 2 as run 1.

3.2 ODLA-PBM

The revived model is based on Online Dual Learning Algorithm (ODLA), which utilizes Plackett-Luce (PL) model to make online interventions and train a DLA model using the clicks collected in each iteration. Following Ai et al. [1], we utilize PBM as the propensity model, and implement the ranking model with deep neural networks. The result is shown in Table 2 as run 2.

In addition, we revive the same ODLA model with a different training strategy. We iteratively submit the ranking lists of train queries via the API provided by the ULTRE task organizers to get simulated clicks, and use all the click data collected so far to update the ODLA model. The result is shown in Table 2 as run 3.

3.3 ODLA-MCM

As we mentioned in section 2.2, the assumption of PBM does not hold for different kinds of simulated click data in the ULTRE task.

Table 2: ULTRE online subtask official results of RUCIR21 1-4 runs

Run	Model	PBM	DCM	UBM	MCM	FUSION	AVG
1	PDGD	0.7316	0.7694	0.7897	0.7622	0.7944	0.7695
2	ODLA-PBM	**0.7956**	0.7858	0.7997	0.7731	0.7904	0.7889
3	ODLA-PBM2	0.7882	0.7900	**0.8082**	**0.7952**	**0.8037**	**0.7971**
4	ODLA-MCM	0.7922	**0.8069**	0.7948	0.7546	0.7806	0.7858

Therefore, we try to use MCM as an alternative to PBM and propose an ODLA model based on MCM (ODLA-MCM). We implement the ranking model with deep neural networks, and use a separate variable $\gamma_{i,d}$ to represent $P(E_i = 1|S_{i-1} = 0)$ for each position i with a distance d to the last clicked document before position i. The attractiveness parameter and satisfaction parameter for each query-document pair are outputted by the ranking model, and the click propensity is computed with Equation (1) in section 2.2. We train the ODLA-MCM model in the same way as we do in run 3. The result is shown in Table 2 as run 4.

3.4 Results

Table 2 shows the individual and average NDCG@5 performance of our four runs submitted for the online subtask on all five click datasets. Run 3 achieves better performance than run 2 on the DCM, UBM, MCM, and FUSION dataset, as well as on average over the five datasets. This illustrates that instead of only using the clicks collected in each iteration, using all the clicks collected so far leads to a better performance for the online DLA model.

The ODLA-MCM model performs best on the DCM dataset, but performs worse than ODLA-PBM on the other four datasets and on average. For dual learning algorithm, the performance of propensity estimation depends on the quality of the ranking model. Therefore, we conjecture that MCM, as a propensity model, is too complicated so that the propensity estimated in the dual learning process is not very reliable.

4 CONCLUSIONS

In the NTCIR-16 ULTRE task, the RUCIR21 team participated in both the offline and online subtasks. We tried MCM-based models in both the offline and online subtasks, we also revived some ULTR benchmarks of high performance. The results show that in the offline task, MCM serves as an effective propensity model for ULTR, especially when the mixed click data follows a variety of user behavior models. In addition, we find that in the online task, instead of only using the clicks collected in each iteration, using all the click data collected so far leads to a better performance for the online DLA model. In the future, we would like to investigate how to estimate the parameters of MCM more accurately, and how to better capture the biases in mixed click data that may follow different user behavior models.

REFERENCES

[1] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018b. Unbiased learning to rank with unbiased propensity estimation. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 385–394.

[2] Georges E. Dupret and Benjamin Piwowarski. 2008. A User Browsing Model to Predict Search Engine Click Data from Past Observations. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, 331–338.

[3] Fan Guo, Chao Liu, and Yi Min Wang. 2009. Efficient multiple-click models in web search. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*. ACM, 124–131.

[4] Ziniu Hu, Yang Wang, Qu Peng, and Hang Li. 2019. Unbiased LambdaMART: An Unbiased Pairwise Learning-to-Rank Algorithm. In *Proceedings of the 28th World Wide Web Conference*. ACM, 2830–2836.

[5] Yiling Jia, Huazheng Wang, Stephen Guo, and Hongning Wang. 2021. PairRank: Online Pairwise Learning to Rank by Divide-and-Conquer. In *Proceedings of the 30th World Wide Web Conference*. ACM, 146–157.

[6] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. 2005. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual ACM SIGIR*. ACM, 154–161.

[7] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems* 25, 2 (2007), 7.

[8] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased learning-to-rank with biased feedback. In *Proceedings of the 10th ACM WSDM*. ACM, 781–789.

[9] Jiaxin Mao, Cheng Luo, Min Zhang, and Shaoping Ma. 2018. Constructing Click Models for Mobile Search. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 775–784.

[10] Harrie Oosterhuis, and Maarten de Rijke. 2018. Differentiable Unbiased Online Learning to Rank. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1293–1302.

[11] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting Clicks: Estimating the Click-Through Rate for New Ads. In *Proceedings of the 16th International Conference on World Wide Web (WWW)*. ACM, 521–530.

[12] Ali Vardasbi, Maarten de Rijke, and Ilya Markov. 2020. Cascade Model-based Propensity Estimation for Counterfactual Learning to Rank. *In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2089–2092.

[13] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In *Proceedings of the 39th ACM SIGIR*. ACM, 115–124.

[14] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position Bias Estimation for Unbiased Learning to Rank in Personal Search. In *Proceedings of the 11th ACM WSDM*(WSDM '18). ACM, New York, NY, USA, 610–618.

[15] Yisong Yue, and Thorsten Joachims. 2009. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 1201–1208.

[16] Yurou Zhao, Zechun Niu, Feng Wang, Jiaxin Mao, Qingyao Ai, Tao Yang, Junqi Zhang, and Yiqun Liu. 2022. Overview of the NTCIR-16 Unbiased Learning to Rank Evaluation (ULTRE) Task. In *Proceedings of the NTCIR-16 Conference on Evaluation of Information Access Technologies*.