Non-stationary Dueling Bandits for Online Learning to Rank

Shiyin Lu[1], Yuan Miao[2], Ping Yang[2], Yao Hu[2], and Lijun Zhang[1(✉)]

[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
{lusy,zhanglj}@lamda.nju.edu.cn
[2] Alibaba Group, Hangzhou 311121, China
{miaoyuan.my,yangping.yangping,yaoohu}@alibaba-inc.com

Abstract. We study online learning to rank (OL2R), where a parameterized ranking model is optimized based on sequential feedback from users. A natural and popular approach for OL2R is to formulate it as a multi-armed dueling bandits problem, where each arm corresponds to a ranker, i.e., the ranking model with a specific parameter configuration. While the dueling bandits and its application to OL2R have been extensively studied in the literature, existing works focus on static environments where the preference order over rankers is assumed to be stationary. However, this assumption is often violated in real-world OL2R applications as user preference typically changes with time and so does the optimal ranker. To address this problem, we propose non-stationary dueling bandits where the preference order over rankers is modeled by a time-variant function. We develop an efficient and adaptive method for non-stationary dueling bandits with strong theoretical guarantees. The main idea of our method is to run multiple dueling bandits gradient descent (DBGD) algorithms with different step sizes in parallel and employ a meta algorithm to dynamically combine these DBGD algorithms according to their real-time performance. With straightforward extensions, our method can also apply to existing DBGD-type algorithms.

Keywords: Online learning to rank · Dueling bandits · Non-stationary environments

1 Introduction

As a powerful ranking optimization paradigm, learning to rank has found applications in a variety of information retrieval scenarios such as web search, online advertising, and recommendation systems [7,15]. In the classical offline learning to rank, a parameterized ranking model is first trained on collected queries and documents with relevance labels, and then deployed to respond to users' queries with predicted relevant documents. A drawback of offline learning to rank is that the process of collecting training data with relevance labels is highly time-consuming and expensive in large-scale applications [4]. Furthermore, as

B. Li et al. (Eds.): APWeb-WAIM 2022, LNCS 13422, pp. 166–174, 2023.
https://doi.org/10.1007/978-3-031-25198-6_13

the ranking model is fixed after being deployed, it cannot track the evolution of user needs [6].

To address these issues, recent advances in information retrieval have introduced online learning to rank (OL2R), where the ranking model is optimized based on its interactions with users on the fly [3]. Compared to its offline counterpart, OL2R has lighter computational overhead and higher updating frequency. At the heart of OL2R lies the trade-off between exploring new rankers and exploiting the seemingly optimal ranker. Thus, a natural and popular approach for OL2R is to formulate it as a dueling bandits problem [13,14], where each ranker is viewed as an arm and the ranking model is optimized through sequential noisy comparisons between rankers. While the dueling bandits based methods have been widely studied for OL2R, they are limited in that the preference order over rankers is assumed to follow stationary probability distributions. However, in real-world scenarios, user preference typically changes with time, making the stationary assumption invalid.

To better cope with real-world ranking tasks, we investigate dueling bandits with non-stationary preference probability distributions for OL2R. Specifically, let $\mathbf{w}$ and $\mathbf{w}'$ be two points in the parameter space of the ranking model. We model the probability that users prefer the ranking results produced by a ranker with parameter $\mathbf{w}$ over those of a ranker with parameter $\mathbf{w}'$ by a composite function $f_t(\mathbf{w}, \mathbf{w}') = \sigma(v_t(\mathbf{w}) - v_t(\mathbf{w}'))$, where σ is a static link function, and v_t denotes the utility function in round t. Compared to the existing works on dueling bandits, the novelty of our model is that the utility function can change with time t, capturing the non-stationarity of user preference. Since v_t and $v_{t'}$ can be different for $t \neq t'$, the optimal parameter $\mathbf{w}_t^*$ that maximizes v_t and hence the optimal ranker can change with time, making the non-stationary dueling bandits much harder to deal with than its stationary counterpart.

Nevertheless, by drawing inspiration from recent progress in dynamic online optimization [16,17], we develop an efficient and adaptive method for non-stationary dueling bandits. Our method follows the prediction with expert advice framework [1] and has a two layer hierarchical structure: multiple dueling bandits gradient descent (DBGD) [14] algorithms running parallel in the bottom and a meta algorithm aggregating the outputs of DBGDs in the top. Generally speaking, DBGDs aim at balancing the exploration-exploitation tradeoff, which also exists in the classical stationary dueling bandits, and the meta algorithm is responsible for tracking the change of utility functions, which is a new task arising only in our non-stationary setting. Under mild assumptions, we prove that our method guarantees no-regret learning, indicating that when the number of rounds goes infinity, the average performance of our method is the same as that of a clairvoyant who knows the optimal ranker in each round. Furthermore, we show that our method, while developed in the context of DBGD, can be also straightforwardly extended to existing variants of DBGD. Finally, we conduct extensive experiments on public datasets to demonstrate the effectiveness and efficiency of our method for OL2R in non-stationary environments.[1]

[1] Due to space limitation, proofs and experiments are postponed to the full version of this paper: www.lamda.nju.edu.cn/lusy/ns-ol2r.pdf.

2 Problem Setup

We study non-stationary dueling bandits for online learning to rank, which proceeds in a sequence of rounds. Let $\mathcal{W} \subseteq \mathbb{R}^d$ be the parameter space of a ranking model and T be the number of rounds. Following previous work [8,11,12], we refer to the ranking model with a specific parameter configuration as a ranker. In each round $t \in [T] = \{1, \ldots, T\}$, firstly a learner chooses two rankers with parameters $\mathbf{w}_t \in \mathcal{W}$ and $\mathbf{w}'_t \in \mathcal{W}$, respectively. Then, the ranking lists produced by the rankers are merged by an interleaving method [5,9]. The merged list is displayed to a user and a noisy preference order over the rankers is inferred from the user's click feedback. Specifically, the ranker whose ranking list receives more clicks is preferred. Finally, the learner updates the parameter of the ranking model based on the inferred preference order.

We denote by $\mathbf{w} \succ \mathbf{w}'$ the event that users prefer the ranking list produced by the ranker $\mathbf{w}$ than that of the ranker $\mathbf{w}'$. While the existing works only consider the setting where the probability of this event is fixed, we allow the probability to change with time so as to capture the non-stationary nature of user preference. Specifically, in round t, the probability of the event $\mathbf{w} \succ \mathbf{w}'$ is defined as

$$\Pr(\mathbf{w} \succ \mathbf{w}'|t) = f_t(\mathbf{w}, \mathbf{w}') = \sigma(v_t(\mathbf{w}) - v_t(\mathbf{w}')) \tag{1}$$

where σ is a static link function, and v_t denotes the utility function in round t. Following previous work [11,14], we make some standard assumptions as follows:

- The parameter space of the ranking model $\mathcal{W}$ is bounded

$$\max_{\mathbf{w} \in \mathcal{W}} \|\mathbf{w}\|_2 \leq R. \tag{2}$$

- The link function σ is rotation-symmetric

$$\sigma(x) = 1 - \sigma(-x). \tag{3}$$

- The link function σ is monotonically increasing and satisfies

$$\sigma(-\infty) = 0, \quad \sigma(0) = 1/2, \quad \sigma(\infty) = 1.$$

- The link function σ is L_σ-Lipschitz, and all utility functions $v_t, t \in [T]$ are L_v-Lipschitz. Furthermore, the link function σ is also second order L_2-Lipschitz.[2]

Denoting $L = L_\sigma L_v$, the above assumptions directly imply the functions $f_t, t \in [T]$ are L-Lipschitz in both arguments.

Let $\mathbf{w}_t^* = \arg\max_{\mathbf{w} \in \mathcal{W}} v_t(\mathbf{w})$ denote the optimal ranker achieving the maximum utility in round t. We adopt dynamic regret as performance metric, defined as

$$\mathrm{DR}(T) = \sum_{t=1}^{T} \big(f_t(\mathbf{w}_t^*, \mathbf{w}_t) + f_t(\mathbf{w}_t^*, \mathbf{w}'_t) - 2 f_t(\mathbf{w}_t^*, \mathbf{w}_t^*) \big).$$

Our goal is to design an online learning method for minimizing the above dynamic regret.

[2] In OL2R, a widely used link function is the sigmoid function $\sigma(x) = 1/\big(1+\exp(-x)\big)$, which satisfies all of our assumptions.

3 Method

In this section, we first review the dueling bandits gradient descent (DBGD) algorithm and derive its dynamic regret bound, then present our method as well as its theoretical guarantee, and finally discuss the extensions of our method to existing DBGD-type algorithms.

3.1 Dueling Bandits Gradient Descent

As outlined in Algorithm 1, DBGD has two hyperparameters δ and γ, corresponding to the step sizes of exploration and exploitation, respectively. In each round t, DBGD first draws a vector $\mathbf{u}_t$ uniformly at random from the unit sphere $\mathbb{S} \triangleq \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 = 1\}$ as an exploratory direction. Then, a candidate ranker is created with parameter

$$\mathbf{w}'_t = \Pi_{\mathcal{W}}[\mathbf{w}_t + \delta \mathbf{u}_t] \tag{4}$$

where $\mathbf{w}_t$ is the current parameter of the ranking model and $\Pi_{\mathcal{W}}[\cdot]$ denotes the operation of projecting a point to the parameter space $\mathcal{W}$. Next, the two rankers $\mathbf{w}_t$ and $\mathbf{w}'_t$ are compared by the probabilistic interleaving method [5], which can merge the ranking lists produced by the two rankers and infer a preference order over the two rankers from user clicks on the merged ranking list. Finally, based on the preference order, DBGD updates the parameter of the ranking model for the next round. Specifically, if $\mathbf{w}'_t$ wins, which reveals that the exploratory direction leads to better ranking performance, then the parameter of the ranking model moves along the exploratory direction with step size γ: $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}[\mathbf{w}_t + \gamma \mathbf{u}_t]$. Otherwise, the ranking model remains unchanged.

We rigorously analyze the learning properties of DBGD and derive a sub-linear dynamic regret bound as follows.

Theorem 1. *Let C_T be the path length of the optimal rankers over T rounds, defined as*

$$C_T = \sum_{t=2}^{T} \|\mathbf{w}_t^* - \mathbf{w}_{t-1}^*\|_2. \tag{5}$$

By setting $\delta = \sqrt{\frac{2\lambda d}{(11+2\lambda)L\sqrt{T}}}$ and $\gamma = \sqrt{\frac{5R^2+2RC_T}{T}}$, the dynamic regret of DBGD satisfies

$$\mathbb{E}[\mathrm{DR}(T)] \leq \sqrt{2(11+2\lambda)\lambda d L}\left(1 + \sqrt{5R^2 + 2RC_T}\right) T^{\frac{3}{4}}.$$

3.2 DBGD Meets Meta Learning

While DBGD can achieve a sub-linear dynamic regret bound for $C_T = o(\sqrt{T})$, it requires the value of the path-length C_T for tuning the step size γ, which is clearly impossible in practice since C_T depends on the unknown optimal rankers

Algorithm 1. DBGD

Require: step sizes of exploration δ and exploitation γ
Initialize a ranker $\mathbf{w}_1 \in \mathcal{W}$ arbitrarily
for $t = 1, 2, \ldots, T$ **do**
  Draw a vector $\mathbf{u}_t$ uniformly at random from $\mathbb{S}$
  Create an exploratory ranker $\mathbf{w}'_t = \Pi_{\mathcal{W}}[\mathbf{w}_t + \delta \mathbf{u}_t]$
  Compare $\mathbf{w}_t$ and $\mathbf{w}'_t$ by probabilistic interleaving
  if $\mathbf{w}'_t \succ \mathbf{w}_t$ **then**
    Set $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}[\mathbf{w}_t + \gamma \mathbf{u}_t]$
  else
    Set $\mathbf{w}_{t+1} = \mathbf{w}_t$
  end if
end for

$\mathbf{w}^*_1, \ldots, \mathbf{w}^*_T$. To address this issue, we employ the meta learning technique to automatically tune the step size γ, which has exhibited successes in online convex optimization [2,16,17]. The basic idea is to run multiple DBGDs in parallel, each of which is configured with a different step size γ and admits the sublinear dynamic regret bound for a class of path length. We develop our method in the prediction with expert advice framework, where each DBGD is viewed as an expert and the outputs of DBGDs are combined by an expert-tracking algorithm.

We now describe our method in detail, which is termed as DBGD Meets Meta Learning (DM^{2}L) and consists of a meta algorithm and an expert algorithm.

Meta Algorithm As outlined in Algorithm 2, at the beginning of the meta algorithm, we invoke the expert algorithm with different step size γ. According to our theoretical analysis, we maintain

$$N = \left\lceil \log_2 \sqrt{1 + 4T/5} \right\rceil + 1 \tag{6}$$

experts and the step size γ of the i-th expert is configured as

$$\gamma_i = 2^{i-1} R \sqrt{5/T}, \quad i = 1, \ldots, N. \tag{7}$$

Each expert $i \in [N]$ is associated with a time-variant weight π^i_t, which is dynamically adjusted according to the real time performance of expert i. For deriving a tighter dynamic regret bound, we take a nonuniform initialization of weights:

$$\pi^i_1 = \frac{N+1}{i(i+1)N}, \quad i = 1, \ldots, N. \tag{8}$$

In each round t, we first receive a ranker $\mathbf{w}^i_t$ from each expert $i \in [N]$ and aggregate these rankers according to the weights of experts $\pi^i_t, i \in [N]$ as $\mathbf{w}_t =$

$\sum_{i=1}^{N} \pi_t^i \mathbf{w}_t^i$. Then, we sample a vector $\mathbf{u}_t$ from the unit sphere $\mathbb{S}$ uniformly at random and compare $\mathbf{w}_t$ with $\mathbf{w}_t' = \Pi_{\mathcal{W}}[\mathbf{w}_t + \delta \mathbf{u}_t]$ by invoking the probabilistic interleaving method, which returns a noisy preference order $\mathbb{I}_{\{\mathbf{w}_t' \succ \mathbf{w}_t\}}$. Next, we update the weight of each expert using an exponential scheme

$$\pi_{t+1}^i = \frac{\pi_t^i \exp(-\alpha \ell_t(\mathbf{w}_t^i))}{\sum_{j=1}^{N} \pi_t^j \exp(-\alpha \ell_t(\mathbf{w}_t^j))}, \quad i = 1, \ldots, N \tag{9}$$

where $\ell_t(\mathbf{w})$ is a surrogate loss function, defined as

$$\ell_t(\mathbf{w}) = -\frac{d}{\delta} \langle \mathbb{I}_{\{\mathbf{w}_t' \succ \mathbf{w}_t\}} \mathbf{u}_t, \mathbf{w} - \mathbf{w}_t \rangle$$

which approximately evaluates the real-time performance of the experts. Finally, both the preference order $\mathbb{I}_{\{\mathbf{w}_t' \succ \mathbf{w}_t\}}$ and the exploratory direction $\mathbf{u}_t$ are sent to each expert so that they can update their own rankers accordingly.

Expert Algorithm. As summarized in Algorithm 3, the expert algorithm is a variant of DBGD. In each round t, each expert $i \in [N]$ first sends its current ranker $\mathbf{w}_t^i$ to the meta algorithm. Then, each expert receives the same preference order $\mathbb{I}_{\{\mathbf{w}_t' \succ \mathbf{w}_t\}}$ and exploratory direction $\mathbf{u}_t$ from the meta algorithm. Finally, each expert updates its own ranker as

$$\mathbf{w}_{t+1}^i = \Pi_{\mathcal{W}}[\mathbf{w}_t^i + \gamma_i \mathbb{I}_{\{\mathbf{w}_t' \succ \mathbf{w}_t\}} \mathbf{u}_t], \quad i = 1, \ldots, N. \tag{10}$$

Different from DBGD, we here take the same updating direction $\mathbb{I}_{\{\mathbf{w}_t' \succ \mathbf{w}_t\}} \mathbf{u}_t$ for all experts so that only two rankers $\mathbf{w}_t, \mathbf{w}_t'$ need to be compared in each round. While the updating direction is no longer opposite to the gradient of the smoothed function $\nabla h_t(\mathbf{w}_t^i)$, it is the inverse of the gradient of the surrogate loss function $\nabla \ell_t(\mathbf{w}_t^i)$. Thus, the updating rule of each expert can still be viewed as gradient descent and the dynamic regret of each expert can be analyzed following the proof of Theorem 1.

We present the theoretical guarantee of our method DM^{2}L in the following theorem. Compared to DBGD, the main advantage of DM^{2}L is that it can achieve the sub-linear dynamic regret bound without prior knowledge of the path length C_T and thus can adapt to unknown non-stationarity of environments.

Theorem 2. *By setting* $\delta = \sqrt{\frac{3\lambda d}{(11+2\lambda)L\sqrt{T}}}$ *and* $\alpha = 4/\sqrt{T}$ *and using the configurations in (6) and (7), DM^{2}L achieves the following dynamic regret bound*

$$\mathbb{E}[\mathrm{DR}(T)] \leq \sqrt{3(11+2\lambda)\lambda d L} \left(1 + \sqrt{5R^2 + 2RC_T}\right) T^{\frac{3}{4}} + \lambda(1 + \ln(N+1))\sqrt{T}.$$

3.3 Extensions to DBGD-Type Algorithms

While our meta learning method is developed in the context of DBGD, it be also straightforwardly extended to existing DBGD-type algorithms such as MGD [10]

Algorithm 2. DM^{2}L: Meta Algorithm

Require: number of experts N, step sizes δ, $\gamma_1, \ldots, \gamma_N$, learning rate α

1: Invoke Algorithm 3 with γ_i for each expert $i \in [N]$
2: Initialize the weights of experts $\pi_1^i, i \in [N]$ by (8)
3: **for** $t = 1, 2, \ldots, T$ **do**
4: Receive ranker $\mathbf{w}_t^i$ from each expert $i \in [N]$
5: Aggregate the rankers as $\mathbf{w}_t = \sum_{i=1}^N \pi_t^i \mathbf{w}_t^i$
6: Draw a vector $\mathbf{u}_t$ uniformly at random from $\mathbb{S}$
7: Create an exploratory ranker $\mathbf{w}_t' = \Pi_{\mathcal{W}}[\mathbf{w}_t + \delta \mathbf{u}_t]$
8: Compare $\mathbf{w}_t$ and $\mathbf{w}_t'$ by probabilistic interleaving
9: Update the weight of each expert $\pi_t^i, i \in [N]$ by (9)
10: Send $\mathbb{I}_{\{\mathbf{w}_t' \succ \mathbf{w}_t\}}$ and $\mathbf{u}_t$ to each expert $i \in [N]$
11: **end for**

Algorithm 3. DM^{2}L: Expert Algorithm

Require: step size of exploitation γ_i

1: Initialize a ranker $\mathbf{w}_1^i \in \mathcal{W}$ arbitrarily
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Send ranker $\mathbf{w}_t^i$ to Algorithm 2
4: Receive $\mathbb{I}_{\{\mathbf{w}_t' \succ \mathbf{w}_t\}}$ and $\mathbf{u}_t$ from Algorithm 2
5: Update ranker $\mathbf{w}_{t+1}^i = \Pi_{\mathcal{W}}[\mathbf{w}_t^i + \gamma_i \mathbb{I}_{\{\mathbf{w}_t' \succ \mathbf{w}_t\}} \mathbf{u}_t]$
6: **end for**

and NSGD-DSP [11,12]. Note that the existing DBGD-type algorithms only differ in the exploratory direction and the updating direction. Thus, we can replace Steps 6–8 at Algorithm 2 with the corresponding exploration pseudocodes of the DBGD-type algorithm and set $\mathbf{u}_t$ used in Steps 9–10 at Algorithm 2 as the updating direction in the DBGD-type algorithm, while keeping Algorithm 3 and the other steps of Algorithm 2 unchanged. We termed the algorithms obtained by applying our meta learning method to MGD and NSGD-DSP as M^{3}L (MGD Meets Meta Learning) and NM^{2}L (NSGD-DSP Meets Meta Learning), respectively.

4 Conclusion

We have formulated a new bandits model for OL2R, termed as non-stationary dueling bandits, where the preference order over rankers can change with time. For this bandits model, we developed a meta learning method, which dynamically aggregates multiple DBGD algorithms with different step sizes. Theoretical analysis showed that under mild assumptions, our meta learning method enjoys a sub-linear dynamic regret bound. We also discuss the extensions of our meta learning method to existing DBGD-type algorithms. Extensive experiments on public datasets demonstrate the effectiveness and efficiency of our meta learning method for OL2R in non-stationary environments.

Acknowledgements. This work was partially supported by NSFC (61976112) and JiangsuSF (BK20200064). We thank the anonymous reviewers for their constructive suggestions.

References

1. Cesa-Bianchi, N., Lugosi, G.: Prediction, Learning, and Games. Cambridge University Press (2006)
2. van Erven, T., Koolen, W.M.: Metagrad: Multiple learning rates in online learning. In: Advances in Neural Information Processing Systems, vol. 29, pp. 3666–3674 (2016)
3. Grotov, A., Rijke, M.: Online learning to rank for information retrieval. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1215–1218 (2016)
4. Hofmann, K.: Fast and reliable online learning to rank for information retrieval. Ph.D. Dissertation (2013)
5. Hofmann, K., Whiteson, S., Rijke, M.: A probabilistic method for inferring preferences from clicks. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 249–258 (2011)
6. Hofmann, K., Whiteson, S., Rijke, M.: Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. Inform. Retriev. **16**(1), 63–90 (2013)
7. Liu, T.Y.: Learning to rank for information retrieval. Found. Trends Inf. Retriev. **3**(3), 225–331 (2009)
8. Oosterhuis, H., Rijke, M.: Differentiable unbiased online learning to rank. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 1293–1302 (2018)
9. Radlinski, F., Kurup, M., Joachims, T.: How does clickthrough data reflect retrieval quality? In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 43–52 (2008)
10. Schuth, A., Oosterhuis, H., Whiteson, S., Rijke, M.: Multileave gradient descent for fast online learning to rank. In: Proceedings of the 9th ACM International Conference on Web Search and Data Mining, pp. 457–466 (2016)
11. Wang, H., Kim, S., McCord-Snook, E., Wu, Q., Wang, H.: Variance reduction in gradient exploration for online learning to rank. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 835–844 (2019)
12. Wang, H., Langley, R., Kim, S., McCord-Snook, E., Wang, H.: Efficient exploration of gradient space for online learning to rank. In: Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 145–154 (2018)
13. Yue, Y., Broder, J., Kleinberg, R., Joachims, T.: The k-armed dueling bandits problem. J. Comput. Syst. Sci. **78**(5), 1538–1556 (2012)
14. Yue, Y., Joachims, T.: Interactively optimizing information retrieval systems as a dueling bandits problem. In: Proceedings of the 26th International Conference on Machine Learning, pp. 1201–1208 (2009)
15. Zang, Y., et al.: GISDCN: A graph-based interpolation sequential recommender with deformable convolutional network. In: International Conference on Database Systems for Advanced Applications, pp. 289–297. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-00126-0_21

16. Zhang, L., Lu, S., Zhou, Z.H.: Adaptive online learning in dynamic environments. In: Advances in Neural Information Processing Systems, vol. 31, pp. 1323–1333 (2018)
17. Zhao, P., Wang, G., Zhang, L., Zhou, Z.H.: Bandit convex optimization in non-stationary environments. In: Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, pp. 1508–1518 (2020)