Hierarchical Learning Algorithms for Multi-scale Expert Problems

Lin Yang
University of Massachusetts Amherst
Amherst, MA, USA
linyang@cs.umass.edu

Yu-zhen Janice Chen
University of Massachusetts Amherst
Amherst, MA, USA
yuzhenchen@cs.umass.edu

Mohammad H. Hajiesmaili
University of Massachusetts Amherst
Amherst, MA, USA
hajiesmaili@cs.umass.edu

Mark Herbster
University College London
London, UK
m.herbster@cs.ucl.ac.uk

Don Towsley
University of Massachusetts Amherst
Amherst, MA, USA
towsley@cs.umass.edu

ABSTRACT

In this work[1], we study the multi-scale expert problem, where the rewards of different experts vary in different reward ranges. The performance of existing algorithms for the multi-scale expert problem degrades linearly proportional to the maximum reward range of any expert or the best expert and does not capture the non-uniform heterogeneity in the reward ranges among experts. In this work, we propose learning algorithms that construct a hierarchical tree structure based on the heterogeneity of the reward range of experts and then determine differentiated learning rates based on the reward upper bounds and cumulative empirical feedback over time. We then characterize the regret of the proposed algorithms as a function of non-uniform reward ranges and show that their regrets outperform prior algorithms when the rewards of experts exhibit non-uniform heterogeneity in different ranges. Last, our numerical experiments verify our algorithms' efficiency compared to previous algorithms.

CCS CONCEPTS

• **Computing methodologies → Online learning settings**; **Sequential decision making**.

ACM Reference Format:

Lin Yang, Yu-zhen Janice Chen, Mohammad H. Hajiesmaili, Mark Herbster, and Don Towsley. 2022. Hierarchical Learning Algorithms for Multi-scale Expert Problems. In *Abstract Proceedings of the 2022 ACM SIGMETRICS/IFIP PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/PERFORMANCE '22 Abstracts), June 6–10, 2022, Mumbai, India.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3489048.3530967

1 INTRODUCTION

This paper studies an extension of the expert problem that deals with experts with scaled rewards in different ranges, which is a practically relevant variant of the Hedge problem, the multi-scale Hedge problem or MSHedge, for short. The basic Hedge problem assumes a homogeneous reward range for each expert, e.g., $[0, 1]$. In practice, however, there is a broad range of applications such as dynamic pricing, portfolio selection, etc., where the rewards of different experts are heterogeneous and scaled in different ranges. This motivates the model of our interest, which involves multi-scale experts, those who possess non-uniform reward ranges. In the MSHedge problem, the reward range of expert i is $[L_i, U_i]$, where U_i and L_i serve as the upper and lower bounds of rewards, whose values are known to the online player in advance. For ease of technical presentation, we consider two different models for MSHedge: (1) MSHedge-U, which only considers heterogeneity in upper bounds with lower bounds set to 0; and (2) MSHedge-LU, which allows both upper and lower bounds to be heterogeneous[2]. It is straightforward to show that the naive extension of the Hedge algorithm to the multi-scale setting leads to a regret of $O(M\sqrt{T \log K})$, which linearly scales with M, the maximum reward upper bound among K experts. A related work in [1] improves the regret to scale linearly proportional to the reward upper bound of the optimal expert instead of the largest reward upper bound among all experts. In this work, we made the following contributions to the MSHedge problem.

1.1 Contributions

DRate*: A hierarchical learning algorithm with differentiated learning rates.* Our key idea is to explicitly capture non-uniform upper bounds into the algorithm design. Towards this, we propose to adaptively change the learning rate, and hence the selection probabilities of the expert, based on both the cumulative feedback and upper bound of the leading experts. We develop two learning algorithms based on Differentiated learning RATEs, called DRate-U and DRate-LU for short, which work within the MSHedge-U and MSHedge-LU models, respectively. For simplicity of presentation, we also refer to those two algorithms together as DRate.

To deal with the heterogeneity of multi-scale reward values, DRate partitions the set of experts into smaller subsets, placing experts with similar reward ranges in the same subset. DRate recursively continues to partition the experts into the new subsets and stops partitioning when the upper bounds in the subset are uniform. With a given tree, the decision-making process of DRate is as follows. At each round, DRate traverses the tree by recursively selecting nodes from the root to a leaf node and possibly running

[1]The full version of this abstract is available in [2].

[2]In the literature, a well-known special model of MSHedge-LU is the Lipschitz Expert problem [3].

ACM ISBN 978-1-4503-9141-2/22/06.
https://doi.org/10.1145/3489048.3530967

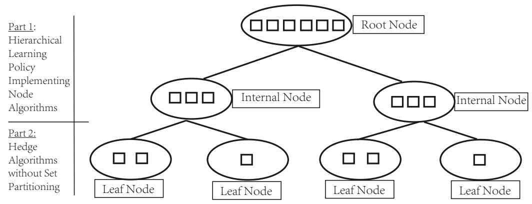

Figure 1: The hierarchical structure of the `DRate-U` algorithm

the Hedge algorithm in the selected leaf node, and eventually, returning an expert associated with the selected leaf node as its final decision.

Regret Bounds. We first characterize the regret of `DRate-U` as a function of the path from root to the node that includes the best expert. By proper parameter setting and with a balanced binary tree, we show that `DRate-U` achieves a regret of $O(\sqrt{U_1}\sum_{l=1}^{\log K}\sqrt{U_{2^l-1}T})$, where U_1, w.l.o.g, is the largest reward upper bound assuming a descending order of experts based on their upper bounds. Then, we propose an algorithm that constructs an underlying tree that minimizes the regret. The regret of `DRate-LU` also depends on how the tree is constructed. Given a tree, `DRate-LU` provides a provable regret, which is the cumulative regret over the path from root to the node that includes the best expert. Specifically, by placing experts with similar reward ranges into the same node, `DRate-LU` can reduce the region that the reward fluctuates in a node, as well as the regret.

2 THE DRATE ALGORITHM

The `DRate` Algorithm can work in both `MSHedge-U` and `MSHedge-LU`. As an example, we only introduce the `DRate-U` algorithm for the first model of `MSHedge`, where the reward upper bounds of experts scale in different values. Our high-level idea is to adopt a hierarchical learning policy to effectively capture the multi-scale upper bounds. More specifically, `DRate-U` leverages a tree structure to categorize the experts with different upper bounds, and tackles the original learning problem hierarchically by traversing through the constructed tree.

In Figure 1, we demonstrate the structure of the decision tree where the expert set $\mathcal{K}$ resides at the root and each internal node represents a subset of experts. Each node in the tree associated with more than one experts with different upper bounds can be further partitioned into two smaller subsets as its children. The algorithm may choose not to further partition the node when the upper bounds of the contained experts are the same. Hence, a leaf node may contain more than one expert. The performance of `DRate-U` closely depends on how the tree is constructed.

With a given tree, the decision making process of `DRate-U` is as follows. At each time slot, `DRate-U` traverses the tree by recursively calling a node algorithm[3] from the root to a leaf node and possibly running the Hedge algorithm in the selected leaf node. Eventually, it returns an expert associated with the selected leaf node as its final decision.

[3]In our terminology, the algorithm executed by a non-leaf node in the tree is called the node algorithm.

In `DRate-U`, the node algorithm plays a critical role in dealing with multi-scale experts in `DRate-U`, where a function parameterized by a learning rate is maintained to generate the selection probabilities of the child nodes. Let $\alpha_t(v)$ be the child of node v with larger cumulative feedback at time t, and $\beta_t(v)$ be the other child. Consider two cases: Case-1 occurs when $\alpha_t(v)$ is the node with the larger upper bound; and Case-2, when $\alpha_t(v)$ is the node with smaller upper bound than that of $\beta_t(v)$. The key idea is to assign aggressive (or larger) learning rate to Case-1 and a conservative (or smaller) learning rate when facing Case-2. The reason is intuitive: with multi-scale upper bounds, an important observation is that when Case-2 occurs, i.e., the cumulative feedback of the expert with larger upper bounds falls behind, the risk of large regret is higher, since the larger upper bound provides a room for the expert to quickly catch up with the cumulative feedback of the other node. Thus, in Case-2, a small or conservative learning rate is preferable. On the other hand, in Case-1, i.e., when the expert with lower upper bound falls behind, it takes longer for it to catch up with the other one, so, it is safe to select a more aggressive learning rate.

Our analysis shows that the non-uniform structure of expert upper bounds can lead to different underlying trees that minimize the regret of `DRate-U`. In this work, we also provide an algorithm to generate the optimal tree for any instance of `MSHedge-U` given the reward upper bounds.

3 MAIN RESULTS

Also, due to the page limitation, we only present the results for the `DRate-U` algorithm.

Theorem 3.1. *(Regret of* `DRate-U`*) With proper learning rates for non-leaf node v and leaf node v', the regret of* `DRate-U` *satisfies the following upper bound.*

$$R_T \le \sum_{v\in \mathtt{path}(v^*)\setminus\{v^*\}} \left(\sqrt{2U_{v_l}U_{v_r}T} + U_{v_r}\sqrt{T}\right) + U_{v^*}\sqrt{T\log|v^*|},$$

where v_l and v_r are left and right children of node v, respectively, $\mathtt{path}(v)$ *denotes the set of nodes on the path from the root to node v, and $v*$ is the leaf node containing the optimal expert.*

4 ACKNOWLEDGMENTS

This research is supported by NSF CPS 2136199, CAREER 2045641, CNS 2102963, CNS 1908298, and CNS 2106299, U.S. Army Research Laboratory Army Research Laboratory under Cooperative Agreement W911NF-17-2-0196 (IoBT CRA), and U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement W911NF-16-3-0001.

REFERENCES

[1] Sebastien Bubeck, Nikhil R Devanur, Zhiyi Huang, and Rad Niazadeh. 2017. Online auctions and multi-scale online learning. In *Proceedings of the 2017 ACM Conference on Economics and Computation.* 497–514.

[2] Lin Yang, Janice Yu-zhen Chen, Mohammad H Hajiesmaili, Mark Herbster, and Don Towsley. 2022. Hierarchical Learning Algorithms for Multi-scale Expert Problems. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 6, 2 (2022), 1–29.

[3] Lin Yang, Lei Deng, Mohammad H Hajiesmaili, Cheng Tan, and Wing Shing Wong. 2018. An optimal algorithm for online non-convex learning. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2, 2 (2018), 1–25.