Faster Simulated Performance of Hedge on Low Rank Experts

Cheng Shi Michael Spece

May 11, 2021

***Abstract*—This paper investigates the relationship between an algorithm's worst-case regret and the rank of an ensemble's loss matrix Y. A previous approach has attempted to study this through simulation. We build on this work by providing faster generation of low rank matrices. We concur with the previous conclusion that a standard exponential weighting algorithm (Hedge) exploits low rank structure effectively.**

***Keywords*—experts, online learning, regret, simulation study.**

I. Introduction

In the field of online learning, there's a basic problem about the prediction with expert advice. N experts generate predictions for T rounds. The learner have a probability vector $p_t \in \Delta_N$ in every round $t = 1 \cdots T$, where Δ_N denotes the N-simplex, namely the set of all distributions over N experts

$$\Delta_N = \left\{ x \in R^N : \forall i, x(i) \geq 0 \wedge \sum_{i=1}^{N} x_i = 1 \right\}.$$

Then Nature chooses a loss vector $y^t \in [0,1]^N$, and sets a loss as $p^t(y^t) = p^t \cdot y^t$. The regret is defined as follows:

$$\mathrm{Reg}(N,T) = \sum_{t=1}^{T} p^t \cdot y^t - \min_{k \in \{1,\ldots,N\}} \sum_{t=1}^{T} y_k^t.$$

According to the overall performance, the expert who got the best score in the history, can be a more reliable trust more. This strategy can be interpreted as a form of empirical risk minimization and is called "follow the leader".

Following [3], stochastic and adversarial are two types of settings in the field of online learning. In the stochastic, a fixed IID distribution generate y^t and the learner pursues control of the expected regret. In the adversarial, the history of round can totally decides the y^t. This study, therefore, set out to assess the adversarial setting, which is less well understood.

A. *Algorithms for Adversarial Data*

Following [4], Exponentially Weighted Averaging(EWA, also known as Hedge) is a form of regularized follow the leader.

EWA is named and defined in [1]. Its basic idea is to weight experts according to their performance. Specifically, when an expert has relatively high loss, it will get an exponentially decreasing proportion of weight. The update rule is

$$p_j^{t+1} = \frac{p_j^t e^{-\eta y_j^t}}{\sum_{i=1}^{N} p_i^t e^{-\eta y_i^t}}. \tag{1}$$

B. *Low Rank Experts*

[2] proposed the Low-Rank Experts setting, in which they assume that there is a low dimensional r ($r << N$) embedding of the loss matrix. Besides [4], the literature has focused on new algorithms for exploiting this low dimensionality, largely ignoring the benefit that may come for free with existing algorithms.

C. *Simulation*

[4] randomly generated matrices of either low or high rank, discarding matrices that do not have exact desired rank. Rather than having two different algorithms for each of these regimes, we decompose the problem into first generating a full rank matrix and secondly dependent rows. Because each of these steps is relatively straight-forward and can therefore be accomplished without discarding many randomly sampled matrices, we expect this approach to be much faster.

II. Learning Algorithms Under Simulation Study

The learning algorithm (Algorithm 1) under consideration is a horizon-adaptive version of EWA, which also appears in [1].

Algorithm 1: Horizon Adaptive EWA

Data: N,T, binary loss matrix Y (the t th column of Y is called y^t,the k th rows and the t th column of Y is called y_k^t)

Result: Get the regret

1. $p_1^1 = 1$ $p_k^1 = 0 (k = 2, \ldots, N)$
2. $p_k^t = \frac{e^{-\eta \sum_{t'=1}^{t-1} y_k^{t'}}}{\sum_{k'=1}^{N} e^{-\eta \sum_{t'=1}^{t-1} y_{k'}^{t'}}} (\eta = \sqrt{\frac{\log(N)}{t} \times 8}, t = 2, \ldots, T, k = 1, \ldots, N)$
3. $\text{Reg}(N, T) = \sum_{t=1}^{T} p^t \cdot y^t - \min_{k \in \{1, \ldots, N\}} \sum_{t=1}^{T} y_k^t$

III. SIMULATION

A. *Generating the Loss Matrix*

Algorithm 2: Generating loss matrix

Data: Rank $r \leq N = T$

Result: Randomly generated 0-1 loss matrix with rank r

1. Iteratively randomly generate r rows until obtaining an independent set R
2. Generate $N - r$ dependent rows
   (a) Iteratively randomly generate a row as a linear binary-weighted combination of rows from R until obtaining a row of 0 and 1s
   (b) Repeat above $N - r - 1$ times

B. *Simulated Worst-Case Regret for $N = T = 20$*

We follow [4] in considering the case $N = T = 20$. However, we only examine ranks of less than 11 because it was observed the regret appears to saturate there, and despite using a larger number of trials all of which are effective, we also observe saturation. Leveraging the increased speed of our simulation, we generate ten times more samples, that is 10^6.

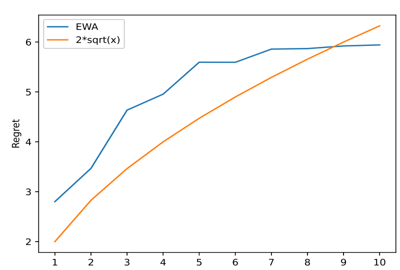

Fig. 1. Worst-case regret of EWA among one million loss simulations per rank.

Figure 1 describes the relationship between the worst regret and the loss matrices' rank r. As in [4] we compare the curve to $y = 2\sqrt{x}$. The larger number of samples does appear to result in a uniformly strictly better approximation of the worst-case regret for ranks greater than ?.[1]

[1]To facilitate future comparisons, we provide the exact values here: ....

IV. CONCLUSION

We have confirmed EWA exploits low rank structure effectively and comparably to the expected square-root behavior. This experimental finding was facilitated by our algorithm for faster generation of low rank matrices.

References

[1] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*.

[2] E. Hazan, T. Koren, R. Livni, and Y. Mansour, 'Online learning with low rank experts', *CoRR*, **abs/1603.06352**, (2016).

[3] W. Liu, M. Spece, and S. Jia, 'Fast learning and low rank experts: Novel adaptive methods', *SSRN*, (October 2019). Available at https://ssrn.com/abstract=3484205.

[4] Weiqi Yang and Michael Spece, 'Implicit adaptation to low rank structure in online learning', *International Journal of Machine Learning and Computing*, **11**(5), (2021).