Dynamic Scheduling with Expert Predictions

Ahan Mishra
Cornell University
Ithaca, NY, USA

ABSTRACT

We consider a non-clairvoyant scheduling problem in which the algorithm has access to a set of experts and must learn in order to identify the useful experts. In particular, each expert predicts a job size for each incoming job and the objective is to minimize total flow time in the M/G/1 setting. With access to an empirical risk minimization oracle over the experts, we are able to discover a good predictor from a large set, rather than assuming we have a good predictor from the beginning.

1. INTRODUCTION

The field of learning-augmented algorithms [8] focuses on adding predictions into traditional algorithms to improve performance when the predictions are good. Most works have focused on using predictors with a specified performance guarantee, and proving results on robustness and consistency with respect to that guarantee (e.g., see the early work of [6]).

A relatively underexplored area is integrating the method by which a predictor is actually obtained, e.g., see [4, 5, 3]. The closest work to ours in considering "explicit predictors" is that of [4], which introduces a setting where there is a hypothesis class $\mathcal{H}$ that can be used to generate predictions. The goal in [4] is to compete with the best hypothesis in hindsight (in the agnostic setting), or the best offline algorithm (in the realizable setting). We also similarly borrow the terminology of "expert" from the experts problem [7] in learning theory.

One weakness of the results in [4] is that the computation time is linear in the size of the hypothesis class. As noted in [2], models with large numbers of parameters have become particularly relevant due to deep learning, and achieving computational time logarithmic in the size of the hypothesis class is a major goal in the contextual bandit setting [1].

In addition, while [4] focuses on static scheduling (where all jobs are released at time 0), we study dynamic scheduling. The settings of static and dynamic scheduling differ significantly in optimizing total flow time, as seen by the fact that without predictions, static scheduling admits a simple 2-competitive algorithm (round robin) while dynamic scheduling admits no constant-factor-competitive algorithm [9]. Note that the total flow time is $F(I) = \sum_{i=1}^{J} c_i - r_i$ where r_i and c_i are the release and completion times, respectively, of job i. An algorithm is said to be c-competitive for constant c if for any input I, $F(I)/F_{\mathrm{OPT}}(I) \leq c$, where $F_{\mathrm{OPT}}(I)$ is the total flow time of the optimal offline algorithm. Note that Shortest Remaining Processing Time (SRPT) is an optimal offline algorithm [10].

The work [11] also studies dynamic scheduling with predictions in a stochastic setting, specifically the M/G/1. We also study the M/G/1 setting. A lower bound of [4] shows that if job arrivals and sizes are adversarial, it is impossible to achieve total flow time loss (with respect to SRPT) that is sublinear in the size of the hypothesis class, even in the static, realizable case. A linear rate is not useful in the setting of large hypothesis classes that we are interested in. In [11], it is assumed that the algorithm has access to a predictor that, for a job of size s, outputs an estimate in the range $[\alpha s, \beta s]$. As discussed earlier, this does not explain how the predictor was obtained, and in addition, it may be unrealistic for a predictor to always output good predictions. A predictor that does poorly on a small portion of the job distribution should be possible to use effectively, and we formalize this intuition.

2. SETTING

We study a stochastic M/G/1 setting. Note that the "M" in the M/G/1 queue refers to inter-arrival times being drawn i.i.d. from an exponential distribution with parameter λ. The "G" refers to job sizes being drawn i.i.d. from some distribution S with mean $\mu = \mathbb{E}[S]$. We also assume that the inter-arrival times and job sizes are independent. The "1" refers to having a single machine to process jobs with. The load on the system is $\rho = \lambda\mu$. We assume that $\rho < 1$, as in [11], specifically letting $\alpha = 1 - \rho$ with $\alpha > 0$. This corresponds to the assumption that jobs arrive slower than the machine processes them, on average.

We will also assume that S is bounded (i.e., there exists some B such that $\mathbb{P}_{s\sim S}(s \leq B) = 1$), as in [4]. Then, without loss of generality, we can assume that the maximum possible job size is 1. In particular, if $B > 1$, then we can work with $A' = A/B$ and $S' = S/B$, Then,

$$\mathbb{E}[S']/\mathbb{E}[A'] = (\mathbb{E}[S]/B)/(\mathbb{E}[A]/B) = \mathbb{E}[S]/\mathbb{E}[A]$$

so the new distributions can be regarded as a time-rescaled version of the originals with the same load parameter ρ.

The set of experts we consider is $\mathcal{F}$, where each $f \in \mathcal{F}$ predicts a job size for each job. In particular, as each job comes in, associated information x_i is known, and $f(x_i)$ is the prediction for job i. We assume that the contexts x_i come in i.i.d. from the context distribution $\mathcal{X}$. Let s_i be the

Learning-Augmented Algorithms: Theory and Applications (LATA) 2026
Ann Arbor, MI, USA

true size of job i. The loss of each expert is the total absolute deviation, specifically, $\text{loss}(f) = \sum_{i=1}^{n} |f(x_i) - s_i|$. It is not possible to achieve sublinear regret with unstructured experts. We assume that structure exists in the form of an ERM oracle [2], specifically an efficiently implementable function that, when given (x_i, s_i) pairs, outputs the expert f^* which minimizes $\text{loss}(f)$ over those pairs.

3. CONTRIBUTIONS

We show that the following algorithm, which simply plays the ERM on each round, has a total flow time that depends smoothly on the loss of the best expert in hindsight.

Algorithm 1 ERM-based Algorithm

```
1: f* ← some arbitrary expert in F
2: for each time instant t do
3:     A ← the set of active jobs at time t
4:     if A is not empty then
5:         Schedule the job argmin_{j∈A} f*(j)
6:         if job j completes then
7:             f* ← ERM({(x_i, s_i) from all completed jobs})
8:         end if
9:     end if
10: end for
```

Our analysis associates regret in the expert setting with total flow time by establishing a structure of job completion times for any work-conserving scheduler. In particular, we show that any loss incurred by an expert affects only a small (compared to n) number of jobs in the scheduler, due to the regular occurrence of idle periods arising from $\alpha > 0$.

In particular, in the agnostic setting, we prove that with probability at least $1 - \delta$, on instance I, the total flow time F achieved by Algorithm 1 satisfies $F(I) \leq F_{\text{OPT}}(I) + O\left(\left(\frac{\log(n/\delta)+\lambda}{\alpha}\right)\left(\text{loss}(f_{\text{OPT}}) + \sqrt{\frac{n\log(n|\mathcal{F}|/\delta)}{2}}\right)\right)$ where $f_{\text{OPT}} = \arg\min_{f\in\mathcal{F}}(\sum_{i=1}^{n} |f(x_i) - s_i|)$. The proof bounds the gap between the estimated and true losses of each expert in a standard way using Hoeffding's inequality and a union bound.

In the realizable setting, where there exists f^* that has true loss of 0, we prove that with probability at least $1 - \delta$, the total flow time F achieved by Algorithm 1 satisfies $F(I) \leq F_{\text{OPT}}(I) + O\left(\left(\frac{\log(n/\delta)+\lambda}{\alpha}\right)(\log(n)\log(n|F|/\delta))\right)$.

In this setting, the ERM computed on line 7 always has an empirical loss of 0, and we apply a variance-based concentration inequality (specifically Freedman's inequality) to obtain a tighter bound than achievable through Hoeffding's.

We also show that the number of ERM calls can be reduced from linear in n to logarithmic with an epoch-based approach.

4. REFERENCES

[1] A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1638–1646, Bejing, China, 22–24 Jun 2014. PMLR.

[2] A. Assos, I. Attias, Y. Dagan, C. Daskalakis, and M. K. Fishelson. Online learning and solving infinite games with an erm oracle. In G. Neu and L. Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 274–324. PMLR, 12–15 Jul 2023.

[3] E. Du, F. Wang, and M. Mitzenmacher. Putting the "learning" into learning-augmented algorithms for frequency estimation. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2860–2869. PMLR, 18–24 Jul 2021.

[4] M. Eliáš, H. Kaplan, Y. Mansour, and S. Moran. Learning-augmented algorithms with explicit predictors. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 97972–98008. Curran Associates, Inc., 2024.

[5] M. Khodak, M.-F. Balcan, A. Talwalkar, and S. Vassilvitskii. Learning predictions for algorithms with predictions. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.

[6] R. Kumar, M. Purohit, and Z. Svitkina. Improving online algorithms via ml predictions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9684–9693, Red Hook, NY, USA, 2018. Curran Associates Inc.

[7] N. Littlestone and M. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[8] M. Mitzenmacher and S. Vassilvitskii. Algorithms with predictions. *Commun. ACM*, 65(7):33–35, June 2022.

[9] R. Motwani, S. Phillips, and E. Torng. Non-clairvoyant scheduling. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '93, page 422–431, USA, 1993. Society for Industrial and Applied Mathematics.

[10] L. Schrage. A proof of the optimality of the shortest remaining processing time discipline. *Operations Research*, 16(3):687–690, 1968.

[11] Z. Scully, I. Grosof, and M. Mitzenmacher. Uniform Bounds for Scheduling with Job Size Estimates. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 114:1–114:30, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.