Dynamic Learning in Large Matching Markets

Anand Kalvit[1] and Assaf Zeevi[2]
Columbia University
{[1]akalvit22,[2]assaf}@gsb.columbia.edu

ABSTRACT

We study a sequential matching problem faced by *large* centralized platforms where "jobs" must be matched to "workers" subject to uncertainty about worker skill proficiencies. Jobs arrive at discrete times (possibly in batches of stochastic size and composition) with "job-types" observable upon arrival. To capture the "choice overload" phenomenon, we posit an unlimited supply of workers where each worker is characterized by a vector of attributes (aka "worker-types") sampled from an underlying population-level distribution. The distribution as well as mean payoffs for possible worker-job type-pairs are unobservables and the platform's goal is to sequentially match incoming jobs to workers in a way that maximizes its cumulative payoffs over the planning horizon. We establish lower bounds on the *regret* of any matching algorithm in this setting and propose a novel rate-optimal learning algorithm that adapts to aforementioned primitives *online*. Our learning guarantees highlight a distinctive characteristic of the problem: achievable performance only has a *second-order* dependence on worker-type distributions; we believe this finding may be of interest more broadly.

Keywords

Dynamic learning, Two-sided markets, Multi-armed bandits, Matching algorithms, Regret analysis.

1. INTRODUCTION

Background and motivation. The problem of sequentially matching "jobs" to "workers" under uncertainty forms the bedrock of many modern operational settings, especially in the online gig economy, see, e.g., applications such as Amazon Mechanical Turk, TaskRabbit, Jobble, and the likes. A simpler instance of the problem dates back to [1] where it is referred to as the sequential stochastic assignment problem (SSAP). A fundamental issue in such settings is that the platform typically is oblivious (at least initially) to the skill proficiencies of individual workers for specific job categories. This complexity is further compounded by the large number of workers usually present on such platforms, tantamount to prohibitively large experimentation costs associated with acquisition of granular information at the level of an individual worker. This issue is commonly mitigated by exploiting structure in the problem (if any), or by positing distributional assumptions on the population of available workers, e.g., workers may be drawn from some distribution $\mathcal{D}$ satisfying certain context-specific desiderata. Such distributional assumptions are vital to designing efficient algorithms for these systems, and as such, traditional literature has largely relied on the availability of ex ante knowledge of $\mathcal{D}$ or certain key aspects thereof (see, e.g., [3, 2], etc.)

Key research question. An important characteristic of the gig economy is that the population of workers may undergo distributional shifts over the course of the platform's planning horizon. These effects may, many a time, fail to register in a timely manner; as a result, there may be delays in tailoring appropriately the matching algorithm (calibrated typically using available distribution-level information) to the changed environment. This has the potential to cause revenue losses as well as catalyze endogenous worker attrition. Such exigencies necessitate designing algorithms that are *agnostic* to $\mathcal{D}$ and whose performance is *robust* to plausible realizations thereof.

The model at a glance. We consider a finite set of possible job-types (denoted by $\mathcal{J}$), an assumption we deem appropriate for settings such as those discussed above. In addition, we model workers as exhibiting discrete skill-levels (aka worker-types), indexed by $\{1, ..., K_j\}$, w.r.t. each job-type $j \in \mathcal{J}$, and assume that $(K_j : j \in \mathcal{J})$ is known a priori. It is not unreasonable to make this assumption since it is common, in practice, for platforms to deploy pilot experiments prior to the actual matching phase in order to gather sufficient information on key primitives such as the size and stability of low-dimensional sub-population clusters, if any exist; one can therefore safely assume in settings where such structure exists that $(K_j : j \in \mathcal{J})$ is well-estimated a priori.

While the demand is constituted by sequential job-arrivals (possibly in batches of stochastic size and composition), we posit availability of an unlimited number of workers on the supply side. This feature encapsulates the *choice overload* phenomenon characteristic of many large market settings where workers are available in a large number relative to the platform's planning horizon. To our best knowledge, extant literature on matching under uncertainty is largely limited to "finite" markets and therefore fails to accommodate this important practical consideration. In our setting, the population of workers, albeit large, is governed by a finitely supported distribution that controls the proportion of each worker-type. Specifically, the K_j distinct worker-types w.r.t. job-type j are distributed according to $\boldsymbol{\alpha_j} := (\alpha_{i,j} : i = 1, ..., K_j)$, where $\sum_{i=1}^{K_j} \alpha_{i,j} = 1$. We note that this is *one* possible model of a matching market that is closer in spirit to SSAP [1]; it differs from other models in the matching literature (see, e.g., [2]) in that it tries to capture a salient aspect of *large* markets, viz., choice overload, as opposed to aspects such as *competition* and *congestion* best elucidated via traditional "finite" market models.

The platform's goal is to maximize its expected cumulative payoffs over a sequence of n rounds of matching, subject to worker-types w.r.t. job-types and their distributions $\{\boldsymbol{\alpha_j} : j \in \mathcal{J}\}$, as well as mean payoffs for possible worker-job type-pairs being latent attributes. As is the norm in settings with incomplete information and imperfect learning, we reformulate this objective as minimizing the *expected cumulative regret* relative to an oracle that is privy to aforementioned primitives.

2. PROBLEM FORMULATION

Job-arrival process. The platform faces an arrival stream of jobs (i.i.d. in time) given by $\{(\Lambda_{j,t} : j \in \mathcal{J}) : t \geqslant 1\}$, where $\mathcal{J}$ is finite and $\Lambda_{j,t}$ is the number of type j jobs arriving at time t. Types and multiplicities of jobs are perfectly observable upon arrival. We assume that there exists some finite constant $M > 0$ s.t. $\mathbb{P}\left(\max_{j\in\mathcal{J}} \sup_{t\geqslant 1} \Lambda_{j,t} \leqslant M\right) = 1$. Note that our algorithms do not require knowledge of M; the assumption only serves to simplify analysis and can be relaxed under suitable conditions on the tail distribution of $\Lambda_{j,t}$'s.

Supply of workers. We assume that workers are distributed on the unit interval $[0,1]$ according to some probability distribution $\mathcal{D}$ that is absolutely continuous w.r.t. the Lebesgue measure on $[0,1]$. Associated with each job-type $j \in \mathcal{J}$, there exists a permutation $\boldsymbol{\sigma_j} := \{\sigma_j(i) : i = 1, ..., K_j\}$ of $\{1, ..., K_j\}$, and a sequence of thresholds $0 =: \lambda_{0,j} < \lambda_{1,j} < ... < \lambda_{K_j-1,j} < \lambda_{K_j,j} := 1$ partitioning the unit interval into K_j disjoint sub-intervals. We posit a payoff model whereby a worker $x \in (\lambda_{i-1,j}, \lambda_{i,j})$ (for some $i \in \{1, ..., K_j\}$) generates a stochastic reward with mean $\mu_{\sigma_j(i),j}$ upon match with a type j job; it is assumed that the K_j mean rewards adhere to the strict order $\mu_{1,j} > ... > \mu_{K_j,j}$. Define $\boldsymbol{\mu_j} := (\mu_{i,j} : i = 1, ..., K_j)$. Also define $\alpha_{i,j} := \mathbb{P}\left(X \in \left(\lambda_{\iota(i,j)-1,j}, \lambda_{\iota(i,j),j}\right)\right)$, where $X \sim \mathcal{D}$ and $\iota(i,j) \in \{1, ..., K_j\}$ is the unique element satisfying $\sigma_j(\iota(i,j)) = i$, as the probability that a worker sampled at random from $\mathcal{D}$ (equivalently, from the *population*), is i^{th} best for job-type j (generates mean reward $\mu_{i,j}$); such a worker is said to have type i w.r.t. job-type j. Thus, a type 1 worker w.r.t. job-type j is *optimal* for jobs of type j. Define $\boldsymbol{\alpha_j} := (\alpha_{i,j} : i = 1, ..., K_j)$. Note that the model allows for *staggered optimality* of worker-types; see Figure 1.

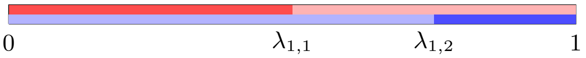

Figure 1: Possible distribution of worker-types for $\mathcal{J} = \{1, 2\}$ and $K_1 = K_2 = 2$. The darker shades represent type 1 (optimal) workers while the lighter shades represent type 2 (inferior) workers w.r.t. each job-type in $\mathcal{J}$. In this example, no worker can simultaneously be optimal for both job-types.

High-level description of the matching problem. Each arriving job may be matched one-to-one to a worker from the available supply. Each match takes one period for execution, it is therefore possible to match jobs arriving in consecutive periods to the same worker. Matched jobs leave the system upon completion and the platform receives a stochastic reward for each completed job; a job that remains unmatched drops out instantaneously. The platform has information neither on individual worker-types w.r.t. job-types nor on their supply distribution, however, it has perfect knowledge of $(K_j : j \in \mathcal{J})$. Subject to this premise, the platform must match incoming jobs to workers in a way that maximizes its expected cumulative payoffs over n rounds of matching.

Adaptive control. For any job that arrives at time t, the platform can match it to: (i) a worker that has matched before, (ii) a *new* worker (one without any history of matches) sampled from the population, or (iii) no worker (job gets dropped in this case). A policy $\pi := (\pi_1(\cdot,\cdot), \pi_2(\cdot,\cdot), ...)$, to this end, is an *adaptive* rule that prescribes the allocation $\pi_t(\cdot,\cdot)$ at time t. Specifically, $\pi_t(j,k)$ encodes the worker that should match with the k^{th} job of type j arriving at time t (provided there are at least k job-arrivals of type j at t and the k^{th} job is not dropped). Upon match, a $[0,1]$-valued stochastic reward with mean $\mu_{\kappa_j(\pi_t(j,k)),j}$ is realized, where $\kappa_j(\pi_t(j,k)) \in \{1, ..., K_j\}$ denotes the type of worker $\pi_t(j,k)$ w.r.t. job-type j. The realized rewards are independent across matches and in time.

Platform's objective. The goal of maximizing the expected cumulative payoffs over n rounds is converted to minimizing the expected *regret* relative to a clairvoyant policy that prescribes an "optimal" match for each arriving job. We are thus interested in the following optimization problem

$$\inf_{\pi\in\Pi} \mathbb{E}R_n^\pi := \inf_{\pi\in\Pi} \mathbb{E}\left[\sum_{t=1}^{n} \sum_{j\in\mathcal{J}:\Lambda_{j,t}\geqslant 1} \sum_{k=1}^{\Lambda_{j,t}} \left(\mu_{1,j} - \mu_{\kappa_j(\pi_t(j,k)),j}\right)\right]. \tag{1}$$

Here, Π is the class of *non-anticipating* policies, i.e., $\pi_{t+1}(\cdot,\cdot)$ is *adapted* to $\mathcal{F}_t$ for each $t \in \{0, 1, ...\}$, where $\mathcal{F}_t$ denotes the natural filtration at time t, i.e., $\mathcal{F}_t := \sigma\left\{(\boldsymbol{\Lambda_s}, \boldsymbol{\pi_s}, \boldsymbol{r_s}) : s \leqslant t\right\}$. Here, $\boldsymbol{\Lambda_s} := (\Lambda_{j,s} : j \in \mathcal{J})$, $\boldsymbol{\pi_s}$ is the set of matches implemented at time s and $\boldsymbol{r_s}$ is the set of collected rewards. The expectation in (1) is w.r.t. the randomness in job-arrivals, worker supply, policy, and rewards.

Going forward, we will adopt standard terminology from the multi-armed bandit literature and refer to workers as "arms" and jobs as "pulls" interchangeably.

3. HIGH-LEVEL OVERVIEW OF RESULTS

On the complexity of the problem. Even with a unique job-type, say $\mathcal{J} = \{j_0\}$, and only one job arriving per period, the ensuing *allocation* problem is challenging to analyze on account of the distribution $\boldsymbol{\alpha_{j_0}}$ and any statistical properties of the rewards being unknown. In the simplest possible formulation, $K_{j_0} = 2$, and the statistical complexity of the corresponding regret minimization problem is governed by three principal primitives: (i) the sub-optimality gap $\underline{\Delta}_{j_0} := \mu_{1,j_0} - \mu_{2,j_0} > 0$ between the mean rewards of the optimal and inferior worker sub-populations; (ii) the probability α_{1,j_0} of sampling an optimal worker from the population; and (iii) the planning horizon n. One may aptly recognize this as an infinitely many-armed bandit problem (where arms are synonymous to workers) with an *arm-reservoir* distribution $(\alpha_{1,j_0}, 1 - \alpha_{1,j_0})$ and a mean reward gap of $\underline{\Delta}_{j_0}$. However, this model differs from the classical literature on infinite-armed bandits in that its arm-reservoir distribution is not endowed with any regularity properties (see, e.g., [3]), instead we only posit a finite support with cardinality known to the decision maker (in this case, a cardinality of two), absent however, knowledge of the associated probability masses (in this case, $\alpha_{j_0,1}$ and $1-\alpha_{j_0,1}$). In our setting, absence of information on α_{1,j_0} significantly exacerbates the difficulty of analysis as calibrating *exploration* becomes challenging (on account of a "large" number of arms). In particular, how many arms must one query from the arm-reservoir in order to have at least one optimal arm in the queried set with high probability, is difficult to answer if (a lower bound on) the proportion α_{1,j_0} of optimal arms is unknown. Consequently, any finite consideration set may only contain inferior arms and as a result, any algorithm limited to such a selection will suffer a *linear regret.* One may contrast this setting with its classical two-armed counterpart with gap $\underline{\Delta}_{j_0}$ where the binary action space is key to designing rate-optimal policies. In our setting, on the other hand, it remains a priori unclear if there even exists a policy capable of achieving *sub-linear regret.*

Contributions. In this work, we resolve several foundational questions pertaining to complexity and achievable performance in the matching problem described earlier. We propose an algorithm that achieves a finite-time instance-dependent expected regret of $\mathcal{O}(\log n)$ after n rounds and prove that this performance cannot be improved w.r.t. n. While the order of regret and complexity of the problem suggests a great degree of similarity to the classical stochastic finite-armed bandit problem, properties of the performance bounds and salient aspects of algorithm design are quite distinct from the latter, as are the key primitives that determine complexity along with the analysis tools needed to study them. In what follows, we will for expositional reasons assume $\mathcal{J}=\{j_0\}$ with jobs arriving one at a time whenever $|\mathcal{J}|=1$. Our theoretical contributions can then summarized as under:

Complexity of regret when $|\mathcal{J}|=1$. We establish information-theoretic lower bounds on regret that are order-wise tight (in the horizon n) in the instance-dependent setting. In addition, we establish a *uniform* lower bound on achievable performance (also tight in n) that captures explicitly the scaling behavior w.r.t. the fraction α_{1,j_0} of optimal arms; this is shown via a novel non-information-theoretic proof based entirely on convex analysis.

Algorithm design and achievable performance. We propose a policy that is rate-optimal (in n) in the instance-dependent sense. Our policy only relies on knowledge of K_{j_0}, is agnostic to the distribution $\boldsymbol{\alpha_{j_0}}$ of worker-types as well as their rewards. Furthermore, its regret only has a *second-order* dependence on $\boldsymbol{\alpha_{j_0}}$ (see below).

Performance bounds for general $\mathcal{J}$. Aforementioned results for $|\mathcal{J}|=1$ and jobs arriving one at a time are then translated to the general (matching) version of the problem described earlier, where $\mathcal{J}$ can be any arbitrary finite set and jobs may arrive in batches of stochastic size and composition. In the matching problem, we establish that regret after any number $n\geqslant 1$ of rounds is bounded above by $\sum_{j\in\mathcal{J}}\left(C_1\left(\boldsymbol{\mu_j}\right)\log n+C_2\left(\boldsymbol{\mu_j},\boldsymbol{\alpha_j}\right)\log\log n\right)$ under our policy tailored to this setting, where the constants $C_1(\cdot), C_2(\cdot,\cdot)$ only depend on their arguments, $\boldsymbol{\mu_j}$ and $\boldsymbol{\alpha_j}$. Moreover, when $K_j=2$ for each $j\in\mathcal{J}$, we improve this guarantee to $\sum_{j\in\mathcal{J}}\left(C_1\left(\boldsymbol{\mu_j}\right)\log n+C_2\left(\boldsymbol{\mu_j},\boldsymbol{\alpha_j}\right)\right)$. It is noteworthy that the upper bound depends on $\{\boldsymbol{\alpha_j}: j\in\mathcal{J}\}$ only through $o(\log n)$ terms (*second-order* dependence). We believe this finding may be of interest more broadly.

4. THEORETICAL RESULTS

THEOREM 1 (INFORMATION-THEORETIC LOWER BOUNDS). *Fix $j\in\mathcal{J}$. Suppose that for each $t=1,2,...,$ we have $\Lambda_{j,t}=1$ and $\Lambda_{j',t}=0\ \forall\ j'\in\mathcal{J}\backslash\{j\}$. Also suppose that $K_j=2$ with $\alpha_{1,j}\leqslant 1/2-\epsilon$, where $\epsilon\in(0,1/2)$ is arbitrary. Let $\Pi_{\mathtt{adm}}$ denote the class of admissible policies. Then, the following is true under any $\pi\in\Pi_{\mathtt{adm}}$:*

(i) For any $\underline{\Delta}_j>0$, there exists a problem instance ν such that $\mathbb{E}R_n^\pi(\nu)\geqslant C\log n/\underline{\Delta}_j$ for n large enough (depending on ϵ), where C is some absolute constant.

(ii) For any $n\in\mathbb{N}$, there exists a problem instance ν such that $\mathbb{E}R_n^\pi(\nu)\geqslant\epsilon C\sqrt{n}$.

THEOREM 2 ($\boldsymbol{\alpha_j}$-DEPENDENT LOWER BOUND). *Fix $j\in\mathcal{J}$. Suppose that for each $t=1,2,...,$ we have $\Lambda_{j,t}=1$ and $\Lambda_{j',t}=0\ \forall\ j'\in\mathcal{J}\backslash\{j\}$. Also suppose that $\alpha_{1,j}\leqslant 1/2$. Denote by $\Pi_{\mathtt{m}}$ the class of "memoryless" policies under which the decision to match an incoming job to a new worker at any time $t\in\{1,2,...\}$ is independent of $\mathcal{F}_{t-1}$. Then, for all problem instances ν with a minimal sub-optimality gap of at least $\underline{\Delta}_j>0$, $\liminf_{n\to\infty}\inf_{\pi\in\Pi_{\mathtt{m}}}\mathbb{E}R_n^\pi(\nu)/\log n\geqslant\underline{\Delta}_j/4\alpha_{1,j}$.*

Remarks. (i) It is not impossible to avoid $1/\alpha_{1,j}$-scaling in the instance-dependent logarithmic regret. We will next show via an upper bound for a policy called `MATCH` that the $\alpha_{1,j}$-dependence can, in fact, be relegated to sub-logarithmic terms (`MATCH` samples new workers from the population *adaptively* based on the sample-history of onboarded workers and therefore does not belong to $\Pi_{\mathtt{m}}$). Importantly, this will establish a somewhat surprising fact that the instance-dependent logarithmic bound in Theorem 1 is optimal w.r.t. to its dependence on $\alpha_{1,j}$ **(ii)** Theorem 2 holds also for any worker supply where the optimal mean reward w.r.t. job-type j is at least $\underline{\Delta}_j$-separated from the rest, the nature of worker-types (countable or uncountable) notwithstanding.

THEOREM 3 (ACHIEVABLE PERFORMANCE). *Denote the policy `MATCH` by π. Then, after any number $n\geqslant 1$ of rounds,*

$$\mathbb{E}R_n^\pi\leqslant CM\sum_{j\in\mathcal{J}}\left[\frac{K_j^3\bar{\Delta}_j}{\beta_{\delta_j,K_j}}\left(\frac{\log n}{{\delta_j}^2}+\frac{\log\log\left(n+2\right)}{K_j!\prod_{i=1}^{K_j}\alpha_{i,j}}\right)\right], \quad (2)$$

where $\bar{\Delta}_j:=\mu_{1,j}-\mu_{K_j,j}$, $\delta_j:=\min_{1\leqslant i<i'\leqslant K_j}\left(\mu_{i,j}-\mu_{i',j}\right)$, β_{δ_j,K_j} is a constant that depends exclusively on (δ_j,K_j), and C is some absolute constant.

Remark. When $K_j=2\ \forall\ j\in\mathcal{J}$, the $\mathcal{O}(\log\log n)$ term in (2) can be improved to $\mathcal{O}(1)$. In fact, we conjecture this to be true also for $K_j>2$; pursuits are left to future work.

5. REFERENCES

[1] DERMAN, C., LIEBERMAN, G. J., AND ROSS, S. M. A sequential stochastic assignment problem. *Management Science 18*, 7 (1972), 349–355.

[2] JOHARI, R., KAMBLE, V., AND KANORIA, Y. Matching while learning. *Operations Research 69*, 2 (2021), 655–681.

[3] WANG, Y., AUDIBERT, J.-Y., AND MUNOS, R. Algorithms for infinitely many-armed bandits. In *Advances in Neural Information Processing Systems* (2009), pp. 1729–1736.