Elastic Job Scheduling with Unknown Utility Functions

Xinzhe Fu and Eytan Modiano

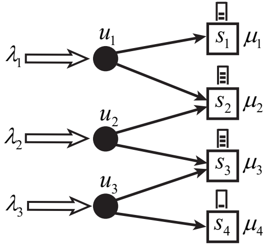
MIT LIDS

November 11, 2021

Elastic Job Scheduling

Problem Setting:

- Decide on sizes of incoming jobs.
- Route the jobs to servers.
- Maximize the total utility.
- Receive observations of utility values upon jobs' completion.



Applications:

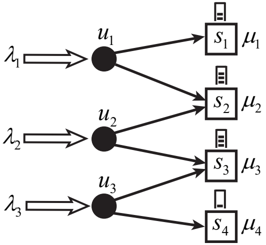
- Jobs are malleable in nature.
  - ML-training tasks [1].
- Utility functions are unknown apriori.

[1] H. Zhang, L. Stafman, A. Or, and M. Freedman. "Slaq: quality-driven scheduling for distributed machine learning." In Proceedings of the 2017 Symposium on Cloud Computing, 2017.

Problem Formulation

Consider a discrete-time system with K classes, for $t = 1, \ldots, T$:

- $a_k(t)$ jobs of classes k arrive at dispatcher u_k.
- u_k is connected to subset of servers S_{u_k}.
- Decide the "job-size" $x_k(t) \in [0, B]$.
- Route the jobs to servers.
- Server s_m's offered service rate $c_m(t)$.
- Queue length at server s_m: $Q_m(t+1) = [Q_m(t) + Arr_m(t) - c_m(t)]^+$.



Problem Formulation

Assumptions:

- Each class k is associated with a utility function f_k.
  - Utility functions are unknown apriori.
  - Each f_k is non-decreasing, concave and L-Lipschitz-continuous.
- For a class k job of size x_k, we observe and obtain $f_k(x_k) + \epsilon_i$ when the job is completed.
  - ϵ_i's are i.i.d. noises.
  - $\mathbb{E}[\epsilon_i] = 0, |\epsilon_i| \leq C$.
- Arrivals $a_k(t)$'s are i.i.d. Offered services $c_m(t)$'s are i.i.d.
  - $\mathbb{E}[a_k(t)] = \lambda_k, \mathbb{E}[c_m(t)] = \mu_m$. $0 \leq a_k(t), c_m(t) \leq C$.
  - $a_k(t)$'s, $c_m(t)$'s are observable but λ_k's, μ_m's are unknown.

Problem Formulation

Goal:

- Design a policy π that decides on the sizes and designated servers of incoming jobs such that the expected utility obtained by the end of the time horizon T is maximized.

Performance Metric: Regret

- $U(\pi, T)$: the expected utility obtained under policy π.
- π^*: the policy that maximizes the expected utility.
- Regret of π: $R(\pi, T) = U(\pi^*, T) - U(\pi, T)$.

Bounds on the Utility

Consider the set Λ:

$$\Lambda := \left\{ (x_1, \ldots, x_K) \;\middle|\; \begin{array}{l} \exists \{\alpha\}_{km}, \text{ s.t.} \\ \sum_k \alpha_{km} \lambda_k x_k \le \mu_m \quad \forall s_m, \\ \alpha_{km} = 0, \quad \forall s_m \notin S_{u_k} \\ \sum_m \alpha_{km} = 1, \forall k \\ \alpha_{km} \ge 0, \quad \forall k, m \\ 0 \le x_k \le B, \quad \forall k \end{array} \right\},$$

- Stability region of the network.
- Λ involves unknown parameters $\boldsymbol{\lambda}, \boldsymbol{\mu}$.

Bounds on the Utility

Consider the optimization problem $\mathcal{P}$:

$$\mathcal{P}: \max_{\{x\}_k} \sum_{k=1}^{K} \lambda_k f_k(x_k)$$
$$\textbf{s.t.} \quad (x_1, \ldots, x_K) \in \Lambda$$
$$x_k \in [0, B], \quad \forall k.$$

Theorem

$U(\pi^*, T) \leq T \cdot OPT(\mathcal{P})$.

- Proof using Jensen's inequality.

Bounds on the Utility

Theorem

For any policy π, $U(\pi, T) \geq \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k \mathbb{E}[f_k(x_k(t))] - L \cdot \sum_{t=1}^{T} \sum_{m=1}^{M} \mathbb{E}[Q_m(T)]$.

- Expected utility of dispatched jobs minus the incomplete jobs in the queues.

Theorem

For any policy π, $R(\pi, T) \leq \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k \mathbb{E}[f_k(x_k^*) - f_k(x_k(t))] + L \cdot \sum_{t=1}^{T} \sum_{m=1}^{M} \mathbb{E}[Q_m(T)]$.

- Utility Regret: $\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k \mathbb{E}[f_k(x_k^*) - f_k(x_k(t))]$.
- Queueing Regret: $\sum_{t=1}^{T} \sum_{m=1}^{M} \mathbb{E}[Q_m(T)]$.
- Convex optimization with bandit feedback and unknown (stochastic) constraints.
- Regret lower bound: $\Omega(\sqrt{T})$.

Related Works

First-order Methods

- Use utility observations to construct approximate gradients of the utility functions [2][3].
- Cannot achieve order-optimal regret due to the variance of approximate gradients.

Zeroth-order Methods

- Maintain a target region that contains the optimal solution, and directly use utility observations to shrink the target region. Robust to noise, but cannot handle stochastic constraints [4].
- **Main Contribution:** Zeroth-order algorithm for elastic job scheduling with order-optimal regret.

---

[2] X. Fu and E. Modiano, "Learning-NUM: Network Utility Maximization with Unknown Utility Functions and Queueing Delay." in Mobihoc 2021.

[3] T. Chen and G. B. Giannakis. "Bandit convex optimization for scalable and dynamic IoT management." in IEEE Internet of Things Journal, 2018.

[4] A. Agarwal, D. Foster, D. Hsu, S. Kakade, and A. Rakhlin, "Stochastic convex optimization with bandit feedback." in SIAM Journal on Optimization, 2013.

The SCBA Algorithm

Stochastic Convex Bandit Algorithm [4]:

$$\max_{\boldsymbol{x}} F(\boldsymbol{x})$$
$$\textbf{s.t.} \quad \boldsymbol{x} \in \mathcal{X}$$

- F is concave and Lipschitz-continuous. item Query $\boldsymbol{x}(t)$, observe $F(\boldsymbol{x}(t)) + \epsilon_t$.
- Achieves $\tilde{O}(\sqrt{T})$-regret if $\mathcal{X}$ is known.
- Output a sequence of query points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ that $\sum_{t=1}^{T} \mathbb{E}[F(\boldsymbol{x}^*) - F(\boldsymbol{x}(t))] \leq \tilde{O}(\sqrt{T})$.

[4] A. Agarwal, D. Foster, D. Hsu, S. Kakade, and A. Rakhlin, "Stochastic convex optimization with bandit feedback." in SIAM Journal on Optimization, 2013.

The SCBA Algorithm

Stochastic Convex Bandit Algorithm:

- Maintain a target region (initialized as $\mathcal{X}$).
- Repeatedly query a set of points in the target region.
- One query corresponds to one x_t. The x_t for multiple t's may take the same value.
- Construct confidence intervals of function values of the queried points.
- Eliminate a fraction of the target region (thereby shrink the target region) based on the constructed confidence intervals.

Transformed Optimization Problem

$$\mathcal{P}: \max_{\boldsymbol{x}} \sum_{k=1}^{K} \lambda_k f_k(x_k) \quad \text{s.t.} \quad \boldsymbol{x} \in \Lambda, \quad x_k \in [0, B], \quad \forall k.$$

$$\tilde{\mathcal{P}}: \max_{\boldsymbol{x}} \; F(\boldsymbol{x}) := \sum_{k=1}^{K} \lambda_k f_k(x_k) - L \cdot \Delta(\boldsymbol{x}, \Lambda) \quad \text{s.t.} \quad x_k \in [0, B], \quad \forall k.$$

- $\Delta(\cdot, \Lambda)$ denotes the l_1-distance from $\boldsymbol{x}$ to Λ.

Theorem

$\mathcal{P}$ and $\tilde{P}$ are equivalent.

- $\mathcal{P}$ and $\tilde{\mathcal{P}}$ have the same optimal solution.
- $\tilde{\mathcal{P}}$ is a convex problem with Lipschitz-continuous objective function.

Transformed Optimization Problem

$$\tilde{\mathcal{P}}: \ \max_{\boldsymbol{x}} \ F(\boldsymbol{x}) := \sum_{k=1}^{K} \lambda_k f_k(x_k) - L \cdot \Delta(\boldsymbol{x}, \Lambda)$$
$$\textbf{s.t.} \quad x_k \in [0, B], \quad \forall k.$$

- $\tilde{\mathcal{P}}$ does not involve stochastic constraints.
- However, there is no unbiased observations available for the objective function $F(\boldsymbol{x})$.
  - Design a special procedure to construct confidence intervals for $F(\boldsymbol{x})$.

Confidence Interval Construction

If $\boldsymbol{x}(t) = \boldsymbol{x}$ for $t \in \{t_1, t_1 + 1, \ldots, t_2\}$, how to construct confidence intervals around $F(\boldsymbol{x})$?

- Assume (for now) utility observation is immediately available after the corresponding job size decision.
- Using empirical means, we have:
  - Confidence interval of $f_k(x_k)$: $[f_k^l(x_k), f_k^u(x_k)]$.
  - Confidence interval of λ_k: $[\lambda_k^l, \lambda_k^u]$.
  - Confidence interval of μ_m: $[\mu_m^l, \mu_m^u]$.
- These confidence intervals can propagate to a confidence interval of $F(\boldsymbol{x})$.

Confidence Interval Construction

Recall the set Λ:

$$\Lambda := \left\{ (x_1, \ldots, x_K) \;\middle|\; \begin{array}{l} \exists\{\alpha\}_{km}, \text{ s.t.} \\ \sum_k \alpha_{km}\lambda_k x_k \le \mu_m \quad \forall s_m, \\ \alpha_{km} = 0, \quad \forall s_m \notin S_{u_k} \\ \sum_m \alpha_{km} = 1, \forall k \\ \alpha_{km} \ge 0, \quad \forall k, m \\ 0 \le x_k \le B, \quad \forall k \end{array} \right\},$$

- $\Delta(\boldsymbol{x}, \Lambda)$ is parameterized by $\Delta(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$, where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_K)$, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_M)$.

Confidence Interval Construction

- $\Delta(\boldsymbol{x}, \Lambda)$ is parameterized by $\Delta(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$, where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_K)$, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_M)$.
- $\Delta(\boldsymbol{x}, \boldsymbol{\lambda}^u, \boldsymbol{\mu}^l) \geq \Delta(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \geq \Delta(\boldsymbol{x}, \boldsymbol{\lambda}^l, \boldsymbol{\mu}^u)$.
- $F^l(\boldsymbol{x}) := \sum_k \lambda_k^l f_k^l(x_k) - L \cdot \Delta(\boldsymbol{x}, \boldsymbol{\lambda}^u, \boldsymbol{\mu}^l)$.
- $F^u(\boldsymbol{x}) := \sum_k \lambda_k^u f_k^u(x_k) - L \cdot \Delta(\boldsymbol{x}, \boldsymbol{\lambda}^l, \boldsymbol{\mu}^u)$.
- $[F^l(\boldsymbol{x}), F^u(\boldsymbol{x})]$ satisfies the requirements of SCBA.
- SCBA algorithm embedded with the confidence interval construction procedure can be used to solve $\tilde{\mathcal{P}}$.

The CI-JSQ Policy

Routing

- The routing is done based on the Join-the-Shortest-Queue (JSQ) rule.
- Send the jobs to the server with the smallest $Q_m(t)$.

Confidence Interval Join-the-Shortest-Queue (CI-JSQ) Policy:

- For each time t, decide on the job size based on the output of SCBA embedded with the confidence interval construction procedure.
- Route the jobs based on the JSQ rule.

The CI-JSQ Policy

Theorem

The CI-JSQ policy achieves $\tilde{O}(\sqrt{T})$-regret.

- $\sum_{t=1}^{T} \mathbb{E}[F(\boldsymbol{x}^*(t)) - F(\boldsymbol{x}(t))] \leq \tilde{O}(\sqrt{T})$ from the guarantee of SCBA.
- Can show that:
  - $\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k \mathbb{E}[f_k(x_k^*) - f_k(x_k(t))] \leq \tilde{O}(\sqrt{T})$ (utility regret).
  - $\sum_{t=1}^{T} \mathbb{E}[\Delta(\boldsymbol{x}(t), \Lambda)] \leq \tilde{O}(\sqrt{T})$ (constraint violation).
- Using properties of JSQ, the queueing regret is also in $\tilde{O}(\sqrt{T})$ from the constrain violation bound.
- An episodic scheme (with the same order of regret) that deals with **feedback delay**.

Conclusion

- We considered the problem of elastic job scheduling with unknown utility functions.
- The problem can be viewed as a problem of convex optimization with bandit feedback and stochastic constraints.
- We combined the SCBA with a confidence interval construction procedure to form a policy with order-optimal regret for the elastic job scheduling problem.

Thank You!

– xinzhe@mit.edu