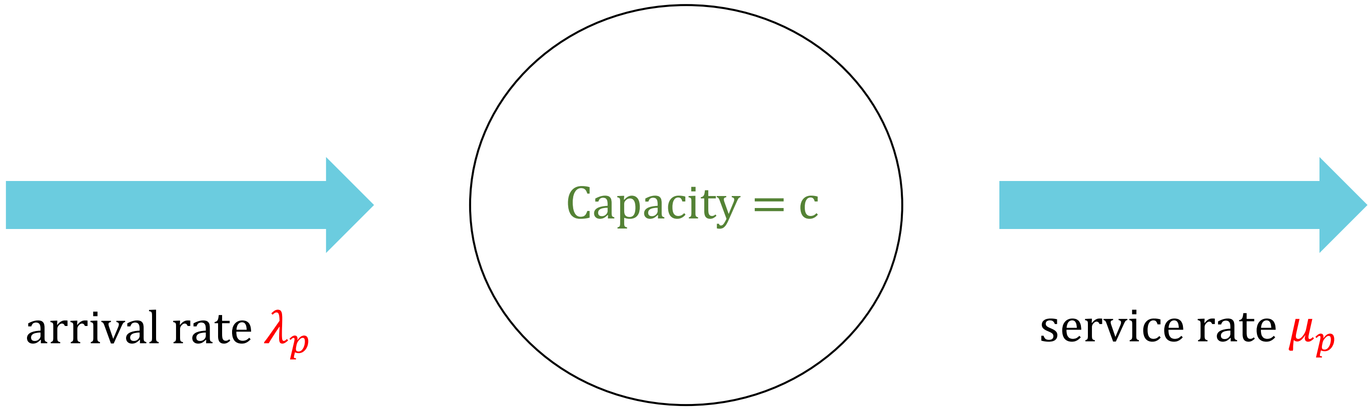
Online Learning and Pricing with Reusable Resources: Linear Bandits with Sub-Exponential Rewards

Huiwen Jia, Cong Shi, Siqian Shen

INDUSTRIAL & OPERATIONS ENGINEERING
UNIVERSITY OF MICHIGAN

Problem Modelling: Online Learning and Pricing with Reusable Resource

- A firm is endowed with a finite capacity of c reusable products
- In each period t, the firm posts a price $p \in [p_L, p_U]$
- Customers arrive according to a Poisson process with rate λ_p and they are served on a first-arrive-first-serve basis by occupying one unit of resource following an exponential distribution with rate μ_p
- Goal: maximizing revenue



The firm does not know the arrival rates λ_p and the service rates μ_p *a priori*

Problem Modelling: Linear Bandits with Sub-Exponential Rewards

Price p $\rightarrow$ Feature Vector $\mathbf{x}_p \in \mathbb{R}^{d_f}$

Assume linear mappings: $\frac{1}{\lambda_p} = \theta_\lambda^T \mathbf{x}_p$, $\frac{1}{\mu_p} = \theta_\mu^T \mathbf{x}_p$

Offer price:

p_1

p_2

...

p_N

Arrival time intervals

- Count: $n_m(p)$
- Observation: $\hat{d}_i(p)$, $i = 1, \ldots, n_m(p)$
- Empirical Mean: $\bar{d}_p = \sum_{i=1}^{n_m(p)} \hat{d}_i(p)/n_m(p)$

- $\bar{d}_p$ follows an Erlang distribution $\mathrm{Erlang}(n_m(p), n_m(p)\lambda_p)$
- $\bar{d}_p \sim \mathrm{SE}(\frac{4}{n_m(p)\lambda_p^2}, \frac{2}{n_m(p)\lambda_p})$
- $\bar{d}_p = \theta_\lambda^T \mathbf{x}_p + \epsilon_p$ and $\epsilon_p \sim \mathrm{SE}(\frac{4}{n_m(p)\lambda_p^2}, \frac{2}{n_m(p)\lambda_p})$

Problem Modelling: Linear Bandits with Sub-Exponential Rewards

Price p → Feature Vector $\mathbf{x}_p \in \mathbb{R}^{d_f}$

Assume linear mappings: $\frac{1}{\lambda_p} = \theta_\lambda^T \mathbf{x}_p$, $\frac{1}{\mu_p} = \theta_\mu^T \mathbf{x}_p$

Offer price:

p_1

p_2

...

p_N

Arrival time intervals

- Count: $n_m(p)$
- Observation: $\hat{d}_i(p)$, $i = 1, \ldots, n_m(p)$
- Empirical Mean: $\bar{d}_p = \sum_{i=1}^{n_m(p)} \hat{d}_i(p)/n_m(p)$

- $\bar{d}_p$ follows an Erlang distribution $\mathrm{Erlang}(n_m(p), n_m(p)\lambda_p)$
- $\bar{d}_p \sim \mathrm{SE}(\frac{4}{n_m(p)\lambda_p^2}, \frac{2}{n_m(p)\lambda_p})$
- $\bar{d}_p = \theta_\lambda^T \mathbf{x}_p + \epsilon_p$ and $\epsilon_p \sim \mathrm{SE}(\frac{4}{n_m(p)\lambda_p^2}, \frac{2}{n_m(p)\lambda_p})$

Data

- $\mathbf{X} = [x_{p_1}, x_{p_2}, \ldots, x_{p_N}]^T$: features
- $\mathbf{d} = [\bar{d}_{p_1}, \bar{d}_{p_2}, \ldots, \bar{d}_{p_N}]^T$ mean arrival time
- $\hat{\Omega}_\lambda$ with i^{th} element $\frac{\bar{d}_{p_i}^2}{n_m(p_i)}$

Estimate

$$\hat{\theta}_\lambda = (\mathbf{X}^T \hat{\Omega}_\lambda^{-1} \mathbf{X})^{-1} \mathbf{X}^T \hat{\Omega}_\lambda^{-1} \mathbf{d}$$

Proposition 2. *Consider N implemented prices with $N \geq d_f$ and $n_m(p) \geq 8\log(T)$ for any implemented price p. Then, for a new feature vector $\mathbf{x}'$:*

$$\mathbb{P}\left(\frac{|\hat{\theta}_\lambda^T \mathbf{x}' - \theta_\lambda^T \mathbf{x}'|}{\sqrt{\mathbf{x}'^T (\mathbf{X}^T \Omega^{-1} \mathbf{X})^{-1} \mathbf{x}'}} \geq \sqrt{32\log(T)}\right) \leq \frac{2}{T^4}.$$

With Service time intervals

Proposition 3. *For price p with a feature vector $\mathbf{x}$, we have:*

$$\mathbb{P}\left(\left|\frac{\lambda_p}{\mu_p} - \frac{\hat{\theta}_\mu^T \mathbf{x}}{\hat{\theta}_\lambda^T \mathbf{x}}\right| \leq \frac{\sqrt{32\log(T)}}{\hat{\theta}_\lambda^T \mathbf{x}} \mathcal{G}\right) \geq 1 - \frac{4}{T^4},$$

where

$$\mathcal{G} = \left(r_{\max}\sqrt{\mathbf{x}^T (\mathbf{X}^T \hat{\Omega}_\lambda^{-1} \mathbf{X})^{-1} \mathbf{x}} + \sqrt{\mathbf{x}^T (\mathbf{X}^T \hat{\Omega}_\mu^{-1} \mathbf{X})^{-1} \mathbf{x}}\right).$$

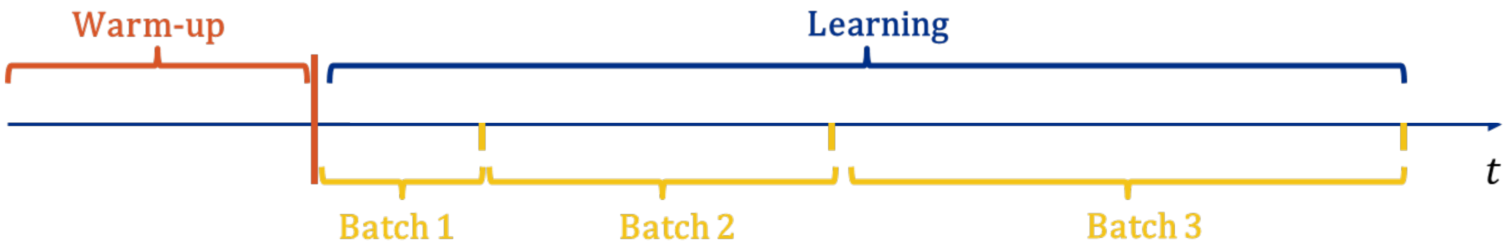
Online Batch Linear Upper Confidence Bound Algorithm

Algorithm 1 Online Batch LinUCB Algorithm (BLinUCB).

Input: T, p_L, p_U, d_f.
Initialize: $\tau, I_m, M, \mathcal{P}_b$ as in Section 4.2.
Warm-up Phase:
for $p \in \mathcal{P}_b$ **do**
  Offer price p, record $\hat{d}_i(p)$ for arriving customers and $\hat{g}_i(p'), \forall p' \in [p_L, p_U]$ for leaving customers.
  if $n_m^s(p) \geq 8\log(T)$ **then**
    Update $\mathbf{X}, \hat{\Omega}_\lambda, \hat{\Omega}_\mu, \mathbf{d}, \mathbf{y}_\mu$
    Continue.
  end if
end for
Compute $\hat{\theta}_\lambda$ and $\hat{\theta}_\mu$ by (5) and (6)
Learning Phase:
for $m = 1, \ldots, M$ **do**
  Choose $p_m = \text{argmax}_{p\in[p_L,p_U]} U_{m-1}(p)$.
  Offer p_m in batch m, i.e., for $I_m\tau$ periods.
  Record $\hat{d}_i(p_m)$ for arriving customers and $\hat{g}_i(p), \forall p \in \mathcal{P}$ for leaving customers.
  Update $\mathbf{X}, \hat{\Omega}_\lambda, \hat{\Omega}_\mu, \mathbf{d}, \mathbf{y}_\mu$; Compute $\hat{\theta}_\lambda$ and $\hat{\theta}_\mu$.
end for

Online Batch Linear Upper Confidence Bound Algorithm

- Initialize parameters $\tau = (log(T))^2, I_m = 2^m$ $\quad I_m\tau$



Definition 3. The upper confidence bound of the revenue rate associated with price p by the end of batch m is:

$$U_m(p) = \left(\frac{\hat{\theta}_\mu^T \mathbf{x}}{\hat{\theta}_\lambda^T \mathbf{x}} + \frac{\sqrt{32 \log(T)}}{\hat{\theta}_\lambda^T \mathbf{x}} \mathcal{G} \right) p.$$

Algorithm 1 Online Batch LinUCB Algorithm (BLinUCB).

Input: T, p_L, p_U, d_f.
Initialize: $\tau, I_m, M, \mathcal{P}_b$ as in Section 4.2.
Warm-up Phase:
for $p \in \mathcal{P}_b$ **do**
  Offer price p, record $\hat{d}_i(p)$ for arriving customers and $\hat{g}_i(p'), \forall p' \in [p_L, p_U]$ for leaving customers.
  if $n_m^s(p) \geq 8 \log(T)$ **then**
    Update $\mathbf{X}, \hat{\Omega}_\lambda, \hat{\Omega}_\mu, \mathbf{d}, \mathbf{y}_\mu$
    Continue.
  end if
end for
Compute $\hat{\theta}_\lambda$ and $\hat{\theta}_\mu$ by (5) and (6)
Learning Phase:
for $m = 1, \ldots, M$ **do**
  Choose $p_m = \text{argmax}_{p \in [p_L, p_U]} U_{m-1}(p)$.
  Offer p_m in batch m, i.e., for $I_m\tau$ periods.
  Record $\hat{d}_i(p_m)$ for arriving customers and $\hat{g}_i(p), \forall p \in \mathcal{P}$ for leaving customers.
  Update $\mathbf{X}, \hat{\Omega}_\lambda, \hat{\Omega}_\mu, \mathbf{d}, \mathbf{y}_\mu$; Compute $\hat{\theta}_\lambda$ and $\hat{\theta}_\mu$.
end for

Performance

Theoretical:

$$\mathcal{R}_T := \mathbb{E}\sum_{t=1}^{T} J_t^{\pi^*} - \mathbb{E}\sum_{t=1}^{T} J_t^{\pi}$$

Clairvoyant OPT

Learning Policy

Theorem 1. *The T-period cumulative regret of BLinUCB is bounded by $\tilde{O}\left(d_f\sqrt{T}\right)$.*

Performance

Theoretical:

$$\mathcal{R}_T := \mathbb{E}\sum_{t=1}^{T} J_t^{\pi^*} - \mathbb{E}\sum_{t=1}^{T} J_t^{\pi}$$

Clairvoyant OPT — Learning Policy

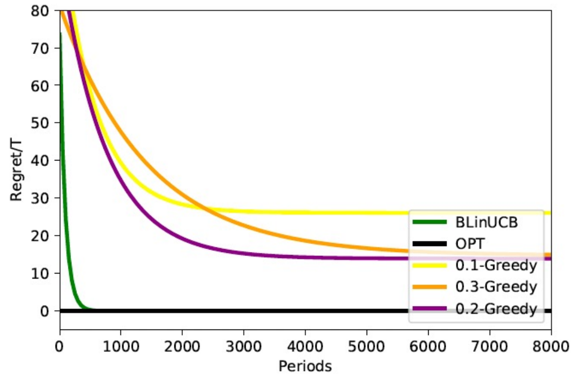
Theorem 1. *The T-period cumulative regret of BLinUCB is bounded by $\tilde{O}\left(d_f\sqrt{T}\right)$.*

Empirical:

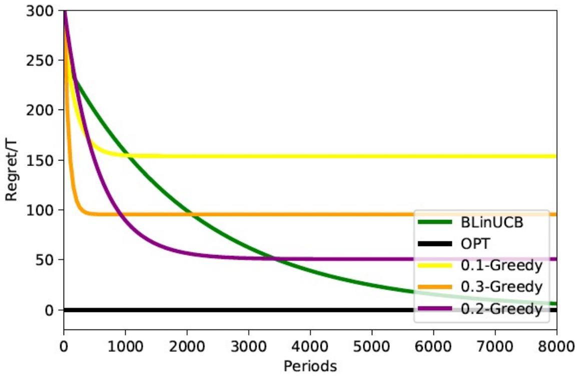
Time Average Regret

$$\frac{\mathbb{E}\sum_{t=1}^{T} J_t^{\pi^*} - \mathbb{E}\sum_{t=1}^{T} J_t^{\pi}}{T}$$

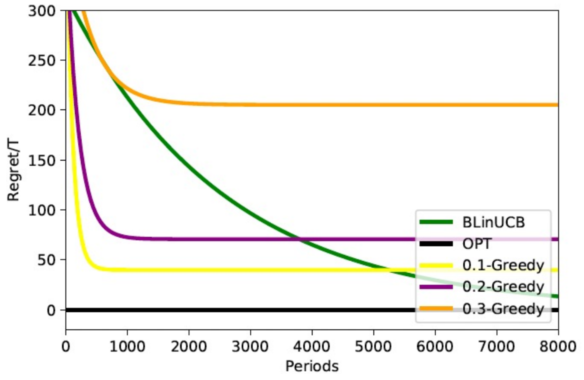
Instances #1



Instances #2



Instances #3



BLinUCB **0.1-greedy** **0.2-greedy** **0.3-greedy** **OPT**

ϵ −Greedy benchmark: chooses $\text{argmax}_p \frac{\hat{\theta}_\mu^T \mathbf{x}}{\hat{\theta}_\lambda^T \mathbf{x}} p$ with probability $1-\epsilon$; randomly chooses other price with ϵ

BLinUCB performs very well. The time average regret converges to 0 quickly.

Thank You!!