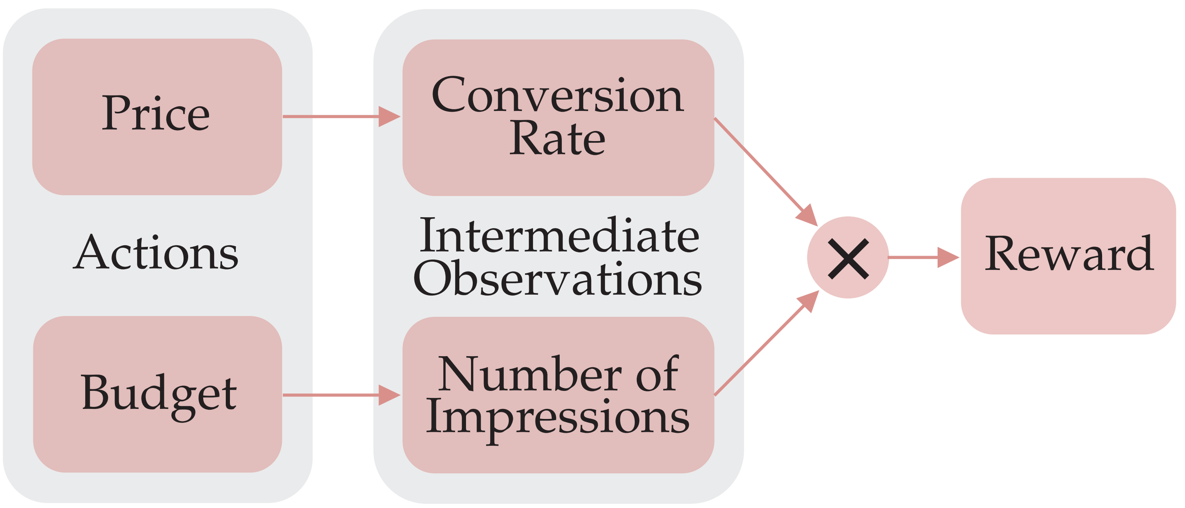
Factored-Reward Bandits with Intermediate Observations

Marco Mussi*, Simone Drago*, Marcello Restelli and Alberto Maria Metelli
{marco.mussi, simone.drago, marcello.restelli, albertomaria.metelli}@polimi.it
Politecnico Milano 1863

Example: Joint Pricing–Advertising



Why not Standard MAB?

We can solve this problem using **standard** Multi-Armed Bandit techniques considering the **price-budget couples** as actions, at the cost of an:

- **unnecessarily large action space** ($|\mathcal{A}| = \prod_{i \in [\![d]\!]} k_i$)
- **amplified heavy-tailed noise** effect

Factored-Reward Bandits (FRB)

We choose an **action vector**:

$$\mathbf{a}(t) = (a_1(t), \dots, a_d(t)) \in \mathcal{A} := [\![k_1]\!] \times \dots \times [\![k_d]\!]$$

We observe a vector of d **intermediate observations**:

$$\mathbf{x}(t) = (x_1(t), \dots, x_d(t))$$

with:

$$x_i(t) = \underbrace{\mu_{i,a_i(t)}}_{\text{Expected intermediate observation of } a_i(t) \text{ (with } \mu_{i,j} \in [0,1])} + \underbrace{\epsilon_i(t)}_{\sigma^2\text{-subgaussian noise}}$$

We receive a **reward**: $r(t) = \prod_{i \in [\![d]\!]} x_i(t)$

We consider $k_i = k, \ \forall i \in [\![d]\!]$ for *simplicity*.

Learning Problem

Optimal **action vector**:

$$\mathbf{a}^* = (a_1^*, \dots, a_d^*) \in \times_{i \in [\![d]\!]} \arg\max_{a_i \in [\![k_i]\!]} \mu_{i,a_i}$$

Optimal **expected reward**:

$$\prod_{i \in [\![d]\!]} \max_{a_i \in [\![k_i]\!]} \mu_{i,a_i} = \prod_{i \in [\![d]\!]} \mu_i^* = \mu^*$$

Suboptimality gaps: $\Delta_{i,a_i} := \mu_i^* - \mu_{i,a_i}$

Goal is to minimize the **expected cumulative regret**:

$$\mathbb{E}\left[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})\right] = T\mu^* - \mathbb{E}\left[\sum_{t \in [\![T]\!]} \prod_{i \in [\![d]\!]} \mu_{i,a_i(t)}\right]$$

Lower Bounds

Worst-Case Lower Bound

$$\mathbb{E}\left[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})\right] \geqslant \Omega\left(\sigma d \sqrt{kT}\right)$$

Instance-Dependent Lower Bound

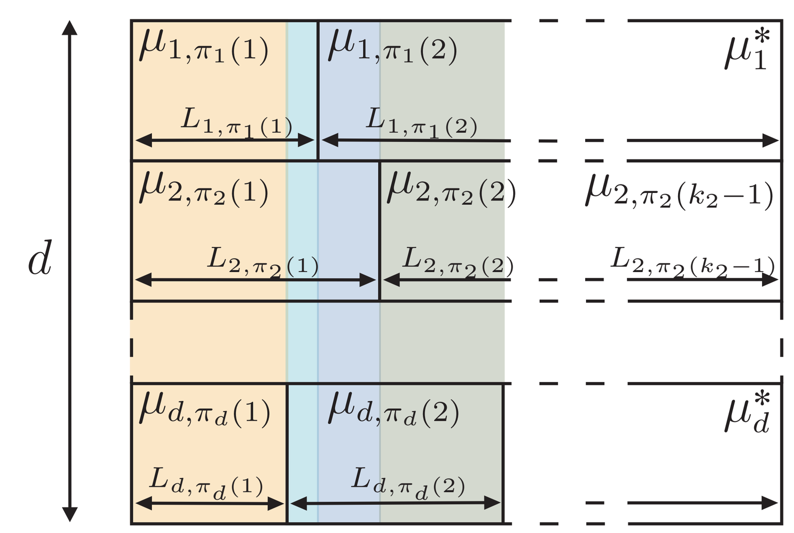
$$\liminf_{T \to +\infty} \frac{\mathbb{E}\left[R_T(\mathfrak{A}, \underline{\boldsymbol{\nu}})\right]}{\log T} \geqslant \underline{C}(\underline{\boldsymbol{\nu}})$$

- Every algorithm $\mathfrak{A}$ has to pull at least:
$$\frac{\mathbb{E}[N_{i,j}]}{\log T} \geqslant \frac{2\sigma^2}{\Delta_{i,j}^2}$$
times every suboptimal action component
- The lower bound is obtained by finding the combination of pulls minimizing the regret
- The naïve approach is to solve a **Linear Programming optimization problem**

Efficient Solution to the LP

Using **Rearrangement Inequality**

$\mathcal{O}(dk\log(k))$ complexity



Solution 1: Factored Upper Confidence Bound (F-UCB)

`F-UCB` is an **anytime optimistic regret minimization** algorithm that plays over the d different dimensions **independently**. In every dimension, the algorithm plays the action defined as:

$$\mathbf{a}(t) = \arg\max_{(a_1, \dots, a_d) \in \mathcal{A}} \prod_{i \in [\![d]\!]} \mathrm{UCB}_{i,a_i}(t)$$

where the **optimistic index** is: $\mathrm{UCB}_{i,a_i}(t) = \hat{\mu}_{i,a_i}(t-1) + \sigma\sqrt{\frac{\alpha \log t}{N_{i,a_i}(t-1)}}$

Worst-Case Upper Bound

$$\mathbb{E}\left[R_T(\texttt{F-UCB}, \underline{\boldsymbol{\nu}})\right] \leqslant \widetilde{\mathcal{O}}\left(\sigma d \sqrt{kT}\right)$$

Instance-Dependent Upper Bound

Implicit Upper Bound

- `F-UCB` pulls at most:
$$\mathbb{E}[N_{i,j}] \leqslant \frac{4\alpha\sigma^2 \log T}{\Delta_{i,j}^2}$$
times every suboptimal action component
- We want to find the worst combination of pulls
- Again, the naïve approach is to solve a Linear Programming optimization problem

Explicit Upper Bound

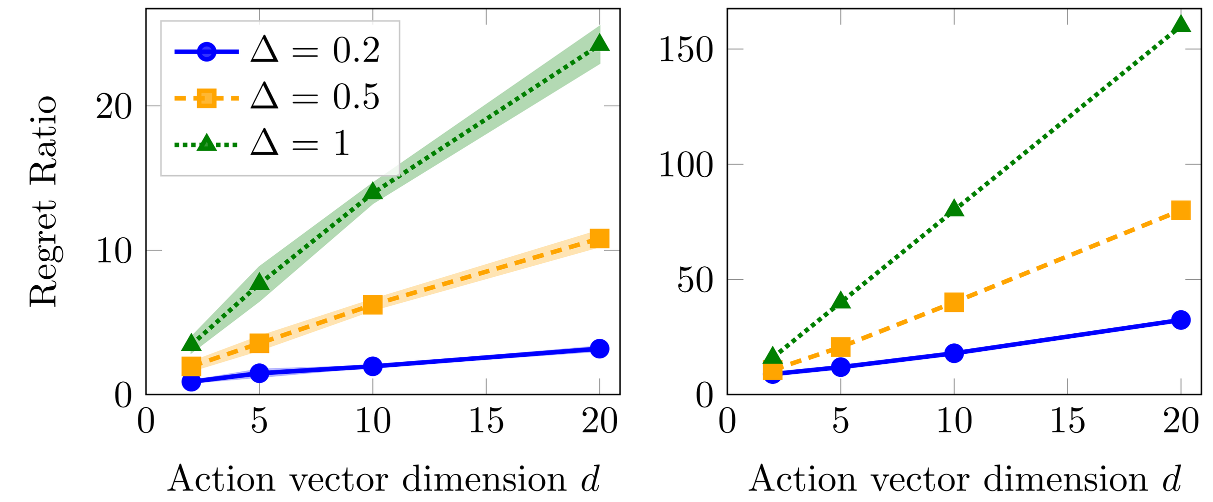
(*Rearrangement Inequality*, opposite direction)

$$\mathbb{E}\left[R_T(\texttt{F-UCB}, \underline{\boldsymbol{\nu}})\right] \leqslant 4\alpha\sigma^2 \log T \sum_{i \in [\![d]\!]} \mu_{-i}^* \sum_{j \in [\![k]\!] \setminus \{a_i^*\}} \Delta_{i,j}^{-1}$$

where $\mu_{-i}^* = \prod_{l \in [\![d]\!] \setminus \{i\}} \mu_l^* \leqslant 1, \forall i \in [\![d]\!]$

Solution 2: F-Track

F-UCB is Instance-Dependent Suboptimal in d



Solution: F-Track

`F-Track` **coordinates** among the d dimensions in three phases:

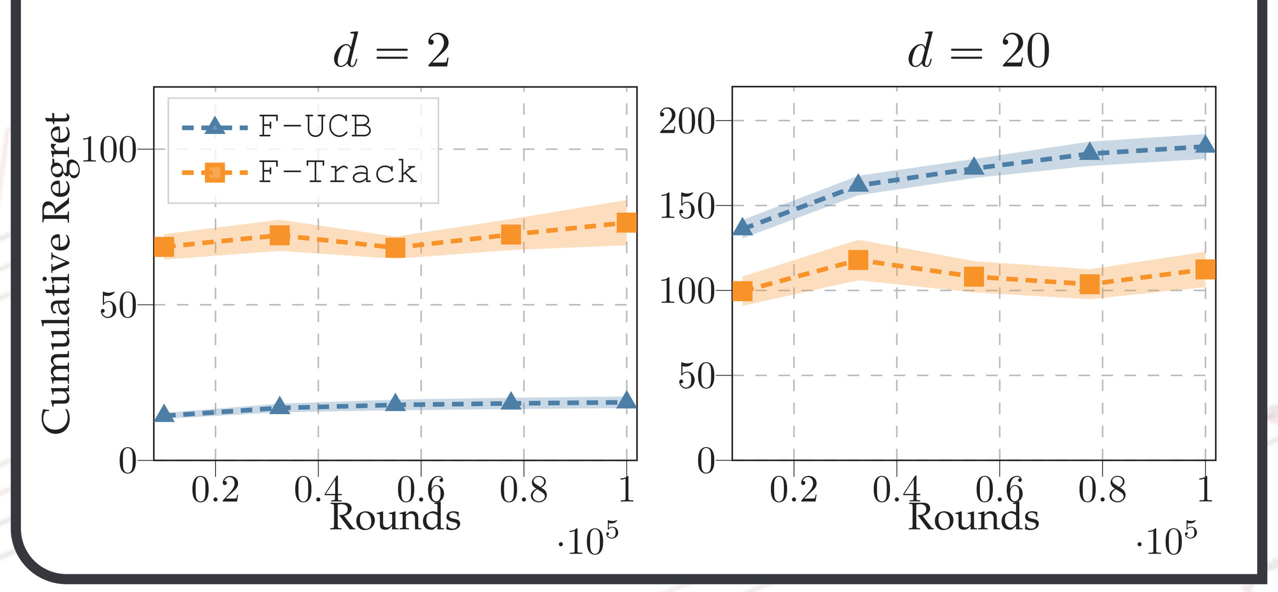
1. **Warm-up**: Play action vectors in round robin until every action component has been pulled at least a minimum amount of times
2. **LB Matching**: Use warm-up data to compute estimates of $\hat{\mu}_{i,j}$ and $\hat{\Delta}_{i,j}$. Solve the lower bound LP to define a pull schedule
3. **Recovery**: If, during phase 2, the estimation error of any $\hat{\mu}_{i,j}$ is discovered to invalidate the scheduling, fall back to `F-UCB` until $t = T$

Instance-Dependent Upper Bound

$$\limsup_{T \to +\infty} \frac{\mathbb{E}\left[R_T(\texttt{F-Track}, \underline{\boldsymbol{\nu}})\right]}{\log T} = \underline{C}(\underline{\boldsymbol{\nu}})$$

Experimental Results

Comparison between **`F-UCB`** and **`F-Track`** for different values of d. Setting: $k = 2$, $\mu^* = 1$, $\Delta = 0.7$.



References

S. Bubeck, N. Cesa-Bianchi, and G. Lugosi. Bandits with heavy tail. *IEEE Trans. Inf. Theory*, 2013.

T. Lattimore and C. Szepesvári. The end of optimism? an asymptotic analysis of finite-armed linear bandits. In *AISTATS*, 2017.

J. Zimmert and Y. Seldin. Factored bandits. In *NeurIPS*, 2018.