Exploiting Structure of Uncertainty for Efficient Matroid Semi-Bandits

Pierre Perrault[1,2], Vianney Perchet[2,3], Michal Valko[1]

[1]SequeL team, INRIA Lille [2]CMLA, ENS Paris-Saclay [3]Criteo Research

Challenge and contribution

Optimism: Play $\arg\max \underbrace{L}_{Empiric} + \underbrace{F}_{Bonus}$

✗ **Issue**: **Inefficient** for **accurate** F when the action space $\mathcal{A}$ is combinatorial.

✓ **Contribution**: **Efficient** and **accurate enough** approximation algorithm for $\mathcal{A}$ given by a *matroid*.

Setting

Semi-bandit feedback: $X_{i,t}$ revealed $\forall i \in A_t$.
Rewards: $\boldsymbol{X} \in \mathbb{R}^n$, means: $\mathbb{E}[\boldsymbol{X}] \triangleq \boldsymbol{\mu}^\star$, action space: $\mathcal{A}$.
Purpose: Design policy minimizing the **expected regret**

$$R_T \triangleq \mathbb{E}\left[\sum_{t\le T} \left(\mathbf{e}_{A^\star} - \mathbf{e}_{A_t}\right)^\top \boldsymbol{\mu}^\star\right].$$

Example: Building a spanning tree for network routing [1].

Semi-bandits confidence regions

Many algorithms [2, 3, 4, 5] minimize R_T applying OFU principle with a *confidence region* $\mathcal{C}_t$ around $\overline{\boldsymbol{\mu}}_{i,t-1}$ (so that $\boldsymbol{\mu}^\star \in \mathcal{C}_t$ w.h.p.).

Empirical average: $\overline{\boldsymbol{\mu}}_{i,t-1} \triangleq \frac{1}{N_{i,t}} \sum_{u\le t-1} \mathbb{I}\{i \in A_u\} X_{i,u}$.

Optimism in Face of Uncertainty (OFU) principle: at each round, solve the bilinear program

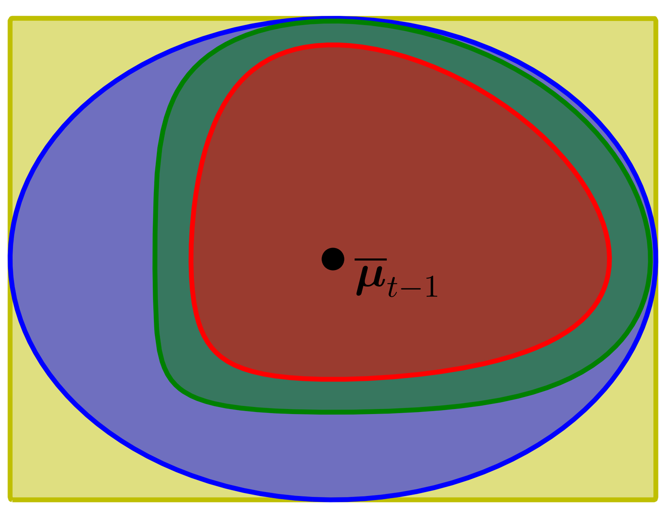
$$(\boldsymbol{\mu}_t, A_t) \in \arg\max_{\boldsymbol{\mu}\in\mathcal{C}_t, A\in\mathcal{A}} \mathbf{e}_A^\top \boldsymbol{\mu}\,. \qquad (1)$$

$\mathcal{C}_t$ is generally of the form

$$\mathcal{C}_t = \overline{\boldsymbol{\mu}}_{t-1} + \left\{\boldsymbol{\delta} \in \mathbb{R}^n, \left\|(g_{i,t}(\delta_i))_i\right\|_p \le 1\right\}, \qquad (2)$$

- $g_{i,t}$ is convex, such that $g_{i,t}(0) = g'_{i,t}(0) = 0$.
- $p \in \{1, \infty\}$.

Examples of $\mathcal{C}_t$:



- Cartesian product of intervals
- Ellipsoid
- Sub-Gaussian based
- *kl* ball

Submodular maximization

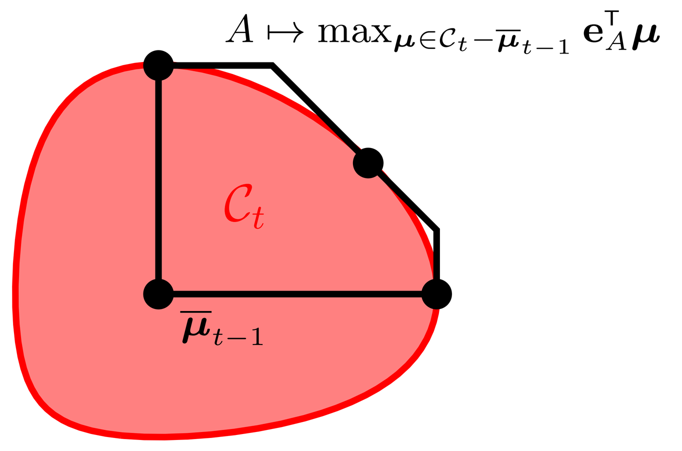
We want to maximize

$$A \mapsto \max_{\boldsymbol{\mu}\in\mathcal{C}_t} \mathbf{e}_A^\top \boldsymbol{\mu} = \mathbf{e}_A^\top \overline{\boldsymbol{\mu}}_{t-1} + \max_{\boldsymbol{\mu}\in\mathcal{C}_t - \overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\top \boldsymbol{\mu} = L(A) + F(A).$$

Theorem. *If $\mathcal{C}_t$ is of the form (2), then F is*

- *linear if $p = \infty$,*
- *submodular if $p = 1$.*

Polymatroid defined by $A \mapsto \max_{\boldsymbol{\mu}\in\mathcal{C}_t - \overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\top \boldsymbol{\mu}$



Example 1. $g_{i,t}(\delta) = \delta^2 \alpha_{i,t}$. $F(A) = \sqrt{\mathbf{e}_A^\top \left(\frac{1}{\alpha_{i,t}}\right)_i}$.

Remark: $L + F$ is either linear or submodular. In the first case, maximization is efficient, in the second it is NP-Hard.

Approximation guarantees

How to *approximately and efficiently* maximize $L + F$?

Remark: Standard $1 - 1/e$-approximition [6] is not satisfying (gives linear regret), since for a very tight confidence region $\mathcal{C}_t$, one expects an approximation factor close to 1.

$\mathcal{A} = \mathcal{I}$ is the family of *independent sets*

Algorithm LocalSearch for maximizing $L + F$ on $\mathcal{I}$.

Input: L, F, $\mathcal{I}$, m, $\varepsilon > 0$.
Initialization: $S_{\text{init}} \in \arg\max_{A\in\mathcal{I}} L(A)$.
if $S_{\text{init}} = \emptyset$ **then**
  if $\exists \{x\} \in \mathcal{I}$ such that $(L+F)(\{x\}) > 0$ **then**
    $S_0 \in \arg\max_{\{x\}\in\mathcal{I},\ (L+F)(\{x\})>0} L(\{x\})$.
  else
    Output $\emptyset$
  end if
else
  $S_0 \leftarrow S_{\text{init}}$
end if
$S \leftarrow S_0$.
Repeatedly perform one of the following local improvements **while** possible:

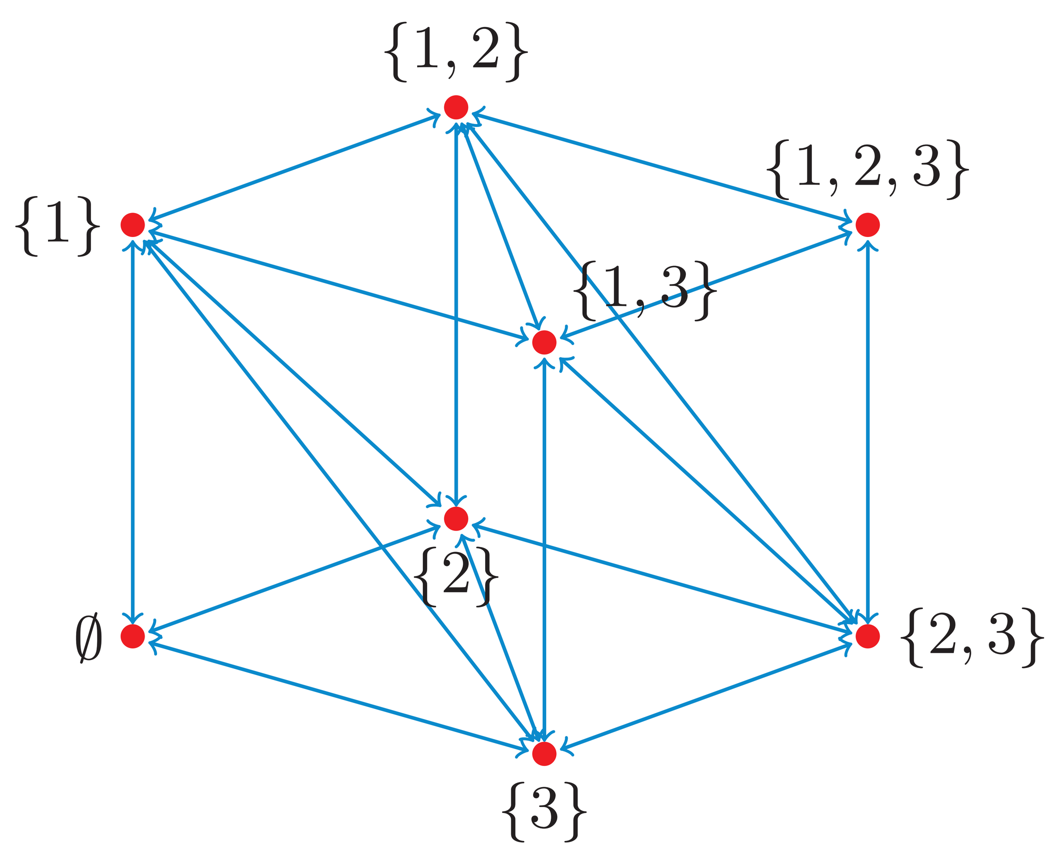
- **Delete an element:**
  if $\exists x \in S$ such that $(L+F)(S\backslash\{x\}) > (L+F)(S) + \frac{\varepsilon}{m} F(S)$,
  then $S \leftarrow S\backslash\{x\}$.
  end if
- **Add an element:**
  if $\exists y \in [n]\backslash S$, $S \cup \{y\} \in \mathcal{I}$, such that $(L+F)(S\cup\{y\}) > (L+F)(S) + \frac{\varepsilon}{m} F(S)$,
  then $S \leftarrow S \cup \{y\}$.
  end if
- **Swap a pair of elements:**
  if $\exists (x, y) \in S \times [n]\backslash S$, $S\backslash\{x\}\cup\{y\} \in \mathcal{I}$, such that $(L+F)(S\backslash\{x\}\cup\{y\}) > (L+F)(S) + \frac{\varepsilon}{m} F(S)$
  then $S \leftarrow S\backslash\{x\}\cup\{y\}$ **end if**

end while
Output: S.

Theorem. *This algorithm outputs $S \in \mathcal{I}$ such that*

$$L(S) + 2(1+\varepsilon) F(S) \ge L(O) + F(O), \quad \forall O \in \mathcal{I}.$$

Its complexity is $\mathcal{O}\left(m^2 n \log(mt)\right)$ (for $\varepsilon = 0.1$ fixed).



$\mathcal{A} = \mathcal{B}$ is the family of *bases*

Algorithm Greedy for maximizing $L + F$ on $\mathcal{B}$.

Input: L, F, $\mathcal{I}$, m.
Initialization: $S \leftarrow \emptyset$.
for $i \in [k]$ **do**
  $x \in \arg\max_{x\notin S,\ S\cup\{x\}\in\mathcal{I}} (L+F)(S\cup\{x\})$.
  $S \leftarrow S \cup \{x\}$.
end for
Output: S.

Theorem. *Algorithm outputs $S \in \mathcal{B}$ such that*

$$L(S) + 2F(S) \ge L(O) + F(O), \quad \forall O \in \mathcal{B}.$$

Its complexity is $\mathcal{O}(mn)$.

Budgeted Matroid Semi-Bandits

Goal: Minimize

$$\left(\frac{L_1 - F_1}{L_2 + F_2}\right)^+.$$

Approximation Lagrangian:

$$\mathcal{L}_\kappa(\lambda, S) \triangleq L_1(S) - \kappa F_1(S) - \lambda\left(L_2(S) + \kappa F_2(S)\right),$$

Remark .

- For $\lambda \ge 0$,

$$\min_{A\in\mathcal{A}} \mathcal{L}(\lambda, A) \text{ and } \lambda^\star - \lambda \text{ have the same sign.}$$

- For a κ−approximation algorithms outputing S (with objective function $-\mathcal{L}$),

$$\min_{A\in\mathcal{A}} \mathcal{L}_\kappa(\lambda, A) \le \mathcal{L}_\kappa(\lambda, S) \le \min_{A\in\mathcal{A}} \mathcal{L}(\lambda, A).$$

Thus,a lower bound λ_1 on $\lambda^\star \triangleq \min\left(\frac{L_1 - F_1}{L_2 + F_2}\right)^+$, and an upper bound λ_2 on $\min_{A\in\mathcal{A}}\left(\frac{L_1(A) - \kappa F_1(A)}{L_2(A) + \kappa F_2(A)}\right)^+$ can be computed.

Algorithm Binary search for minimizing the ratio $(L_1 - F_1)^+ / (L_2 + F_2)$.

Input: L_1, L_2, F_1, F_2, Algo$_\kappa$, $\eta > 0$.
$\delta \leftarrow \frac{\eta \min_{\{s\}\in\mathcal{A}} F_1(\{s\})}{L_2(B) + \kappa^2 F_2(B)}$ with $B = \text{Algo}_\kappa(L_2 + \kappa F_2)$.
$A \leftarrow A_0 \in \mathcal{A}\backslash\{\emptyset\}$ arbitrary.
if $\mathcal{L}_\kappa(0, A) > 0$ **then**
  $\lambda_1 \leftarrow 0$, $\lambda_2 \leftarrow \frac{L_1(A) - F_1(A)}{L_2(A) + F_2(A)}$.
  while $\lambda_2 - \lambda_1 \ge \delta$ **do**
    $\lambda \leftarrow \frac{\lambda_1 + \lambda_2}{2}$.
    $S \leftarrow \text{Algo}_\kappa(-\mathcal{L}(\lambda, \cdot))$.
    if $\mathcal{L}_\kappa(\lambda, S) \ge 0$ **then**
      $\lambda_1 \leftarrow \lambda$.
    else
      $\lambda_2 \leftarrow \lambda$.
      $A \leftarrow S$.
    end if
  end while
end if
Output: A.

Theorem. *Algorithm outputs A such that*

$$\left(\frac{L_1(A) - (\kappa + \eta) F_1(A)}{L_2(A) + \kappa F_2(A)}\right)^+ \le \lambda^\star,$$

the complexity is of order $\log(mt/\eta)$ times the complexity of Algo$_\kappa$.

Experiments

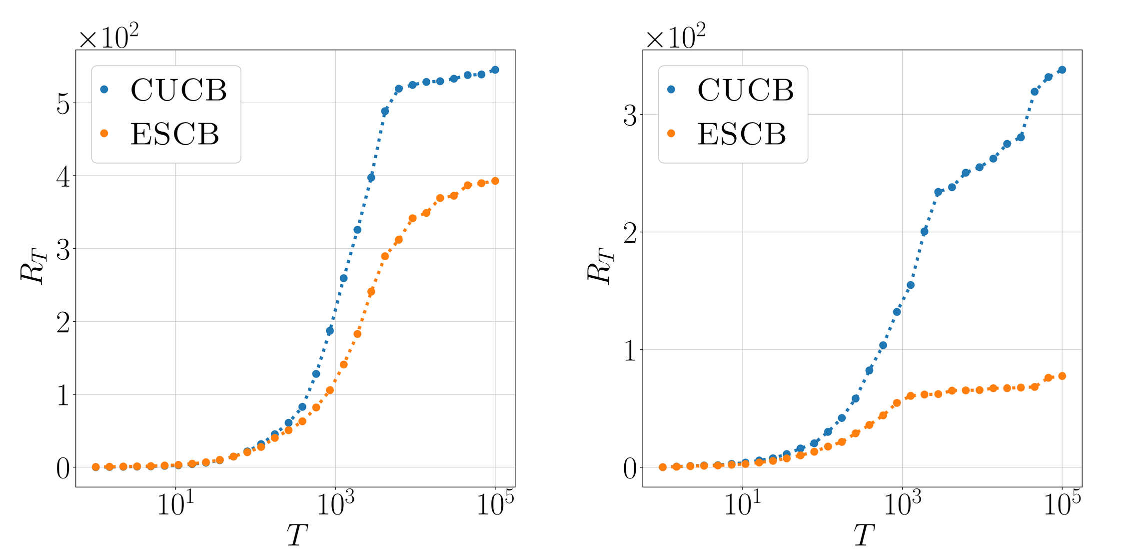


Figure 1: Cumulative regret for the minimum spanning tree setting in up to 10^5 rounds, averaged over 100 independent simulations. **Left**: for $\mathcal{A} = \mathcal{B}$. **Right**: for $\mathcal{A} = \mathcal{I}$.

References

[1] Oliveira, C. A. and Pardalos, P. M. A survey of combinatorial optimization problems in multicast routing, 2005.

[2] Chen, W., Wang, Y., Yuan, Y., and Wang, Q. Combinatorial Multi-Armed Bandit and Its Extension to Probabilistically Triggered Arms. jul 2014.

[3] Kveton, B., Wen, Z., Ashkan, A., Eydgahi, H., and Eriksson, B. Matroid bandits: Fast combinatorial optimization with learning. In *Uncertainty in Artificial Intelligence*, 2014.

[4] Combes, R., Shahi, M. S. T. M., Proutiere, A., and Others. Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems*, pp. 2116–2124, 2015.

[5] Degenne, R. and Perchet, V. Combinatorial semi-bandit with known covariance. dec 2016.

[6] Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions-I. *Mathematical Programming*, 14(1):265–294, 1978.