Matroid Semi-bandits in Sublinear Time

Ruo-Chun Tzeng[1], Naoto Ohsaka[2], and Kaito Ariu[2]

ICML 2024

[1]EECS, KTH Royal Institute of Technology, Sweden
[2]AI Lab, CyberAgent, Japan

Matroid Semi-Bandits in Sublinear Time (ICML'24)

Outline

Matroid Semi-Bandits in Sublinear Time (ICML'24)

- Matroids
- Matroid semi-bandit problem
- CUCB algorithm
- FasterCUCB algorithm

Conclusion and future works

Matroids

A matroid is a pair $\mathcal{M} = ([K], \mathcal{I})$, where

- $[K] = \{1, \cdots, K\}$ is the ground set
- $\mathcal{I}$ is the set of independent sets satisfying
  - (downward closure) For any $Y \in \mathcal{I}$, if $X \subset Y$, then $X \in \mathcal{I}$
  - (augmentation property) For $X, Y \in \mathcal{I}$, if $|X| < |Y|$, then there exists $e \in Y$ such that $X \cup \{e\} \in \mathcal{I}$
- A set X is called a basis of $\mathcal{M}$ if $|X|$ is maximum
- All bases have the same cardinality which is known as the rank of the matroid.

Examples of matroids

- Uniform matroid: $\mathcal{I} = \{\boldsymbol{x} \in \{0,1\}^K : |\mathrm{supp}\,(\boldsymbol{x})| \leq m\}$
- Partition matroid: Given a partition $S_1, \cdots, S_D$ of $[K]$, $\mathcal{I} = \{\boldsymbol{x} \in \{0,1\}^K : |\mathrm{supp}\,(\boldsymbol{x}) \cap S_i| \leq 1, \forall i \in [D]\}$
- Graphical matroid: $\mathcal{I} = \{\boldsymbol{x} \in \{0,1\}^K : \mathrm{supp}\,(\boldsymbol{x}) \text{ is a forest}\}$
- Transversal matroid: $\mathcal{I} = \{\boldsymbol{x} \in \{0,1\}^K : \mathrm{supp}\,(\boldsymbol{x}) \text{ is the left vertices in a matching}\}$

Outline

Problem Setting of Matroid Semi-bandits

An instance of matroid semi-bandit is described by $([K], \mathcal{X}, \boldsymbol{\mu})$,

- $[K] = \{1, \cdots, K\}$ is the ground set;
- $\mathcal{X} \subseteq \{0,1\}^K$ is the set of bases of $\mathcal{M} = ([K], \mathcal{I})$[1];
- $\boldsymbol{\mu} \in (0,1)^K$ is the mean of the arms' rewards $(\nu_1, \cdots, \nu_K)$;[2]

(Rule) At each round t, the learner pulls an action $\boldsymbol{x}(t) \in \mathcal{X}$ and observes a noisy reward $y_k(t) \sim \nu_k$ for each arm $k \in \operatorname{supp}(\boldsymbol{x}(t))$.

(Goal) Minimize the expected cumulative regret

$$R(T) = T \langle \boldsymbol{i}^\star(\boldsymbol{\mu}), \boldsymbol{\mu} \rangle - \sum_{t=1}^{T} \mathbb{E}[\langle \boldsymbol{x}(t), \boldsymbol{\mu} \rangle],$$

where $\boldsymbol{i}^\star(\boldsymbol{\mu}) \in \operatorname{argmax}_{\boldsymbol{x} \in \mathcal{X}} \langle \boldsymbol{x}, \boldsymbol{\mu} \rangle$.

[1] $\mathcal{I}$ satisfy downward closure and augmentation property.

[2] We assume for each $k \in [K]$, the support of ν_k is $[a, b] \subseteq (0,1)$.

Existing Matroid Semi-bandit Algorithms

Fix a best action $\boldsymbol{i}^\star \in \operatorname{argmax}_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{x}, \boldsymbol{\mu}\rangle$.
Define $\triangle_{\min} = \min_{i\in\mathrm{supp}(\boldsymbol{i}^\star), j\notin\mathrm{supp}(\boldsymbol{i}^\star): \mu_i-\mu_j>0}(\mu_i - \mu_j)$.

- `CUCB` [KWA+14] achieves $R(T) = \mathcal{O}(\frac{(K-D)\log T}{\triangle_{\min}})$, which matches the gap-dependent LB $R(T) = \Omega(\frac{(K-D)\log T}{\triangle_{\min}})$
- `KL-OSM` [TP16] achieves $\limsup_{T\to\infty} \frac{R(T)}{\log T} \leq c_{\mathcal{M}}(\boldsymbol{\mu})$, matching instance-specific LB $\liminf_{T\to\infty} \frac{R(T)}{\log T} \geq c_{\mathcal{M}}(\boldsymbol{\mu})$

Per-round time complexity:

- `CUCB` and `KL-OSM` require at least $\Omega(K)$
- `CUCB` takes $\mathcal{O}(K(\log K + \mathcal{T}_{\mathrm{member}}))$ and `KL-OSM` takes $\mathcal{O}(K(\log K + \mathcal{T}_{\mathrm{member}} + \text{line search}))$.

Existing Matroid Semi-bandit Algorithms

Fix a best action $\boldsymbol{i}^\star \in \mathrm{argmax}_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{x}, \boldsymbol{\mu} \rangle$.
Define $\triangle_{\min} = \min_{i\in\mathrm{supp}(\boldsymbol{i}^\star), j\notin\mathrm{supp}(\boldsymbol{i}^\star):\mu_i-\mu_j>0} (\mu_i - \mu_j)$.

- `CUCB` [KWA+14] achieves $R(T) = \mathcal{O}(\frac{(K-D)\log T}{\triangle_{\min}})$, which matches the gap-dependent LB $R(T) = \Omega(\frac{(K-D)\log T}{\triangle_{\min}})$
- `KL-OSM` [TP16] achieves $\limsup_{T\to\infty} \frac{R(T)}{\log T} \le c_{\mathcal{M}}(\boldsymbol{\mu})$, matching instance-specific LB $\liminf_{T\to\infty} \frac{R(T)}{\log T} \ge c_{\mathcal{M}}(\boldsymbol{\mu})$

(This work) `FasterCUCB` achieves $\lim_{T\to\infty} \frac{R(T)}{\log T} = \mathcal{O}(\frac{(K-D)}{\triangle_{\min}})$ and has per-round time complexity sublinear in K.

Summary of Our Results

- Nearly optimal per-round time complexity of `FasterCUCB` on uniform matroid, partition matroid, and graphical matroid
- `FasterCUCB` has the same regret guarantee as `CUCB`

	`CUCB`	`FasterCUCB`
Per-round Time Complexity	$\mathcal{O}(K(\log K + \mathcal{T}_{\mathrm{member}}))$	$\mathcal{O}(D \operatorname{polylog}(T)\, \mathcal{T}_{\mathrm{update}}(\mathcal{A}))$
Uniform Matroid	$\mathcal{O}(K \log K)$	$\mathcal{O}(D \log K \operatorname{polylog}(T))$
Partition Matroid	$\mathcal{O}(K \log K)$	$\mathcal{O}(D \log K \operatorname{polylog}(T))$
Graphical Matroid	$\mathcal{O}(K \log K)$	$\mathcal{O}(D \operatorname{polylog}(K) \operatorname{polylog}(T))$
Transversal Matroid	$\mathcal{O}(K(\log K + DK))$	$\mathcal{O}(D\sqrt{K} \operatorname{polylog}(T))$

Table 1: Per-round time complexity on different matroids. K is the number of arms and $D = \max_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x}\|_0$.

Outline

Solve CUCB's Sampling Rule by a Greedy Algorithm

Let $\boldsymbol{f}_k = (\hat{\mu}_k(t-1), \frac{1}{\sqrt{N_k(t-1)}})$ and $\boldsymbol{q} = (1, \sqrt{\log t})$.

$$\boldsymbol{x}(t) \in \operatorname*{argmax}_{\boldsymbol{x} \in \mathcal{X}} \sum_{k=1}^{K} \langle \boldsymbol{f}_k, \boldsymbol{q} \rangle x_k \tag{CUCB}$$

A greedy algorithm works as follows:

(1) Find permutation $\pi : [K] \to [K]$ s.t. $\left\langle \boldsymbol{f}_{\pi(k)}, \boldsymbol{q} \right\rangle \geq \left\langle \boldsymbol{f}_{\pi(k+1)}, \boldsymbol{q} \right\rangle$, $\forall k$

(2) $\boldsymbol{x} = \boldsymbol{0}_K$; While($|\mathrm{supp}\,(\boldsymbol{x})| < D$) If($\boldsymbol{x} + \boldsymbol{e}_{\pi(k)} \in \mathcal{I}$) $\boldsymbol{x} = \boldsymbol{x} + \boldsymbol{e}_{\pi(k)}$;

Time complexity: $\mathcal{O}(K(\log K + \mathcal{T}_{\mathrm{member}}))$

Solve CUCB's Sampling Rule by a Greedy Algorithm

Let $\boldsymbol{f}_k = (\hat{\mu}_k(t-1), \frac{1}{\sqrt{N_k(t-1)}})$ and $\boldsymbol{q} = (1, \sqrt{\log t})$.

$$\boldsymbol{x}(t) \in \operatorname*{argmax}_{\boldsymbol{x} \in \mathcal{X}} \sum_{k=1}^{K} \langle \boldsymbol{f}_k, \boldsymbol{q} \rangle x_k \tag{CUCB}$$

A greedy algorithm works as follows:

(1) Find permutation $\pi : [K] \to [K]$ s.t. $\langle \boldsymbol{f}_{\pi(k)}, \boldsymbol{q} \rangle \geq \langle \boldsymbol{f}_{\pi(k+1)}, \boldsymbol{q} \rangle$, $\forall k$

(2) $\boldsymbol{x} = \boldsymbol{0}_K$; While($|\mathrm{supp}(\boldsymbol{x})| < D$) If($\boldsymbol{x} + \boldsymbol{e}_{\pi(k)} \in \mathcal{I}$) $\boldsymbol{x} = \boldsymbol{x} + \boldsymbol{e}_{\pi(k)}$;

Time complexity: $\mathcal{O}(K(\log K + \mathcal{T}_{\mathrm{member}}))$

(Observation) For the greedy solution, the ordering of the arm weights matters, rather than their absolute values!

gent
ab

Outline

Matroid Semi-Bandits in Sublinear Time (ICML'24)

- Matroids
- Matroid semi-bandit problem
- CUCB algorithm
- FasterCUCB algorithm

Conclusion and future works

Our Approach: Rounding + Minimum Hitting Set (1/4)

Let $\boldsymbol{f}_k = (\hat{\mu}_k(t-1), \frac{1}{\sqrt{N_k(t-1)}})$ and $\boldsymbol{q} = (1, \sqrt{\log t})$.

$$\boldsymbol{x}(t) \in \operatorname*{argmax}_{\boldsymbol{x} \in \mathcal{X}} \sum_{k=1}^{K} \langle \boldsymbol{f}_k, \boldsymbol{q} \rangle x_k \tag{CUCB}$$

(Challenge) Every arm k's weight $\langle \boldsymbol{f}_k, \boldsymbol{q} \rangle$ changes at each round.

CyberAgent AI Lab

KTH VETENSKAP OCH KONST

Our Approach: Rounding + Minimum Hitting Set (1/4)

Let $\boldsymbol{f}_k = (\hat{\mu}_k(t-1), \frac{1}{\sqrt{N_k(t-1)}})$ and $\boldsymbol{q} = (1, \sqrt{\log t})$.

$$\boldsymbol{x}(t) \in \operatorname*{argmax}_{\boldsymbol{x} \in \mathcal{X}} \sum_{k=1}^{K} \langle \boldsymbol{f}_k, \boldsymbol{q} \rangle x_k \tag{CUCB}$$

(Idea) $\boldsymbol{q}$ is "fixed" $\Rightarrow$ only $D = \max_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x}\|_0$ arms will change

Our Approach: Rounding + Minimum Hitting Set (1/4)

Let $\boldsymbol{f}_k = (\hat{\mu}_k(t-1), \frac{1}{\sqrt{N_k(t-1)}})$ and $\boldsymbol{q} = (1, \sqrt{\log t})$.

$$\boldsymbol{x}(t) \in \operatorname*{argmax}_{\boldsymbol{x} \in \mathcal{X}} \sum_{k=1}^{K} \langle \boldsymbol{f}_k, \boldsymbol{q} \rangle x_k \tag{CUCB}$$

(Idea) $\boldsymbol{q}$ is "fixed" $\Rightarrow$ only $D = \max_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x}\|_0$ arms will change

- [OT23]: Construct a set $\mathcal{H}$ such that for any $\boldsymbol{q} \in \mathbb{R}_+^2$, $\exists \boldsymbol{h} \in \mathcal{H}$ s.t.

  $$\langle \boldsymbol{f}_k, \boldsymbol{h} \rangle > \langle \boldsymbol{f}_\ell, \boldsymbol{h} \rangle \implies \langle \boldsymbol{f}_k, \boldsymbol{q} \rangle \geq \langle \boldsymbol{f}_\ell, \boldsymbol{q} \rangle$$

  for all possible $(k, \ell) \in [K] \times [K]$ and $k \neq \ell$
- $\forall \boldsymbol{h} \in \mathcal{H}$, create a dynamic alg. $\mathcal{A}_{\boldsymbol{h}}$ with arm k's weight $\langle \boldsymbol{f}_k, \boldsymbol{h} \rangle$:
  - fast computation of a maximum-weight base (in $\mathcal{O}(D)$-time)
  - fast update of an arm's weight (in $\mathcal{T}_{\mathrm{update}}$-time)
- Time complexity: $\mathcal{O}(D \cdot |\mathcal{H}| \cdot \mathcal{T}_{\mathrm{update}})$

Our Approach: Rounding + Minimum Hitting Set (1/4)

Let $\boldsymbol{f}_k = (\hat{\mu}_k(t-1), \frac{1}{\sqrt{N_k(t-1)}})$ and $\boldsymbol{q} = (1, \sqrt{\log t})$.

$$\boldsymbol{x}(t) \in \operatorname*{argmax}_{\boldsymbol{x} \in \mathcal{X}} \sum_{k=1}^{K} \langle \boldsymbol{f}_k, \boldsymbol{q} \rangle x_k \qquad \text{(CUCB)}$$

(Idea) $\boldsymbol{q}$ is "fixed" $\Rightarrow$ only $D = \max_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x}\|_0$ arms will change

- [OT23]: Construct a set $\mathcal{H}$ such that for any $\boldsymbol{q} \in \mathbb{R}^2_+$, $\exists \boldsymbol{h} \in \mathcal{H}$ s.t.

  $$\langle \boldsymbol{f}_k, \boldsymbol{h} \rangle > \langle \boldsymbol{f}_\ell, \boldsymbol{h} \rangle \implies \langle \boldsymbol{f}_k, \boldsymbol{q} \rangle \geq \langle \boldsymbol{f}_\ell, \boldsymbol{q} \rangle$$

  for all possible $(k, \ell) \in [K] \times [K]$ and $k \neq \ell$
- $\forall \boldsymbol{h} \in \mathcal{H}$, create a dynamic alg. $\mathcal{A}_{\boldsymbol{h}}$ with arm k's weight $\langle \boldsymbol{f}_k, \boldsymbol{h} \rangle$:
  - fast computation of a maximum-weight base (in $\mathcal{O}(D)$-time)
  - fast update of an arm's weight (in $\mathcal{T}_{\mathrm{update}}$-time)
- Time complexity: $\mathcal{O}(D \cdot |\mathcal{H}| \cdot \mathcal{T}_{\mathrm{update}})$ Naive: $|\mathcal{H}| = \mathcal{O}(K^2)$

Our Approach: Rounding + Minimum Hitting Set (2/4)

Let $\boldsymbol{f}_k = (\hat{\mu}_k(t-1), \frac{1}{\sqrt{N_k(t-1)}})$ and $\boldsymbol{q} = (1, \sqrt{\log t})$.

$$\boldsymbol{x}(t) \in \operatorname*{argmax}_{\boldsymbol{x}\in\mathcal{X}} \sum_{k=1}^{K} \langle \boldsymbol{f}_k, \boldsymbol{q} \rangle x_k \tag{CUCB}$$

(Rounding) Generate polylog (T) bins with $\epsilon = \frac{1}{\log^m T}$.

Represent each $\mathrm{BIN}(q,r) = (a(1+\epsilon)^{q-1}, a(1+\epsilon)^q] \times (\frac{(1+\epsilon)^{r-1}}{\sqrt{T}}, \frac{(1+\epsilon)^r}{\sqrt{T}}]$ by $\mathrm{dom}_{q,r} = (a(1+\epsilon)^q, \frac{1}{\sqrt{T}}(1+\epsilon)^r)$.

$$\forall \boldsymbol{f} \in \mathrm{BIN}(q,r), \quad \frac{\langle \mathrm{dom}_{q,r}, \boldsymbol{q} \rangle}{1+\epsilon} < \langle \boldsymbol{f}, \boldsymbol{q} \rangle \le \langle \mathrm{dom}_{q,r}, \boldsymbol{q} \rangle , \tag{1}$$

(Hitting set) Generate $\mathcal{H}$ of size polylog (T) s.t. $\forall \boldsymbol{q} \in \mathbb{R}_+^2$, $\exists \boldsymbol{h} \in \mathcal{H}$ s.t.

$$\langle \mathrm{dom}_{q,r}, \boldsymbol{h} \rangle > \langle \mathrm{dom}_{q',r'}, \boldsymbol{h} \rangle \implies \langle \mathrm{dom}_{q,r}, \boldsymbol{q} \rangle \ge \langle \mathrm{dom}_{q',r'}, \boldsymbol{q} \rangle \tag{2}$$

for any $(q,r) \neq (q',r')$, and $\boldsymbol{h}$ can be found in polylog (T) time.

Our Approach: Rounding + Minimum Hitting Set (3/4)

Let $\mathrm{dom}(\boldsymbol{f}) = \mathrm{dom}_{q,r}$ with the unique (q, r) s.t. $\boldsymbol{f} \in \mathrm{BIN}(q, r)$.

Initialization:

- Generate polylog (T) bins
- Generate the minimum hitting set $\mathcal{H}$ of size polylog (T)
- For each $\boldsymbol{h} \in \mathcal{H}$, instantiate a dynamic algorithm $\mathcal{A}_{\boldsymbol{h}}$ with arm k's weight $\langle \mathrm{dom}(\boldsymbol{f}_k), \boldsymbol{h} \rangle$

FindBase($\boldsymbol{q}$): Find $\boldsymbol{h} \in \mathcal{H}$ such that

$$\langle \mathrm{dom}_{q,r}, \boldsymbol{h} \rangle > \langle \mathrm{dom}_{q',r'}, \boldsymbol{h} \rangle \implies \langle \mathrm{dom}_{q,r}, \boldsymbol{q} \rangle \geq \langle \mathrm{dom}_{q',r'}, \boldsymbol{q} \rangle$$

Call $\mathcal{A}_{\boldsymbol{h}}$ to output a $(1 + \epsilon)$-approx maximum-weight base

UpdateFeature($\boldsymbol{f}$,k): $\forall \boldsymbol{h} \in \mathcal{H}$, update arm k's weight to $\langle \mathrm{dom}(\boldsymbol{f}), \boldsymbol{h} \rangle$

Our Approach: Rounding + Minimum Hitting Set (4/4)

Let $\boldsymbol{f}_k = (\hat{\mu}_k(t-1), \frac{1}{\sqrt{N_k(t-1)}})$ and $\boldsymbol{q} = (1, \sqrt{\log t})$.

$$\boldsymbol{x}(t) \in \operatorname*{argmax}_{\boldsymbol{x}\in\mathcal{X}} \sum_{k=1}^{K} \langle \boldsymbol{f}_k, \boldsymbol{q} \rangle x_k \qquad \text{(CUCB)}$$

`FasterCUCB`: At round t,

- Compute $\boldsymbol{x}(t)$ by **FindBase**($\boldsymbol{q}$)
- For each $k \in \operatorname{supp}(\boldsymbol{x}(t))$, observe $y_k(t) \sim \nu_k$ and **UpdateFeature**$\left(\left(\frac{(t-1)\mu_k(t-1)+y_k(t)}{t}, \frac{1}{\sqrt{N_k(t-1)+1}}\right), k\right)$

Time complexity: $\mathcal{O}(D \operatorname{polylog}(T)\, \mathcal{T}_{\mathrm{update}}(\mathcal{A}))$

Regret bound: $\lim_{T\to\infty} \frac{R(T)}{\log T} = \mathcal{O}\left(\frac{K-D}{\triangle_{\min}}\right)$.

Conclusion and future works

Conclusion and Future Works

We have developed the first sublinear-time algorithm for matroid semi-bandits. There are several directions:

- Our developed techniques might be use to speedup UCB-style algorithms for other problems, e.g., combinatorial best-arm identification [CLK$^+$14, DKC21] and nonstationary semi-bandits [ZWVL20, CWZZ21]
- Another direction is to study the possibility of speeding up other forms of weights, such as those derived from gradients [TWPL23] and those in the follow-the-perturbed-leader algorithm [NB16]

Reference i

- Shouyuan Chen, Tian Lin, Irwin King, Michael R Lyu, and Wei Chen, *Combinatorial pure exploration of multi-armed bandits*, Proc. of NeurIPS, 2014.
- Wei Chen, Liwei Wang, Haoyu Zhao, and Kai Zheng, *Combinatorial semi-bandit in the non-stationary environment*, Proc. of UAI, 2021.
- Yihan Du, Yuko Kuroki, and Wei Chen, *Combinatorial pure exploration with full-bandit or partial linear feedback*, Proc. of AAAI, 2021.
- Branislav Kveton, Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Brian Eriksson, *Matroid bandits: Fast combinatorial optimization with learning*, Proc. of UAI, 2014.

Reference ii

- Gergely Neu and Gábor Bartók, *Importance weighting without importance weights: An efficient algorithm for combinatorial semi-bandits*, JMLR (2016).
- Naoto Ohsaka and Riku Togashi, *Curse of" low" dimensionality in recommender systems*, Proc. of SIGIR, 2023.
- Mohammad Sadegh Talebi and Alexandre Proutiere, *An optimal algorithm for stochastic matroid bandit optimization*, Proc. of AAMAS, 2016.
- Ruo-Chun Tzeng, Po-An Wang, Alexandre Proutiere, and Chi-Jen Lu, *Closing the computational-statistical gap in best arm identification for combinatorial semi-bandits*, Proc. of NeurIPS, 2023.

Reference iii

Huozhi Zhou, Lingda Wang, Lav Varshney, and Ee-Peng Lim, *A near-optimal change-detection based algorithm for piecewise-stationary combinatorial semi-bandits*, Proc. of AAAI, 2020.

Generating the Minimum Hitting Set

- Generate a line orthogonal to $\overleftrightarrow{\boldsymbol{f}_i\boldsymbol{f}_j}$ for each $i \neq j$
- Each region corresponds a distinct ordering of $\{\boldsymbol{f}_i\}_{i\in[K]}$