Matroid Semi-bandits in Sublinear Time

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

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

Problem Setting: Matroid Semi-bandits

Setting : A matroid semi-bandit instance 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 $\mathcal{M} = ([K], \mathcal{I})^a$;
- $\boldsymbol{\mu} \in (0,1)^K$ is the mean of the arms' rewards $(\nu_1, \cdots, \nu_K)$;^{b}

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

Prior Works

Fix a best action $\boldsymbol{i}^\star \in \operatorname{argmax}_{\boldsymbol{x}\in\mathcal{X}} \langle \boldsymbol{x}, \boldsymbol{\mu}\rangle$, and define $\triangle_{\min} = \min_{i\in\operatorname{supp}(\boldsymbol{i}^\star), j\notin\operatorname{supp}(\boldsymbol{i}^\star): \mu_i-\mu_j>0}(\mu_i - \mu_j)$. Denote by $\mathcal{T}_{\text{member}}$ the time to query whether a set is in $\mathcal{I}$.

CUCB [2] :

- achieves $R(T) = \mathcal{O}\left(\frac{(K-D)\log T}{\triangle_{\min}}\right)$
- the per-round time complexity is $\mathcal{O}(K(\log K + \mathcal{T}_{\text{member}}))$

KL-OSM [3] :

- achieves $\limsup_{T\to\infty} \frac{R(T)}{\log T} \le c(\boldsymbol{\mu})$ matching instance-specific lower bound, where $c(\boldsymbol{\mu}) \le \frac{K-D}{\triangle_{\min}}$
- the per-round time complexity is $\mathcal{O}(K(\log K + \mathcal{T}_{\text{member}} + \text{line search}))$

Both CUCB and KL-OSM requires per-round time complexity $\Omega(K)$.

$^a\mathcal{I}$ satisfy hereditary property and augmentation property.
^{b}We assume for each $k \in [K]$, the support of ν_k is $[a, b] \subseteq (0, 1)$.

Summary of Our Results

We propose FasterCUCB, which

- achieves nearly optimal per-round time complexity on uniform matroid, partition matroid, and graphical matroid
- maintains the same regret guarantee as CUCB

	CUCB	FasterCUCB
Per-round Time Complexity	$\mathcal{O}(K(\log K + \mathcal{T}_{\text{member}}))$	$\mathcal{O}(D \operatorname{polylog}(T)\, \mathcal{T}_{\text{update}}(\mathcal{A}))$
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: Per-round time complexity on different matroids. K is the number of arms and $D = \max_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{x}\|_0$.

Main Ideas: Rounding and Minimum Hitting Set

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

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

Rounding: Generate $\operatorname{polylog}(T)$ 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 $\mathrm{dom}_{q,r} = (a(1+\epsilon)^q, \frac{1}{\sqrt{T}}(1+\epsilon)^r)$.

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

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

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

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

Algorithm: FasterCUCB

At round $t \in \mathbb{N}$,

- 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**$((\frac{(t-1)\mu_k(t-1)+y_k(t)}{t}, \frac{1}{\sqrt{N_k(t-1)+1}}), k)$

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 $\operatorname{polylog}(T)$ bins
- Generate the minimum hitting set $\mathcal{H}$ of size $\operatorname{polylog}(T)$
- 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

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

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 $\langle \mathrm{dom}(\boldsymbol{f}), \boldsymbol{h}\rangle$

Future Works

- Our developed techniques might be use to speedup UCB-style algorithms for other problems, e.g., nonstationary semi-bandits [4, 1]
- Another direction is to study the possibility of speeding up other forms of weights

References

[1] W. Chen, L. Wang, H. Zhao, and K. Zheng. Combinatorial semi-bandit in the non-stationary environment. In *Proc. of UAI*, 2021.

[2] W. Chen, Y. Wang, and Y. Yuan. Combinatorial multi-armed bandit: General framework and applications. In *Proc. of ICML*, 2013.

[3] M. S. Talebi and A. Proutiere. An optimal algorithm for stochastic matroid bandit optimization. In *Proc. of AAMAS*, 2016.

[4] H. Zhou, L. Wang, L. Varshney, and E.-P. Lim. A near-optimal change-detection based algorithm for piecewise-stationary combinatorial semi-bandits. In *Proc. of AAAI*, 2020.