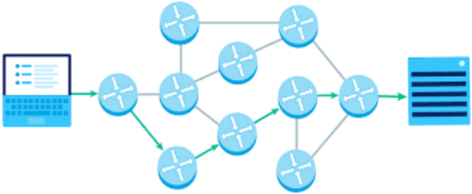
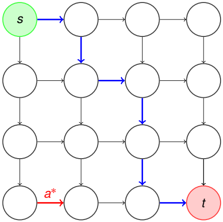
ETH zürich

Efficient Pure Exploration for Combinatorial Bandits with Semi-Bandit Feedback

Marc Jourdan, Mojmír Mutný, Johannes Kirschner, Andreas Krause

Routing





Arm	Action	Feedback	Answer	Oracle
edge	(s, t)-path	latency per edge	worst edge	Dijkstra's algorithm

Combinatorial semi-bandits

- Unstructured d-armed bandit: $\mu \in \mathbb{R}^d$
- Actions: $\mathcal{A} \subset \{0,1\}^d$
- Semi-bandit feedback: pull $A_t \in \mathcal{A}$ and observe

$$Y_{t,A_t} \in \mathbb{R}^{|A_t|} \sim \Pi_{a \in A_t} \nu_a$$

  where ν_a is a one-parameter exponential family with mean μ_a
- Efficient oracle to solve $\text{argmax}_{A \in \mathcal{A}} \langle \mathbf{1}_A, c \rangle$ for $c \in \mathbb{R}^d$

Pure exploration for combinatorial semi-bandits

Goal: Best-answer, $I^*(\mu) := \text{argmax}_{I \in \mathcal{I}} \langle \mathbf{1}_I, \mu \rangle$ where $\mathcal{I} \subset \{0, 1\}^d$

Three rules:

- *sampling* rule, $A_t \in \mathcal{A}$
- *recommendation* rule, $I_t \in \mathcal{I}$
- *stopping* rule, τ_δ

Pure exploration for combinatorial semi-bandits

Goal: Best-answer, $I^*(\mu) := \text{argmax}_{I \in \mathcal{I}} \langle \mathbf{1}_I, \mu \rangle$ where $\mathcal{I} \subset \{0, 1\}^d$

Three rules:

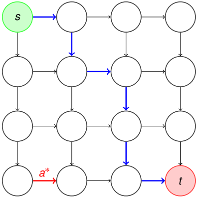
- *sampling* rule, $A_t \in \mathcal{A}$
- *recommendation* rule, $I_t \in \mathcal{I}$
- *stopping* rule, τ_δ

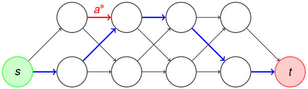
Objectives:

- Minimize $\mathbb{E}_\nu[\tau_\delta]$ among δ-PAC algorithm, $\mathbb{P}_\nu[\tau_\delta = \infty \vee I_{\tau_\delta} \neq I^*] \leqslant \delta$
- Computationally efficient implementation

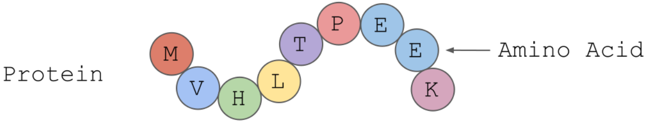
Applications

- Routing





- Batch experiments
- Protein design



Contributions

- Game approach for pure exploration [Degenne et al., 2019] to study combinatorial semi-bandits with **arbitrary** $\mathcal{A}$ and $\mathcal{I}$.
- CombGame meta-algorithm, **asymptotically optimal** algorithms with **finite-time** guarantees.
- **Computationally efficient** algorithm for best-arm identification, being asymptotically optimal and **empirically competitive**.

Sample complexity lower bound

Theorem

Let $\delta \in (0,1)$. For all δ-PAC strategy, for all bandit ν,

$$\frac{\mathbb{E}_\nu[\tau_\delta]}{\ln(1/(2.4\delta))} \geqslant D_\nu^{-1} \quad \textit{and} \quad \limsup_{\delta \to 0} \frac{\mathbb{E}_\nu[\tau_\delta]}{\ln(1/\delta)} \geqslant D_\nu^{-1}$$

Sample complexity lower bound

Theorem

Let $\delta \in (0,1)$. For all δ-PAC strategy, for all bandit ν,

$$\frac{\mathbb{E}_\nu[\tau_\delta]}{\ln(1/(2.4\delta))} \geqslant D_\nu^{-1} \quad \text{and} \quad \limsup_{\delta \to 0} \frac{\mathbb{E}_\nu[\tau_\delta]}{\ln(1/\delta)} \geqslant D_\nu^{-1}$$

$$D_\nu := \max_{\tilde{w} \in \mathcal{S}_\mathcal{A}} \inf_{\lambda \in \Theta^\complement_{I^*(\mu)}} \langle \tilde{w}, d_{\mathrm{KL}}(\mu, \lambda) \rangle$$

CombGame meta-algorithm

μ unknown: μ_t, MLE

- *Recommendation* rule: $I_t = I^*(\mu_{t-1})$
- *Stopping* rule: GLRT

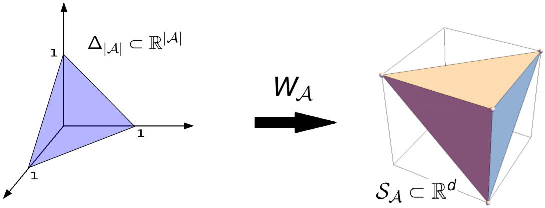
CombGame meta-algorithm

μ unknown: μ_t, MLE

- *Recommendation* rule: $I_t = I^*(\mu_{t-1})$
- *Stopping* rule: GLRT
- *Sampling* rule, **two-player zero sum-game**
  - *A*-player (MAX): w_t based on online learners
  - λ-player (MIN): given w_t, λ_t using a best-response oracle

Sampling rule

- Tracking: deterministic A_t from w_t
- Optimism: optimistic reward $r_t \in \mathbb{R}^d$
- Learner:
  - Minimize $R_t^{\mathcal{L}} := \max_{A \in \mathcal{A}} \sum_{s=1}^{t} \langle \mathbf{1}_A, r_s \rangle - \langle \tilde{w}_s, r_s \rangle$
  - Update $w_t \in \Delta_{|\mathcal{A}|}$ or $\tilde{w}_t \in \mathcal{S}_\mathcal{A}$



Comparison of learners instantiating CombGame

	Update	Sparse	Computational cost	Learner's $R_t^{\mathcal{L}}$
Hedge[1]	$\Delta_{\lvert\mathcal{A}\rvert}$	✗	$O(\lvert\mathcal{A}\rvert)$	$O\left(\ln(t)\sqrt{t}\right)$
AdaHedge[2]	$\Delta_{\lvert\mathcal{A}\rvert}$	✗	$O(\lvert\mathcal{A}\rvert)$	$O\left(\ln(t)\sqrt{t}\right)$
OFW[3]	$\mathcal{S}_{\mathcal{A}}$	✓	$O(\lvert B_t\rvert)$	$O\left(\ln(t)^2 t^{3/4}\right)$
LLOO[4]	$\mathcal{S}_{\mathcal{A}}$	✓	$O(\lvert B_t\rvert(d+\ln(\lvert B_t\rvert)))$	$O\left(\ln(t)\sqrt{t}\right)$

with $B_t := \mathrm{supp}\left(\sum_{s=1}^{t} w_s\right)$

[1] [Cesa-Bianchi et al., 2005]
[2] [Rooij et al., 2014]
[3] [Hazan and Kale, 2012]
[4] [Garber and Hazan, 2013]

Sample complexity upper bound

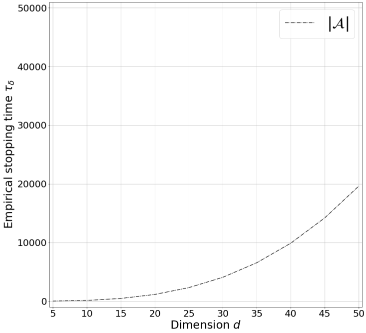
Theorem

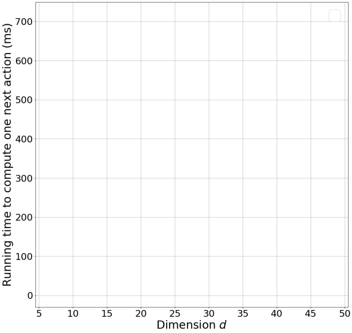
Let $\mu \in \mathcal{M}$ bounded and an online learner with sublinear regret. The instantiated CombGame meta-algorithm is asymptotically optimal:

$$\limsup_{\delta \to 0} \frac{\mathbb{E}_\nu[\tau_\delta]}{\ln(1/\delta)} \leqslant D_\nu^{-1}$$

Experiment

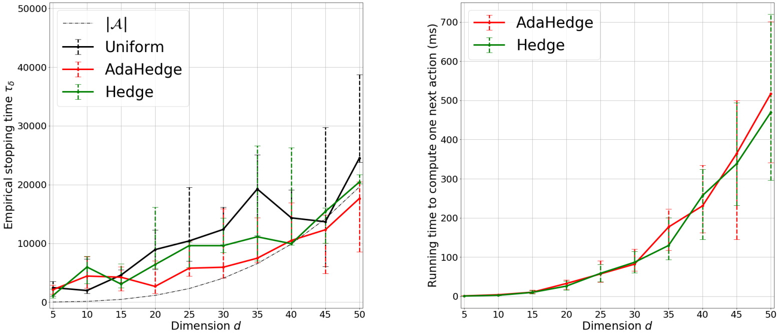
Best-arm identification by playing batches of size $k = 3$ for a Gaussian bandit with $\sigma = 0.1$, $\delta = 0.1$, $N = 750$ runs.





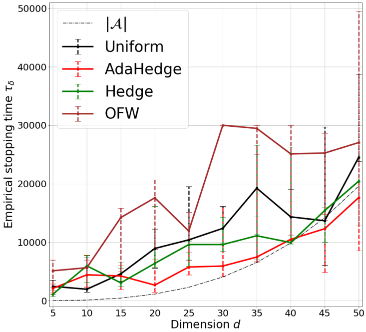
Results

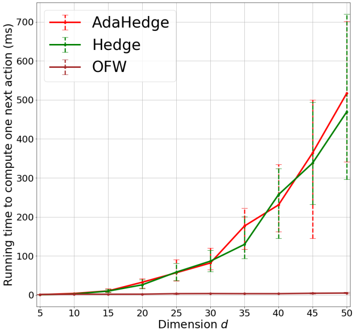
- Uniform has suboptimal sample complexity. AdaHedge outperforms Hedge, **but** both have high computational cost.



Results

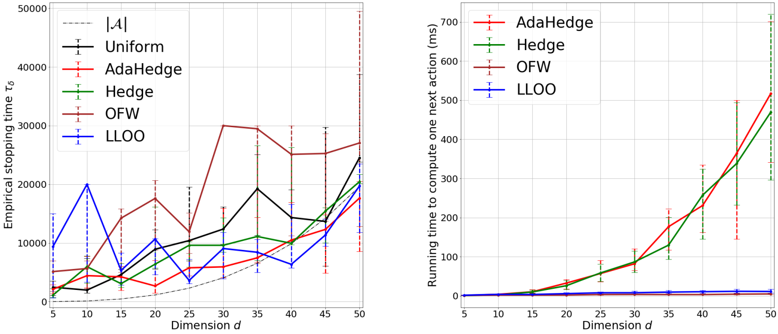
- OFW has low and almost constant computational cost, **but** has suboptimal sample complexity.





Results

- **Main:** LLOO has competitive sample complexity for a low and almost constant computational cost.



Summary

- Asymptotically optimal CombGame meta-algorithm for transductive combinatorial semi-bandits
- Computationally efficient algorithm for best-arm identification by playing actions, being asymptotically optimal and empirically competitive.

Appendix

References

Cesa-Bianchi, N., Mansour, Y., and Stoltz, G. (2005).
Improved Second-Order Bounds for Prediction with Expert Advice.
In *Learning Theory*, pages 217–232, Berlin, Heidelberg.

Degenne, R., Koolen, W. M., and Ménard, P. (2019).
Non-Asymptotic Pure Exploration by Solving Games.
In *Advances in Neural Information Processing Systems 32*, pages 14492–14501.

Garber, D. and Hazan, E. (2013).
A linearly convergent conditional gradient algorithm with applications to online and stochastic optimization.
SIAM Journal on Optimization, 26.

Hazan, E. and Kale, S. (2012).
Projection-free online learning.
In *Proceedings of the 29th International Conference on Machine Learning*, page 1843–1850, Madison, WI, USA.

Rooij, S. d., Erven, T. v., Grünwald, P. D., and Koolen, W. M. (2014).
Follow the Leader If You Can, Hedge If You Must.
Journal of Machine Learning Research, 15(37):1281–1316.

Algorithm: CombGame meta-algorithm

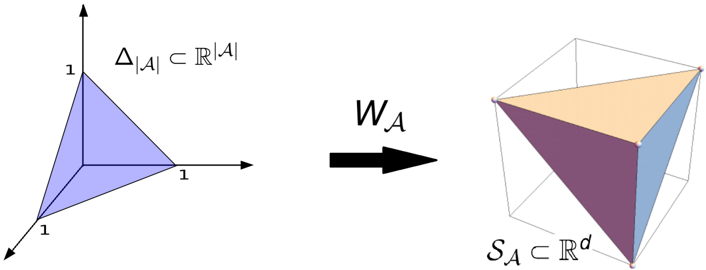
Input: Learner $\mathcal{L}$

Output: Answer I_{τ_δ}

```
Perform initialization;
for t = n_0 + 1, ··· do
    Compute candidate answer I_t ;
    If the stopping criterion met then return I_t ;
    Get w_t from L_{I_t} ;                                   ▷A-player
    Compute λ_t by using Best-Response Oracle ;              ▷λ-player
    Compute optimistic reward r_t ;                          ▷optimism
    Feed L_{I_t} with the reward r_t;
    Compute A_t by using sparse C-Tracking ;
    Observe a sample Y_{t,A_t} and update estimator ;
end
```

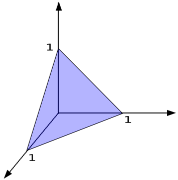
Transformed Simplex

- Linear operator $W_{\mathcal{A}} : w \in \Delta_{|\mathcal{A}|} \mapsto \tilde{w} = W_{\mathcal{A}} w \in \mathcal{S}_{\mathcal{A}}$.



- Transform the high dimensional simplex into a low dimensional transformed simplex, $d \ll |\mathcal{A}|$.

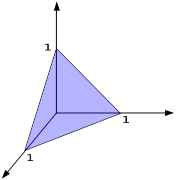
Learners on $\Delta_{|\mathcal{A}|}$



Hedge-type Learners, Hedge [Cesa-Bianchi et al., 2005] and AdaHedge [Rooij et al., 2014]:

$$\forall A \in \mathcal{A}, \quad U_{t,A} = \langle \mathbf{1}_A, r_t \rangle$$

Learners on $\Delta_{|\mathcal{A}|}$



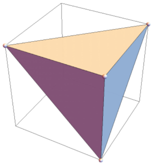
Hedge-type Learners, Hedge [Cesa-Bianchi et al., 2005] and AdaHedge [Rooij et al., 2014]:

$$\forall A \in \mathcal{A}, \quad U_{t,A} = \langle \mathbf{1}_A, r_t \rangle$$

Computationally inefficient due to non sparse:

- Initialization: *full*.
- $\mathcal{L}_{I_t}$ update: $U_t \in \mathbb{R}^{|\mathcal{A}|}$.
- Tracking: dense support.

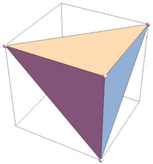
Learners on $\mathcal{S}_{\mathcal{A}}$, requirements



Online Convex Optimization (OCO) on a convex polytope:

- projection-free,
- one call to efficient oracle per round,
- efficient and incrementally sparse representation.

Learners on $\mathcal{S}_{\mathcal{A}}$, requirements



Online Convex Optimization (OCO) on a convex polytope:

- projection-free,
- one call to efficient oracle per round,
- efficient and incrementally sparse representation.

Algorithms:

- Online Frank-Wolfe (OFW) [Hazan and Kale, 2012]
- Local Linear Optimization Oracle-based OCO (LLOO) [Garber and Hazan, 2013].

Learners on $\mathcal{S}_{\mathcal{A}}$

Use the efficient oracle, $\text{argmax}_{A \in \mathcal{A}} \langle \mathbf{1}_A, r_t \rangle$

- OFW: one FW step.
- LLOO: iterative pairwise FW steps.

Learners on $\mathcal{S}_{\mathcal{A}}$

Use the efficient oracle, $\text{argmax}_{A \in \mathcal{A}} \langle \mathbf{1}_A, r_t \rangle$

- OFW: one FW step.
- LLOO: iterative pairwise FW steps.

Computationally efficient due to sparse:

- Initialization: *covering*.
- $\mathcal{L}_{I_t}$ update: efficient oracle.
- Tracking: incrementally sparse.

Finite-time Upper Bound

Theorem

Let $\mathcal{M}$ bounded, $\mu \in \mathcal{M}$. The instantiated CombGame meta-algorithm satisfies:

$$\mathbb{E}_\mu[\tau_\delta] \leqslant T_0(\delta) + \frac{2ed}{c^2}$$

$$\text{with } T_0(\delta) = \max\left\{ t \in \mathbb{N} : t \leqslant \frac{\beta(t,\delta)}{D_\mu} + C_\mu \left(R_t^{\mathcal{L}} + h(t) \right) \right\}$$

where $h(t) = O\left(\sqrt{t \ln(t)}\right)$. $R_t^{\mathcal{L}}$ *is the online Learner's cumulative regret.*

Algorithm: Sparse Tracking

Input: Weights, $w_t \in \Delta_{|\mathcal{A}|}$, the associated support, $B_t \subset \mathcal{A}$, and the tracking mode

Output: Action to sample, A_t

```
if tracking = "D" then
    A_t = argmin_{A∈B_t} N_{t−1,A} / w_{t,A} ;
else if tracking = "C" then
    A_t = argmin_{A∈B_t} N_{t−1,A} / (∑_{s=1}^{t} w_{s,A}) ;
Return A_t;
```

if *tracking = "D"* **then**

$$A_t = \text{argmin}_{A \in B_t} \frac{N_{t-1,A}}{w_{t,A}} ;$$

else if *tracking = "C"* **then**

$$A_t = \text{argmin}_{A \in B_t} \frac{N_{t-1,A}}{\sum_{s=1}^{t} w_{s,A}} ;$$

Return A_t;

Algorithm: Stopping rule

Input: Candidate answer I_t

Output: True if the stopping condition is met

for $J \in N(I_t)$ **do**

$$Z_{t,I_t,J} = \begin{cases} \inf_{\lambda \in \bar{\Theta}_J^{I_t}} \sum_{a \in [d]} N_{t-1,a} d_{\mathrm{KL}}(\mu_{t-1,a}, \lambda_a) & \text{(a)} \\ \frac{((\mathbf{1}_J - \mathbf{1}_{I_t})^\mathsf{T} \mu_{t-1})^2}{2 \sum_{J \triangle I_t} \frac{\sigma_a^2}{N_{t-1,a}}} & \text{(b)} \end{cases}$$

    if $Z_{t,I_t,J} \leqslant \beta(t, \delta)$ **then**

        Return False;

    end

end

Return True;

Algorithm: OFW

Input: $D_{\mathcal{A}}$, diameter of $\mathcal{S}_{\mathcal{A}}$

if *Get* **then**

  Return $(w_t, \tilde{w}_t, B_t)$;

if *Feed* **then**

  $F_t(x) = \frac{1}{t}\left(\sum_{s=1}^{t} \frac{1}{D_{\mathcal{A}}} s^{-1/4} ||x - \tilde{w}_{n_0}||_2^2 - \langle x, r_s \rangle\right)$;

  $\tilde{A}_t = \mathrm{argmin}_{A \in \mathcal{A}} \langle \mathbf{1}_A, \nabla F_t(\tilde{w}_t) \rangle$;

  $(\tilde{w}_{t+1}, w_{t+1}) = (1 - t^{-1/4})(\tilde{w}_t, w_t) + t^{-1/4}\left(\mathbf{1}_{\tilde{A}_t}, \delta_{\tilde{A}_t}\right)$;

  $B_{t+1} = B_t \cup \{\tilde{A}_t\}$;

Algorithm: LLOO

Input: Horizon T, upper bound on gradients G_T, $D_{\mathcal{A}}$ and $\rho_{\mathcal{A}} = \sqrt{d}\mu_{\mathcal{A}}$ depending on the geometry of $\mathcal{S}_{\mathcal{A}}$

Let $\gamma = (3\rho_{\mathcal{A}}^2)^{-1}$, $\eta = \frac{D_{\mathcal{A}}}{18 G_T \rho_{\mathcal{A}} \sqrt{T}}$ and $M = \min\left\{ \frac{\rho}{D} \frac{D_{\mathcal{A}}}{\sqrt{T}} \left(\rho_{\mathcal{A}} + \frac{1}{18\rho_{\mathcal{A}}} \right), 1 \right\}$;

if *Get* **then**

  Return $(w_t, \tilde{w}_t, B_t)$;

if *Feed* **then**

  $F_t(x) = ||x - \tilde{w}_{n_0}||_2^2 - \eta \left(\sum_{s=1}^{t} \langle x, r_s \rangle \right)$;

  $\tilde{A}_t = \mathrm{argmin}_{A \in \mathcal{A}} \langle \mathbf{1}_A, \nabla F_t(\tilde{w}_t) \rangle$;

  $(\tilde{w}_{t,-}, w_{t,-}) = \mathcal{A}^{\text{reduce}}(w_t, B_t, M, \nabla F_t(\tilde{w}_t))$;

  $(\tilde{w}_{t+1}, w_{t+1}) = (\tilde{w}_t, w_t) + \gamma \left(M \left(\mathbf{1}_{\tilde{A}_t}, \delta_{\tilde{A}_t} \right) - (\tilde{w}_{t,-}, w_{t,-}) \right)$;

  $B_{t+1} = B_t \cup \{\tilde{A}_t\}$;

Algorithm: LLOO's $\mathcal{A}^{\text{reduce}}$

Input: $w \in \Delta_{|\mathcal{A}|}$ with sparse support B, probability mass $M \in \mathbb{R}$ and cost vector $c \in \mathbb{R}^d$

$\forall A \in B, \quad l_A = \langle \mathbf{1}_A, c \rangle$;

Let $i_1, \cdots, i_{|B|}$ be a permutation such that $l_{A_{i_1}} \geqslant \cdots \geqslant l_{A_{i_{|B|}}}$;

Let k be the smallest integer such that $\sum_{j=1}^{k} w_{A_{i_j}} \geqslant M$;

$(\tilde{w}_-, w_-) = \sum_{j=1}^{k-1} w_{A_{i_j}} \left(\mathbf{1}_{A_{i_j}}, \delta_{A_{i_j}}\right) + \left(M - \sum_{j=1}^{k-1} w_{A_{i_j}}\right) \left(\mathbf{1}_{A_{i_k}}, \delta_{A_{i_k}}\right)$;

Return $(\tilde{w}_-, w_-)$;