Statistically Efficient, Polynomial-Time Algorithms for Combinatorial Semi-Bandits

Thibaut Cuvelier, Richard Combes, Eric Gourdin

Statistically Efficient, Polynomial-Time Algorithms for Combinatorial Semi-Bandits

Thibaut Cuvelier
Centrale-Supelec, L2S and Orange Labs
Gif-sur-Yvette, France
thibaut.cuvelier@centralesupelec.fr

Richard Combes
Centrale-Supelec, L2S
Gif-sur-Yvette, France
richard.combes@centralesupelec.fr

Eric Gourdin
Orange Labs
Chatillon, France
eric.gourdin@orange.com

ABSTRACT

We consider combinatorial semi-bandits over a set $\mathcal{X} \subset \{0,1\}^d$ where rewards are uncorrelated across items. For this problem, the algorithm ESCB yields the smallest known regret bound $R(T) = O\left(\frac{d(\ln m)^2(\ln T)}{\Delta_{\min}}\right)$ after T rounds, where $m = \max_{x\in\mathcal{X}} \mathbb{1}^\top x$. However, ESCB has computational complexity $O(|\mathcal{X}|)$, which is typically exponential in d, and cannot be used in large dimensions. We propose the first algorithm that is both computationally and statistically efficient for this problem with regret $R(T) = O\left(\frac{d(\ln m)^2(\ln T)}{\Delta_{\min}}\right)$ and computational asymptotic complexity $O(\delta_T^{-1}\mathbf{poly}(d))$, where δ_T is a function which vanishes arbitrarily slowly. Our approach involves carefully designing AESCB, an approximate version of ESCB with the same regret guarantees. We show that, whenever budgeted linear maximization over $\mathcal{X}$ can be solved up to a given approximation ratio, AESCB is implementable in polynomial time $O(\delta_T^{-1}\mathbf{poly}(d))$ by repeatedly maximizing a linear function over $\mathcal{X}$ subject to a linear budget constraint, and showing how to solve these maximization problems efficiently. Additional algorithms, proofs and numerical experiments are given in the complete version of this work.

CCS CONCEPTS

• **Mathematics of computing** → **Combinatorial optimization**; • **Theory of computation** → **Reinforcement learning**;

KEYWORDS

Bandits, Combinatorial Bandits, Combinatorial Optimization

ACM Reference Format:
Thibaut Cuvelier, Richard Combes, and Eric Gourdin. 2021. Statistically Efficient, Polynomial-Time Algorithms for Combinatorial Semi-Bandits. In *Abstract Proceedings of the 2021 ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '21 Abstracts), June 14–18, 2021, Virtual Event, China.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3410220.3453926

SIGMETRICS '21 Abstracts, June 14–18, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8072-0/21/06.
https://doi.org/10.1145/3410220.3453926

1 COMBINATORIAL SEMI-BANDITS

We consider combinatorial semi-bandits: time is discrete, and at times $t = 1,\dots,T$ a learner chooses a decision $x(t) \in \mathcal{X}$, where $\mathcal{X} \subset \{0,1\}^d$ is a combinatorial set which is known to the learner. Set $\mathcal{X}$ may be any combinatorial set, including the bases of a matroid, the set of paths in some graph, the set of matchings in a bipartite graph, etc. The problem dimension is d, and we define $m = \max_{x\in\mathcal{X}} \mathbb{1}^\top x$ the size of the largest decision. After selecting decision $x(t)$, the learner then receives a reward $Z(t)^\top x(t)$ and observes a feedback vector $Y(t) = (x_1(t)Z_1(t), \dots, x_d(t)Z_d(t))$, where $Z(t) \in [0,1]^d$ is a random vector.

We assume that $(Z(t))_t$ are i.i.d. with mean $\theta \in [0,1]^d$ and that the entries of $Z(t)$ are independent as well. Vector θ is initially unknown to the learner, and must be learnt by repetitively selecting decisions and observing subsequent feedback. For $i \in \{1,\dots,d\}$, if $x_i(t) = 1$, then the learner obtains a noisy realization of θ_i and nothing otherwise, so that decisions must be carefully selected to obtain a good estimate of θ. This is the "semi-bandit feedback" model. Since θ is unknown to the learner, decision $x(t)$ must be selected solely as a function of the feedback information available at time t, i.e. $Y(t-1),\dots,Y(1)$.

The expected reward received by selecting decision $x \in \mathcal{X}$ is $\theta^\top x$ (i.e. rewards are linear in the decision), so that θ_i represents the amount of reward received by selecting $x_i = 1$. The optimal decision is $x^\star \in \arg\max_{x\in\mathcal{X}}\{\theta^\top x\}$ (there may be several optimal decisions). We define the reward gap $\Delta_x = \theta^\top(x^\star - x)$, i.e. the amount of regret incurred to the learner by selecting decision x instead of $x^\star$. We denote by $\Delta_{\min} = \min_{x:\Delta_x>0} \Delta_x$ the smallest non-null gap.

The goal of the learner is to minimize the regret, which is simply the difference in terms of expected cumulative rewards between the learner and an oracle who knows the latent vector θ in advance and who always selects the optimal decision $x^\star$, that is:

$$R(T) = \sum_{t=1}^{T} \mathbb{E}(\Delta_{x(t)}).$$

Known algorithms for this problem include CUCB [2], ESCB [1] and TS [3].

2 THE AESCB ALGORITHM

We now propose AESCB (Approximate-ESCB), an algorithm that approximates ESCB and enjoys the same regret bound, while being implementable with polynomial complexity (unlike ESCB). The AESCB algorithm requires two sequences $(\varepsilon_t, \delta_t)$, which quantify the level of approximation at each time step. We define the following statistics, for $i = 1, \ldots, d$:

$$n_i(t) = \sum_{t'=1}^{t-1} x_i(t')$$

$$\hat{\theta}(t) = \frac{\sum_{t'=1}^{t-1} x_i(t') Z_i(t')}{\max(1, \sum_{t'=1}^{t-1} x_i(t'))}$$

$$\sigma_i^2(t) = \begin{cases} \frac{f(t)}{2n_i(t)} & \text{if } n_i(t) \geq 1 \\ +\infty & \text{otherwise.} \end{cases}$$

where, at time t, $n_i(t)$ is the number of samples obtained for θ_i, $\hat{\theta}_i(t)$ is the estimate of θ_i, and $\sigma_i^2(t)$ is proportional to the variance of estimate $\hat{\theta}_i(t)$. $f(t)$ is defined as $\ln t + 4m \ln\ln t$. We denote by $n(t) = (n_i(t))_{i=1,\ldots,d}$, $\hat{\theta}(t) = (\hat{\theta}_i(t))_{i=1,\ldots,d}$, and $\sigma^2(t) = (\sigma_i^2(t))_{i=1,\ldots,d}$ the corresponding vectors.

Definition 2.1 (AESCB). *The AESCB algorithm with approximation factors $(\varepsilon_t, \delta_t)_{t\geq 1}$ is the policy which at any time $t \geq 1$ selects a decision $x(t)$ verifying:*

$$\arg\max_{x \in \mathcal{X}} \{\hat{\theta}(t)^\top x + \sqrt{\sigma^2(t)^\top x}\} \leq \delta_t + \hat{\theta}(t)^\top x(t) + \frac{1}{\varepsilon_t} \sqrt{\sigma^2(t)^\top x(t)}$$

where ties are broken arbitrarily.

When $(\varepsilon_t, \delta_t) = (1, 0)$ for all $t \geq 1$, AESCB reduces to ESCB. Our first main result is Theorem 2.2, which provides a regret upper bound for AESCB. We show that, if one chooses approximation parameters $(\varepsilon_t, \delta_t)$ with $\varepsilon_t = \varepsilon > 0$ some fixed number and δ_t any sequence such that $\lim_{t\to\infty} \delta_t = 0$, then AESCB verifies the same (state-of-the-art) regret as ESCB up to a multiplicative constant. For m-sets, knapsack sets, and source destination paths, we choose $\varepsilon = 1$. For spanning trees, matroids, matchings, and matroid intersection, we choose $\varepsilon = \frac{1}{2}$ (see Section 3). This choice of parameters does not require any knowledge about the time horizon T, nor about the unknown problem parameters θ, nor about the minimal gap $\Delta_{\min}$. Nevertheless, if $\Delta_{\min}$ is known as well, we can select δ_t to yield an even better algorithm; however, knowing this parameter is by no means required. We can show that, with this choice of parameters, AESCB can be implemented in polynomial time.

Theorem 2.2 (Regret of AESCB). *The regret of AESCB with parameters $(\varepsilon_t, \delta_t)$ admits the following upper bound for all $T \geq 1$:*

$$R(T) \leq C_4(m) + \frac{2\,d\,m^3}{\Delta_{\min}^2} + \frac{24 d\, f(T)}{(\min_{t\leq T} \varepsilon_t)^2 \Delta_{\min}} \left\lceil \frac{\ln m}{1.61} \right\rceil^2 + 4 \sum_{t=1}^{T} \delta_t \mathbb{1}(\Delta_{\min} \leq 4\delta_t).$$

with $C_4(m)$ a positive number that solely depends on m. By corollary, for $\varepsilon_t = \varepsilon$ and $\lim_{t\to\infty} \delta_t = 0$, we have:

$$R(T) = O\left(d\,(\ln m)^2 \frac{1}{\Delta_{\min}} \ln T\right) \quad \text{as} \quad T \to \infty.$$

Similarly, with $\varepsilon_t = \varepsilon$ and $\delta_t < \frac{1}{4}\Delta_{\min}$, we have, for all $T \geq 1$:

$$R(T) \leq C_4(m) + \frac{2\,d\,m^3}{\Delta_{\min}^2} + \frac{24 d\, f(T)}{\varepsilon^2 \Delta_{\min}} \left\lceil \frac{\ln m}{1.61} \right\rceil^2.$$

3 AESCB IN POLYNOMIAL TIME

We now show a technique to implement AESCB that ensures polynomial time complexity. While our methodology is generic, the precise value of the computational complexity depends on the combinatorial set $\mathcal{X}$. Our approach involves three steps: rounding and scaling to ensure that the weights are integer, then solving a budgeted linear maximization over $\mathcal{X}$ several times, and finally maximizing over the budget to obtain the result. Given time t, statistics $\hat{\theta}(t)$ and $\sigma^2(t)$, and approximation factors $(\varepsilon_t, \delta_t)$, the method works as follows.

Step 1: rounding and scaling. Define $a(t)$ and $b(t)$:

$$\xi(t) = \lceil m/\delta_t \rceil.$$

$$a_i(t) = \lceil \xi(t) \hat{\theta}_i(n) \rceil \ , \ i \in \{1, \ldots, d\}$$

$$b_i(t) = \xi(t)^2 \sigma_i^2(t) \ , \ i \in \{1, \ldots, d\}$$

Step 2: budgeted linear maximization. For all $s \in \{0, \ldots, m\xi(t)\}$, compute $\bar{x}^s(t)$, an ε_t-optimal solution to budgeted linear maximization problem:

$$\bar{x}^s(t) \geq \varepsilon_t \left(\max_{x \in \mathcal{X}: a(t)^\top x \geq s} \{b(t)^\top x\} \right) \quad \text{and} \quad a(t)^\top \bar{x}^s(t) \geq s.$$

Step 3: optimizing over a budget. Return decision $x(t)$:

$$x(t) = \bar{x}^{s^\star(t)}(t) \text{ with}$$

$$s^\star(t) \in \arg\max_{s=0,\ldots,m\xi(t)} \left\{ s + \frac{1}{\varepsilon_t} \sqrt{b(t)^\top \bar{x}^s(t)} \right\}.$$

$a(t)$ is defined using a ceiling operation in order to ensure that $a(t)^\top x$ has an integer value for any $x \in \mathcal{X}$, while $b(t)$ does not need to have integer entries. Theorem 3.1 states that this technique returns the decision chosen by AESCB, in a time proportional to solving budgeted linear maximization at most $m\xi(t)$ times (where $\xi(t)$ is bounded by a polynomial in d), and that the input parameters $a(t)$ and $b(t)$ are positive vectors and where the entries of $a(t)$ are in $\{1, \ldots, \xi(t)\}$. For many combinatorial sets of interest, budgeted linear maximization over $\mathcal{X}$ can be done in polynomial time in the dimension, so that AESCB is indeed implementable in polynomial time, see the complete version of this work where we provide algorithms to do so.

Theorem 3.1. *The above algorithm returns a decision $x(t) \in \mathcal{X}$ verifying the AESCB definition. It does so by maximizing $x^\top b(t)$ subject to $x^\top a(t) \geq s$ over $\mathcal{X}$ at most $m\xi(t)$ times with input parameters $a(t)$ and $b(t)$, where $a(t) \in \{1, \ldots, \xi(t)\}^d$ and $b(t) \in \mathbb{R}^d$.*

4 CONCLUSION

We propose AESCB, the first algorithm which enjoys both the state-of-the art regret bound of ESCB and polynomial computational complexity.

REFERENCES

[1] Richard Combes, M. Sadegh Talebi, Alexandre Proutiere, and Marc Lelarge. 2015. Combinatorial Bandits Revisited. In *Proc. of NIPS.*

[2] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. 2015. Tight Regret Bounds for Stochastic Combinatorial Semi-Bandits. In *Proc. of AISTATS.*

[3] Siwei Wang and Wei Chen. 2018. Thompson Sampling for Combinatorial Semi-Bandits. In *Proc. of ICML.*