Active clustering with bandit feedback

joint work with Alexandra Carpentier[1], Christophe Giraud[2] and Nicolas Verzelen[3]

Victor Thuot[3]

[1]University Potsdam

[2]LMO Orsay

[3]INRAE Montpellier

2nd of May 2024, seminar ML-MTP

Introduction : from batch to active clustering

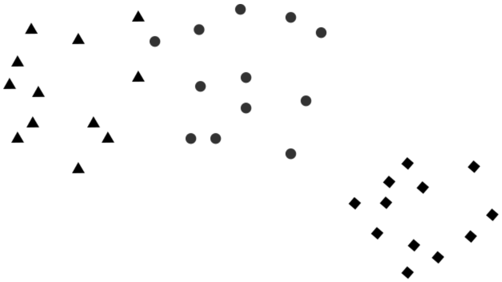

Figure: $X_1, \ldots, X_T$, data points in $\mathbb{R}^d$ partitioned in $K = 3$ groups

Introduction : from batch to active clustering

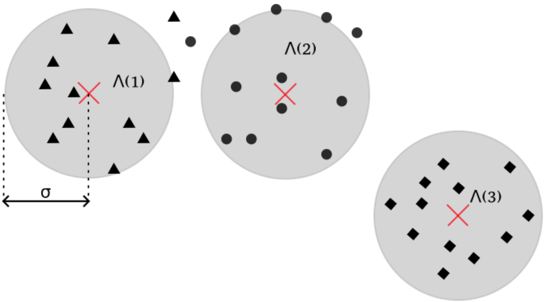

Figure: $\forall t \in [T]$, $X_t \sim \mathcal{N}(\Lambda(k_t)), \sigma^2 I_d)$ with $k_t \in [K]$

Introduction : from batch to active clustering

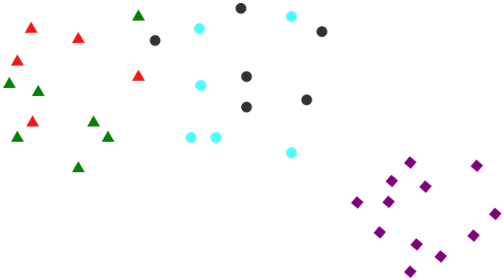

Figure: $N = 5$ arms, partitioned in $K = 3$ groups

Introduction : from batch to active clustering

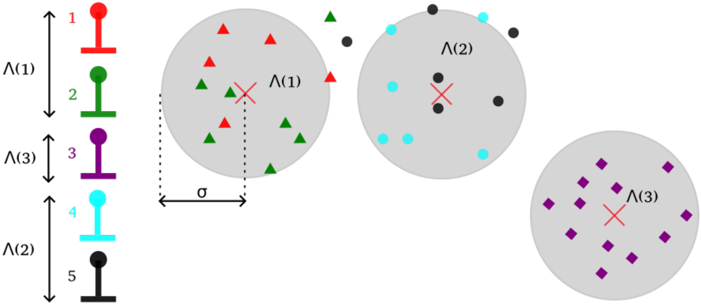

Figure: $N = 5$ arms, partitioned in $K = 3$ groups

Introduction : from batch to active clustering

- What if the data points are collected sequentially ?
- What if the learner can chose the order of the observations ?

Active setting

An algorithm collects sequentially and actively data, by interacting with a stochastic (Gaussian) bandit.

Active setting

An algorithm collects sequentially and actively data, by interacting with a stochastic (Gaussian) bandit.

- N arms
- each arm $a \in [N] \leftrightarrow$ probability distribution $\mathcal{N}(\mu_a, \sigma^2 I_d)$
- mean-vector $\mu_a \in \mathbb{R}^d$

Active setting

An algorithm collects sequentially and actively data, by interacting with a stochastic (Gaussian) bandit.

- N arms
- each arm $a \in [N] \leftrightarrow$ probability distribution $\mathcal{N}(\mu_a, \sigma^2 I_d)$
- mean-vector $\mu_a \in \mathbb{R}^d$

Active setting

An algorithm collects sequentially and actively data, by interacting with a stochastic (Gaussian) bandit.

- N arms
- each arm $a \in [N] \leftrightarrow$ probability distribution $\mathcal{N}(\mu_a, \sigma^2 I_d)$
- mean-vector $\mu_a \in \mathbb{R}^d$
- at each time $t \geqslant 1$, the algorithm chooses arm $A_t \in [N]$ (based on passed observations)
- the algorithm receives X_t, s.t.,
  conditionally on $A_t = a$, $X_t \sim \mathcal{N}(\mu_a, \sigma^2 I_d)$

Active setting

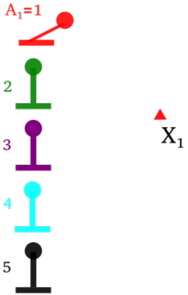

Figure: $A_1 = 1$ and receives $X_1 \sim \mathcal{N}(\mu_{A_1}, \sigma^2 I_d)$

Active setting

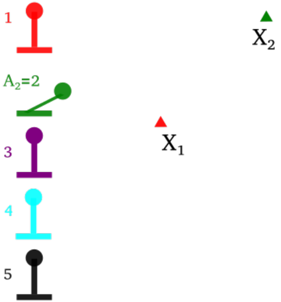

Figure: $A_2 = 2$ and receives $X_2 \sim \mathcal{N}(\mu_{A_2}, \sigma^2 I_d)$

Active setting

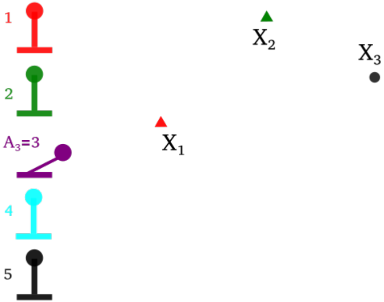

Figure: $A_3 = 3$ and receives $X_3 \sim \mathcal{N}(\mu_{A_3}, \sigma^2 I_d)$

Active setting

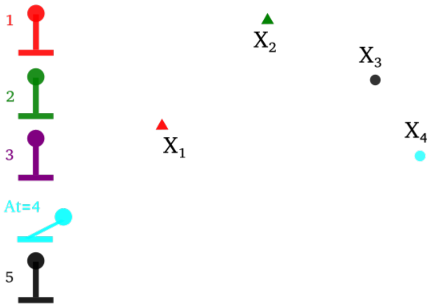

Figure: $A_4 = 4$ and receives $X_4 \sim \mathcal{N}(\mu_{A_4}, \sigma^2 I_d)$

Active seting

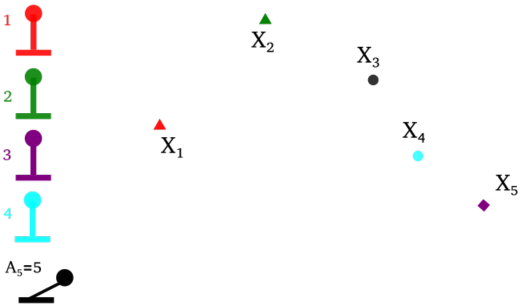

Figure: $A_5 = 5$ and receives $X_5 \sim \mathcal{N}(\mu_{A_5}, \sigma^2 I_d)$

Active setting

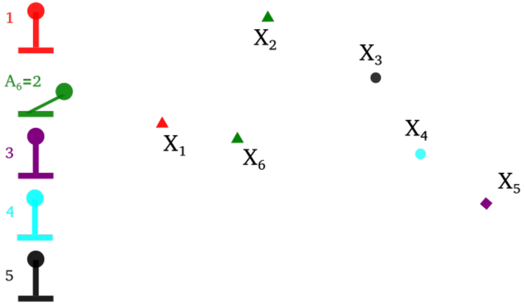

Figure: multiple sampling

Active clustering problem

The Active Clustering Problem (ACP) ([Yang et al., 2024]):

- a bandit with N arms with means $\mu_1, \ldots, \mu_N$

Active clustering problem

The Active Clustering Problem (ACP) ([Yang et al., 2024]):

- a bandit with N arms with means $\mu_1, \ldots, \mu_N$

Assumption 1: hidden partition

There exists a hidden partition G^* of $[N]$ into K groups.

For $a, b \in [N]$, a, b are in the same group $\Leftrightarrow \mu_a = \mu_b$.

Active clustering problem

The Active Clustering Problem (ACP) ([Yang et al., 2024]):

- a bandit with N arms with means $\mu_1, \ldots, \mu_N$

Assumption 1: hidden partition

There exists a hidden partition G^* of $[N]$ into K groups.

For $a, b \in [N]$, a, b are in the same group $\Leftrightarrow \mu_a = \mu_b$.

Objective

- recover the unknown partition G^* using as few requests as possible

Active clustering problem

The Active Clustering Problem (ACP) ([Yang et al., 2024]):

- a bandit with N arms with means $\mu_1, \dots, \mu_N$

Assumption 1: hidden partition

There exists a hidden partition G^* of $[N]$ into K groups.

For $a, b \in [N]$, a, b are in the same group $\Leftrightarrow \mu_a = \mu_b$.

Objective

- recover the unknown partition G^* using as few requests as possible
- sample arms until time T (budget)

Active clustering problem

The Active Clustering Problem (ACP) ([Yang et al., 2024]):

- a bandit with N arms with means $\mu_1, \ldots, \mu_N$

Assumption 1: hidden partition

There exists a hidden partition G^* of $[N]$ into K groups.

For $a, b \in [N]$, a, b are in the same group $\Leftrightarrow \mu_a = \mu_b$.

Objective

- recover the unknown partition G^* using as few requests as possible
- sample arms until time T (budget)
- output: $\hat{G}$ estimate of G^*.

Remarks

- assume the groups nonempty and disjoints
- K is known (and also N and d)
- $\Lambda(k)$ is the common mean of arms in G_k^* (for $k \in [K]$)
- the problem is defined up to permutation of the groups
- σ^2 known

Active clustering problem

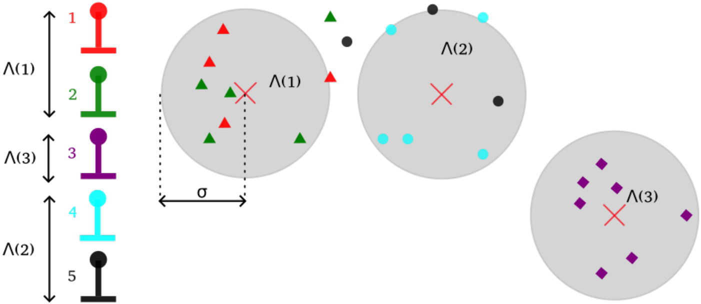

Figure: At time t, estimating $\hat{G}$ or sampling a new point.

Active clustering problem

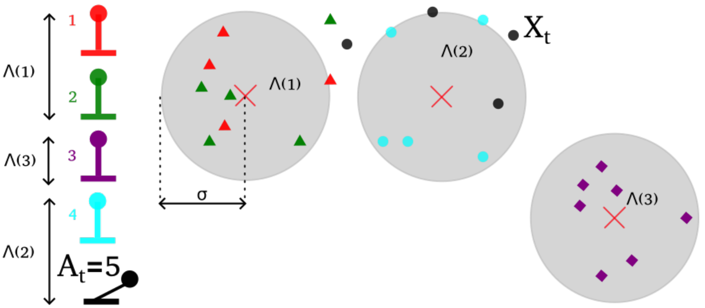

Figure: Based on $X_1, \ldots, X_{t-1}$, choose A_t and observe X_t.

PAC setting

δ-PAC algorithm

Given $\delta \in (0,1)$, an algorithm π for the ACP is said to be δ-PAC on a collection of environments $\mathcal{E}$ if for any ν, then

$$\mathbb{P}_{\pi,\nu}(\hat{G} \sim G^*) \geqslant 1 - \delta \ .$$

where $\sim$ means that the partition is exact up to permutation of the groups.

- We define the optimal worst case (expected) budget on the collection $\mathcal{E}$ as

$$T^*(\delta, \mathcal{E}) = \inf_{\pi} \sup_{\nu \in \mathcal{E}} \mathbb{E}_{\pi,\nu}[T] \ ,$$

where π is δ-PAC on $\mathcal{E}$.

PAC setting

δ-PAC algorithm

Given $\delta \in (0, 1)$, an algorithm π for the ACP is said to be δ-PAC on a collection of environments $\mathcal{E}$ if for any ν, then

$$\mathbb{P}_{\pi,\nu}(\hat{G} \sim G^*) \geqslant 1 - \delta \ .$$

where $\sim$ means that the partition is exact up to permutation of the groups.

- We define the optimal worst case (expected) budget on the collection $\mathcal{E}$ as

$$T^*(\delta, \mathcal{E}) = \inf_{\pi} \sup_{\nu \in \mathcal{E}} \mathbb{E}_{\pi,\nu}[T] \ ,$$

where π is δ-PAC on $\mathcal{E}$.

PAC setting

δ-PAC algorithm

Given $\delta \in (0,1)$, an algorithm π for the ACP is said to be δ-PAC on a collection of environments $\mathcal{E}$ if for any ν, then

$$\mathbb{P}_{\pi,\nu}(\hat{G} \sim G^*) \geqslant 1 - \delta \ .$$

where $\sim$ means that the partition is exact up to permutation of the groups.

- We define the optimal worst case (expected) budget on the collection $\mathcal{E}$ as

$$T^*(\delta, \mathcal{E}) = \inf_{\pi} \sup_{\nu \in \mathcal{E}} \mathbb{E}_{\pi,\nu}[T] \ ,$$

where π is δ-PAC on $\mathcal{E}$.

PAC setting

δ-PAC algorithm

Given $\delta \in (0,1)$, an algorithm π for the ACP is said to be δ-PAC on a collection of environments $\mathcal{E}$ if for any ν, then

$$\mathbb{P}_{\pi,\nu}(\hat{G} \sim G^*) \geqslant 1 - \delta \ .$$

where $\sim$ means that the partition is exact up to permutation of the groups.

- We define the optimal worst case (expected) budget on the collection $\mathcal{E}$ as

$$T^*(\delta, \mathcal{E}) = \inf_{\pi} \sup_{\nu \in \mathcal{E}} \mathbb{E}_{\pi,\nu}[T] \ ,$$

where π is δ-PAC on $\mathcal{E}$.

Parameters of interest

First, we consider $\Delta_* = \Delta_*(\nu)$ for the minimal gap,

$$\Delta_* = \min_{k \neq k'} \|\Lambda(k) - \Lambda(k')\| > 0 \ .$$

Besides, we denote m_* as the size of the smallest group,

$$m_* = \min_{k \in [K]} |G_k^*| \ .$$

Parameters of interest

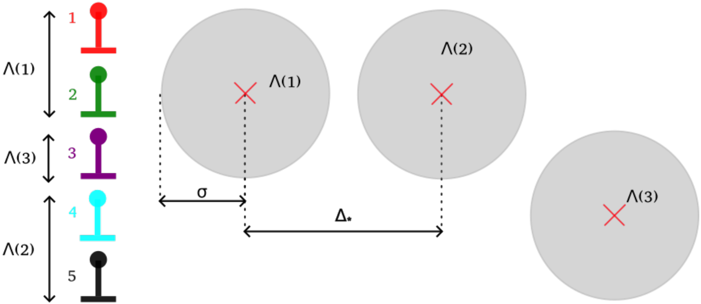

Figure: Here $N = 5$, $K = 3$, $d = 2$, $m_* = 1$.

Construction of a family of environments

- Consider $\Delta > 0$, and $m \geqslant 1$, we define

$$\mathcal{E} := \mathcal{E}(\Delta, m) \quad ,$$

as the collection of environments ν such that $\Delta_*(\nu) \geqslant \Delta$, $m_*(\nu) \geqslant m$.

Construction of a family of environments

- Consider $\Delta > 0$, and $m \geqslant 1$, we define

$$\mathcal{E} := \mathcal{E}(\Delta, m) \ ,$$

as the collection of environments ν such that $\Delta_*(\nu) \geqslant \Delta$, $m_*(\nu) \geqslant m$.

- We study the optimal worst case budget on $\mathcal{E}(\Delta, m)$

$$T^*(\delta, \mathcal{E}(\Delta, m)) \ .$$

Benchmark: the uniform sampling strategy

Uniform sampling strategy (US):

1. sample U times each arm
2. use (batch) clustering on the empirical means

Benchmark: the uniform sampling strategy

Uniform sampling strategy (US):

1. sample U times each arm
2. use (batch) clustering on the empirical means

- (shrinkage $\sigma^2 \mapsto \sigma^2/U$)

Benchmark: the uniform sampling strategy

Uniform sampling strategy (US):

1. sample U times each arm
2. use (batch) clustering on the empirical means

- (shrinkage $\sigma^2 \mapsto \sigma^2/U$)
- Benchmark: for a confidence bound $\delta = 1/N$, balanced group $m = N/K$, uniform sampling is $\delta - PAC$ on $\mathcal{E}(\Delta, m)$ as long as

$$T = NU \gtrsim \frac{\sigma^2}{\Delta^2}\left[N(\log(N) \vee K) + \sqrt{dKN(\log(N) \vee K)}\right]$$

([Royer, 2017, Giraud and Verzelen, 2019, Vempala and Wang, 2004])

Benchmark: the uniform sampling strategy

Uniform sampling strategy (US):

1. sample U times each arm
2. use (batch) clustering on the empirical means

- (shrinkage $\sigma^2 \mapsto \sigma^2/U$)
- Benchmark: for a confidence bound $\delta = 1/N$, balanced group $m = N/K$, uniform sampling is $\delta - PAC$ on $\mathcal{E}(\Delta, m)$ as long as

$$T = NU \gtrsim \frac{\sigma^2}{\Delta^2}\left[N(\log(N) \vee K) + \sqrt{dKN(\log(N) \vee K)}\right]$$

([Royer, 2017, Giraud and Verzelen, 2019, Vempala and Wang, 2004])

- Information-computation gap

Benchmark: optimal result in the asymptotic regime $\delta \to 0$

The problem was introduced in [Yang et al., 2024], they provide:

1. Instance-dependent lower bound
2. An algorithm optimal in the asymptotic regime $\delta \to 0$

- for instance, with $m = N/K$, equal distance between the groups,

$$\lim_{\delta \to 0} \frac{T^*(\delta, \mathcal{E}(\Delta, m))}{\log(1/\delta)} = 2\frac{\sigma^2}{\Delta^2}(N + K)$$

- methodology: ([Garivier and Kaufmann, 2016])

Contributions

1. Can we improve the budget of a simple uniform sampling strategy ?

Contributions

1. Can we improve the budget of a simple uniform sampling strategy ?

2. Can we achieve $T^*(\delta, \mathcal{E}(\Delta, m))$?

Contributions

1. Can we improve the budget of a simple uniform sampling strategy ?

2. Can we achieve $T^*(\delta, \mathcal{E}(\Delta, m))$?

3. Is there an information-computation gap for ACP?

Contributions

1. Can we improve the budget of a simple uniform sampling strategy ?

Yes, we provide the ACB Algorithm, a polynomial-time algorithm which outperforms the uniform sampling strategy.

2. Can we achieve $T^*(\delta, \mathcal{E}(\Delta, m))$?

3. Is there an information-computation gap for ACP?

Contributions

1. Can we improve the budget of a simple uniform sampling strategy ?

Yes, we provide the ACB Algorithm, a polynomial-time algorithm which outperforms the uniform sampling strategy.

2. Can we achieve $T^*(\delta, \mathcal{E}(\Delta, m))$?

Yes, $T^* \simeq \frac{\sigma^2}{\Delta^2}\left[N \log\left(\frac{N}{\delta}\right) + \sqrt{dKN \log\left(\frac{N}{\delta}\right)}\right]$.

3. Is there an information-computation gap for ACP?

Contributions

1. Can we improve the budget of a simple uniform sampling strategy ?

Yes, we provide the ACB Algorithm, a polynomial-time algorithm which outperforms the uniform sampling strategy.

2. Can we achieve $T^*(\delta, \mathcal{E}(\Delta, m))$?

Yes, $T^* \simeq \frac{\sigma^2}{\Delta^2}\left[N \log\left(\frac{N}{\delta}\right) + \sqrt{dKN \log\left(\frac{N}{\delta}\right)}\right].$

3. Is there an information-computation gap for ACP?

No, there is no computational gap, ACB is optimal in most emblematic regimes (e.g., balanced setting).

Information-theoretic lower bound

Theorem 1

There exists a universal constant $c > 0$, such that the following holds for any $\Delta > 0$, any $m \geq 2$, any $\delta \in (0, 1/12)$, and any $N \geq mK$:

$$T^*(\delta, \mathcal{E}(\Delta, m)) \geqslant c \left[N + \frac{\sigma^2}{\Delta^2} N \log\left(\frac{N}{\delta}\right) + \frac{\sigma^2}{\Delta^2} \sqrt{dKN \log\left(\frac{N}{\delta}\right)} \right].$$

Comments

Dichotomy between low-dimensional and high-dimensional clustering problems:

$$T^* \geqslant c\left[\frac{\sigma^2}{\Delta^2} N \log\left(\frac{N}{\delta}\right) + \frac{\sigma^2}{\Delta^2}\sqrt{dKN \log\left(\frac{N}{\delta}\right)}\right] \quad .$$

Comments

Dichotomy between low-dimensional and high-dimensional clustering problems:

$$T^* \geqslant c \left[\frac{\sigma^2}{\Delta^2} N \log\left(\frac{N}{\delta}\right) + \frac{\sigma^2}{\Delta^2}\sqrt{dKN \log\left(\frac{N}{\delta}\right)} \right] .$$

- In low-dimension, reduction to the thresholding bandit problem ([Cheshire et al., 2020, Chen and Li, 2015, Chen et al., 2014]).
- Simpler problem: $K = 2, d = 1$, $\mu_a \in \{0, \Delta\}$ with Δ known

Comments

$$T^* \geqslant c\left[N + \frac{\sigma^2}{\Delta^2} N \log\left(\frac{N}{\delta}\right) + \frac{\sigma^2}{\Delta^2}\sqrt{dKN \log\left(\frac{N}{\delta}\right)}\right] .$$

- In high-dimension, we use a series of reduction, driven by two ideas:
  1. Bayesian approach: chose a "good" prior on the unknwon centers
  2. ACP is at least as “difficult”, as the (active) supervised problem

Upper bound: the ACB algorithm

We derive an algorithm called Active Clustering with Bandit (ACB).
Main structure of the algorithm:

1. identify $\hat{S}$ a set of K arms with exactly one arm from each group

Upper bound: the ACB algorithm

We derive an algorithm called Active Clustering with Bandit (ACB). Main structure of the algorithm:

1. identify $\hat{S}$ a set of K arms with exactly one arm from each group
2. estimate the unknown centres of the groups using $\hat{S}$

Upper bound: the ACB algorithm

We derive an algorithm called Active Clustering with Bandit (ACB).
Main structure of the algorithm:

1. identify $\hat{S}$ a set of K arms with exactly one arm from each group
2. estimate the unknown centres of the groups using $\hat{S}$
3. sample uniformly the remaining arms and use a distance-based classifier

First step: identification of representatives

Imagine that $\hat{S}$ contains $k < K$ arms from different clusters. Until $|\hat{S}| < K$,

- take randomly a new candidate $b \in [N]$
- perform a sequence of tests to decide whether b should be added to $\hat{S}$ or not
- the objective is to reject quickly arms whose groups are already represented

We use for the tests sub-sampling and high dimensional two-sample testing.
Similar to an elimination technique
[de Heide et al., 2021, Jamieson and Nowak, 2014].

Upper bound: main theorem

Theorem 2

Let $\delta > 0$. The ACB algorithm is δ-PAC on $\mathcal{E}(\Delta, m)$.
If we assume that $m \geqslant \log(K)$, then

$$\mathbb{E}_{ACB,\nu}[T] \leqslant c\frac{\sigma^2}{\Delta^2}\left[N\log\left(N/\delta\right) + \sqrt{dNK\log\left(N/\delta\right)} + \sqrt{d}\frac{N\log(K)}{m}\right]$$

Upper bound: main theorem

Theorem 2

Let $\delta > 0$. The ACB algorithm is δ-PAC on $\mathcal{E}(\Delta, m)$.
If we assume that $m \geqslant \log(K)$, then

$$\mathbb{E}_{ACB,\nu}[T] \leqslant c \frac{\sigma^2}{\Delta^2}\left[N \log\left(N/\delta\right) + \sqrt{dNK \log\left(N/\delta\right)} + \sqrt{d}\frac{N \log(K)}{m}\right]$$

- c is independent of all parameters

Upper bound: main theorem

Theorem 2

Let $\delta > 0$. The ACB algorithm is δ-PAC on $\mathcal{E}(\Delta, m)$.
If we assume that $m \geqslant \log(K)$, then

$$\mathbb{E}_{ACB,\nu}[T] \leqslant c\frac{\sigma^2}{\Delta^2}\left[N\log\left(N/\delta\right) + \sqrt{dNK\log\left(N/\delta\right)} + \sqrt{d}\frac{N\log(K)}{m}\right]$$

- c is independent of all parameters
- we also give a bound with high probability

Comments

$$\mathbb{E}[T] \leqslant c\frac{\sigma^2}{\Delta^2}\left[N\log\left(N/\delta\right) + \sqrt{dNK\log\left(N/\delta\right)} + \sqrt{d}\frac{N\log(K)}{m}\right]$$

Comments

$$\mathbb{E}[T] \leqslant c\frac{\sigma^2}{\Delta^2}\left[N\log\left(N/\delta\right) + \sqrt{dNK\log\left(N/\delta\right)} + \sqrt{d}\frac{N\log(K)}{m}\right]$$

1. the condition $m \geqslant \log(K)$ is not too restrictive

Comments

$$\mathbb{E}[T] \leqslant c\frac{\sigma^2}{\Delta^2}\left[N\log\left(N/\delta\right) + \sqrt{dNK\log\left(N/\delta\right)} + \sqrt{d}\frac{N\log(K)}{m}\right]$$

1. the condition $m \geqslant \log(K)$ is not too restrictive
2. the upper bound is optimal if m is large enough, $\left(m \geqslant \sqrt{\frac{N}{K}}\frac{\log(K)}{\sqrt{\log(N/\delta)}}\right)$

Comments

$$\mathbb{E}[T] \leqslant c\frac{\sigma^2}{\Delta^2}\left[N\log\left(N/\delta\right) + \sqrt{dNK\log\left(N/\delta\right)} + \sqrt{d}\frac{N\log(K)}{m}\right]$$

1. the condition $m \geqslant \log(K)$ is not too restrictive
2. the upper bound is optimal if m is large enough, $\left(m \geqslant \sqrt{\frac{N}{K}}\frac{\log(K)}{\sqrt{\log(N/\delta)}}\right)$
3. the algorithm is polynomial $\rightarrow$ no information-computation gap

Take home message

1. There is no computational gap for the ACP.
2. We provide a lower bound on the budget

$$T^* \geqslant c\left[\frac{\sigma^2}{\Delta^2} N \log\left(\frac{N}{\delta}\right) + \frac{\sigma^2}{\Delta^2}\sqrt{dKN\log\left(\frac{N}{\delta}\right)}\right] \quad .$$

3. We provide a polynomial time δ-PAC algorithm called ACB, together with an upper bound on its budget which matches the lower bound for mild assumption on m_*.

Thank you !

Chen, L. and Li, J. (2015).
On the optimal sample complexity for best arm identification.
arXiv preprint arXiv:1511.03774.

Chen, S., Lin, T., King, I., Lyu, M. R., and Chen, W. (2014).
Combinatorial pure exploration of multi-armed bandits.
Advances in neural information processing systems, 27.

Cheshire, J., Ménard, P., and Carpentier, A. (2020).
The influence of shape constraints on the thresholding bandit problem.
In *Conference on Learning Theory*, pages 1228–1275. PMLR.

de Heide, R., Cheshire, J., Ménard, P., and Carpentier, A. (2021).
Bandits with many optimal arms.
Advances in Neural Information Processing Systems, 34:22457–22469.

Garivier, A. and Kaufmann, E. (2016).
Optimal best arm identification with fixed confidence.
In *Conference on Learning Theory*, pages 998–1027. PMLR.

Giraud, C. and Verzelen, N. (2019).
Partial recovery bounds for clustering with the relaxed k-means.
Mathematical Statistics and Learning, 1(3):317–374.

Jamieson, K. and Nowak, R. (2014).
Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting.
In *2014 48th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE.

Royer, M. (2017).
Adaptive clustering through semidefinite programming.
Advances in Neural Information Processing Systems, 30.

Vempala, S. and Wang, G. (2004).
A spectral algorithm for learning mixture models.
Journal of Computer and System Sciences, 68(4):841–860.

Yang, J., Zhong, Z., and Tan, V. Y. (2024).
Optimal clustering with bandit feedback.
Journal of Machine Learning Research, 25:1–54.

Sub-Gaussian noise

A mean-zero random variable Z is subGaussian if, for any $t > 0$, we have $\mathbb{E}[\exp(tZ)] \leq \exp(t^2/2)$.

Assumption 2: sub-Gaussian noise

- For any $a \in [N]$, if X is sampled from the arm a,

$$E = \Sigma_a^{-1/2}[X - \mu_a]$$

  is made of independent subGaussian random variables,
- Σ_a is a $d \times d$ symmetric matrix associated to a,
- there exists σ such that $\max_{a \in [N]} \|\Sigma_a\|_{op} \leqslant \sigma^2$.

- Exemples : bounded noise or Gaussian noise.

First step: identification of representatives

Consider for $s \in [r]$ the s-th test performed on the candidate b.
For $a \in \hat{S}$, we compute the statistic

$$\langle \bar{\mu}_{b,s} - \hat{\mu}_a, \bar{\mu}'_{b,s} - \hat{\mu}'_a \rangle$$

- $\hat{\mu}_{b,s}$, $\hat{\mu}'_{b,s}$ are two independent estimation of μ_b computed with n_s samples
- $\hat{\mu}_a$, $\hat{\mu}_a$ are estimates of μ_a computed with n_{max} samples
- the expectation of this statistic is $\|\mu_a - \mu_b\|^2$
- we reject b if it is smaller than $\Delta^2/2$ fo some $a \in \hat{S}$.

We use sub-Gaussian concentration to choose the tuning parameters n_s, n_{max}, r.

Second step: classification

Imagine that $\hat{S} = \{a_1, \ldots, a_K\}$ contains one arm from each group.

1. First, for $j \in [K]$, label a_j with j and estimate $\mu(j)$ with two independent means using $2J$ samples.
2. Then, for each $b \in [N] \setminus \hat{S}$, labels b in the group

$$\mathrm{argmin}_{j=1,\ldots,K}\left\langle \hat{\mu}_b - \hat{\mu}(j), \hat{\mu}'_b - \hat{\mu}'(j)\right\rangle$$

$\hat{\mu}_b$,$\hat{\mu}'_b$ are computed with $I = KJ/N$ samples