Clustering Items through Bandit Feedback

Finding the Right Feature out of Many

Maximilian Graf* **Victor Thuot**† Nicolas Verzelen†

June 18, 4th ASCAI (closing) Workshop, Orsay

* Institut für Mathematik, Universität Potsdam, Potsdam, Germany

† INRAE, Mistea, Institut Agro, Univ Montpellier, Montpellier, France

Setting

Motivating example

- a set of forest patches
- a set of automatic biodiversity sensors

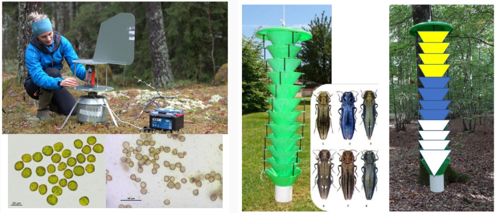

Figure 1: DNA sensor (left), optical sensor (right)[1]

- Objective: partition the forest patches by their biodiversity
- Limitations: cost, lack of specialists, unknown sensors

1. C. Bouget et al. **"Bioc@pt : Capteurs automatiques de biodiversité en forêt"**. In: 2021.

Clustering Items ...

- n: number of forest patches
- d: number of sensors

Clustering Items . . .

- n: number of forest patches
- d: number of sensors
- $M_{i,j}$: mean value of the j-th sensor on the i-th patch

$$M = \begin{bmatrix} M_{1,1} & \cdots & M_{1,j} & \cdots & M_{1,d} \\ \vdots & & \vdots & & \vdots \\ M_{i,1} & \cdots & M_{i,j} & \cdots & M_{i,d} \\ \vdots & & \vdots & & \vdots \\ M_{n,1} & \cdots & M_{n,j} & \cdots & M_{n,d} \end{bmatrix} \leftarrow M_{i,\cdot} \in \{\mu_0, \mu_1\}$$

Clustering Items . . .

- n: number of forest patches (items)
- d: number of sensors (features)
- $M_{i,j}$: mean value of the j-th sensor on the i-th patch

$$M = \begin{bmatrix} M_{1,1} & \cdots & M_{1,j} & \cdots & M_{1,d} \\ \vdots & & \vdots & & \vdots \\ M_{i,1} & \cdots & M_{i,j} & \cdots & M_{i,d} \\ \vdots & & \vdots & & \vdots \\ M_{n,1} & \cdots & M_{n,j} & \cdots & M_{n,d} \end{bmatrix} \leftarrow M_{i,\cdot} \in \{\mu_0, \mu_1\}$$

- n: number of forest patches (items)
- d: number of sensors (features)
- $M_{i,j}$: mean value of the j-th sensor on the i-th patch

$$M = \begin{bmatrix} M_{1,1} & \cdots & M_{1,j} & \cdots & M_{1,d} \\ \vdots & & \vdots & & \vdots \\ M_{i,1} & \cdots & M_{i,j} & \cdots & M_{i,d} \\ \vdots & & \vdots & & \vdots \\ M_{n,1} & \cdots & M_{n,j} & \cdots & M_{n,d} \end{bmatrix} \leftarrow M_{i,\cdot} \in \{\mu_0, \mu_1\}$$

- Hidden partition: there are two groups of patches

$$\boxed{\exists \mu_0 \neq \mu_1 \in \mathbb{R}^d;\ \forall i \in \{1, \dots, n\},\ M_{i,\cdot} \in \{\mu_0, \mu_1\}}$$

Clustering Items . . .

- n: number of forest patches (items)
- d: number of sensors (features)
- $M_{i,j}$: mean value of the j-th sensor on the i-th patch

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \leftarrow M_{i,\cdot} \in \{0, \Delta\}$$

- Hidden partition: there are two groups of patches

$$\boxed{\exists \mu_0 \neq \mu_1 \in \mathbb{R}^d;\ \forall i \in \{1, \dots, n\},\ M_{i,\cdot} \in \{\mu_0, \mu_1\}}$$

- Gap vector Denote $\Delta = \mu_1 - \mu_0$ (w.l.o.g, $\mu_0 = 0$)

Clustering Items ...

- n: number of forest patches (items)
- d: number of sensors (features)
- $M_{i,j}$: mean value of the j-th sensor on the i-th patch

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \leftarrow M_{i,\cdot} \in \{0, \Delta\}$$

- Hidden partition: there are two groups of patches

$$\boxed{\exists \mu_0 \neq \mu_1 \in \mathbb{R}^d;\ \forall i \in \{1, \ldots, n\},\ M_{i,\cdot} \in \{\mu_0, \mu_1\}}$$

- Gap vector Denote $\Delta = \mu_1 - \mu_0$ (w.l.o.g, $\mu_0 = 0$)

- **Objective 1**: recover the partition of patches

... through Bandit feedback

Vanilla clustering: observe the entire matrix

Bandit clustering: adapt the learning strategy on the fly

Vanilla clustering: observe the entire matrix

Bandit clustering: adapt the learning strategy on the fly

Learning protocol

At each time step t,

- choose a patch $I_t \in 1, \ldots, n$ (*based on the past*)
- choose a sensor $J_t \in 1, \ldots, d$ (*based on the past*)
- receive X_t, s.t., $X_t = M_{I_t,J_t}$+random noise

...through Bandit feedback

Vanilla clustering: observe the entire matrix

Bandit clustering: adapt the learning strategy on the fly

Learning protocol

At each time step t,

- choose a patch $I_t \in 1, \ldots, n$ (*based on the past*)
- choose a sensor $J_t \in 1, \ldots, d$ (*based on the past*)
- receive X_t, s.t., $X_t = M_{I_t, J_t}$+random noise

Objective 1: recover the partition of the patches ...

Objective 2: ... while minimizing the budget

$\rightarrow$ focusing the budget on the most informative sensor

Sampling protocol (example)

$$\begin{bmatrix} \boxed{0} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Time t	1	2	3	4
(patch,sensor)	(1,1)			
$X_t = \boxed{} + \text{noise}$				

Sampling protocol (example)

$$\begin{bmatrix} \boxed{0} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Time t	1	2	3	4
(patch,sensor)	(1,1)			
$X_t = \boxed{} + \text{noise}$	0.1			

Sampling protocol (example)

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \boxed{0} & 1 & 0 & 0.5 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Time t	1	2	3	4
(patch,sensor)	(1,1)	(2,1)		
$X_t = \boxed{} + \text{noise}$	0.1			

Sampling protocol (example)

$$\begin{array}{cc} & \downarrow \\ & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ \boxed{0} & 1 & 0 & 0.5 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{array}{c} \\ \leftarrow \\ \\ \\ \\ \end{array} \end{array}$$

Time t	1	2	3	4
(patch,sensor)	(1,1)	(2,1)		
$X_t = \boxed{} + \text{noise}$	0.1	-0.05		

Sampling protocol (example)

$$\begin{bmatrix} 0 & 0 & 0 & \boxed{0} & 0 & 0 \\ 0 & 1 & 0 & 0.5 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Time t	1	2	3	4
(patch,sensor)	(1,1)	(2,1)	(1,4)	
$X_t = \boxed{} + \text{noise}$	0.1	-0.05		

Sampling protocol (example)

$$\begin{bmatrix} 0 & 0 & 0 & \boxed{0} & 0 & 0 \\ 0 & 1 & 0 & 0.5 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Time t	1	2	3	4
(patch,sensor)	(1,1)	(2,1)	(1,4)	
$X_t = \boxed{} + \text{noise}$	0.1	-0.05	0.1	

Sampling protocol (example)

$$
\begin{array}{c}
\begin{array}{cccccc} & & & \downarrow & & \end{array} \\
\begin{array}{cc}
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0.5 & 0.1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & \boxed{0.5} & 0.1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
&
\begin{array}{c} \\ \\ \\ \leftarrow \\ \\ \end{array}
\end{array}
\end{array}
$$

Time t	1	2	3	4
(patch,sensor)	(1,1)	(2,1)	(1,4)	(4,4)
$X_t = \boxed{} + \text{noise}$	0.1	-0.05	0.1	

Sampling protocol (example)

$$
\begin{array}{cc}
\begin{array}{cccccc} & & & \downarrow & & \end{array} & \\
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0.5 & 0.1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & \boxed{0.5} & 0.1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} & \begin{array}{c} \\ \\ \\ \leftarrow \\ \\ \end{array}
\end{array}
$$

Time t	1	2	3	4
(patch,sensor)	(1,1)	(2,1)	(1,4)	(4,4)
$X_t = \boxed{} + \text{noise}$	0.1	-0.05	0.1	0.4

Sampling protocol (example)

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.5 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Time t	1	2	3	4	...
(patch,sensor)	(1,1)	(2,1)	(1,4)	(4,4)	
$X_t = \boxed{} +$ noise	0.1	-0.05	0.1	0.4	...

$\rightarrow$ At time T, output a partition of the patches

PAC-setting

Learning protocol

Input: prescribed probability δ

While $t \leqslant T$, (T *a stopping time*)

- choose a patch $I_t \in 1, \ldots, n$
- choose a sensor $J_t \in 1, \ldots, d$
- receive X_t, s.t., $X_t = M_{I_t, J_t}$+noise[a]

Output: $\hat{g}$ partition of the patches

a. $\epsilon_t := (X_t - M_{I_t, J_t})$ is 1-sub-Gaussian

PAC-setting

Learning protocol

Input: prescribed probability δ

While $t \leqslant T$, (T *a stopping time*)

- choose a patch $I_t \in 1, \ldots, n$
- choose a sensor $J_t \in 1, \ldots, d$
- receive X_t, s.t., $X_t = M_{I_t, J_t}$+noise[a]

Output: $\hat{g}$ partition of the patches

a. $\epsilon_t := (X_t - M_{I_t, J_t})$ is 1-sub-Gaussian

Definition

$\mathcal{A}$ is δ-PAC if $\mathbb{P}_{\mathcal{A},M}(\hat{g} \text{ is correct}) \geqslant 1 - \delta$

PAC-setting

Learning protocol

Input: prescribed probability δ

While $t \leqslant T$, (T *a stopping time*)

- choose a patch $I_t \in 1, \dots, n$
- choose a sensor $J_t \in 1, \dots, d$
- receive X_t, s.t., $X_t = M_{I_t,J_t}$+noise[a]

Output: $\hat{g}$ partition of the patches

a. $\epsilon_t := (X_t - M_{I_t,J_t})$ is 1-sub-Gaussian

Definition

$\mathcal{A}$ is δ-PAC if $\mathbb{P}_{\mathcal{A},M}(\hat{g} \text{ is correct}) \geqslant 1 - \delta$

Main objective

find $\mathcal{A}$, δ-PAC algorithm with a budget T as small as possible

Connection to pure exploration litterature

- The problem[2] consists in balancing the exploration
$$\begin{cases} \textbf{over the patches} & \rightarrow \quad \text{to classify} \\ \textbf{over the sensors} & \rightarrow \quad \text{to learn the structure of } \Delta \end{cases}$$
- We combine ideas from (active) signal detection[3][4], and good-arm-identification[5].

---

2. K. Ariu et al. **"Optimal clustering from noisy binary feedback".** In: *Machine Learning* (2024).
3. R. M Castro. **"Adaptive sensing performance lower bounds for sparse signal detection and support estimation".** In: *Bernoulli* (2014).
4. M. Saad, N. Verzelen, and A. Carpentier. **"Active ranking of experts based on their performances in many tasks".** In: *ICML*. PMLR. 2023.
5. Y. Zhao et al. **"Revisiting simple regret: Fast rates for returning a good arm".** In: *ICML*. PMLR. 2023.

Contribution

Our Contribution

1. Introduction of a δ-PAC Algorithm `BanditClustering`, which
   a) identifies the most discriminative sensor
   b) classify patches based on this sensor

Our Contribution

1. Introduction of a δ-PAC Algorithm `BanditClustering`, which
   a) identifies the most discriminative sensor
   b) classify patches based on this sensor
2. Answers to the questions:
   - What is the sampling budget required ?
     (upper bound on the budget of `BanditClustering`)
   - Is our approach optimal?
     (matching lower in most relevent cases)

Our Contribution

1. Introduction of a δ-PAC Algorithm `BanditClustering`, which
   a) identifies the most discriminative sensor
   b) classify patches based on this sensor
2. Answers to the questions:
   - What is the sampling budget required ?
     (upper bound on the budget of `BanditClustering`)
   - Is our approach optimal?
     (matching lower in most relevent cases)
3. Illustrative numerical experiments

Algorithm: BanditClustering

1. Identification of representatives patches r_0, r_1
   ($\sim$ signal detection)

$$\Delta = M_{r_1,\cdot} - M_{r_0,\cdot}$$

2. Selection of j a good discriminative sensor
   ($\sim$ good-arm identification)

$$\Delta_j = M_{r_1,j} - M_{r_0,j}$$

3. Classification of patches based on the sensor
   ($\sim$ binary classification)

First step : representative identification

- **Objective**: For r_0 fixed, find r_1 s.t. $M_{r_1,\cdot} \neq M_{r_0,\cdot}$
  (w.l.o.g, $M_{r_0,\cdot} = 0$, and $M_{r_1,\cdot} = \Delta$)

First step : representative identification

- **Objective**: For r_0 fixed, find r_1 s.t. $M_{r_1,\cdot} \neq M_{r_0,\cdot}$
  (w.l.o.g, $M_{r_0,\cdot} = 0$, and $M_{r_1,\cdot} = \Delta$)
- **Method**:
  1. subsampling: select randomly L entries of M
  2. Sequential halving: identify entry (i,j) with $M_{i,j}$ large (with SH and budget N)
  3. stopping condition: if $M_{i,j} \neq 0$ w.h.p

     output the patch of this entry

     Else increase (L, N) and repeat $1, 2, 3$

First step : representative identification

- **Objective**: For r_0 fixed, find r_1 s.t. $M_{r_1,\cdot} \neq M_{r_0,\cdot}$
  (w.l.o.g, $M_{r_0,\cdot} = 0$, and $M_{r_1,\cdot} = \Delta$)
- **Method**:
  1. subsampling: select randomly L entries of M
  2. Sequential halving: identify entry (i,j) with $M_{i,j}$ large (with SH and budget N)
  3. stopping condition: if $M_{i,j} \neq 0$ w.h.p

     output the patch of this entry

     Else increase (L, N) and repeat $1, 2, 3$
- **Budget**:

$$\frac{d}{\theta \|\Delta\|_2^2} \log(1/\delta) \quad \text{(up to log terms)}$$

with θ proportion of patches in the smallest group

Second step : sensor selection

- **Objective**: We know r_1 s.t. $\Delta = M_{r_1,\cdot} \neq M_{r_0,\cdot} = 0$, we want j (sensor) s.t. Δ_j is large enough

Second step : sensor selection

- **Objective**: We know r_1 s.t. $\Delta = M_{r_1,\cdot} \neq M_{r_0,\cdot} = 0$, we want j (sensor) s.t. Δ_j is large enough
- **Method**:
  1. subsampling: select randomly L sensors among $\{1, , d\}$
  2. Sequential halving: identify entry j with Δ_j large (with SH and budget N)
  3. Estimation: estimate Δ_j
  4. stopping condition: if $\frac{n}{\hat{\Delta}_j^2} \log(n/\delta) \lesssim N$ w.h.p

classification

Else increase (L, N) and repeat $1, 2, 3$

Second step (and third step): sensor selection

- **Objective**: We know r_1 s.t. $\Delta = M_{r_1,\cdot} \neq M_{r_0,\cdot} = 0$, we want j (sensor) s.t. Δ_j is large enough
- **Method**:
  1. subsampling: select randomly L sensors among $\{1, , d\}$
  2. Sequential halving: identify entry j with Δ_j large (with SH and budget N)
  3. Estimation: estimate Δ_j
  4. stopping condition: if $\frac{n}{\hat{\Delta}_j^2} \log(n/\delta) \lesssim N$ w.h.p

     Sample each patch i with sensor j, N/n times

     Use a linear classifier

     Else increase (L, N) and repeat $1, 2, 3$

Thereom (Simpler case)

recall: matrix $(M_{i,j})_{\substack{i=1,\ldots,n\\ j\in 1,\ldots,d}}$ with rows $M_{i,\cdot} \in \{0, \Delta\}$

Thereom (Simpler case)

recall: matrix $(M_{i,j})_{\substack{i=1,\ldots,n \\ j\in 1,\ldots,d}}$ with rows $M_{i,\cdot} \in \{0, \Delta\}$

sparse setting: Δ equals $h > 0$ in s entries and 0 elsewhere

Thereom (Simpler case)

recall: matrix $(M_{i,j})_{\substack{i=1,\dots,n \\ j\in 1,\dots,d}}$ with rows $M_{i,\cdot} \in \{0, \Delta\}$

sparse setting: Δ equals $h > 0$ in s entries and 0 elsewhere

Theorem

With probability $1-\delta$ `BanditClustering` returns the good partition after $T \lesssim \log(1/\delta)\cdot H$, where

$$H := \frac{1}{\theta}\frac{d}{\|\Delta\|^2} + \frac{n}{h^2}\ .$$

Theorem (general case)

Theorem

Define

$$H := \frac{d}{\theta}\left(\frac{1}{\|\Delta\|^2}\right) + \min_{s\in[d]}\left(\frac{d}{s} + n\right)\left(\frac{1}{\Delta_{(s)}^2}\right) \quad ,$$

with $|\Delta_{(1)}| \geqslant |\Delta_{(2)}| \geqslant \dots$

With probability $1-\delta$, BanditClustering returns the good partition after

$$T \lesssim \log\left(\frac{1}{\delta}\right) \cdot H$$

Lower bound

Theorem

If $\mathcal{A}$ is δ-PAC,then there exists a permutation of M, M_{per}, s.t.,

$$\mathbb{P}_{M_{per},\mathcal{A}}\left(T \geqslant \frac{2d}{\theta\|\Delta\|^2}\log(1/6\delta) \vee \frac{2(n-2)}{\|\Delta\|_\infty^2}\log(1/4.8\delta)\right) \geqslant \delta \ .$$

Lower bound

Theorem

If $\mathcal{A}$ is δ-PAC,then there exists a permutation of M, M_{per}, s.t.,

$$\mathbb{P}_{M_{per},\mathcal{A}}\left(T \geqslant \frac{2d}{\theta\|\Delta\|^2}\log(1/6\delta) \vee \frac{2(n-2)}{\|\Delta\|_\infty^2}\log(1/4.8\delta)\right) \geqslant \delta \ .$$

Assume that $\Delta = (\underbrace{h, \ldots, h}_{s}, 0, \ldots, 0)$

Lower bound

Theorem

If $\mathcal{A}$ is δ-PAC,then there exists a permutation of M, M_{per}, s.t.,

$$\mathbb{P}_{M_{per},\mathcal{A}}\left(T \geqslant \frac{2d}{\theta\|\Delta\|^2}\log(1/6\delta) \vee \frac{2(n-2)}{\|\Delta\|_\infty^2}\log(1/4.8\delta)\right) \geqslant \delta \ .$$

Assume that $\Delta = (\underbrace{h, \dots, h}_{s}, 0, \dots, 0)$

1. Finding a patch in the smallest group $\quad \rightarrow$ explore $\frac{1}{\theta}\log(1/\delta)$

Lower bound

Theorem

If $\mathcal{A}$ is δ-PAC,then there exists a permutation of M, M_{per}, s.t.,

$$\mathbb{P}_{M_{per},\mathcal{A}}\left(T \geqslant \frac{2d}{\theta\|\Delta\|^2}\log(1/6\delta) \vee \frac{2(n-2)}{\|\Delta\|_\infty^2}\log(1/4.8\delta)\right) \geqslant \delta \ .$$

Assume that $\Delta = (\underbrace{h, \ldots, h}_{s}, 0, \ldots, 0)$

1. Finding a patch in the smallest group $\rightarrow$ explore $\frac{1}{\theta}\log(1/\delta)$
2. Detecting a sensor j with $\Delta_j = h$ $\rightarrow$ explore $\frac{d}{s}\log(1/\delta)$

Lower bound

Theorem

If $\mathcal{A}$ is δ-PAC,then there exists a permutation of M, M_{per}, s.t.,

$$\mathbb{P}_{M_{per},\mathcal{A}}\left(T \geqslant \frac{2d}{\theta\|\Delta\|^2}\log(1/6\delta) \vee \frac{2(n-2)}{\|\Delta\|_\infty^2}\log(1/4.8\delta)\right) \geqslant \delta \ .$$

Assume that $\Delta = (\underbrace{h,\dots,h}_{s},0,\dots,0)$

1. Finding a patch in the smallest group $\rightarrow$ explore $\frac{1}{\theta}\log(1/\delta)$
2. Detecting a sensor j with $\Delta_j = h$ $\rightarrow$ explore $\frac{d}{s}\log(1/\delta)$
3. Testing $\Delta_j = 0$ VS $\Delta_j = h$ $\rightarrow$ sample $\frac{1}{h^2}\log(1/\delta)$

Lower bound

Theorem

If $\mathcal{A}$ is δ-PAC,then there exists a permutation of M, M_{per}, s.t.,

$$\mathbb{P}_{M_{per},\mathcal{A}}\left(T \geqslant \frac{2d}{\theta\|\Delta\|^2}\log(1/6\delta) \vee \frac{2(n-2)}{\|\Delta\|_\infty^2}\log(1/4.8\delta)\right) \geqslant \delta \ .$$

Lower bound

Theorem

If $\mathcal{A}$ is δ-PAC,then there exists a permutation of M, M_{per}, s.t.,

$$\mathbb{P}_{M_{per},\mathcal{A}}\left(T \geqslant \frac{2d}{\theta\|\Delta\|^2}\log(1/6\delta) \vee \frac{2(n-2)}{\|\Delta\|_\infty^2}\log(1/4.8\delta)\right) \geqslant \delta \ .$$

- Let j such that Δ_j is maximal, for each item, we need at least $\frac{1}{\Delta_j^2}\log(1/\delta)$ samples from to classify it

Lower bound

Theorem

If $\mathcal{A}$ is δ-PAC,then there exists a permutation of M, M_{per}, s.t.,

$$\mathbb{P}_{M_{per},\mathcal{A}}\left(T \geqslant \frac{2d}{\theta\|\Delta\|^2}\log(1/6\delta) \vee \frac{2(n-2)}{\|\Delta\|_\infty^2}\log(1/4.8\delta)\right) \geqslant \delta \ .$$

- Let j such that Δ_j is maximal, for each item, we need at least $\frac{1}{\Delta_j^2}\log(1/\delta)$ samples from to classify it
- **Remark.** Still a gap with $\min_{s\in[d]}\left(\frac{d}{s}+n\right)\frac{1}{\Delta_{(s)}^2}$, however, optimal if Δ takes two values.

Numerical experiments

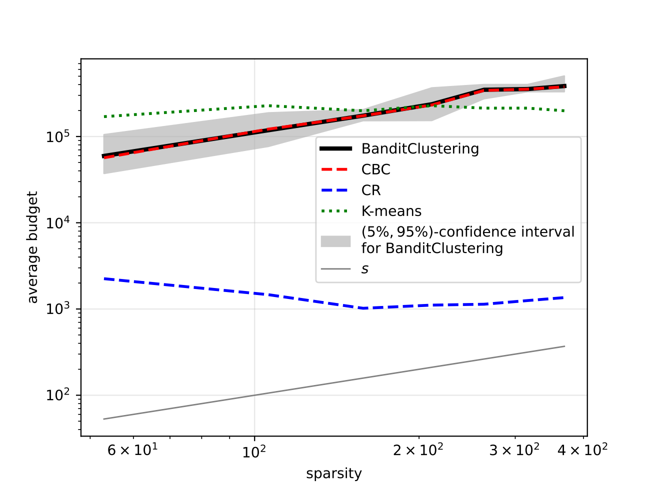

Figure 2: Budgets for increasing sparsity s with fixed $\|\Delta\|$

$n = 20, d = 1000$, $\Delta^s = (\underbrace{h_s, \ldots, h_s}, 0, \ldots, 0)$, and $h_s = \frac{15}{\sqrt{s}}$

Take home message

1. **Algorithmic solution**: new online clustering alg. of budget
$\frac{d}{\theta}\left(\frac{1}{\|\Delta\|^2}\right)\log\left(\frac{1}{\delta}\right) + \min_{s\in[d]}\left(\frac{d}{s}+n\right)\left(\frac{1}{\Delta_{(s)}^2}\right)\log\left(\frac{1}{\delta}\right)$

Take home message

1. **Algorithmic solution**: new online clustering alg. of budget
   $\frac{d}{\theta}\left(\frac{1}{\|\Delta\|^2}\right)\log\left(\frac{1}{\delta}\right) + \min_{s\in[d]}\left(\frac{d}{s}+n\right)\left(\frac{1}{\Delta_{(s)}^2}\right)\log\left(\frac{1}{\delta}\right)$
2. **Optimality**: matching lower bound in the sparse setting

Take home message

1. **Algorithmic solution**: new online clustering alg. of budget
$$\frac{d}{\theta}\left(\frac{1}{\|\Delta\|^2}\right)\log\left(\frac{1}{\delta}\right) + \min_{s\in[d]}\left(\frac{d}{s}+n\right)\left(\frac{1}{\Delta_{(s)}^2}\right)\log\left(\frac{1}{\delta}\right)$$
2. **Optimality**: matching lower bound in the sparse setting
3. **Perspectives**: refine the lower bounds, generalize to $K > 2$

Take home message

1. **Algorithmic solution**: new online clustering alg. of budget
   $\frac{d}{\theta}\left(\frac{1}{\|\Delta\|^2}\right)\log\left(\frac{1}{\delta}\right) + \min_{s\in[d]}\left(\frac{d}{s}+n\right)\left(\frac{1}{\Delta_{(s)}^2}\right)\log\left(\frac{1}{\delta}\right)$
2. **Optimality**: matching lower bound in the sparse setting
3. **Perspectives**: refine the lower bounds, generalize to $K > 2$
4. **Reference:**
   M. Graf, V. Thuot, and N. Verzelen. ***Clustering Items through Bandit Feedback: Finding the Right Feature out of Many.*** Accepted at the 42nd International Conference on Machine Learning. 2025. arXiv: 2503.11209

Thank you

Sequential Halving

Input: Number of arms M, budget T
Set $S \leftarrow \{1, 2, \ldots, M\}$ ▷ Set of active arms
Set $R \leftarrow \lfloor \log_2 M \rfloor$ ▷ Number of rounds
for $r = 1$ to R **do**
    $n_r \leftarrow \left\lfloor \frac{T}{|S| \cdot R} \right\rfloor$ ▷ $n_{r+1} = 2 \cdot n_r$
    for each arm a in S **do**
        Sample arm a for n_r times, compute empirical mean $\hat{\mu}_a$
    end for
    Keep top $\left\lfloor \frac{|S|}{2} \right\rfloor$ arms in S
end for
Output: Return the remaining arm in S