Clustering Experts with Bandit Feedback of their Performance in Multiple Tasks

Maximilian Graf[1] Victor Thuot[2] Nicolas Verzelen[2]

[1]Institut für Mathematik, Universität Potsdam, Potsdam, Germany

[2]INRAE, Mistea, Institut Agro, Univ Montpellier, Montpellier, France

March 11th, GPSD 2025, Dresden

1 Setting: Clustering with Bandit Feedback

2 Contributions

Model: feature matrix

- Consider n items

Model: feature matrix

- Consider n items
- Consider d features

Model: feature matrix

- Consider n items
- Consider d features
- Each item i is characterized by a d-dimensional feature vector $\mu_i = [\mu_{i,1}, \ldots, \mu_{i,d}] \in \mathbb{R}^d$

Model: feature matrix

- Consider n items
- Consider d features
- Each item i is characterized by a d-dimensional feature vector $\mu_i = [\mu_{i,1}, \ldots, \mu_{i,d}] \in \mathbb{R}^d$

$$M = \begin{bmatrix} \mu_{1,1} & \cdots & \mu_{1,j} & \cdots & \mu_{1,d} \\ \vdots & & \vdots & & \vdots \\ \mu_{i,1} & \cdots & \mu_{i,j} & \cdots & \mu_{i,d} \\ \vdots & & \vdots & & \vdots \\ \mu_{n,1} & \cdots & \mu_{n,j} & \cdots & \mu_{n,d} \end{bmatrix} \begin{array}{l} \\ \\ \leftarrow \mu_i \\ \\ \\ \end{array}$$

Model: feature matrix

- Consider n items (experts)
- Consider d features (tasks)
- Each item i is characterized by a d-dimensional feature vector $\mu_i = [\mu_{i,1}, \ldots, \mu_{i,d}] \in \mathbb{R}^d$ (performance)

$$M = \begin{array}{cc} \begin{bmatrix} \mu_{1,1} & \cdots & \mu_{1,j} & \cdots & \mu_{1,d} \\ \vdots & & \vdots & & \vdots \\ \mu_{i,1} & \cdots & \mu_{i,j} & \cdots & \mu_{i,d} \\ \vdots & & \vdots & & \vdots \\ \mu_{n,1} & \cdots & \mu_{n,j} & \cdots & \mu_{n,d} \end{bmatrix} & \begin{array}{c} \\ \\ \leftarrow \mu_i \\ \\ \\ \end{array} \end{array}$$

Model: hidden partition

Assumption 1: hidden partition

We assume that there exists two different vector $\mu^a \in \mathbb{R}^d$, and $\mu^b \in \mathbb{R}^d$, such that, for any item $i \in \{1, \ldots, n\}$, $\mu_i \in \{\mu^a, \mu^b\}$.

$$M = \begin{bmatrix} \mu_{1,1} & \cdots & \mu_{1,j} & \cdots & \mu_{1,d} \\ \vdots & & \vdots & & \vdots \\ \mu_{i,1} & \cdots & \mu_{i,j} & \cdots & \mu_{i,d} \\ \vdots & & \vdots & & \vdots \\ \mu_{n,1} & \cdots & \mu_{n,j} & \cdots & \mu_{n,d} \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_i \\ \vdots \\ \mu_d \end{bmatrix} \begin{array}{l} \leftarrow \mu_1 = \mu^a \\ \vdots \\ \leftarrow \mu_i \in \mu^a, \mu^b \\ \\ \\ \end{array}$$

Model: hidden partition

Assumption 1: hidden partition

We assume that there exists two different vector $\mu^a \in \mathbb{R}^d$, and $\mu^b \in \mathbb{R}^d$, such that, for any item $i \in \{1, \ldots, n\}$, $\mu_i \in \{\mu^a, \mu^b\}$.

- The items are partitioned into two unknown, non empty and non overlapping groups.
- Let $g \in \{0,1\}^n$ be the label vector such that $g(1) = 0$, and

$$M_{i,j} = \begin{cases} \mu^a_j & \text{if } g(i) = 0 \ , \\ \mu^b_j & \text{if } g(i) = 1 \ . \end{cases}$$

Model: hidden partition

Assumption 1: hidden partition

We assume that there exists two different vector $\mu^a \in \mathbb{R}^d$, and $\mu^b \in \mathbb{R}^d$, such that, for any item $i \in \{1, \dots, n\}$, $\mu_i \in \{\mu^a, \mu^b\}$.

- The items are partitioned into two unknown, non empty and non overlapping groups.
- Let $g \in \{0, 1\}^n$ be the label vector such that $g(1) = 0$, and

$$M_{i,j} = \begin{cases} \mu_j^a & \text{if } g(i) = 0 \;, \\ \mu_j^b & \text{if } g(i) = 1 \;. \end{cases}$$

- The objective is to recover perfectly g

Model: bandit feedback

An algorithm collects sequentially and adaptively feedback - which consists on noisy observations of entries of matrix M.

Model: bandit feedback

An algorithm collects sequentially and adaptively feedback - which consists on noisy observations of entries of matrix M.

Learning protocol

At each time step t,

- choose an item $I_t \in 1, \ldots, n$ (*based on the past*)
- choose a feature $J_t \in 1, \ldots, d$ (*based on the past*)
- receive X_t, s.t., $X_t \sim \nu_{I_t, J_t}$ (*conditionally on* (I_t, J_t))

Model: bandit feedback

An algorithm collects sequentially and adaptively feedback - which consists on noisy observations of entries of matrix M.

Learning protocol

At each time step t,

- choose an item $I_t \in 1, \dots, n$ (*based on the past*)
- choose a feature $J_t \in 1, \dots, d$ (*based on the past*)
- receive X_t, s.t., $X_t \sim \nu_{I_t, J_t}$ (*conditionally on* (I_t, J_t))

- For each couple item/feature, $(i, j) \in [n] \times [d] \leftrightarrow$ probability distribution $\nu_{i,j}$ with mean $\mu_{i,j}$.

Model: bandit feedback

An algorithm collects sequentially and adaptively feedback - which consists on noisy observations of entries of matrix M.

Learning protocol

At each time step t,

- choose an item $I_t \in 1, \dots, n$ (*based on the past*)
- choose a feature $J_t \in 1, \dots, d$ (*based on the past*)
- receive X_t, s.t., $X_t \sim \nu_{I_t, J_t}$ (*conditionally on* (I_t, J_t))

- For each couple item/feature, $(i, j) \in [n] \times [d] \leftrightarrow$ probability distribution $\nu_{i,j}$ with mean $\mu_{i,j}$.

Assumption 2: sub-Gaussian noise

If $X \sim \nu_{i,j}$, then $\mathbb{E}[X] = \mu_{i,j}$, and $(X - \mu_{i,j})$ is 1-sub-Gaussian.

Application: image classification within crowd-sourcing platform [Ariu et al., 2024]

- each item ↔ one image (e.g., cats and dogs)
- each feature ↔ one binary question (e.g., "Does the animal has a long fair?")
- $\mu_{i,j}$ ↔ probability of answering yes to question j on image i

Figure: Image I_t, Question J_t "Does the animal has a long fair?"

PAC-setting

Learning protocol

Input: prescribed probability δ

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

- choose an item I_t and a feature J_t (*based on the past*)
- receive X_t, s.t., $X_t \sim \nu_{I_t,J_t}$ (*conditionally on* (I_t, J_t))

Output: $\hat{g}$ estimation of g

PAC-setting

Learning protocol

Input: prescribed probability δ

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

- choose an item I_t and a feature J_t (*based on the past*)
- receive X_t, s.t., $X_t \sim \nu_{I_t,J_t}$ (*conditionally on* (I_t, J_t))

Output: $\hat{g}$ estimation of g *with probability of error* $\leqslant \delta$

Objective (δ-PAC setting)

An algorithm $\mathcal{A}$ is called δ-PAC if $\mathbb{P}_{\mathcal{A},M}(\hat{g} = g) \geqslant 1 - \delta$.

PAC-setting

Learning protocol

Input: prescribed probability δ

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

- choose an item I_t and a feature J_t (*based on the past*)
- receive X_t, s.t., $X_t \sim \nu_{I_t,J_t}$ (*conditionally on* (I_t, J_t))

Output: $\hat{g}$ estimation of g *with probability of error* $\leqslant \delta$

Objective (δ-PAC setting)

An algorithm $\mathcal{A}$ is called δ-PAC if $\mathbb{P}_{\mathcal{A},M}(\hat{g} = g) \geqslant 1 - \delta$.

- T is called the budget
- Our objective is to construct a δ-PAC algorithm with a budget as small as possible.

Pure exploration

- The difficulty of the task is driven by two parameters:

$$\begin{array}{lcll} \Delta & = & \mu^a - \mu^b \in \mathbb{R}^d & \text{gap vector} \\ \theta & = & \frac{1}{n}\min\left(\sum_{i=1}^n \mathbb{1}_{g(i)=0}, \sum_{i=1}^n \mathbb{1}_{g(i)=1}\right) & \text{balancedness} \end{array}$$

- The problems consists in balancing the exploration

$$\begin{cases} \textbf{over the items} & \rightarrow \quad \text{to classify the items} \\ \textbf{over the features} & \rightarrow \quad \text{to learn the structure of } \Delta \end{cases}$$

- We combine ideas from (active) signal detection [Castro, 2014, Saad et al., 2023], and good-arm-identification [Zhao et al., 2023].

Pure exploration

- The difficulty of the task is driven by two parameters:

$$\Delta = \mu^a - \mu^b \in \mathbb{R}^d \quad \text{gap vector}$$
$$\theta = \frac{1}{n}\min\left(\sum_{i=1}^n \mathbb{1}_{g(i)=0}, \sum_{i=1}^n \mathbb{1}_{g(i)=1}\right) \quad \text{balancedness}$$

-

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0.5 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0.5 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- $n = 5,\ d = 6,\ g = (0, 1, 0, 1, 0),\ \theta = 2/5,$
$\Delta = [0, 1, 1, 0.5, 0.05, 0]$

Main structure of the algorithm: BanditClustering

1. identify an item $\hat{i}$ such that $\mu_{\hat{i}} \neq \mu_1$ ($\sim$ signal detection)
2. using items 1 and $\hat{i}$, use these two items to learn the structure of $\Delta = \mu^a - \mu^b$, and choose a feature j such that $|\Delta_j|$ is large ($\sim$ good-arm identification)
3. classify each item based on samples from feature j (binary classification)

First step : representative identification

The first objective is to detect an item $\hat{i}$ such that $\mu_i =\neq \mu_1$ (recall that $\mu_1 = \mu^a$).

- With Sequential Halving (and sub-sampling), we prove that we can find $\hat{i}$ with a budget $\frac{d}{\theta\|\Delta\|_2^2} \log(1/\delta)$ (up to log terms).

Second step : feature selection

For the second step, we use $\hat{i}$ (and the first item) to find j such that $|\Delta_j|$ is large.

- Apply Sequential halving with a budget T_k on a bandit with d arms with means $\Delta_1, \dots, \Delta_d$
- Define $|\Delta_{(1)}| \geqslant \cdots \geqslant |\Delta_{(d)}|$
- With a budget $T_k \sim \frac{d}{s}\frac{1}{\Delta^2_{(s)}}\log(1/\delta)$, the output of SH j is such that $\Delta_j \geqslant \Delta_{(s)}/2$*
- Estimate $\hat{\Delta}_j^2 \leqslant \Delta_j^2$
- If $T_k \geqslant C\frac{n}{\hat{\Delta}_j^2}\log(n/\delta)$, we stop and chose j, else, we apply SH again, with a double budget.

Final step: classification

One we know $(\hat{i}, j)$ such that $|M_{i,j} - M_{1,j}|$ is large, together with a lower bound $\hat{\Delta}_j \leqslant |\Delta_{(j)}|$ (w.h.p), then:

- Sample each entry (i, j) $\frac{C}{\hat{\Delta}_j^2} \log(n/\delta)$ for $i = 1, \ldots, n$
- classify each arm

Main theorem

Theorem

For $\delta \in (0, 1/e)$. *Assume that* $\Delta \in [-1, 1]^d$. *Define*

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

With a probability of at least $1 - \delta$, *BanditClustering returns* $\hat{g} = g$ *with a budget of at most*

$$T \leqslant \tilde{C} \cdot \log\left(\frac{1}{\delta}\right) \cdot H,$$

where $\tilde{C}$ *is a logarithmic factor in* d, n, Δ *and poly-logarithmic in* δ.

Lower bound

Theorem 2

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 2

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 $\mu^a = (0,\dots,0)$ and $\mu^b = (\underbrace{\mu,\dots,\mu}_{s},0,\dots,0)$

Lower bound

Theorem 2

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 $\mu^a = (0, \dots, 0)$ and $\mu^b = (\underbrace{\mu, \dots, \mu}_{s}, 0, \dots, 0)$

1. To detect an item i with $g(i) = 1 \rightarrow$ explore $\frac{1}{\theta}\log(1/\delta)$ items.

Lower bound

Theorem 2

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 $\mu^a = (0,\ldots,0)$ and $\mu^b = (\underbrace{\mu,\ldots,\mu}_{s},0,\ldots,0)$

1. To detect an item i with $g(i) = 1 \to$ explore $\frac{1}{\theta}\log(1/\delta)$ items.
2. To detect a feature j with $\mu_j^a = \mu, \to$ explore $\frac{d}{s}\log(1/\delta)$ feat..

Lower bound

Theorem 2

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 $\mu^a = (0, \dots, 0)$ and $\mu^b = (\underbrace{\mu, \dots, \mu}_{s}, 0, \dots, 0)$

1. To detect an item i with $g(i) = 1 \to$ explore $\frac{1}{\theta}\log(1/\delta)$ items.
2. To detect a feature j with $\mu_j^a = \mu$, $\to$ explore $\frac{d}{s}\log(1/\delta)$ feat..
3. To test if an entry is equal to μ VS 0, $\to$ sample it $\frac{1}{\mu^2}\log(1/\delta)$.

Second lower bound

Theorem 2

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

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

Second lower bound

Theorem 2

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

Second lower bound

Theorem 2

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

Take home message

1. We introduce a pure exploration problem in a matrix whose row have to be clustered with bandit feedback.
2. We provide an algorithm with a budget $H \log(1/\delta)$ with

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

3. We provide a lower bound matching for a vector gap taking two values.
4. Ongoing work and perspectives:
   - close the gap
   - generalize to $K > 2$ groups

Take home message

1. We introduce a pure exploration problem in a matrix whose row have to be clustered with bandit feedback.
2. We provide an algorithm with a budget $H \log(1/\delta)$ with

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

3. We provide a lower bound matching for a vector gap taking two values.
4. Ongoing work and perspectives:
   - close the gap
   - generalize to $K > 2$ groups

Thank you for your attention !

Ariu, K., Ok, J., Proutiere, A., and Yun, S. (2024).
Optimal clustering from noisy binary feedback.
Machine Learning, 113(5):2733–2764.

Castro, R. M. (2014).
Adaptive sensing performance lower bounds for sparse signal detection and support estimation.
Bernoulli, 20(4):2217–2246.

Saad, E. M., Verzelen, N., and Carpentier, A. (2023).
Active ranking of experts based on their performances in many tasks.
In *International Conference on Machine Learning*, pages 29490–29513. PMLR.

Zhao, Y., Stephens, C., Szepesvári, C., and Jun, K.-S. (2023).
Revisiting simple regret: Fast rates for returning a good arm.
In *International Conference on Machine Learning*, pages 42110–42158. PMLR.