A Clustering Expert with Bandit Feedback of their Performance in Many Tasks

Maximilian Graf and Victor Thuot

Group seminar: Mathematical Statistics.

Potsdam, 22nd November 2024

University of Potsdam

OAL

Model : Performance Matrix

Consider n "experts"

Consider d "tasks"

The performance of experts $i \in \{1, \dots, n\}$ on the task $j \in \{1, \dots, d\}$ is given by $\mu_{i,j} \in \mathbb{R}$.

Performance Matrix : $M =$

$$M = \begin{array}{cc} \begin{array}{ccccc} \phantom{\mu_{1,1}} & & \text{task } j & & \phantom{\mu_{1,d}} \\ & & \downarrow & & \end{array} & \\ \begin{pmatrix} \mu_{1,1} & \cdots & \cdot & \cdots & \mu_{1,d} \\ \vdots & & \vdots & & \vdots \\ \mu_{i,1} & \cdots & \mu_{i,j} & \cdots & \mu_{i,d} \\ \vdots & & \vdots & & \vdots \\ \mu_{n,1} & \cdots & \cdots & \cdots & \mu_{n,d} \end{pmatrix} & \leftarrow \text{expert } i \end{array}$$

OK

Model : Performance Matrix

Consider n "experts"
Consider d "tasks"

The performance of experts $i \in \{1, \dots, n\}$ on the task $j \in \{1, \dots, d\}$ is given by $\mu_{i,j} \in \mathbb{R}$.

Performance Matrix : $M =$

$$
\begin{array}{ccc}
 & \text{task } j & \\
 & \downarrow & \\
 & \begin{pmatrix}
\mu_{1,1} & \cdots & \cdots & \mu_{1,d} \\
\vdots & & \vdots & \vdots \\
\mu_{i,1} & \cdots & \mu_{i,j} & \cdots \; \mu_{i,d} \\
\vdots & & \vdots & \vdots \\
\mu_{n,1} & \cdots & \cdots & \mu_{n,d}
\end{pmatrix} & \\
\mu_i \; \{ & \boxed{\mu_{i,1} \; \cdots \; \mu_{i,j} \; \cdots \; \mu_{i,d}} & \leftarrow \text{expert } i
\end{array}
$$

OK

Model : Sequential and adaptive learning protocol

Learning protocol

For each round $t = 1, \dots, T$

- chooses $\begin{cases} \text{an expert } I_t \in \{1, \dots, n\} \\ \text{a task } J_t \in \{1, \dots, d\} \end{cases}$
- receives from the environment a feedback X_t

Rq 1.
- sequential: one data at a time
- adaptive: the choice (I_t, J_t) is based on the passed

Model : Sequential and adaptive learning protocol

Learning protocol

For each round $t \geq 1, \ldots, T$

- chooses $\begin{cases} \text{an expert } I_t \in \{1, \ldots, N\} \\ \text{a task } J_t \in \{1, \ldots, d\} \end{cases}$
- receives from the environment a feedback X_t

$$X_t = \underbrace{\mu_{I_t, J_t}}_{\text{performance}} + \underbrace{\varepsilon_t}_{\text{noise}}$$

Rq 1.
- sequential: one data at a time
- adaptive: the choice (I_t, J_t) is based on the passed

OK

Model : Stationary and subGaussian feedback

Assumption 1 (stationarity)

There exists $n \times d$ distributions on $\mathbb{R}$: $\{\nu_{i,j}\}_{i,j}$

- $$\mathcal{L}(\underbrace{X_t}_{\text{feedback}} \mid \underbrace{I_1, J_1, X_1, \dots, I_{t-1}, J_{t-1}, X_{t-1}}_{\text{passed decisions and observations}}, \underbrace{I_t, J_t}_{\text{decision at time } t}) = \nu_{I_t, J_t}$$

- $$\mathbb{E}_{X \sim \nu_{i,j}}[X] = \mu_{i,j}$$

Assumption 2 (subGaussian noise)

For all (i,j), if $X \sim \nu_{i,j}$
then $X - \mu_{i,j}$ is 1-sub-Gaussian

Model : Hidden partition

Assumption 1 : assume that there exists $\begin{cases} \mu^a \in \mathbb{R}^d \\ \mu^b \in \mathbb{R}^d \end{cases}$

such that, for any expert $i \in \{1, \dots, n\}$, $\mu_i \in \{\mu^a, \mu^b\}$

$$M = \begin{pmatrix} \mu_{1,1} & \cdots & \cdots & \cdots & \mu_{1,d} \\ \vdots & & \vdots & & \vdots \\ \mu_{i,1} & \cdots & \mu_{i,j} & \cdots & \mu_{i,d} \\ \vdots & & \vdots & & \vdots \\ \mu_{n,1} & \cdots & \cdots & \cdots & \mu_{n,d} \end{pmatrix} \begin{matrix} \\ \\ \}\,\mu_i \\ \\ \\ \end{matrix}$$

Model : Hidden partition

<u>Assumption 1</u> : assume that these exists $\begin{cases} \mu^a \in \mathbb{R}^d \\ \mu^b \in \mathbb{R}^d \end{cases}$

such that, for any expert $i \in \{1, \dots, n\}$, $\mu_i \in \{\mu^a, \mu^b\}$

<u>Notation</u>

- $S^a = \{ i \in \{1, \dots, d\} ; \mu_i = \mu^a \}$ [resp S^b]
- $S^a \sqcup S^b = \{1, \dots, n\}$ (hidden partition)

Model : Hidden partition

Assumption 1 : assume that there exists $\begin{cases} \mu^a \in \mathbb{R}^d \\ \mu^b \in \mathbb{R}^d \end{cases}$

such that, for any expert $i \in \{1, \dots, n\}$, $\mu_i \in \{\mu^a, \mu^b\}$

Notation

- $S^a = \{ i \in \{1, \dots, d\} \,;\, \mu_i = \mu^a \}$ [resp S^b]
- $S^a \sqcup S^b = \{1, \dots, n\}$ (hidden partition)
- $\Theta := \dfrac{|S^a| \wedge |S^b|}{n}$ (balancedness)

OK

Model : Sequential and adaptive learning protocol

Learning protocol

- For each round $t = 1, \dots, T$

  chooses $\begin{cases} \text{an expert } I_t \in \{1, \dots, N\} \\ \text{a task } J_t \in \{1, \dots, d\} \end{cases}$

  receives from the environment a feedback X_t

▷ sampling rule : $(I_t, J_t)_{t \geq 1}$

Model : Sequential and adaptive learning protocol

Learning protocol

- For each round $t \geq 1, \dots, T$

  chooses $\begin{cases} \text{an expert } I_t \in \{1, \dots, N\} \\ \text{a task } J_t \in \{1, \dots, d\} \end{cases}$

  receives from the environment a feedback X_t

- At time T, the algorithm stops, and outputs $\hat{S}_T$ estimation of the partition

▷ sampling rule : $(I_t, J_t)_{t \geq 1}$

▷ decision rule : $\hat{S}_T$

OK

Objective : PAC setting

First objective

Given $\delta \in (0,1)$, an algorithm $\mathcal{A}$ is δ-PAC if.

$$\mathbb{P}_{\mathcal{A},\nu}\left(\hat{S} = S^a \text{ or } \hat{S} = S^b\right) \overset{*}{\geq} 1-\delta$$

Objective : PAC setting

First objective

Given $\delta \in (0,1)$, an algorithm $\mathcal{A}$ is δ-PAC if.

$$\underbrace{\mathbb{P}_{\mathcal{A},\nu}}_{\text{proba induced by algorithm } \mathcal{A} \text{ and environment } \nu}\Big(\underbrace{\hat{S} = S^a \text{ or } \hat{S} = S^b}_{\hat{S} \text{ should recover } S^a \sqcup S^b \text{ up to permutation}}\Big) \overset{*}{\geq} 1 - \delta$$

proba induced by algorithm $\mathcal{A}$ and environment ν

$\hat{S}$ should recover $S^a \sqcup S^b$ up to permutation

Objective : PAC setting

First objective

Given $\delta \in (0,1)$, an algorithm $\mathcal{A}$ is δ-PAC if:

$$\mathbb{P}_{\mathcal{A},\nu}\left(\hat{S} = S^a \text{ or } \hat{S} = S^b\right) \geq 1-\delta \quad *$$

Second objective

Construct $\mathcal{A}$ with T as small as possible while maintaining $*$

Rq: T is a stopping time designed

Model : Sequential and adaptive learning protocol

Learning protocol, (δ fixed)

- For each round $t \geq 1, \dots, T$
  chooses $\begin{cases} \text{an expert } I_t \in \{1, \dots, N\} \\ \text{a task } J_t \in \{1, \dots, d\} \end{cases}$
  receives from the environment a feedback X_t

- At time T, the algorithm stops, and outputs $\hat{S}_T$ estimation of the partition

▷ sampling rule : $(I_t, J_t)_{t \geq 1}$

▷ decision rule : $\hat{S}_T$

▷ stopping rule : T stopping time.

Remarques:

ⓐ main objective :

balance { exploration to learn the gap vector Δ (explo with d)
{ exploration to classify the arms (explo with n)

ⓑ links with (adaptive) signal detection __

ⓒ link to the previous paper

Lower bound :

Theorem

For any algorithm $\mathcal{A}$, δ-PAC for the clustering problem, there exists σ, τ two permutations such that, if $M_{\sigma,\tau}$ is constructed with M by permutation (σ, τ), then

$$\mathbb{P}_{\mathcal{A}, M_{\sigma,\tau}}\left(T \geq \frac{2d}{\theta \|\Delta\|_2^2} \log\left(\frac{1}{6\delta}\right)\right) \geq \delta$$

where $\Delta = \mu^a - \mu^b \in \mathbb{R}^d$ is the gap vector

Lower bound : intuition

$$\mathbb{P}_{\mathcal{A}, M_{s;\tau}}\left(T \geq \frac{2d}{\theta \|\Delta\|_2^2} \log\left(\frac{1}{6\delta}\right)\right) \geq \delta$$

- imagine that $\mu^a = (0, \dots, 0)$

  $\mu^b = (\underbrace{\mu, \dots, \mu}_{s}, 0, \dots, 0)$ and $|S^b| = \theta n$

- $\|\Delta\|_2^2 = s\mu^2$

in order to detect one expert in S^b,

* we need to explore $\frac{1}{\theta}$ experts to have with constant probability an expert in S^b

* we need to sample for each expert $\frac{d}{s}$ tasks to find an interesting task

* we need $\frac{1}{\mu^2} \log\left(\frac{1}{\delta}\right)$ samples from an (expert, task)-couple to decide whether the entry of M is 0 or μ

Algorithm : three-steps procedure

Main structure of the algorithm :

Step 1 : identify one expert in each group $\begin{cases} \hat{\imath}_a \in S^a \\ \hat{\imath}_b \in S^b \end{cases}$

Step 2 : collect information on the structure of Δ (gap vector $\mu^a - \mu^b$) and choose a task $\hat{\jmath} \in \{1, \dots, d\}$ such that $|\Delta_{\hat{\jmath}}|$ is large

Step 3 : classify each expert based on its performance on the task $\hat{\jmath}$

Step 1	**Step 2**	**Step 3**
$\downarrow$	$\downarrow$	$\downarrow$
Signal detection	ε - Best arm identification (and variants)	binary classification

Sequential Halving: algorithm

- Sequential Halving is a procedure for BAI which has guarantees for:
  - → BAI
  - → ε - best arm identification
  - → (ε, m) - BAI

- If we consider a k-armed bandit, with arms $\{a_1, \dots, a_k\}$

<u>SH</u>

Entrée : budget T, $S_0 = \{a_1, \dots, a_k\}$ set of arms.

For $u = 0, \dots, \lceil \log_2(k) \rceil$:

- sample $\frac{T}{\lceil \log_2(k) \rceil \# S_u}$ times each arm in S_u
- S_{u+1} = half best arm from S_u

Sequential Halving: guaranties

k arms with means $\{\lambda_1, \dots, \lambda_k\}$ ordered as $\lambda_{(1)}, \dots, \lambda_{(k)}$

ε-BAI

(ε, m)-BAI

Remark : we have to adapt the procedure as:

→ we have positive and negative entries.

→ our objective is signal detection and not BAI.

Step 1 : representatives identification (expert)

Objective : identify two experts $\hat{\pi}_a, \hat{\pi}_b \in \{1, \dots, n\}^2$ such that $\hat{\pi}_a$ and $\hat{\pi}_b$ are in different groups w.h.p. $\left(\geq 1 - \frac{\delta}{2}\right)$

Algo
a) Sequential Halving
b) doubling trick because Δ and Θ are unknown
c) subsampling : SH only on a subset of expert-task couples
d) verification through two sample testing

Garanties
1) algo which identifies $\hat{\pi}_a, \hat{\pi}_b$ which are guaranteed to be representatives whp
2) budget (in $(1-\delta)$-quantile)

$$\approx \frac{\log^2(d)}{\Theta} \frac{d}{\|\Delta\|_2^2} \log\left(\frac{d}{\delta}\right)$$

Step 2: learning the gap vector

Balance $\nearrow$ the budget used to learn the structure of $\Delta = \mu^a - \mu^b$
$\rightarrow$ the budget spent to classify the n arms.

With step 1, we have access to $\begin{cases} \hat{\pi}_a \in S^a & \text{expert with performance } \mu^a \in \mathbb{R}^d \\ \hat{\pi}_b \in S^b & \quad \text{〃} \qquad \mu^b \in \mathbb{R}^d \end{cases}$

We want to learn the structure of $\Delta = \mu^a - \mu^b$
in particular, we have to identify a task $\hat{j} \in \{1, \dots, d\}$
such that $|\Delta_{\hat{j}}|$ is large.

Again we use <u>SH</u> and we are interested in (ε, m)-BAI

Step 3 : classification

- We know $\begin{cases} \hat{r}_a \in S^a & \text{expert with performance } \mu^a \\ \hat{r}_b \in S^b & \text{" } \mu^b \\ \hat{j} & \text{task such that } |\Delta_{\hat{j}}| \text{ large.} \end{cases}$

Ideas $\to$ estimate $\mu^a_{\hat{j}}$ and $\mu^b_{\hat{j}}$

$\to$ sample $\underline{n} \quad \dfrac{1}{|\Delta_{\hat{j}}|^2} \log\left(\dfrac{n}{\delta}\right)$ each expert on task $\hat{j}$

Upper bound

Theorem The three-steps procedure is δ-PAC and with probability at least $1-\delta$, it holds that

$$\underbrace{T \lesssim}_{\substack{\text{up to a polylog} \\ \text{in } \Theta, d, n, \Delta}} \underbrace{\frac{d}{\Theta \|\Delta\|_2^2}}_{\substack{\text{expert representation} \\ \text{identification}}} \log\left(\frac{1}{\delta}\right) + \min_{s \in \{1, \dots, d\}} \left\{ \underbrace{\frac{d}{s \Delta_{(s)}^2} \log\left(\frac{1}{\delta}\right)}_{\text{best task identification}} + \underbrace{\frac{n}{\Delta_{(s)}^2} \log\left(\frac{n}{\delta}\right)}_{\text{expert classification}} \right\}$$

where $|\Delta_{(1)}| \geq |\Delta_{(2)}| \geq \dots \geq |\Delta_{(d)}|$ are the entries of Δ ordered by absolute value.

Lower bound

Theorem

For any δ-PAC algorithm, there exists σ, τ permutations of the rows and columns, such that

$$T \geq \frac{d}{\Theta \|\Delta\|_2^2} \log\left(\frac{1}{\delta}\right) + \min_{s \in \{1, \dots, d\}} \left\{ \frac{d}{s \Delta_{(s)}^2} \log\left(\frac{1}{\delta}\right) + \frac{n}{\Delta_{(s)}^2} \log\left(\frac{n}{\delta}\right) \right\}$$

with $\mathbb{P}_{A, \pi, \sigma} \geq \delta$

where $|\Delta_{(1)}| \geq |\Delta_{(2)}| \geq \dots \geq |\Delta_{(d)}|$ are the entries of Δ ordered by absolute value.

Finding Candidates

- assume Θ was known and $\Delta = \mu^a - \mu^b$ was s-sparse such that every non-zero entry is $h > 0$
- sampling $(i_1, j_1), \dots, (i_\phi, j_\phi) \overset{iid}{\sim} \mathcal{U}([N] \times [d])$ with $\phi \gtrsim \frac{d}{s\Theta} \log(\frac{1}{\delta})$ yields

$$\Delta_{i_k j_k} = h \quad \text{for some } k = 1, \dots, \phi$$

- $\frac{1}{h^2} \log(\frac{1}{\delta})$ samples required to decide $\mu_{ij} = 0$ or $\mu_{ij} = h$ correct w.p. $\geq 1 - \delta$
- finding $(i_k j_k)$ requires $\frac{d}{\Theta \|\Delta\|^2}$ (up to log)

Finding Candidates

- in general, Δ is not s-sparse, but $\exists\, s \in [d]$ such that $\|\Delta\|_2^2 \leq \log(2d)\, s \cdot \Delta_{(s)}^2$

  $\{|\Delta_{(1)}| \geq |\Delta_{(2)}| \geq \dots \geq |\Delta_{(d)}|\}$ Before, used $\|\Delta\|_2^2 = s \cdot h^2$

- still: $\|\Delta\|_2^2$, s and Θ unknown

$\longrightarrow$ adaptivity by doubling trick!

Finding candidates

- again: assume $\|\Delta\|_2^2$, s and Θ were known

  consider $h = \frac{\|\Delta\|_2^2}{s \log(2d)}$

- recall: sampling $(i_1, j_1), \ldots, (i_\phi, j_\phi) \overset{iid}{\sim} \mathcal{U}([N]\times[d])$
  with $\phi \gtrsim \frac{d}{s\Theta} \log(\frac{1}{\delta})$ yields

$$\Delta_{i_k j_k} \geq h \quad \text{for } c \cdot \log(\tfrac{1}{\delta}) \quad k = 1, \ldots, \phi$$

- finding one entry with $|\Delta_{i_k j_k}| > \frac{h}{2}$ would require a total budget

$$\gtrsim \phi \cdot \frac{\log(\phi)}{h^2} \gtrsim \frac{d}{\|\Delta\|_2^2} \log\left(\frac{1}{\delta}\right)^{2}$$

up to polylog in d and N

$\sim$ can we do better?

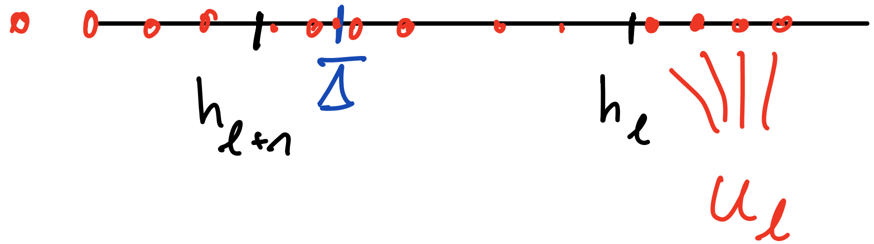
Sequential Halving (intuition)

- recall: $\phi \sim \frac{d}{s\theta} \log(\frac{1}{\delta})$ (i_k, j_k) from which $\Delta_{i_k j_k} \geq h$ for at least $c \cdot \log(\frac{1}{\delta})$ arms
- idea: - sample τ times each arm (i,j)
  - obtain $\hat{\Delta}_{ij}$ as averages and $\bar{\Delta}$ as corresponding median
  - remove all arms with $\hat{\Delta}_{ij} < \bar{\Delta}$
  - $\tau \leftarrow 2 \cdot \tau$, repeat until only one arm left
- algorithm terminates after $L = \lceil \log_2(\phi) \rceil$ steps
- S_ℓ set after ℓ halving steps,

  $U_\ell = \{(i,j) \in S_\ell : |\Delta_{ij}| \geq (1 - \frac{\ell}{2L})h\}$
- We show: $\frac{|U_\ell|}{|S_\ell|}$ is non-decreasing w.h.p. $\Rightarrow S_L \neq \emptyset$!

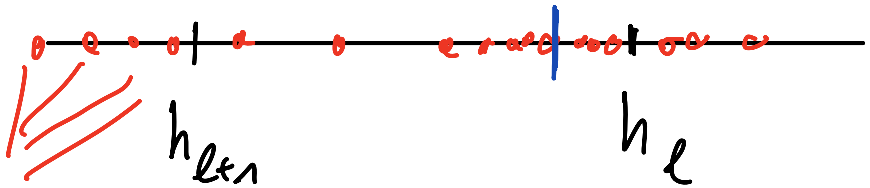
Case 1: $\bar{\Delta} \le \frac{1}{2}(h_{\ell+1} + h_\ell)$

$h_k := \left(1 - \frac{k}{2L}\right) h$



$\to$ at least half of U_ℓ remains in $S_{\ell+1}$

Case 2: $\bar{\Delta} > \frac{1}{2}(h_{\ell+1} + h_\ell)$



a) number of $(i,j) \in S_\ell$ with $|\Delta_{ij}| < h_{\ell+1}$ large

$\to$ at least half of them not in $S_{\ell+1}$

b) if small

$\to$ $U_{\ell+1}$ automatically large enough