Active Ranking of Experts Based on their Performances in Many Tasks

El Mehdi Saad [1] **Nicolas Verzelen** [1] **Alexandra Carpentier** [2]

Abstract

We consider the problem of ranking n experts based on their performances on d tasks. We make a monotonicity assumption stating that for each pair of experts, one outperforms the other on all tasks. We consider the sequential setting where in each round, the learner has access to noisy evaluations of actively chosen pair of expert-task, given the information available up to the actual round. Given a confidence parameter $\delta \in (0, 1)$, we provide strategies allowing to recover the correct ranking of experts and develop a bound on the total number of queries made by our algorithm that hold with probability at least $1 - \delta$. We show that our strategy is adaptive to the complexity of the problem (our bounds are instance dependent), and develop matching lower bounds up to a poly-logarithmic factor. Finally, we adapt our strategy to the relaxed problem of best expert identification and provide numerical simulation consistent with our theoretical results.

1. Introduction

Consider the problem of ranking n experts based on noisy evaluations of their performances in d tasks. This problem arises in many modern applications such as recommender systems (Zhou et al., 2012) and crowdsourcing (Snow et al., 2008; Raykar et al., 2010; Karger et al., 2011; Lu et al., 2015), where the objective is to recommend, for instance, films, music, books, etc based on the product ratings. Sports is another field where ranking plays an important role via the task of player ranking based on their data-driven performances (Morgulev et al., 2018; Pappalardo et al., 2019). In many situations, it is possible to rank experts in an active fashion by sequentially auditing the performance of a chosen expert on a specific task, based on the information collected previously.

In this paper, we consider such a sequential and active ranking setting. For a positive integer N we write $[\![N]\!] = \{1, \dots, N\}$. We consider that the performance of each expert $i \in [\![n]\!]$ on task $j \in [\![d]\!]$ is modeled via random variable following an unknown 1-sub-Gaussain distribution ν_{ij} with mean M_{ij} - and the matrix $M = (M_{i,j})_{i,j}$ encodes the mean performance of each expert on each task. Samples are collected sequentially in an active way: at each round t, the learner chooses a pair expert-task and receives a sample from the corresponding distribution - and we assume that the obtained samples are mutually independent conditional on the chosen distribution.

Our setting is related to the framework of comparison-based ranking considered in the active learning literature (Jamieson & Nowak, 2011; Heckel et al., 2019). This literature stream is tightly connected to the *dueling bandit setting* (Yue et al., 2012; Ailon et al., 2014), where the learner receives binary and noisy feedback on the relative performance of any pair of experts - which is a specific case of our setting when $n = d$. An important decision for setting the ranking problem is then on defining a notion of order between the experts based on the mean performance M - which is ambiguous for a general M. In the active ranking literature, as well as in the dueling bandit literature, it is customary to define a notion of order among any two experts, for instance, through a score as e.g. the *Borda score*, or through another mechanism as e.g. Condorcet winners. E.g. the widely used *Borda scores* corresponds, for each expert i, to the average performance of expert i on all questions - so that this corresponds to ranking experts based on their average performance across all tasks. We discuss this literature in more detail in Section 3.

In this paper, we are inspired by recent advances in the batch literature on this setting - where we receive one sample for each entry expert-task pair. Beyond the historical parametric Bradley-Luce-Terry (BLT) model (Bradley & Terry, 1952), recent papers have proposed batch ranking algorithms in more flexible settings, where no parametric assumption is made, but where a monotonicity assumption, up to some unknown permutation, is made on the matrix M. A minimal

[1]INRAE, Mistea, Institut Agro, Univ Montpellier, Montpellier, France. [2]Institut für Mathematik, Universität Potsdam, Germany.. Correspondence to: El Mehdi Saad <el-mehdi.saad@inrae.fr>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

such assumption was made in (Flammarion et al., 2019) where it is assumed that the matrix M is monotone up to a permutation of its lines, namely of the experts. More precisely, they suppose that there is a complete ordering of the n experts, and that expert i is better than expert k if the first outperforms the latter in all tasks:

$$\text{“}\, i \;\overset{\bullet}{\geq}\; k \,\text{”}: \quad \forall j \in [\![d]\!],\ M_{ij} \geq M_{kj}. \tag{1}$$

Such a shape constraint is arguably restrictive, yet realistic in many applications - e.g. when the tasks are of varying difficulty yet of same nature. There have been recently very interesting advances in the batch literature on this setting, through the construction and analysis of efficient algorithm that fully leverage such a shape constraint. And importantly, these approaches vastly overperform on most problems with a naive strategy that would just rank the experts according to their Borda scores. Beyond the Borda score, this remark is true for any fixed score, as the recent approaches mentioned ultimatively resort in adapting the computed scores to the problem in order to put more emphasis on informative questions.See Section 3 for an overview of this literature.

In this paper, we start from the above remark in the batch setting - namely that in the batch setting and under a monotonicity assumption, ranking according to fixed scores as e.g. Borda is vastly sub-optimal - and we aim at exploring whether this is also the case in the sequential and active setting. We therefore make the monotonicity assumption of Equation (1), and aim at recovering the exact ranking in the active and online setting described above. More precisely, given a confidence parameter $\delta \in (0,1)$, our objective is to rank all the experts using as few queries as possible and consequently adapt to the unknown distributions $(\nu_{ij})_{ij}$.

In this paper, we make the following contributions: First, In Section 4, we consider the problem of comparing two expert ($n = 2$) based on their performances on d tasks under a monotonicity assumption. We provide a sampling strategy and develop distribution dependent upper bound on the total number of queries made by our algorithm to output the correct ranking with probability at least $1-\delta$, $\delta \in (0,1)$ being prescribed confidence. We then consider the problem of ranking estimation for a general n in Section 5, we use the previous algorithm for pairwise comparison as a building block and provide theoretical guarantees on the correctness of our output and a bound on the total number of queries made that holds with high probability. Next, we consider the relaxed objective of identifying the best expert out of n, we provide a sampling strategy for this problem and a bound on the query budget that holds with high probability. In Section 6 we give some instance-dependent lower bounds for the three problems above, showing that all our algorithms are optimal up to a poly-logarithmic factor. In Section 7 of the appendix, we make numerical simulations on synthetic data. The proofs of the theorems are in the appendix.

2. Problem formulation and notation

Consider a set of n experts evaluated on d tasks. As noted in the introduction, the performance of each expert $i \in [\![n]\!]$ on task $j \in [\![d]\!]$, is modeled via a random variable following an unknown distribution ν_{ij} with mean M_{ij} - and write $M = (M_{ij})_{i \leq n, j \leq d}$. We refer to $\nu = (\nu_{ij})_{i \leq n, j \leq d}$ as the (expert-task) performance distribution, and to $M = (M_{ij})_{i \leq n, j \leq d}$ as the mean (expert-task) performance. We have two aims in this work: (i) Ranking identification $(\boldsymbol{R})$, namely identifying the permutation that ranks the expert, and (ii) Best-expert identification $(\boldsymbol{B})$, namely identifying the best expert - we will define precisely these two objectives later.

On top of assuming as explained in the introduction the existence of a total ordering of experts following (1), we also make an identifiability assumption, so that we have a unique assumption for our problems of ranking $(\boldsymbol{R})$ and best-expert identification $(\boldsymbol{B})$; this is summarised in the following assumption.

Assumption 2.1. Suppose that the following assumption holds:

- **Monotonicity:** there exists a permutation $\pi : [\![n]\!] \to [\![n]\!]$, such that $\forall i \in [\![n-1]\!], \forall j \in [\![d]\!] : M_{\pi(i)j} \geq M_{\pi(i+1)j}$.
- **Bounded mean performance:** we assume that the mean performance M takes value in $[0,1]$, namely $M_{i,j} \in [0,1]$ for all $i \in [\![n]\!]$ and $j \in [\![d]\!]$.

We will then assume one of these two identifiability assumptions, depending on whether we are considering the ranking identification problem $(\boldsymbol{R})$, or the best expert identification problem $(\boldsymbol{B})$.

- **Identifiability for $(\boldsymbol{R})$:** for $i, k \in [\![n]\!]$, if $i \neq k$ then $\exists j \in [\![d]\!] : M_{i,j} \neq M_{k,j}$.
- **Identifiability for $(\boldsymbol{B})$:** for $i \in \{2, \dots, n\}$, $\exists j \in [\![d]\!] : M_{\pi(i),j} \neq M_{\pi(1),j}$.

Note that under the identifiability assumption for $(\boldsymbol{R})$, there exists a unique ranking, and that under the identifiability assumption for $(\boldsymbol{B})$, there exists a unique best expert (and that the identifiability assumption for $(\boldsymbol{R})$ implies the the identifiability assumption for $(\boldsymbol{B})$).

We write $\pi : [\![n]\!] \to [\![n]\!]$ for the corresponding permutation such that: $\pi(1) \overset{\bullet}{\geq} \pi(2) \overset{\bullet}{\geq} \dots \overset{\bullet}{\geq} \pi(n)$. In what follows, we write $\mathcal{M}_{n,d} \subset \mathbb{R}^{n \times d}$ for the set of matrices M satisfying Assumption 2.1. Moreover, we are also going to assume in what follows that the samples collected by the learner are sub-Gaussian.

Assumption 2.2. For each $i \in [\![n]\!], j \in [\![d]\!]$: ν_{ij} is 1-sub-Gaussian.

It is satisfied for e.g. random variables taking values in $[0, 1]$, or for Gaussian distributions of variance bounded by 1.

We consider the *fixed confidence setting* presented in the sequential and active identification literature (Heckel et al., 2019; Kaufmann et al., 2016; Garivier & Kaufmann, 2016). At each time t, the learner tasks an expert on a task - we write (I_t, J_t) for this pair of expert and task - based on previous observations, and receive an independent sample Z_t following the distribution of $\nu_{I_t J_t}$. Based on this information, the learner also decides whether it terminates, or continues sampling - and we write N for the termination time, to which we also refer to as number of queries. Upon termination, the learner outputs an estimate, and we consider here the two problems of ranking $(\boldsymbol{R})$, and best-expert identification $(\boldsymbol{B})$:

- (i) Ranking identification $(\boldsymbol{R})$: the learner aims at outputting a ranking $\hat{\pi}$ that estimates π. For a given confidence parameter $\delta \in (0, 1)$, we say that it is δ-accurate for ranking $(\boldsymbol{R})$ if it satisfies:

$$(\boldsymbol{R}): \quad \mathbb{P}(\hat{\pi} \neq \pi) \leq \delta$$

- (ii) Best-expert identification $(\boldsymbol{B})$: the learner aims at outputting an expert $\hat{k} \in [\![d]\!]$ that estimates the best performing expert, namely $\pi(1)$. For a given confidence parameter $\delta \in (0, 1)$, we say that it is δ-accurate for best-expert identification $(\boldsymbol{B})$ if it satisfies:

$$(\boldsymbol{B}): \quad \mathbb{P}(\hat{k} \neq \pi(1)) \leq \delta.$$

The performance of any δ-accurate algorithm is then measured through the total number N of queries made when the procedure terminates. The emphasis is then put on developing high probability guarantees on N i.e., bounds on N on the event of probability at least $1 - \delta$ where the δ-accurate algorithm is correct .

Notation: Let $A \in \mathbb{R}^{n \times d}$ be a rectangular matrix, for $i \in [\![n]\!]$, we write A_i for its ith line. Let $\|.\|_2$ denote the euclidean norm and $\|.\|_1$ denotes the l_1 norm on $\mathbb{R}^d$. For two numbers x, y, denote $x \vee y = \max\{x, y\}$.

3. Related work

Best-arm identification and Top-k bandit problems: Active ranking is related to many works in the vast literature of identification in the multi-armed bandit (MAB) model (Mannor & Tsitsiklis, 2004; Kaufmann et al., 2016; Garivier & Kaufmann, 2016), where each arm is associated to a univariate random variable. The learner's objective is to build a sampling strategy from the arms' distributions to identify the one with the largest mean - which would resemble our objective of best expert identification. Other related works consider the more general objective of identifying the top-k arms with the largest means - and in the case where the problem is solved for all k, it resembles the ranking problem. In this work, we consider a more general setting where instead of having a univariate distribution that characterizes the performance of an expert (akin to an arm in the aforementioned literature), we have a multivariate distribution, corresponding to the d questions.
Many of the previous works rely on a successive elimination approach by discarding arms that are seemingly sub-optimal. This idea is not directly applicable in our setting due to the multi-dimensional aspect of the rewards/performances of each candidate expert. Perhaps, the most natural way to cast our setting into the standard MAB framework is to associate each expert with the average of its performances in all tasks. For expert $i \in [\![n]\!]$, let $Y_i := \sum_{j=1}^{d} X_{ij}/d$ and $m_i := \mathbb{E}[Y_i]$ and observe that due to the monotonicity assumption on the matrix M, the ranking experts with respect to the means m_i leads to the correct ranking. Moreover, the learner has access to samples of Y_i by sampling the performance of expert i in a task chosen uniformly at random from d. Using the last scheme, one can exploit MAB methods to recover the correct ranking. However, we argue that such methods are sub-optimal: consider the problem of comparing two experts associated with distributions Y_1 and Y_2 defined previously. The minimum number of queries necessary to decide the better expert is of order $\mathcal{O}(1/(m_1 - m_2)^2) = \mathcal{O}(d^2/(\sum_{j=1}^{d} M_{1j} - M_{2j})^2)$, in contrast, the procedure presented in Section 4, decides the optimal expert using at most $\mathcal{O}(d/\sum_{j=1}^{d}(M_{1j} - M_{2j})^2)$ up to logarithmic factors. We can show using Jensen's inequality that our bound matches in the worst case the former. However, in general, the improvement can be up to a factor of d. This is due to the fact that our strategy uses a more refined choice of tasks to sample from.

On comparison-based ranking algorithms and duelling bandits: Many previous works consider the problem of ranking based on comparisons between experts in the online learning literature (Ailon et al., 2014; Chen et al., 2020; Heckel et al., 2019; Jamieson & Nowak, 2011; Jamieson et al., 2015; Urvoy et al., 2013; Yue et al., 2012). For instance, in (Heckel et al., 2019), the authors consider a setting where data consist of noisy binary outcomes of comparisons between pairs of experts. More precisely, the outcome of a queried pair of expert $(i, j) \in [\![n]\!] \times [\![n]\!]$ is a Bernoulli variable with mean M_{ij}. The experts are ranked with respect to their Borda scores defined for expert i by $\tau_i := \sum_{k \in [\![n]\!] \setminus \{i\}} M_{ik}/(n-1)$. The authors provide a successive elimination-based procedure leading to optimal guarantees up to logarithmic factors. Their bound on the total number of queries to recover the correct ranking is of

order $\tilde{\mathcal{O}}(\sum_{i=1}^{n-1} /(\tau_{\pi(i)} - \tau_{\pi(i+1)})^2)$, where π is the permutation corresponding to the correct ranking. Their setting can be harnessed into ours by considering that $d = n$ and that for each expert i, the task j consists of outperforming expert j. However, contrary to duelling bandit, our model underlies the additional assumption that experts' performances are monotone which is common in the batch ranking literature – see the next paragraph. In the duelling bandit framework, our monotonicity assumption is equivalent to the strong stochastic transitivity assumption (Shah et al., 2016), in the sense that $M_{\pi(i+1),k} \leq M_{\pi(i),k}$ for all k. In this work, our main idea is to build upon the monotonoticity assumption to drastically reduce in some problem instances the number of queries. Existing approaches based on fixed scores cannot be optimal on all problem instances as they do not adapt to the problem - e.g. applying Borda-scores algorithms in our setting leads to a total number of queries of order $d^2/(\sum_{j=1}^{d} M_{1,j} - M_{2,j})^2$, which compared to our bound is sub-optimal, with a difference up to a factor d.

On batch ranking: In Batch learning, the problem of ranking has attracted a lot of attention since the seminal work of Bradley & Terry, 1952. In this setting, the learner either observes noisy comparisons between experts or the performances of experts in given tasks. Observing that parametric models such as Bradley-Luce-Terry are sometimes too restrictive Shah et al., 2016 have initiated a stream of literature on non-parametric models under shape constraints (Flammarion et al., 2019; Shah et al., 2020; Mao et al., 2018; Liu & Moitra, 2020; Shah et al., 2019; Pananjady et al., 2020; Pananjady & Samworth, 2022; Pilliat et al., 2022). Our monotonicity assumption inspired from Flammarion et al., 2019 is the weakest one in this literature. That being said, our results and methods differ importantly from this literature as we aim at recovering the true ranking with a sequential strategy while these works aim at estimating an approximate ranking according to some loss function in the batch setting.

Link to adaptive signal detection: Consider two expert ($n = 2$) and let $M_1, M_2 \in \mathbb{R}^d$ be their mean performance vectors on d tasks. A closely related problem to the objective of identifying the best expert out of the two (under monotonicity assumption) is signal detection performed on the differences vector $\Delta = M_1 - M_2$, where the aim is to decide whether $\{M_1 = M_2\}$ or $\{M_1 \neq M_2\}$. There is a vast literature in the batch setting for the last testing problem (Baraud, 2002; Ingster et al., 2012; Poor, 1998).(Castro, 2014) considered the signal detection problem in the active setting: given a budget T, the learner's objective is to decide between the two hypotheses $\{\Delta = 0\}$ and $\{\Delta \neq 0\}$, under the assumption that Δ is s-sparse. When the magnitude of non-zero coefficients is μ, they proved that reliable signal detection requires a budget of order $\frac{d \log(1/\delta)}{s\mu^2}$, where δ is a prescribed bound on the risk of hypothesis testing. Our theoretical guarantees are consistent with the last bound and are valid for any difference vector Δ.

Link to bandits with infinitely many arms: As discussed earlier, a key feature of our setting, is the ability of the learner to pick the task to assess the performance of chosen experts. When comparing two experts, since the tasks underlying the greatest performance gaps are unknown, the learner should balance between exploring different tasks and committing to a small number of tasks to build a reliable estimate of the underlying differences. This type of breadth versus depth trade-off in pure exploration arises in the context of best-arm identification in the infinitely-many armed bandit problem. It was introduced by (Berry et al., 1997) and analysed in many subsequent works (Jamieson et al., 2016; Aziz et al., 2018; Katz-Samuels & Jamieson, 2020; de Heide et al., 2021). While comparing two experts in our setting includes dealing with similar challenges in the previous literature, note that we are particularly interested in detecting the existence (and the sign) of the gaps between experts' performances rather than identifying tasks with the largest performance difference.

4. Comparing two experts ($n = 2$)

We start by considering the case where $n = 2$, which we will then use as a building block for the general case. In this case, the ranking problem $(\boldsymbol{R})$ is equivalent to the best-expert identification problem $(\boldsymbol{B})$. Algorithm 1 takes as input two parameters: a confidence level $\delta \in (0, 1)$, and a precision parameter $\epsilon > 0$, and outputs the best expert if the L_2 distance between the compared experts is greater than ϵ.

One wants ideally to focus on the task displaying the greater gap in performance in order to quickly identify the best expert, however, such knowledge is not available to the learner. This raises the challenge of balancing between sampling as many different tasks as possible to pick one that has a large gap and focusing on one task to be able to distinguish between the two experts based on this task. Such width versus depth trade-off arises in many works of best arm identification with infinitely many arms (de Heide et al., 2021; Jamieson et al., 2016), where the proportion of optimal arms is $p \in (0, 1)$ and the gap between optimal and sub-optimal arms is $\Delta > 0$. In contrast, the gaps between tasks (equivalent to arms in the previous problem) may be dense, in order to bypass this difficulty we make the following observation: For $j \in [\![d]\!]$ let x_j denote the gap in task j: $x_j := |M_{1,j} - M_{2,j}|$. Denote $x_{(j)}$ the corresponding decreasing sequence, we have:

$$\max_{s \in [\![d]\!]} s x_{(s)}^2 \leq \sum_{j=1}^{d} x_j^2 \leq \log(2d) \max_{s \in [\![d]\!]} s x_{(s)}^2. \qquad (2)$$

The last inequality was presented in (Audibert et al., 2010) in the context of fixed budget best arm identification (see Lemma F.3 in the appendix). It suggests that, up to a logarithmic factor in d, the L_2 distance is mainly concentrated in the top s^* elements with the largest magnitude, where s^* is an element in $[\![d]\!]$ satisfying the maximum in equation (2). We exploit this observation by focusing our exploration effort on tasks with the largest s^* gaps.

In the first part of our strategy, presented in Algorithm 1, we discretize the set of possible values of s^* and $x_{(s^*)}$ using a doubling trick. The last scheme was adapted from (Jamieson et al., 2016), and allows to be adaptive to the unknown quantities $(s^*, x_{(s^*)})$. In the second part, given a prior guess (s, h), we run Algorithm 2 consisting of two main ingredients: (i) sampling strategy and (ii) stopping rule. We start by sampling a large number of tasks (with replacement) to ensure that with large probability, a proportion s/d of "good tasks" - i.e. relevant for ranking the two experts, namely the tasks where the experts differ the most - where chosen, then we proceed by median elimination by keeping only half of the tasks with the largest empirical mean in each iteration and doubling the number of queries made to have a more precise estimate of the population means. The aim of this process is to focus the sampling force on the "good tasks", by gradually eliminating tasks where the two experts perform similarly. Also, at each iteration, we run a test on the average of the kept tasks to potentially conclude on which one of the two experts is best and terminate the algorithm.

Algorithm 1 `Compare`(δ, ϵ)

Input: δ, ϵ.
Initialize: $\rho = 1$, output $\hat{k} = \texttt{null}$.
while $\{\hat{k} = \texttt{null}\}$ and $\epsilon^2 < 4\log(2d)d\, 2^{-\rho}$ **do**
  for $r = 0, \dots, \rho$ **do**
    Set $s_r = 2^r d/2^\rho$, $h_r = 1/\sqrt{2^r}$.
    Run: `Try-compare`$(\delta/(10\rho^3\log(d)), s_k, h_k)$ and **Set** : $\hat{k}$ its output.
    If $\hat{k} \neq \texttt{null}$, **break**.
  end for
  $\rho \leftarrow \rho + 1$
end while
Output: $\hat{k}$.

We now turn to the analysis of the performance of `compare`. We first show that with probability at least $1-\delta$, Algorithm 1 does not make the wrong diagnostic. Based on the precision parameter ϵ, we prove that if the unknown squared L_2 distance between the experts' performance vectors is larger than ϵ, then the algorithm identifies the optimal expert out of the two, with a large probability. We also bound on the same event the total number of queries N made by the procedure.

Algorithm 2 `Try-compare`(δ, s, h)

Input: $\delta \in (0,1), s \leq d, h > 0$.
Initialize: $\phi = 2^{\lceil \log_2[26\log(1/\delta)d/s]\rceil}$, $n_0 = 64/h^2$, $\hat{k} \leftarrow \texttt{null}$.
Draw a set (denoted S_1) of ϕ elements from $[\![d]\!]$ uniformly at random *with replacement*.
Initialize: $S_1^{(12)} = S_1$ and $S_1^{(21)} = S_1$.
for $\ell = 1, \dots, \lceil \log_{4/3}(d/s)\rceil + 1$ **do**
  Sample: For each element $j \in S_\ell^{(12)} \cup S_\ell^{(21)}$, sample the entries $(1,j)$ and $(2,j)$ $t_\ell = n\phi/\left|S_\ell^{(12)}\right|$ times, denote $(X_{1,j}^{(u,\ell)}, X_{2,j}^{(u,\ell)})_{u \leq t_\ell}$ the obtained samples.
  Compute: the means

$$\hat{\mu}_{j,\ell}^{(12)} = \frac{1}{t_\ell}\sum_{u=1}^{t_\ell}\left(X_{1,j}^{(u,\ell)} - X_{2,j}^{(u,\ell)}\right)$$

$$\hat{\mu}_{\ell}^{(12)} = \frac{1}{n_0\phi}\sum_{u=1}^{t_\ell}\sum_{j\in S_\ell}\left(X_{1,j}^{(u,\ell)} - X_{2,j}^{(u,\ell)}\right).$$

  Denote $\hat{\mu}_\ell^{(21)} = -\hat{\mu}_\ell^{(12)}$
  if $\hat{\mu}_\ell^{(12)} \geq \sqrt{\frac{2\log(2/\delta)}{n_0\phi}}$ **then**
    $\hat{k} = 1$, **break**
  else if $\hat{\mu}_\ell^{(21)} \geq \sqrt{\frac{2\log(2/\delta)}{n_0\phi}}$ **then**
    $\hat{k} = 2$, **break**
  else
    Find the median of $(\hat{\mu}_{j,\ell}^{(12)})_{i\in S_\ell^{(12)}}$ and $(\hat{\mu}_{j,\ell}^{(21)})_{i\in S_\ell^{(21)}}$, denoted $m_\ell^{(12)}$ and $m_\ell^{(21)}$ resp.
    $S_{\ell+1}^{(12)} \leftarrow S_\ell^{(12)} \setminus \{i \in S_\ell^{(12)} : \hat{\mu}_{j,\ell}^{(12)} < m_\ell^{(12)}\}$.
    $S_{\ell+1}^{(21)} \leftarrow S_\ell^{(21)} \setminus \{i \in S_\ell^{(21)} : \hat{\mu}_{j,\ell}^{(21)} < m_\ell^{(21)}\}$.
  end if
end for
Output: $\hat{k}$

Theorem 4.1. *Suppose Assumption 2.1 holds. For $\delta \in (0,1)$, $\epsilon > 0$, consider Algorithm 1 with input (δ, ϵ). Define*

$$H := \frac{d}{\|M_1 - M_2\|_2^2}\,.$$

With probability at least $1-\delta$, we have

- *The output $\hat{k}$ satisfies: $\hat{k} \in \{\texttt{null}, \pi(1)\}$.*
- *If $\epsilon < \|M_1 - M_2\|_2$, the output satisfies: $\hat{k} = \pi(1)$.*

Moreover, with probability at least $1-\delta$, its total number of queries, denoted N, satisfies:

$$N \leq \tilde{c} \times \log(1/\delta) H_\epsilon,$$

where $\tilde{c} = \log^2(d)\log(H_\epsilon)\log(\log(H_\epsilon \vee d))$*, where* $H_\epsilon := \min\{H, d/\epsilon^2\}$ *and where* c *is a numerical constant.*

We now discuss some properties of the algorithms and its corresponding guarantees. The sample complexity of Algorithm 1 is of order $H = d/\|M_1 - M_2\|_2^2$, matching the lower bound presented in Theorem 6.1 in Section 6 up to a poly-logarithmic factor. The dependence on the L_2 distance between experts is mainly due to the sampling scheme introduced in Algorithm 2, relying on median elimination. We illustrate the gain of our algorithm with respect to a uniform allocation strategy across all tasks. Such sampling schemes rely on the average performances of each expert across all tasks as a criterion (Heckel et al., 2019; Jamieson et al., 2015; Urvoy et al., 2013). Suppose that the difference vector is s-sparse and the amplitude of the non-zero gaps is equal to Δ. Then, sampling a task uniformly at random leads to an expected gap of $s\Delta/d$, which leads to a sample complexity of $\mathcal{O}(d^2/(s\Delta)^2)$ in order to identify the optimal expert. On the other hand, our bound suggests that the total number of queries scales as $\mathcal{O}(d/(s\Delta^2))$. This gain is due to the median elimination strategy allowing the concentration of the sampling effort on tasks with large gaps. More precisely, we prove that the initially sampled set of tasks S_1 contains a proportion of s/d of non-zero gaps and that at each iteration in median elimination the last proportion increases by at least a constant factor $r > 1$. Consequently, after roughly $\log(d/s)$ iteration, "good tasks" constitute a constant proportion of the active tasks, which allows the algorithm to conclude by satisfying the stopping rule condition.

5. Ranking estimation (general n)

5.1. Ranking identification

In this section, we consider the task of ranking identification $(\boldsymbol{R})$. Algorithm 3 takes as input $\delta \in (0,1)$ and outputs a ranking $\hat{\pi}$ of all the experts. We use Algorithm 1 with precision $\epsilon = 0$ to compare any pair experts. Given Algorithm 1 as a building block, we proceed using the binary insertion sort procedure: experts are inserted sequentially and the location of each expert is determined using a binary search. Algorithm 3 presents the procedure, where in each iteration (corresponding to the insertion of a new expert) a call to the procedure `Binary-search` is made. A detailed implementation of the last procedure (using Algorithm 1) is presented in Section C in the appendix.

The following theorem presents the theoretical guarantees on the output and sample complexity of Algorithm 3.

Theorem 5.1. *Suppose Assumption 2.1 holds. Let* $\delta \in (0,1)$*, and define for* $i \in [\![n-1]\!]$*:*

$$H_i := \frac{d}{\left\|M_{\pi(i)} - M_{\pi(i+1)}\right\|_2^2}.$$

Let $H^* = \max_{i \le n-1} H_i$*. With probability at least* $1-\delta$*, Algorithm 3 outputs the correct ranking, and its total number of queries (denoted* N*) is upper bounded by:*

$$N \le \tilde{c} \times \log(1/\delta) \sum_{i=1}^{n-1} H_i \ ,$$

where $\tilde{c} = c\log(n)\log^2(d)\log(H^*)\log(n\log(H^* \vee d))$ *and where* c *is a numerical constant.*

Algorithm 3 `Active-ranking`(δ)

Input: δ.
Initialize: Define $\hat{\pi}$ by $\hat{\pi}(1) = 1$.
for $i = 2, \dots, n$ **do**
  $j \leftarrow$ `Binary-search`$(\frac{\delta}{n\lceil \log_2(n)\rceil}), i, \hat{\pi}, 1, i-1)$.
  Shift the rank of $\hat{\pi}[k]$ for $k \ge j$ by one.
  $\hat{\pi}[j] \leftarrow i$
end for
Output: $\hat{\pi}$.

The first result states that Algorithm 3 with input δ is δ-accurate. The second guarantee is a control on the total number of queries made by our procedure, with a large probability. As one would expect, the cost of full ranking of experts underlies the cost of distinguishing between two consecutive experts $\pi(i)$ and $\pi(i+1)$ for $i \in [\![n-1]\!]$. The last cost is characterized by the sample complexity $H_i = d/\left\|M_{\pi(i)} - M_{\pi(i+1)}\right\|_2^2$. Consequently, the total number of queries made by Algorithm 3 is of order of $\mathcal{O}\left(\sum_{i=1}^{n-1} H_i\right)$ up to a poly-logarithmic factor.

5.2. Best expert identification:

Algorithm 4 takes as input the confidence parameter δ and outputs a candidate for the best expert. Note that while ranking the experts using Algorithm 3 would lead to identifying the top item, one would expect to use fewer queries for this relaxed objective. Algorithm 4 builds on a variant of the `Max-search` algorithm (a detailed implementation is presented in Section D). In the last algorithm, given a subset of expert S and a precision ϵ, we initially select an arbitrary element as a candidate for the best expert, then we perform comparisons with the remaining elements of S using `compare`$(\delta/n, \epsilon)$ and update the candidate for the best expert accordingly. This method leads to eliminating sub-optimal experts that have an L_2 distance with respect to expert $\pi(1)$ larger than $\mathcal{O}(\epsilon)$. Therefore, performing sequentially `Max-search` and dividing the prescribed precision

ϵ by 4 after each iteration allows identifying the optimal expert after roughly $\log_4(1/\|M_{\pi(1)} - M_{\pi(2)}\|_2^2)$ iterations.

Algorithm 4 `Best-expert`(δ)

Input: δ, $S = [\![n]\!]$, $\epsilon = 1$, $\delta' = \delta/2$
while $|S| > 1$ **do**
  $S \leftarrow$ `Max-search`(δ', S, ϵ).
  $\epsilon \leftarrow \epsilon/4$
  $\delta' \leftarrow \delta'/2$
end while
Output: S.

The theorem below analyses the performance of Algorithm 4.

Theorem 5.2. *Suppose Assumption 2.1 holds. Let $\delta \in (0,1)$, and define for $i \in [\![2,n]\!]$:*

$$G_i := \frac{d}{\|M_{\pi(1)} - M_{\pi(i)}\|_2^2}.$$

Let $G^* = \max_{i\in[\![2,n]\!]} G_i$. With probability at least $1-\delta$, Algorithm 4 outputs $\pi(1)$, and the total number of queries (denoted N) satisfies

$$N \leq \tilde{c} \times \log(1/\delta) \sum_{i=2}^{n} G_i,$$

with $\tilde{c} = c\log^2(d)\log^2(G^*)\log(n\log(d))$ and c a numerical constant.

In the first result, we show that Algorithm 4 is δ-accurate for $(\boldsymbol{B})$. The second result presents a bound on the total number of queries made by the algorithm. Observe that for each $i \in \{2,\dots,n\}$ the quantity $G_i = d/\|M_{\pi(1)} - M_{\pi(i)}\|_2^2$ characterizes the sample complexity to distinguish expert $\pi(i)$ from the optimal expert. In order to identify the correct expert, a number of queries of order G_i is required for each suboptimal expert $\pi(i)$, which leads to a total number of queries of order $\sum_{i=2}^n G_i$, up to a poly-logarithmic factor.

5.3. Discussion

Algorithms 3 and 4 are proven to be δ accurate for the ranking problem $(\boldsymbol{R})$ and best expert $(\boldsymbol{B})$. Moreover, Theorems 6.2 and 6.3 in Section 6 below show that their sample complexities are optimal up to a poly-logarithmic factor. While both procedures rely on comparing pairs of experts, their use of the `compare` procedure presented in Algorithm 1 is different. The ranking procedure builds on `compare`$(\delta, 0)$, i.e., we need the exact order between compared experts. In contrast, for best expert identification, we only need approximate comparisons output by `compare`(δ, ϵ) for an adequately chosen precision ϵ. This difference is due to the fact that a complete ranking underlies comparing the closest pair of experts (in L_2 distance) while best expert identification requires distinguishing only the optimal expert from the sub-optimal ones.

6. Lower bounds

In this section, we provide some lower bounds on the number of queries of any δ-accurate algorithm. As for the upper bound, we first consider the case of 2 experts ($n = 2$) in which ranking identification is the same as best expert identification. Then we consider the general case (any $n \geq 2$).

6.1. Lower bound in the case of two experts ($n = 2$)

Problem-dependent lower bounds, i.e. lower bounds that depend on the problem instance at hand[1], are generally obtained by considering slight changes of any fixed problem, and proving that it is not possible to have an algorithm that performs well enough simultaneously on all the resulting problems. In the ranking problem, our monotonicity assumption constraint heavily restricts the nature of problem changes that we are allowed to consider.

For a given matrix $M_0 \in \mathcal{M}_{2,d}$ representing the mean performance of a problem, a minimal and very natural class is the one containing $M_0 = ((M_0)_{1,\cdot}, (M_0)_{2,\cdot})^T$, and also the matrix $((M_0)_{2,\cdot}, (M_0)_{1,\cdot})^T$ where the two rows (experts) are permuted. In this class, however, we know the position of the question leading to maximal difference of performance between the two experts, as it is the question j such that $|(M_0)_{1,j} - (M_0)_{2,j}|$ is maximised. So that an optimal strategy over this class would leverage this information by sampling only this question, and would be able to terminate using a number of query N smaller in order than $\log(1/\delta)/\max_j \left[\left((M_0)_{1,j} - (M_0)_{2,j}\right)^2\right]$, with probability larger than $1-\delta$. Note that this is significantly smaller than our upper bound in Theorem 4.1, but that the algorithm that we alluded to is dependent on the exact knowledge of the matrix M_0 - and in particular the positions of the informative questions - which is not available to the learner, and also very difficult to estimate. In order to include this absence of knowledge in the lower bound, we have to make the class of problems larger, by ensuring in particular that the position of informative questions are not available to the learner. A natural enlargement of the class of problem that takes this into account, but that is still very natural and tied to the matrix M_0, is to consider the class of all matrices whose gaps between experts are equal to those in M_0 up to a permutation of rows and columns (experts and questions). This is precisely the class that we consider in our lower bound, and that we detail below.

Let $M_0 \in \mathcal{M}_{2,d}$. Write $\underline{\Delta} = |(M_0)_{1,\cdot} - (M_0)_{2,\cdot}|$ for the vector of gaps between experts, and π_0 for the permutation that makes M_0 monotone - and we remind that $(M_0)_{\pi_0(2),\cdot}$ is the least performant of the two experts. Write $\mathbb{D}_{M_0}$ for the set of performance distributions, namely of distributions

[1] which is what we need in order to match the upper bound in Theorem 4.1.

of $X = (X_{i,j})_{i\in[\![2]\!],j\in[\![d]\!]}$ corresponding to means M, such that: (i) Assumption 2.2 is satisfied for X, and (ii) The mean performance $M_\pi \in \mathcal{M}_{2,d}$, with associated permutation π that transforms it into a monotone matrix, is such that $M_{\pi(2),.} = (M_0)_{\pi_0(2),.}$, and such that there exists a permutation σ of $[\![d]\!]$ such that $M_{\pi(1),.} = (M_0)_{\pi_0(2),.} + \underline{\Delta}_{\sigma(.)}$.

The set $\mathbb{D}_{M_0}$ is therefore the set of all distributions of X that are 1-sub-Gaussian, and where while one expert in M is equal to the worst expert $(M_0)_{\pi_0(2),.}$ of M_0, the best expert is equal to $(M_0)_{\pi_0(2),.}$ plus a permutation of the gap vector $\underline{\Delta}$. This ensures that the gap structure over the mean performance is the same for all problems in $\mathbb{D}_{M_0}$.

The following theorem establishes a high probability lower bound on the termination time N over the class of problems $\mathbb{D}_{M_0}$.

Theorem 6.1. *Fix $n = 2$. Let $d \geq 1$ and $\delta \in (0, 1/4)$. Consider any matrix $M_0 \in \mathcal{M}_{2,d}$. For any δ-accurate algorithm A for either the ranking identification, or best expert identification (which is the same for $n = 2$), we have:*

$$\max_{\mathcal{B}\in\mathbb{D}_{M_0}} \mathbb{P}_{\mathcal{B},A}\left[N \geq \frac{d}{2\|(M_0)_{1,.} - (M_0)_{2,.}\|_2^2} \log\left(\frac{1}{6\delta}\right)\right] \geq \delta,$$

where $\mathbb{P}_{\mathcal{B},A}$ is the probability corresponding to samples collected by algorithm A on problem $\mathcal{B}$.

This theorem lower bounds the budget of any δ-accurate algorithm by $\frac{d}{\|M_1-M_2\|_2^2} \log\left(\frac{1}{6\delta}\right)$, which matches up to logarithmic terms the upper bound in Theorem 4.1. Interestingly, the query complexity depends therefore only on $\|(M_0)_1 - (M_0)_2\|_2^2$, independently of the gap profile of $(M_0)_1 - (M_0)_2$ - i.e. of whether there are many small differences in performances across tasks, or a few large differences. Of course, a related optimal algorithm would solve differently the width versus depth trade-off on a sparse or dense problem, as discussed in Section 4, yet it does not show in the final bound on the query complexity thanks to the adaptivity of the sampling. A related phenomenon was already observed - albeit in a different regime and context - in (Castro, 2014).
The bound on N in Theorem 6.1 is in high probability, on an event of high probability $1 - \delta$ - where $1 - \delta$ is also the minimal probability of being accurate for the algorithm. This matches our upper bound in Theorem 4.1, where we also provide high probability upper bounds for N.

6.2. Lower bound in the general case (any n)

We now consider the general problem of ranking and best expert identification when $n > 2$. As these two problems are not equivalent anymore, we provide two lower bounds.

In this part, we will consider classes of problems for constructing the lower bound which are wider than the one constructed for the case $n = 2$, see Theorem 6.1 and the class $\mathbb{D}_{M_0}$. Driven by the fact that the quantity that appears there is the L_2 norm between experts, we will define the classes of problems by imposing constraints on the L_2 distance between pairs of experts.

6.2.1. Ranking identification

Fix any $\mathbf{\Delta} = (\Delta_1, \dots, \Delta_{n-1})$, such that $\Delta_i > 0$ for each $i \leq n-1$. Write $\mathbb{D}^{n,d}_{\mathbf{\Delta}}$ for the set of performance distributions, namely of distributions of $X = (X_{i,j})_{i\in[\![n]\!],j\in[\![d]\!]}$ corresponding to means M, such that: (i) Assumption 2.2 is satisfied for X, and (ii) The mean performance $M \in \mathcal{M}_{n,d}$ satisfies

$$\|M_{\pi(i)} - M_{\pi(i+1)}\|_2^2 = \Delta_i^2, \text{ for } i = 1, \dots, n-1 .$$

The class $\mathbb{D}^{n,d}_{\mathbf{\Delta}}$ is such that the L_2 distance between the i-th best expert $\pi(i)$ and the $i+1$-th best expert $\pi(i+1)$ is fixed to Δ_i. We however do not make further assumption on the gap structure within questions, as is done in Theorem 6.1 through the class $\mathbb{D}_{M_0}$.

Next, we provide a minimax lower bound for general n. The following theorem lower bound the expected budget when we fix the sequence of L_2 distances between consecutive rows.

Theorem 6.2. *Let $n, d \geq 1$, $\delta \in (0, 1)$. For any δ-accurate algorithm A for ranking identification $(\boldsymbol{R})$, we have:*

$$\max_{\mathcal{B}\in\mathbb{D}^{n,d}_{\mathbf{\Delta}}} \mathbb{E}_{\mathcal{B},A}[N] \geq \sum_{i=1}^{n-1} \frac{d}{\Delta_i^2} \log(1/(4\delta)).$$

where $\mathbb{E}_{\mathcal{B},A}$ is the probability corresponding to samples collected by algorithm A on problem $\mathcal{B}$.

6.2.2. Best expert identification

Fix any positive and non-decreasing sequence $\overline{\mathbf{\Delta}} = (\overline{\Delta}_1, \dots, \overline{\Delta}_{n-1})$, such that $\Delta_i > 0$ for each $i \leq n-1$. Write $\overline{\mathbb{D}}^{n,d}_{\overline{\mathbf{\Delta}}}$ for the set of performance distributions, namely of distributions of $X = (X_{i,j})_{i\in[\![n]\!],j\in[\![d]\!]}$ corresponding to means M, such that:

the set of distributions of experts performances such that: (i) Assumption 2.2 is satisfied for X, and (ii) The mean performances matrix $M \in \mathcal{M}_{n,d}$ satisfies

$$\|M_{\pi(1)} - M_{\pi(i+1)}\|_2^2 \geq \overline{\Delta}_i^2, \text{ for } i = 1, \dots, n-1 .$$

The class $\overline{\mathbb{D}}^{n,d}_{\overline{\mathbf{\Delta}}}$ is such that the L_2 distance between the best expert $\pi(1)$ and the i-th best expert $\pi(i+1)$ is fixed to $\overline{\Delta}_i$. It is related to the construction for Theorem 6.2 of the set $\mathbb{D}^{n,d}_{\mathbf{\Delta}}$, yet here we only consider the distance to the best expert.

Theorem 6.3. *Let $n, d \geq 1$, $\delta \in (0,1)$. For any δ-accurate algorithm A for best expert identification $(\boldsymbol{B})$, we have:*

$$\min_{A \in \boldsymbol{A}(\delta)} \max_{\mathcal{B} \in \overline{\mathbb{D}}^{n,d}_{\boldsymbol{\Delta}}} \mathbb{E}_{\mathcal{B},A}[N] \geq \sum_{i=1}^{n-1} \frac{d}{\Delta_i^2} \log(1/(4\delta)),$$

where $\mathbb{E}_{\mathcal{B},A}$ is the probability corresponding to samples collected by algorithm A on bandit problem $\mathcal{B}$.

7. Numerical simulations

In this section, we perform some numerical simulations on synthetic data to compare our Algorithm with a benchmark procedure from the literature. We chose AR algorithm from (Heckel et al., 2019) since they considered the problem of ranking experts in an active setting. Their method is based on pairwise comparisons and uses Borda scores as a criterion to rank experts. In order to harness their model into ours, we proceed as follows: when querying a pair of experts (i, j), we sample a task uniformly at random from $[\![d]\!]$, and sample the performances of experts on this task then output the result.

We focus on the specific problem of identifying the best out of two experts ($n = 2$) and $d = 10$ tasks. For each $s \in [\![d]\!]$, we consider the following scenario: the performances of both experts in each task follow a normal distribution with unit variance. The means of performances of the sub-optimal expert $M_{\pi(2)}$ are drawn from $[0, 1/2]$ following the uniform law. We build a gaps vector Δ_s that is s-sparse, the non-zero tasks are drawn uniformly at random from $[\![d]\!]$, and the value of the kth non-zero gap is set to $\left(\frac{k}{3s}\right)^2$. Then we consider $M_{\pi(1)} = M_{\pi(2)} + \Delta_s$. Figure7 presents the sample complexity of Algorithm 1 with parameters $(\delta, 0)$ and AR from (Heckel et al., 2019), as a function of the sparsity ratio s/d for $s \in [\![d]\!]$. The results are averaged over 20 simulations for each scenario, in all simulations both algorithms made the correct output.

Figure 7 presents the empirical sample complexity of Algorithm 1 and AR as a function of the sparsity rate s/d. The results show, as suggested by theoretical guarantees, that Algorithm 1 with parameters $(\delta, 0)$ outperforms AR for small sparsity rates, mainly due to its ability to detect large gaps, as discussed previously. For large sparsity rates, the gaps vector is dense, and evaluating experts using their average performance across all tasks proves to be efficient. In the last regime, AR procedure outperforms Algorithm 1.

8. Conclusion

In this paper, we have addressed the challenge of ranking a set of n experts based on sequential queries of their performance variables in d tasks. By assuming the monotonicity of the mean performance matrix, we have introduced strategies that effectively determine the correct ranking of experts. These strategies optimize the allocation of queries to tasks with larger gaps between experts, resulting in a considerable improvement compared to traditional measures like Borda Scores.

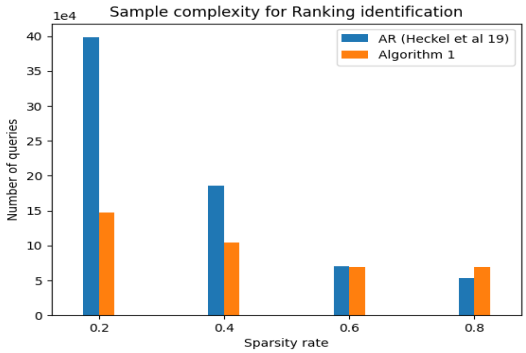

Figure 1. Empirical sample complexity of AR by (Heckel et al., 2019) and Algorithm 1 in this paper. We varied the sparsity rate of the gaps vector $s/d \in [5\%, 100\%]$. The presented results are averaged over 20 simulations.

Our research has unveiled several promising avenues for future exploration. One notable direction involves narrowing the poly-logarithmic gap in n between our upper and lower bounds for both full ranking and best expert identification. Achieving this goal will require the development of more refined ranking strategies, which we leave for future investigation. Additionally, relaxing the monotonicity assumption considered in this study and adopting a more inclusive framework that accommodates diverse practical applications would be an intriguing area to explore. It would be worthwhile to scrutinize the assumptions made in the study conducted by (Bengs et al., 2021) as a potential direction for further research.

Acknowledgments: The work of A. Carpentier is partially supported by the Deutsche Forschungsgemeinschaft (DFG) Emmy Noether grant MuSyAD (CA 1488/1-1), by the DFG – 314838170, GRK 2297 MathCoRe, by the FG DFG, by the DFG CRC 1294 'Data Assimilation', Project A03, by the Forschungsgruppe FOR 5381 "Mathematical Statistics in the Information Age – Statistical Efficiency and Computational Tractability", Project TP 02, by the Agence Nationale de la Recherche (ANR) and the DFG on the French-German PRCI ANR ASCAI CA 1488/4-1 "Aktive und Batch-Segmentierung, Clustering und Seriation: Grundlagen der KI" and by the UFA-DFH through the French-German Doktorandenkolleg CDFA 01-18 and by the SFI Sachsen-Anhalt for the project RE-BCI. The work of E.M. Saad and N. Verzelen is supported by ANR-21-CE23-0035 (ASCAI).

References

Ailon, N., Karnin, Z., and Joachims, T. Reducing dueling bandits to cardinal bandits. In *International Conference on Machine Learning*, pp. 856–864. PMLR, 2014.

Audibert, J.-Y., Bubeck, S., and Munos, R. Best arm identification in multi-armed bandits. In *COLT*, pp. 41–53, 2010.

Aziz, M., Anderton, J., Kaufmann, E., and Aslam, J. Pure exploration in infinitely-armed bandit models with fixed-confidence. In *Algorithmic Learning Theory*, pp. 3–24. PMLR, 2018.

Baraud, Y. Non-asymptotic minimax rates of testing in signal detection. *Bernoulli*, pp. 577–606, 2002.

Bengs, V., Busa-Fekete, R., El Mesaoudi-Paul, A., and Hüllermeier, E. Preference-based online learning with dueling bandits: A survey. *The Journal of Machine Learning Research*, 22(1):278–385, 2021.

Berry, D. A., Chen, R. W., Zame, A., Heath, D. C., and Shepp, L. A. Bandit problems with infinitely many arms. *The Annals of Statistics*, 25(5):2103–2116, 1997.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

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

Chen, W., Du, Y., Huang, L., and Zhao, H. Combinatorial pure exploration for dueling bandit. In *International Conference on Machine Learning*, pp. 1531–1541. PMLR, 2020.

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

Flammarion, N., Mao, C., and Rigollet, P. Optimal rates of statistical seriation. *Bernoulli*, 25(1), 2019.

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

Heckel, R., Shah, N. B., Ramchandran, K., and Wainwright, M. J. Active ranking from pairwise comparisons and when parametric assumptions do not help. *The Annals of Statistics*, 47(6):3099–3126, 2019.

Ingster, Y. I., Sapatinas, T., and Suslina, I. A. Minimax signal detection in ill-posed inverse problems. *The Annals of Statistics*, 40(3):1524–1549, 2012.

Jamieson, K., Katariya, S., Deshpande, A., and Nowak, R. Sparse dueling bandits. In *Artificial Intelligence and Statistics*, pp. 416–424. PMLR, 2015.

Jamieson, K. G. and Nowak, R. Active ranking using pairwise comparisons. *Advances in neural information processing systems*, 24, 2011.

Jamieson, K. G., Haas, D., and Recht, B. The power of adaptivity in identifying statistical alternatives. *Advances in Neural Information Processing Systems*, 29, 2016.

Karger, D., Oh, S., and Shah, D. Iterative learning for reliable crowdsourcing systems. *Advances in neural information processing systems*, 24, 2011.

Katz-Samuels, J. and Jamieson, K. The true sample complexity of identifying good arms. In *International Conference on Artificial Intelligence and Statistics*, pp. 1781–1791. PMLR, 2020.

Kaufmann, E., Cappé, O., and Garivier, A. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17 (1):1–42, 2016.

Liu, A. and Moitra, A. Better algorithms for estimating non-parametric models in crowd-sourcing and rank aggregation. In *Conference on Learning Theory*, pp. 2780–2829. PMLR, 2020.

Lu, J., Wu, D., Mao, M., Wang, W., and Zhang, G. Recommender system application developments: a survey. *Decision Support Systems*, 74:12–32, 2015.

Mannor, S. and Tsitsiklis, J. N. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun):623–648, 2004.

Mao, C., Pananjady, A., and Wainwright, M. J. Breaking the $1/\sqrt{n}$ barrier: Faster rates for permutation-based models in polynomial time. In *Conference On Learning Theory*, pp. 2037–2042. PMLR, 2018.

Morgulev, E., Azar, O. H., and Lidor, R. Sports analytics and the big-data era. *International Journal of Data Science and Analytics*, 5(4):213–222, 2018.

Pananjady, A. and Samworth, R. J. Isotonic regression with unknown permutations: Statistics, computation and adaptation. *The Annals of Statistics*, 50(1):324–350, 2022.

Pananjady, A., Mao, C., Muthukumar, V., Wainwright, M. J., and Courtade, T. A. Worst-case versus average-case design for estimation from partial pairwise comparisons. *The Annals of Statistics*, 48(2):1072–1097, 2020.

Pappalardo, L., Cintia, P., Ferragina, P., Massucco, E., Pedreschi, D., and Giannotti, F. Playerank: data-driven performance evaluation and player ranking in soccer via a machine learning approach. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(5):1–27, 2019.

Pilliat, E., Carpentier, A., and Verzelen, N. Optimal permutation estimation in crowd-sourcing problems. *arXiv preprint arXiv:2211.04092*, 2022.

Poor, H. V. *An introduction to signal detection and estimation*. Springer Science & Business Media, 1998.

Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L., and Moy, L. Learning from crowds. *Journal of machine learning research*, 11(4), 2010.

Shah, N. B., Balakrishnan, S., Guntuboyina, A., and Wainwright, M. J. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. *IEEE Transactions on Information Theory*, 63(2):934–959, 2016.

Shah, N. B., Balakrishnan, S., and Wainwright, M. J. Feeling the bern: Adaptive estimators for bernoulli probabilities of pairwise comparisons. *IEEE Transactions on Information Theory*, 65(8):4854–4874, 2019.

Shah, N. B., Balakrishnan, S., and Wainwright, M. J. A permutation-based model for crowd labeling: Optimal estimation and robustness. *IEEE Transactions on Information Theory*, 67(6):4162–4184, 2020.

Snow, R., O'connor, B., Jurafsky, D., and Ng, A. Y. Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pp. 254–263, 2008.

Tsybakov, A. B. Introduction to nonparametric estimation, 2009. *URL https://doi. org/10.1007/b13794. Revised and extended from the*, 9(10), 2004.

Urvoy, T., Clerot, F., Féraud, R., and Naamane, S. Generic exploration and k-armed voting bandits. In *International Conference on Machine Learning*, pp. 91–99. PMLR, 2013.

Yue, Y., Broder, J., Kleinberg, R., and Joachims, T. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Zhou, X., Xu, Y., Li, Y., Josang, A., and Cox, C. The state-of-the-art in personalized recommender systems for social networking. *Artificial Intelligence Review*, 37(2): 119–132, 2012.

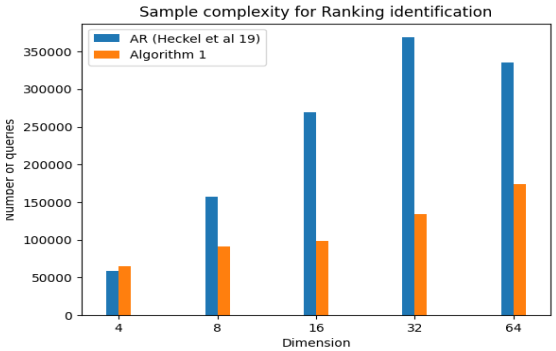

Figure 2. Empirical sample complexity of AR by (Heckel et al., 2019) and Algorithm 1 in this paper. We varied the number of tasks $d \in [4, 8, 16, 32, 64]$ while keeping the sparsity rate constant $1/3$. The presented results are averaged over 20 simulations.

A. Additional numerical simulations

We add a numerical simulation on synthetically generated data. We consider the task of comapring two experts and suppose that the performance gaps vector is sparse, with a fixed sparsity rate of $1/3$. We conduct simulations for various dimension (number of tasks d) and plot the sample complexities of our algorithm and the benchmark algorithm AR ((Heckel et al., 2019)). Figure A displays the results.

B. Proof of Theorem 4.1

Suppose Assumption 2.1 holds and that (w.l.o.g) expert 1 is the optimal expert, that is $\pi(1) = 1$.

Additional notation: Define $x_j := M_{1,j} - M_{2,j}$. Let $(x_{(j)})_{j\in[\![d]\!]}$ denotes the $(x_j)_{j\in[\![d]\!]}$ ordered in a decreasing order. Next, we introduce the effective sparsity s^* of the vector x. Lemma F.3 states that there exists $s^* \in [\![d]\!]$ such that

$$\|x\|_2^2 \leq s^* \log(2d)\, x_{(s^*)}^2 \,.$$

If x had been s-sparse and if all the non-zero components of x had been equal, then we would have $s = s^*$ and the above inequality would be valid (without the $\log(2d)$). Here, the virtue of the effective sparsity s^* is that it is defined for arbitrary vectors x. Then, we denote

$$x_* := x_{(s^*)} \,, \qquad \Delta_*^2 := s^* x_*^2 \,, \qquad \text{and} \quad \mathcal{S}^* := \{i \in [\![d]\!] : x_i \geq x_*\} \,,$$

where Δ_*^2 plays the role of the square norm x at the scale s^* and $\mathcal{S}^*$ is the corresponding set of size s^* of coordinates larger or equal to x_*.

In order to prove Theorem 4.1, we will follow three steps:

- Part 1: In Lemma B.3, we show that Algorithm 1 outputs $\hat{k} \in \{\texttt{null}, 1\}$ with probability at least $1 - \delta$.
- Part 2: In Lemma B.8 we consider Algorithm 2 with input (δ, s, h). If $h < x_*$ and $s < s^*$, then, with probability at least $1 - \log(d)\delta$, the output of the algorithm is $\hat{k} = 1$.
- Part 3: As a conclusion, we will gather previous lemmas to derive a bound on the total number of queries until the termination of Algorithm 1.

.

Part 1: Let us start by introducing the following concentration result for the empirical means computed in Algorithm 2. In the lemma below, the set S is considered as fixed. More generally, S can be any random set independent of the samples used to compute the means $(\hat{\mu}_{j,T})_{j,T}$

Lemma B.1. *Let $T \geq 1$, fix $j \in [\![d]\!]$ and $S \subset [\![d]\!]$. We have*

$$\mathbb{P}\left(\frac{1}{|S|}\sum_{j\in S}\hat{\mu}_{j,T}^{(21)} > \sqrt{\frac{2\log(1/\delta)}{|S|T}}\right) \leq \delta,$$

where $\hat{\mu}_{j,T}^{(21)} = \frac{1}{T}\sum_{u=1}^{T}\left(X_{2,j}^{(u)} - X_{1,j}^{(u)}\right)$.

Proof. The proof is a straightforward consequence of Chernoff's inequality. Recall that the variable $X_{2,j} - X_{1,j}$ is $\sqrt{2}$-sub-Gaussian for any $j \in S$. Moreover, all samples used in the sum $\sum_{j\in S}\hat{\mu}_{j,T}^{(21)}$ are independent. We have

$$\mathbb{P}\left(\frac{1}{|S|}\sum_{j\in S}\hat{\mu}_{j,T}^{(21)} > \sqrt{\frac{2\log(1/\delta)}{|S|T}}\right) \leq \mathbb{P}\left(\frac{1}{|S|}\sum_{j\in S}\hat{\mu}_{j,T}^{(21)} + \frac{1}{|S|}\sum_{j\in S}x_j > \sqrt{\frac{2\log(1/\delta)}{|S|T}}\right) \leq \delta,$$

where the first inequality follows from the fact $x_j \geq 0$ (recall that the optimal expert is 1), and the second is a direct consequence of Chernoff's concentration inequality.

□

Lemma B.2. *Consider Algorithm 2 with input (δ, s, h). We have*

$$\mathbb{P}\Big(\hat{k} = 2\Big) \leq 1.75\log(d)\delta.$$

Proof. We have

$$\mathbb{P}\Big(\hat{k} = 2\Big) = \mathbb{P}\left(\exists \ell \leq \log_{4/3}(d/s) : \hat{\mu}_{\ell}^{(21)} > \sqrt{\frac{2\log(2/\delta)}{n_0\phi}}\right) \leq \log_{4/3}(d/s)\frac{\delta}{2} \leq 1.75\log(d)\delta\ ,$$

where we used Lemma B.1. □

Lemma B.3. *Consider Algorithm 1 with input (δ, ϵ). The probability of outputting the wrong result satisfies*

$$\mathbb{P}\Big(\hat{k} = 2\Big) \leq 0.6\delta\ .$$

Proof. When Algorithm 1 with input δ halts, we denote $\hat{k}$ its returned ranking. For $\rho \geq 1$ and $r \leq \rho$, denote $\hat{k}_{\rho,r}$ the output of Algorithm 2 with input (δ_ρ, s_r, h_r). We have

$$\mathbb{P}\Big(\hat{k} = 2\Big) = \mathbb{P}\Big(\exists \rho \geq 1, \exists r \leq \rho : \hat{k}_{\rho,r} = 2\Big) \leq \sum_{\rho=1}^{\infty}\sum_{r=0}^{\rho}\mathbb{P}\Big(\hat{k}_{\rho,r} = 2\Big)\ .$$

Using Lemma B.2, we have

$$\mathbb{P}\Big(\hat{k} = 2\Big) \leq \sum_{\rho=1}^{\infty}\sum_{r=0}^{\rho}1.75\log(d)\frac{\delta}{10\log(d)\rho^3} \leq \frac{1.75\delta}{10}\sum_{\rho=1}^{\infty}\frac{\rho+1}{\rho^3} \leq 0.6\delta\ .$$

□

Part 2: To ease notation, we write S_ℓ instead of $S_\ell^{(12)}$ in the remainder of this proof. Introduce the following notation

$$\mathcal{T}^* := \{j \in [\![d]\!] : x_j^{(12)} \geq \frac{1}{2}x_*\}\,; \qquad M_\ell := \left|\{j \in S_\ell \cap \mathcal{S}^* : \hat{\mu}_{j,\ell}^{(12)} \geq \frac{3}{4}x_*\}\right|\,;$$

$$N_\ell := \left|\{j \in S_\ell \setminus \mathcal{T}^* : \hat{\mu}_{j,\ell}^{(12)} \geq \frac{3}{4}x_*\}\right|\,; \qquad q_\ell := \exp(-2^\ell);$$

$$\xi_\ell := \left\{\frac{|S_\ell \cap \mathcal{S}^*|}{|S_\ell|} \geq \frac{s^*}{2d}\left(\frac{4}{3}\right)^{\ell-1}\right\}\,; \qquad \xi'_{\ell+1} := \left\{\frac{|S_{\ell+1} \cap \mathcal{S}^*|}{|S_{\ell+1}|} \geq \frac{4}{3}\frac{|S_\ell \cap \mathcal{S}^*|}{|S_\ell|}\right\}\,;$$

$$\Theta_\ell := \left\{\frac{|S_\ell \cap (\mathcal{T}^* \setminus \mathcal{S}^*)|}{|S_\ell|} < \frac{1}{4}\right\}\,; \qquad \Lambda_\ell := \left\{\frac{|S_\ell \cap \mathcal{S}^*|}{|S_\ell|} < \frac{1}{4}\right\}\,.$$

While the set $\mathcal{S}^*$ (introduced earlier) stands for the collection of coordinates that are larger or equal to x_*, the set $\mathcal{T}^*$ contains all moderate coordinates. The other new notation deal with the constant with the ℓ-th iteration in the algorithm. M_ℓ corresponds to the set of 'significant' coordinates that lie in S_ℓ and such that the empirical mean is large enough, while N_ℓ is the set of 'small' coordinates that are in S_ℓ and whose empirical mean is large. The events ξ_ℓ, ξ'_ℓ, Θ_ℓ, and Λ_ℓ are discussed later. The following lemma states the set S_1 at step $\ell = 1$ contains a non-vanishing proportion of significant coordinates.

Lemma B.4. *Consider Algorithm 2 with input (δ, s, h). Suppose that $s < s^*$. Recall that S_1 is the set of sampled questions. We have*

$$\mathbb{P}\Big[|S_1 \cap \mathcal{S}^*| \geq \frac{s_*}{2d}|S_1|\Big] \geq 1 - \delta\,.$$

Proof. By definition, $|S_1 \cap S^*|$ follows a binomial distribution with parameters $(|S_1|, s^*/d)$. Hence, Chernoff's inequality (Lemma F.4 with $\kappa = 1/2$) enforces that

$$\mathbb{P}\Big(|S_1 \cap \mathcal{S}^*| \leq \frac{s^*}{2d}|S_1|\Big) \leq \exp\Big(-\frac{s^*}{8d}|S_1|\Big) \leq \exp\Big(-\frac{s^*}{s}\log(1/\delta)\Big) \leq \delta\,,$$

where we used $|S_1| = \phi \geq 26\log(1/\delta)d/s$ and $s < s^*$. □

The next lemma roughly states that, provided the event ξ_ℓ holds, then, with high probability, S_{l+1} will contain a larger proportion of significant questions. More precisely, we prove that the number M_ℓ of significant questions with a large empirical mean is high and the number S_ℓ of non-significant questions with a large empirical mean is small.

Lemma B.5. *Consider Algorithm 2 with inputs (δ, s, h) such that $s < s^*$ and $h < x_*$. For any $\ell \geq 1$, we have*

$$\mathbb{P}\Big(M_\ell \leq \frac{2}{3}|S_\ell \cap \mathcal{S}^*|\Big|\xi_\ell\Big) \leq \delta\,; \qquad \mathbb{P}\Big(N_\ell \geq \frac{1}{4}|S_\ell|\Big|\xi_\ell\Big) \leq \delta\,.$$

Proof. Define $\bar{M}_\ell := |S_\ell \cap \mathcal{S}^*|$. Using the definition of M_ℓ, we have

$$\bar{M}_\ell = \sum_{j \in S_\ell \cap \mathcal{S}^*} \mathbb{1}\Big(\hat{\mu}_{j,\ell}^{(12)} < \frac{3}{4}x_*\Big).$$

Let $j \in S_\ell \cap \mathcal{S}^*$, recall that $\mathbb{E}[\hat{\mu}_{j,\ell}^{(12)}] \geq x_*$, and the samples $(X_{1,j}^{(u)} - X_{2,j}^{(u)})$ are $\sqrt{2}$-sub-Gaussian. Therefore, using Chernoff's bound

$$\mathbb{P}\Big(\hat{\mu}_{j,\ell}^{(12)} < \frac{3}{4}x_*\Big) \leq \exp\Big(-t_\ell\frac{x_*^2}{32}\Big) = \exp\Big(-2^{\ell-1}n_0\frac{x_*^2}{32}\Big) \leq \exp(-2^\ell) = q_\ell\,,$$

since $h \leq x_*$ and by definition of ℓ and of q_ℓ. Thus, the variables $\mathbb{1}(\hat{\mu}_{j,\ell}^{(12)} < \frac{3}{4}\Delta^*)$ for $j \in S_\ell \cap \mathcal{S}^*$ are stochastically dominated by $\mathcal{B}(q_\ell)$ and independent. Therefore $\bar{M}_\ell$ is stochastically dominated by $\mathcal{B}(|S_\ell \cap \mathcal{S}^*|, q_\ell)$, where $\mathcal{B}(a, b)$ denotes the binomial distribution with parameters (a, b). Let $\kappa := 1/(3q_\ell) - 1 > 0$, we have

$$\begin{aligned}\mathbb{P}\Big(M_\ell \leq \frac{2}{3}|S_\ell \cap \mathcal{S}^*|\Big|\xi_\ell\Big) &= \mathbb{P}\Big(\bar{M}_\ell > \frac{1}{3}|S_\ell \cap \mathcal{S}^*|\Big|\xi_\ell\Big)\\ &= \mathbb{P}\Big(\bar{M}_\ell > (1+\kappa)q_\ell|S_\ell \cap \mathcal{S}^*|\Big|\xi_\ell\Big)\end{aligned}$$

Using Lemma F.4 and the definition of the event ξ_ℓ

$$\mathbb{P}\Big(M_\ell \le \frac{2}{3}|S_\ell \cap \mathcal{S}^*|\Big|\xi_\ell\Big) \le \exp\{\kappa q_\ell |S_\ell \cap \mathcal{S}^*| - (1+\kappa) q_\ell |S_\ell \cap \mathcal{S}^*| \log(1+\kappa)\}$$
$$\le \exp\Big\{\Big(\frac{1}{3} - q_\ell - \frac{1}{3}\log\Big(\frac{1}{3q_\ell}\Big)\Big)|S_\ell \cap \mathcal{S}^*|\Big\} .$$

Recall that $q_\ell = \exp(-2^\ell)$ so that $\kappa > 1/5$. Using the expression $|S_\ell| = \frac{|S_1|}{2^{\ell-1}} \ge \frac{26d\log(1/\delta)}{2^{\ell-1}s}$, the definition of the event ξ_ℓ, and $s > s^*$, we deduce that

$$\mathbb{P}\Big(M_\ell \le \frac{2}{3}|S_\ell \cap \mathcal{S}^*|\Big|\xi_\ell\Big) \le \exp\Big\{\Big(\frac{1}{3} - \exp(-2^\ell) - \frac{2^l - \log(3)}{3}\Big)\Big(\frac{2}{3}\Big)^{\ell-1} 13\log(1/\delta)\Big\}.$$

For $\ell \ge 3$, we have $2^{\ell-1} \ge \log(3) + 3$, which leads to

$$\mathbb{P}\Big(M_\ell \le \frac{2}{3}|S_\ell \cap \mathcal{S}^*|\Big|\xi_\ell\Big) \le \exp\Big\{-\Big(\frac{4}{3}\Big)^{\ell-1} 13\log(1/\delta)\Big\} \le \delta .$$

One easily check that this bound is still true for $\ell = 1$, and $\ell = 2$. We have proved the first part of the lemma.

For the second result, we start from

$$N_\ell = \sum_{j \in S_\ell \setminus \mathcal{T}^*} \mathbb{1}\Big(\hat{\mu}_{j,\ell}^{(12)} \ge \frac{3}{4}x_*\Big).$$

Arguing as in the first part, we easily check that the variables $\mathbb{1}(\hat{\mu}_{j,\ell}^{(12)} \ge \frac{3}{4}x_*)$, for $j \in S_\ell \setminus \mathcal{T}^*$ are independent and stochastically dominated by $\mathcal{B}(q_\ell)$. Hence N_ℓ is stochastically dominated by $\mathcal{B}(|S_\ell|, q_\ell)$. Let $\kappa' = \frac{1}{4q_\ell} - 1 \ge 0$. Using again Lemma F.4, we have

$$\mathbb{P}\Big(N_\ell \ge \frac{1}{4}|S_\ell|\Big|\xi_\ell\Big) = \mathbb{P}\Big(N_\ell \ge (1+\kappa)q_\ell|S_\ell|\Big|\xi_\ell\Big)$$
$$\le \exp\{\kappa q_\ell |S_\ell| - (1+\kappa) q_\ell |S_\ell| \log(1+\kappa)\}$$
$$\le \exp\Big\{\Big(\frac{1}{4} - q_\ell - \frac{1}{4}\log\Big(\frac{1}{4q_\ell}\Big)\Big)|S_\ell|\Big\}.$$

Next, we use the expression of $|S_\ell|$ and obtain

$$\mathbb{P}\Big(N_\ell \ge \frac{1}{4}|S_\ell|\Big|\xi_\ell\Big) \le \exp\Big\{\Big(\frac{1}{4} - \exp(-2^\ell) - \frac{1}{4}\log\Big(\frac{\exp(2^\ell)}{4}\Big)\Big)\frac{26d\log(1/\delta)}{s2^{\ell-1}}\Big\} .$$

Recall that $d \ge s$. As in the first part of the proof, we easily check that $\ell = 1$ and $\ell = 2$ the above expression is smaller or equal to δ. For $\ell \ge 3$, we have $2^{l-1} \ge 1 + \log(4)$, which implies also that

$$\mathbb{P}\Big(N_\ell \ge \frac{1}{4}|S_\ell|\Big|\xi_\ell\Big) \le \exp\Big[-\frac{2^{\ell-1}}{4} \cdot \frac{26}{2^{\ell-1}}\log(1/\delta)\Big] \le \delta .$$

□

The event Λ_ℓ and Θ_ℓ respectively state that the proportion of significant (and moderately significant) questions in $|S_\ell|$ is smaller or equal to $1/4$. The following lemma roughly states that as long ξ_ℓ, $\Theta_{\ell+1}$, and $\Lambda_{\ell+1}$, then the proportion of significant questions in $S_{\ell+1}$ is significantly reduced.

Lemma B.6. *Let any integer $\ell \ge 1$, we have*

$$\mathbb{P}\Big(\Theta_{\ell+1} \textit{ and } \Lambda_{\ell+1} \textit{ and } \neg\xi'_{\ell+1}\Big|\xi_\ell\Big) \le 2\delta \ .$$

Proof. Recall that $m_\ell^{(12)}$ denotes the median computed in Algorithm 2, we will denote m_ℓ instead of $m_\ell^{(12)}$ in this proof for the sake of simplicity. We have

$$\mathbb{P}\Big(\Theta_{\ell+1} \text{ and } \Lambda_{\ell+1} \text{ and } \neg\xi'_{\ell+1}\Big|\xi_\ell\Big) = \mathbb{P}\Big(\Theta_{\ell+1} \text{ and } \Lambda_{\ell+1} \text{ and } \neg\xi'_{\ell+1} \text{ and } \{m_\ell < \frac{3}{4}x_*\}\Big|\xi_\ell\Big) + \mathbb{P}\Big(\Theta_{\ell+1} \text{ and } \Lambda_{\ell+1} \text{ and } \neg\xi'_{\ell+1} \text{ and } \{m_\ell \geq \frac{3}{4}x_*\}\Big|\xi_\ell\Big). \tag{3}$$

Upper bound for the first term in the rhs of (3). The event $m_\ell < (3/4)x_*$ implies that $\{j \in S_\ell \cap \mathcal{S}^* : \hat{\mu}_{j,\ell}^{(12)} \geq \frac{3}{4}x_*\} \subset S_{\ell+1}$. Hence,

$$\Big\{j \in S_\ell \cap \mathcal{S}^* : \hat{\mu}_{j,\ell}^{(12)} \geq \frac{3}{4}x_*\Big\} \subset S_{\ell+1} \cap \mathcal{S}^*,$$

which gives $M_\ell \leq |S_{\ell+1} \cap \mathcal{S}^*|$. This leads us to

$$\begin{aligned}
\mathbb{P}\Big(\Theta_{\ell+1} \text{ and } \Lambda_{\ell+1} \text{ and } \neg\xi'_{\ell+1} \text{ and } \{m_\ell < \frac{3}{4}x_*\}\Big|\xi_\ell\Big) &\leq \mathbb{P}\Big(\neg\xi'_{\ell+1} \text{ and } \{m_\ell < \frac{3}{4}x_*\}\Big|\xi_\ell\Big) \\
&= \mathbb{P}\Big(\{|S_{\ell+1} \cap \mathcal{S}^*| \leq \frac{2}{3}|S_\ell \cap \mathcal{S}^*|\} \text{ and } \{m_\ell < \frac{3}{4}x_*\}\Big|\xi_\ell\Big) \\
&\leq \mathbb{P}\Big(\{M_\ell < \frac{2}{3}|S_\ell \cap \mathcal{S}^*|\} \text{ and } \{m_\ell < \frac{3}{4}x_*\}\Big|\xi_\ell\Big) \\
&\leq \mathbb{P}\Big(M_\ell < \frac{2}{3}|S_\ell \cap \mathcal{S}^*|\Big|\xi_\ell\Big) \leq \delta\,,
\end{aligned}$$

where we used Lemma B.5 in the last line.

Upper bound for the second term in the rhs of (3). The event $\Theta_{\ell+1}$ implies in particular that

$$|S_{\ell+1} \cap \mathcal{S}^*| + |S_{\ell+1} \setminus \mathcal{T}^*| \geq \frac{3}{4}|S_{\ell+1}|.$$

Moreover, we have that the event $\{m_\ell \geq \frac{3}{4}x_*\}$ implies that

$$\Big\{j \in S_\ell \setminus \mathcal{T}^* : \hat{\mu}_{j,\ell}^{(12)} \geq \frac{3}{4}x_*\Big\} \supset S_{\ell+1} \setminus \mathcal{T}^*,$$

which gives $|S_{\ell+1} \cap \mathcal{S}^*| + N_\ell \geq |S_{\ell+1} \cap \mathcal{S}^*| + |S_{\ell+1} \setminus \mathcal{T}^*| \geq \frac{3}{4}|S_{\ell+1}|$. We conclude that

$$\begin{aligned}
\mathbb{P}\Big(\Theta_{\ell+1} \text{ and } \Lambda_{\ell+1} \text{ and } \neg\xi'_{\ell+1} \text{ and } \{m_\ell \geq \frac{3}{4}x_*\}\Big|\xi_\ell\Big) &\leq \mathbb{P}\Big(\Lambda_{\ell+1} \text{ and } \{|S_{\ell+1} \cap \mathcal{S}^*| + N_\ell \geq \frac{3}{4}|S_{\ell+1}|\}\Big|\xi_\ell\Big) \\
&\leq \mathbb{P}\Big(N_\ell \geq \frac{1}{2}|S_{\ell+1}|\Big|\xi_\ell\Big) = \mathbb{P}\Big(N_\ell \geq \frac{1}{4}|S_\ell|\Big|\xi_\ell\Big) \leq \delta,
\end{aligned}$$

where we used the definition of $\Lambda_{\ell+1}$ and Lemma B.5 in the second line. We conclude that

$$\mathbb{P}\Big(\Theta_{\ell+1} \text{ and } \Lambda_{\ell+1} \text{ and } \neg\xi'_{\ell+1}\Big|\xi_\ell\Big) \leq 2\delta\,.$$

□

Consider Algorithm 2 with input (δ, s, h). Let $\bar{\ell} := \lceil \log_{4/3}(d/s^*) \rceil + 1$. Introduce the following event:

$$E_\ell := \{\text{Algorithm 2 made at least } \ell \text{ iterations}\} \text{ and } \bigcap_{u=1}^{u=\ell} (\Theta_u \cap \Lambda_u). \tag{4}$$

Lemma B.7. *Consider Algorithm 2 with input (δ, s, h). Suppose that $h < x_*$ and $s < s^*$. We have*

$$\mathbb{P}(E_{\bar{\ell}}) \leq 2\bar{\ell}\delta.$$

Proof. First, let us prove by induction that, for any $\ell \leq \bar{\ell}$, we have

$$\mathbb{P}(\neg\xi_\ell \text{ and } E_\ell) \leq 2\ell\delta.$$

The assertion for $\ell = 1$ is a direct consequence of Lemma B.4. Suppose that the result for some $\ell \leq \bar{\ell}$. Observe that, by definition, the event $\neg\xi_{\ell+1}$ implies that either we have $\neg\xi'_{\ell+1}$ or we have $\neg\xi_\ell$. This leads us to

$$\begin{aligned}
\mathbb{P}(\neg\xi_{\ell+1} \text{ and } E_{\ell+1}) &\leq \mathbb{P}\big(\neg\xi'_{\ell+1} \text{ and } E_{\ell+1}\big) + \mathbb{P}(\neg\xi_\ell \text{ and } E_{\ell+1}) \\
&\leq \mathbb{P}\big(\neg\xi'_{\ell+1} \text{ and } \Lambda_{\ell+1} \text{ and } \Theta_{\ell+1}\big) + \mathbb{P}(\neg\xi_\ell \text{ and } E_\ell) \\
&\leq 2\delta + 2\ell\delta,
\end{aligned}$$

where we used Lemma B.6 and induction hypothesis in the last line.

Observe that

$$\neg\xi_{\bar{\ell}} = \left\{ \frac{|S_{\bar{\ell}} \cap \mathcal{S}^*|}{|S_{\bar{\ell}}|} < \frac{s^*}{2d}\left(\frac{4}{3}\right)^{\bar{\ell}-1} \right\}; \qquad \Lambda_{\bar{\ell}} = \left\{ \frac{|S_{\bar{\ell}} \cap \mathcal{S}^*|}{|S_{\bar{\ell}}|} < \frac{1}{4} \right\}.$$

Recall that $\bar{\ell} = \lceil \log_{4/3}(d/s^*) \rceil + 1$, which implies that $\frac{s^*}{2d}\left(\frac{4}{3}\right)^{\bar{\ell}} \geq \frac{1}{4}$. Hence, we have $\neg\xi_{\bar{\ell}} \subset \Lambda_{\bar{\ell}}$ and we conclude that $\mathbb{P}(E_{\bar{\ell}}) = \mathbb{P}(\neg\xi_{\bar{\ell}} \text{ and } E_{\bar{\ell}}) \leq 2\ell\delta$. □

The following result is built upon the previous lemmas and states that Algorithm 2 returns `null` with a very small probability provided that h and s are small enough.

Lemma B.8. *Consider Algorithm 2 with input (δ, s, h). Suppose that $h < x_*$ and $s < s^*$. We have*

$$\mathbb{P}\Big(\{\hat{k} = \texttt{null}\}\Big) \leq 3\bar{\ell}\delta.$$

Proof. We have:

$$\begin{aligned}
\mathbb{P}\Big(\{\hat{k} = \texttt{null}\}\Big) &= \mathbb{P}\Big(\{\hat{k} = \texttt{null}\} \text{ and } \neg E_{\bar{\ell}}\Big) + \mathbb{P}\Big(\{\hat{k} = \texttt{null}\} \text{ and } E_{\bar{\ell}}\Big) \\
&\leq \mathbb{P}\Big(\{\hat{k} = \texttt{null}\} \text{ and } \neg E_{\bar{\ell}}\Big) + 2\bar{\ell}\delta,
\end{aligned}$$

where we used Lemma B.7. Now Suppose $E_{\bar{\ell}}$ is false. Hence, there exists an iteration $\ell \leq \bar{\ell}$ such that either Θ_ℓ is false or Λ_ℓ is false. Recall that $\{\hat{k} = \texttt{null}\}$ implies that $\hat{\mu}_\ell^{(12)} < \sqrt{\frac{2\log(2/\delta)}{n_0\phi}}$ (otherwise, the algorithm halts at iteration ℓ and outputs $\hat{k} \in \{1, 2\}$). Then

$$\begin{aligned}
\mathbb{P}\Big(\{\hat{k} = \texttt{null}\} \text{ and } \neg E\Big) &\leq \sum_{\ell=1}^{\bar{\ell}} \mathbb{P}\left(\left\{ \hat{\mu}_\ell^{(12)} < \sqrt{\frac{2\log(2/\delta)}{n_0\phi}} \right\} \text{ and } \{\neg\Theta_\ell \cup \neg\Lambda_\ell\} \right) \\
&= \sum_{\ell=1}^{\bar{\ell}} \mathbb{P}\left(\left\{ \frac{1}{|S_\ell|}\sum_{j\in S_\ell} x_j - \hat{\mu}_\ell^{(12)} > \frac{1}{|S_\ell|}\sum_{j\in S_\ell} x_j - \sqrt{\frac{2\log(2/\delta)}{n_0\phi}} \right\} \text{ and } \{\neg\Theta_\ell \cup \neg\Lambda_\ell\} \right) \\
&\leq \sum_{\ell=1}^{\bar{\ell}} \mathbb{P}\left(\frac{1}{|S_\ell|}\sum_{j\in S_\ell} x_j - \hat{\mu}_\ell^{(12)} > \frac{1}{8}x_* - \sqrt{\frac{2\log(2/\delta)}{n_0\phi}} \right) \\
&\leq \sum_{\ell=1}^{\bar{\ell}} \mathbb{P}\left(\frac{1}{|S_\ell|}\sum_{j\in S_\ell} x_j - \hat{\mu}_\ell^{(12)} > \sqrt{\frac{2\log(2/\delta)}{n_0\phi}} \right) \leq \bar{\ell}\delta\,,
\end{aligned}$$

where we used the definition of Θ_ℓ and Λ_ℓ in the third line, the assumption $h < x_*$ and $s < s^*$ in the fourth line and Chernoff's bound in the last line. □

Part 3: Here, we gather all the previous lemmas to establish the three claims of the theorem.

Conclusion for the first claim: Lemma B.3 leads to the first claim of Theorem 4.1.

Conclusion for the second claim: Let us prove that if $\epsilon^2 < d/H$ then Algorithm 1 with input (δ, ϵ) outputs $\hat{k} = 1$ with probability at least $1 - \delta$. By Lemma B.3, we have $\mathbb{P}(\hat{k} = 2) \leq 0.6\delta$. Hence, it suffices to prove that $\mathbb{P}(\hat{k} = \texttt{null}) \leq 0.4\delta$. Define

$$\rho^* := \left\lfloor \log_2\left(\frac{16d}{s^* x_*^2}\right)\right\rfloor \qquad \text{and} \qquad r^* := \left\lceil \log_2\left(\frac{1}{x_*^2}\right)\right\rceil . \tag{5}$$

By assumption, we have $x_*^2 \leq 1$, so that $r^* \geq 1$ and $\rho^* \geq 1$. Recall that $\epsilon^2 < \frac{d}{H} \leq \log(2d) s^* x_*^2 \leq 16 \log(2d) d\, 2^{-\rho^*}$. If Algorithm 1 returns $\hat{k} = \texttt{null}$, then it implies in particular that Algorithm 2 returns $\hat{k} = \texttt{null}$ at the iteration ρ^* and $r^* \leq \rho^*$. At this step, the inputs $(\delta_{r^*}, s_{r^*}, h_{r^*})$ of Algorithm 2 satisfy $\delta_{r^*} \leq \delta/(10 r^{*3} \log(d))$, $h_{r^*} = 1/\sqrt{2^{r^*}} \leq x_*$, and $s_{r^*} = 2^{r^*} d/2^{\rho^*} \leq s^*$. We then deduce from Lemma B.8 that $\hat{k} = \texttt{null}$ with probability less or equal to $\frac{3}{10 r^{*3} \log(d)}(\lceil \log_{4/3}(d/s^*)\rceil + 1)\delta \leq 0.4\delta$. We conclude that $\mathbb{P}[\hat{k} = 1] \geq 1 - \delta$.

Conclusion for the third claim: The total number of queries made by Algorithm 2 with inputs (δ, s, h) is at most

$$N_{s,h} = 4096 \frac{\log_{4/3}(16d/(9s)) \log(1/\delta)}{sh^2}.$$

Consider Algorithm 1 with input (δ, ϵ). Suppose that $\epsilon^2 \geq d/H$, therefore the maximum number of iterations is less than $\rho' = \lfloor \log_2(4 \log(2d) d/\epsilon^2)\rfloor$. Therefore the total number of queries satisfies:

$$\begin{aligned}
N &\leq \sum_{\rho=1}^{\rho'} \sum_{r=0}^{\rho} N_{s_r, h_r} \\
&\leq 4096[\log_{4/3}(d) + 2] \sum_{\rho=1}^{\rho'} \sum_{r=0}^{\rho} \frac{\log(1/\delta_r)}{s_r h_r^2} \\
&\leq 10^4 \log(d) \log\left(\frac{10 \log(d) \rho'^3}{\delta}\right) \sum_{\rho=1}^{\rho'} (\rho + 1) 2^{\rho} \\
&\leq 8.10^4 \log^2(2d) \log_2\left(\frac{\log(8d \log(d))}{\epsilon^2}\right) \log\left(\frac{10 \log(d) \rho'^3}{\delta}\right) \frac{d}{\epsilon^2} \\
&\leq c' \log^2(2d) \log\left(\frac{\log(2d)}{\epsilon^2}\right) \log\left(\frac{2 \log(2d/\epsilon^2)}{\delta}\right) \frac{d}{\epsilon^2} ,
\end{aligned}$$

where c' is a numerical constant.

Now suppose that $\epsilon^2 < d/H$. We have shown previously that, with probability at least $1 - \delta$, the algorithm stops no later

than at iteration ρ^* and r^* (defined in (5)). Under such an event, the total number of iterations satisfies

$$
\begin{aligned}
N &\leq \sum_{\rho=1}^{\rho^*-1} \sum_{r=0}^{\rho} N_{s_r,h_r} + \sum_{r=0}^{r^*} N_{s_r,h_r} \\
&\leq 4096[\log_{4/3}(d)+2] \left(\sum_{\rho=1}^{\rho^*-1} \sum_{r=0}^{\rho} \frac{\log(1/\delta_r)}{s_r h_r^2} + \sum_{r=0}^{r^*} \frac{\log(1/\delta_r)}{s_r h_r^2} \right) \\
&\leq 10^4 \log(d) \log\left(\frac{10 \log(d) \rho^{*3}}{\delta} \right) \sum_{\rho=1}^{\rho^*} (\rho+1) 2^{\rho} \\
&\leq 10^4 \log(d) \log\left(\frac{10 \log(d) \rho^{*3}}{\delta} \right) (\rho^*+1) 2^{\rho^*+1} \\
&\leq 8 \cdot 10^4 \log(d) \log\left(\frac{10 \log(d)}{\delta} \log_2^3\left(\frac{4d}{s^* x_*^2} \right) \right) \log_2\left(\frac{8d}{s^* x_*^2} \right) \frac{d}{s^* x_*^2} \\
&\leq c' \log^2(d) \log(H) \log\left(\frac{\log(H)\log(d)}{\delta} \right) H \ ,
\end{aligned}
$$

where c' is a numerical constant and where we used that $\|x\|_2^2 \leq s^* \log(2d) x_{s^*}^2$.

C. Full Ranking

C.1. Binary search algorithm

Algorithm 5 `Binary-search`(δ, i, r, a, b)

Input: δ, r, i expert to be inserted, start, end (start/end of array r).
if start $=$ end **then**
  if `compare`$(\delta, 0, i, r[\text{start}]) = i$ **then**
    Output: start $+ 1$.
  else
    Output: start.
  end if
end if
if start $>$ end **then**
  Output: start.
end if
Let mid $\leftarrow \lfloor (\text{start} + \text{end})/2 \rfloor$
if `compare`$(\delta, 0, i, r[\text{mid}]) = i$ **then**
  Output: `Binary-search`$(\delta, i, r, \text{mid}+1, \text{end})$.
else if `compare`$(\delta, 0, i, r[\text{mid}]) = r[\text{mid}]$ **then**
  Output: `Binary-search`$(\delta, i, r, \text{start}, \text{mid}-1)$.
else
  Output: mid.
end if

C.2. Proof of Theorem 5.1

For $i \in [\![n-1]\!]$, define:

$$
H_i := \frac{d}{\sum_{j=1}^{d} (M_{\pi(i),j} - M_{\pi(i+1),j})^2} \ .
$$

By convention, we define $H_0 = 0$. Binary insertion sort procedure makes at most $n\lceil \log_2(n) \rceil$ comparisons, hence using an union bound, we conclude that all calls to `compare` algorithm output a correct result with probability at least $1-\delta$.

Moreover, inserting any expert i costs at most $\lceil \log_2(n) \rceil$ calls to `compare`. For any $k \in [\![n]\!] \setminus \{i\}$, we apply Theorem 4.1. Hence, the total number of queries made by `Binary-search`$(\delta/(n\lceil \log_2(n) \rceil), i, r, 1, i-1)$ is upper bounded by

$$c \ \log^2(d) \log(n) \log(H_i \vee H_{i-1}) \log[n \log(d) \log(H_i \vee H_{i-1})/\delta](H_i \vee H_{i-1})$$

with probability at least $1 - \delta/n$. Here, c stands for a numerical constant. Summing for $i \in [\![n-1]\!]$ gives the desired result.

D. Best expert identification

D.1. Max search algorithm

The max-search routine with precision ϵ is described in Algorithm 6 below.

Algorithm 6 `Max-search`(δ, S, ϵ)

Input: δ confidence parameter, S a set of experts, ϵ a precision parameter.
Output: Set C of experts.
Let $\ell = |S|$ and $a_1, \dots, a_\ell$ be the elements of S.
$C \leftarrow \{a_1\}$, $m \leftarrow a_1$.
for $r = 2, \dots, \ell$ **do**
  if `compare`$(\frac{\delta}{2|S|}, \epsilon, m, a_r) =$ `null` **then**
    $C \leftarrow C \cup \{a_r\}$.
  else if `compare`$(\frac{\delta}{2|S|}, \epsilon, m, a_r) = a_r$ **then**
    $C \leftarrow \{a_r\}$, $m \leftarrow a_r$.
  end if
end for
for $i \in C \setminus \{m\}$ **do**
  if `compare`$(\frac{\delta}{2|S|}, \epsilon, m, i) = m$ **then**
    $C \leftarrow C \cup \{i\}$
  end if
end for
Output: C.

D.2. Proof of Theorem 5.2

Let $i^* \in [\![d]\!]$ denote the optimal expert. For $\epsilon \in (0,1)$, define $B_\epsilon \subset [\![d]\!]$ as follows:

$$B_\epsilon := \{i \in [\![n]\!] : \|M_{i^*} - M_i\|_2^2 \leq \epsilon^2\}.$$

Lemma D.1. *Consider Algorithm 6 with input (δ, S, ϵ) such that $\epsilon \leq 1$ and $i^* \in S$. Denote C its output. With probability at least $1 - \delta$, we have $i^* \in C$ and $C \subseteq B_{2\epsilon}$.*

Proof. Fix $\delta \in (0,1)$ and $\epsilon \in (0,1)$. To ease notation, we denote $R_1(a,b)$ (resp. $R_2(a,b)$), the output of `compare`(δ, ϵ, a, b) for $a \in S$ and $b \in S$ in the first (resp. second) loop of Algorithm 6. Also, we we write $\hat{m}$ the element with which items are compared in the second loop of Algorithm 6.

Using Theorem 4.1, we have that, on an event of probability $1 - \delta$, all the results of $R_1(m,i)$ and $R_2(m,i)$ in Algorithm 6 are such that, for $s = 1, 2$, $R_s(m,i) \in \{i, \texttt{null}\}$ if i is above m, $R_s(m,i) \in \{m, \texttt{null}\}$ if m is above i, and $R_s(m,i) \neq \texttt{null}$ if $\|M_m - M_i\|_2 \geq \epsilon$.

Let us show that, under this event, we have $i^* \in C$ and $C \subseteq B_{2\epsilon}$. Indeed, if $i^* \notin C$, this implies that in the second loop we had $R_2(\hat{m}, i^*) = \hat{m}$, which is not possible by definition.

Besides, we easily check that $\hat{m}$ satisfies $\|M_{\hat{m}} - M_i\|_2 \leq \epsilon$. Since C only contains the elements j such that we have found $R_s(\hat{m}, j) \in \{j, \mathtt{m}\}$ for $s = 1$ or $s = 2$, this implies that either j is above $\hat{m}$ or that $\|M_{\hat{m}} - M_i\|_2 \leq \epsilon$. By triangular inequality, we have proved that $C \subseteq B_{2\epsilon}$.

□

Lemma D.2. *Consider Algorithm 6 with input (δ, S, ϵ) such that $\epsilon \leq 1$ and $i^* \in S$. Denote N_ϵ the total number of queries made. We have*

$$N_\epsilon \leq c \log^2(d) \log\left(\frac{d}{\epsilon^2}\right) \log\left(\frac{|S| \log(\frac{d}{\epsilon^2})}{\delta}\right) \frac{d|S|}{\epsilon^2},$$

where c is a numerical constant.

Proof. Observe that in Algorithm 6, there are at most $2|S|$ calls to the procedure `compare` with precision parameter ϵ. Using Theorem 4.1, we get the result. □

Conclusion: Fix $\delta \in (0,1)$ and denote N the total number number of queries made by Algorithm 4.

Write $r = 1, \dots, \hat{r}$ for the iterations of Algorithm Algorithm 4 and write S_r for the corresponding result of `Max-search` algorithm. We write $S_0 = [n]$. Applying Lemma D.1, we know that on event of probability higher than $1-\delta$, we have,

$$i \in S_r \text{ and } S_r \subseteq B_{2 \cdot 4^{-r+1}} \ . \tag{6}$$

simultaneously for all $r = 1, \dots, \hat{r}$.

We work henceforth under this event. First, this implies that $S_{\hat{r}} = \{i^*\}$. Hence, the procedure recovers the best expert.

Write $\Delta^* := \min_{i \neq i^*} \|M_i - M_{i^*}\|_2^2$ for the minimum distance between i^* and another expert. Denote $r^* = \lceil \log_4(8/\Delta^*) \rceil$. By (6), we have $\hat{r} \leq r^*$. By Lemma D.2, the total number of queries made at iteration r is no larger than

$$c \log^2(d) \log\left(\frac{d}{4^{-2r}}\right) \log\left(\frac{n 2^k \log(\frac{d}{4^{-2r}})}{\delta}\right) \frac{d|S_{r-1}|}{4^{-2r}} \ ,$$

where c is a numerical constant and $|S_{r-1}| \leq |B_{2 \cdot 4^{-r+2}}|$ for $r \geq 2$ and $|S_0| = n$.

As a consequence, the total number N of queries from iteration 2 to r^* satisfies

$$\begin{aligned}
N &\leq c' d \log^2(d) \log\left(\frac{n \log(d)}{\delta}\right) \left[n + \sum_{r=2}^{r^*} r \log(d 4^{2r}) 4^{2r} |B_{2 \cdot 4^{-r+2}}| \right] \\
&\leq c' d \log^2(d) \log\left(\frac{n \log(d)}{\delta}\right) \left[n + \sum_{i \neq i^*} \sum_{r=2}^{r^*} k \log(d 4^{2r}) 4^{2r} 1\{\|M_i - M_{i^*}\|_2 \leq 2 \cdot 4^{-r+2}\} \right] \\
&\leq c'' d \log^2(d) \log\left(\frac{n \log(d)}{\delta}\right) \left[n + \sum_{i \neq i^*} \frac{1}{\|M_i - M_{i^*}\|_2^2} \log^2\left(\frac{d}{\|M_i - M_{i^*}\|_2^2 \vee 1}\right) \right] .
\end{aligned}$$

The result follows.

E. Proofs of the lower bounds

Proof of Theorem 6.1. Fix $d \geq 1$, $\delta \in (0, 1/4)$, and $\underline{\Delta} \in (\mathbb{R}^+)^d$ such that $\|\underline{\Delta}\|^2 > 0$, and let A be a δ-accurate algorithm. Define the $2 \times d$ matrix M^* by $M^*_{1,j} = \underline{\Delta}_j$ for $j \in [\![d]\!]$ and $M^*_{2,j} = 0$. For any permutation π of $[\![2]\!]$ and any permutation σ of $[\![d]\!]$, we write $M^*_{\pi,\sigma}$ for the permuted matrix such that $(M^*_{\pi,\sigma})_{i,j} = (M^*_{\pi,\sigma})_{\pi^{-1}(i),\sigma^{-1}(j)}$. Obviously, we have $M^*_{\pi,\sigma}$ belongs to $\mathcal{M}_{n,d}$ and π is the permutation that order the rows of $M^*_{\pi,\sigma}$.

For any permutations π and σ, we write $\mathbb{P}^{(\pi,\sigma)}$ for the distribution of the data such that $X_{i,j} \sim \mathcal{N}[(M^*_{\pi,\sigma})_{i,j}, 1]$. We also introduce the 'null' distribution $\mathbb{P}^0$ such that $X_{i,j} \sim \mathcal{N}[0,1]$. There exist only 2 permutations on $[\![2]\!]$, that we respectively denote π_0 (for the identity permutation) and π_1. Since the strategy A is δ-accurate, we have, for any permutation σ of $[\![d]\!]$, that

$$\mathbb{P}^{(\pi_0,\sigma)}(\hat{\pi} \neq \pi_0) \leq \delta \text{ and } \mathbb{P}^{(\pi_1,\sigma)}(\hat{\pi} \neq \pi_1) \leq \delta \ . \tag{7}$$

Denote N the total budget of the algorithm A. Let x be the smallest integer such that

$$\max_{\sigma} \max_{z=0,1} \mathbb{P}^{(\pi_z,\sigma)}[N > x] \leq \delta$$

Next, we claim that $\mathbb{P}^0(N > x) \geq 1 - 2\delta$. Indeed, for any $\kappa > 0$, consider the $2 \times d$ matrix M' such that $M'_{1,1} = \kappa$ and $M'_{i,j} = 0$ otherwise. Write M'_{π_0} and M'_{π_1} for the matrix M' where we have permuted the rows according to π_0 and π_1. Write $\mathbb{P}'^{(\pi_0)}$ and $\mathbb{P}'^{(\pi_1)}$ for the corresponding distribution of the data. Since the strategy A is δ-accurate, we have

$$\mathbb{P}'^{(\pi_0)}(\hat{\pi} \neq \pi_0) \leq 1 - \delta \text{ and } \mathbb{P}'^{(\pi_1)}(\hat{\pi} \neq \pi_0) \leq 1 - \delta$$

As a consequence,

$$\mathbb{P}'^{(\pi_0)}(\hat{\pi} = \pi_1 \text{ and } N \leq x) + \mathbb{P}'^{(\pi_1)}(\hat{\pi} = \pi_0 \text{ and } N \leq x) \leq 2\delta$$

On the event where $N \leq x$, the distributions $\mathbb{P}'^{(\pi_0)}$ and $\mathbb{P}'^{(\pi_1)}$ converges in total variation distance towards $\mathbb{P}^0$ when κ goes to zero. This implies that $\mathbb{P}^0[N \leq x] \leq 2\delta$.

Consider the new algorithm A' defined as follows. If the total budget of A is smaller or equal to x, then it returns $\widehat{H} = H_1$, if the total budget is higher than, then it returns $\widehat{H} = H_0$". With a slight abuse of notation, we still write N for the total budget of A' and $\mathbb{P}^{(\pi,\sigma)}$ for the corresponding distributions. By definition of x, we have

$$\max_{\sigma} \max_{z=0,1} \mathbb{P}^{(\pi_z,\sigma)}[\widehat{H} = H_1] \geq 1 - \delta \text{ and } \mathbb{P}^0[\widehat{H} = H_1] \geq 1 - 2\delta \ .$$

By definition of the total variance distance TV between distributions (see Theorem 2.2 of (Tsybakov, 2004)), we derive that

$$\min_{z=0,1} \frac{1}{d!} \sum_{\sigma} \mathrm{TV}\Big(\mathbb{P}^{(\pi_z,\sigma)}, \mathbb{P}^0\Big) \geq 1 - 3\delta$$

By Lemma F.2, we control the total variation distance in terms of Kullback-Leibler discrepancy.

$$\mathrm{TV}\Big(\mathbb{P}^{(\pi,\sigma)}, \mathbb{P}^0\Big) \leq 1 - \frac{1}{2} \exp\left\{ - \sum_{i=1}^{2} \sum_{j=1}^{d} \mathbb{E}^0[N_{i,j}] \ \mathrm{KL}\Big(\mathbb{P}^{(0)}_{i,j}, \mathbb{P}^{\pi,\sigma}_{i,j}\Big) \right\}, \tag{8}$$

Under $\mathbb{P}^{\pi,\sigma}$, the distribution the (i,j)-th entry is $\mathcal{N}(M^*_{\pi^{-1}(i),\sigma^{-1}(j)}, 1)$. Fixing $\pi = \pi_0$ and averaging over all permutation σ leads to

$$\frac{1}{d!} \sum_{\sigma} \exp\left[- \sum_{j=1}^{d} \mathbb{E}^0[N_{1,j}] \underline{\Delta}^2_{\sigma^{-1}(j)} \right] \leq 6\delta$$

By Jensen's inequality, we deduce that

$$\frac{1}{d!} \sum_{\sigma} \sum_{j=1}^{d} \mathbb{E}^0[N_{1,j}] \underline{\Delta}^2_{\sigma^{-1}(j)} \geq \log(1/(6\delta)) \ .$$

Arguing similarly for the permutation π_1, we arrive at

$$\frac{1}{d!} \sum_{\sigma} \sum_{j=1}^{d} \mathbb{E}^0[N_{1,j} + N_{2,j}] \underline{\Delta}^2_{\sigma^{-1}(j)} \geq 2 \log(1/(6\delta)) \ .$$

Then, by Lemma F.1, this left-hand-side term of this equation is larger or equal to $\mathbb{E}^0[N] \|\underline{\Delta}\|_2^2 / d$. Since $\mathbb{E}^0[N] \geq (1 - 2\delta)x$, we conclude that

$$x \geq \frac{d \log(1/6\delta)}{2\|\underline{\Delta}\|_2^2} \ ,$$

which conclude the proof.

□

Proof of Theorem 6.2. We introduce the $n \times d$ matrix M_0 by $(M_0)_{1,j} = 1$ and $(M_0)_{i,j} = 1 - \sum_{l=1}^{i-1} \Delta_l / \sqrt{d}$ for $i = 2, \ldots, n-1$. All the rows of M_0 are constant. Obviously, the true ranking π is the identity, while $\|M_i - M_{i+1}\|_2^2 = \Delta_i^2$. Given $i \in [\![n_0 - 1]\!]$, let $\pi_{i,i+1}$ denote the transposition that exchanges i and $i+1$. Any δ-accurate algorithm A is able to decipher with probability higher than $1 - \delta$ between the matrix M_0 and the permuted matrix $(M_0)_{\pi_{i,i+1}}$, which, since the

rows of M_0 are constant, is equivalent to best arm identification in a two-arm problem with gap $\Delta_i/\sqrt{d}$. Assume that we observe the matrix M_0 with a standard Gaussian noise and denote $\mathcal{B}_0$ for the corresponding distribution. By (Kaufmann et al., 2016), we have $\mathbb{E}_{\mathcal{B}_0,A}[N_{i+1}] \geq d\log[1/4\delta)]/\Delta_i^2$. By linearity, we deduce that

$$\mathbb{E}_{\mathcal{B}}[N] \geq \sum_{i=1}^{n-1} \frac{d}{\Delta_i^2} \log[1/(4\delta)] \ ,$$

which concludes the proof. □

Proof of Theorem 6.3. This theorem is a straightforward consequence of existing lower bounds in multi-armed bandits for best arm identification. Indeed, the set $\overline{\mathbb{D}}_{\boldsymbol{\Delta}}^{n,d}$ contains in particular problem instances that are constant over questions. For these instances, our d-dimensional problem is akin to a one dimensional problem, i.e. to a standard multi-armed bandit problem. Existing lower bounds in this case imply the bounds, see e.g. (Kaufmann et al., 2016).

□

F. Technical results

The first lemma is a slight generalization of Lemma 3.1 in Castro, 2014

Lemma F.1. *Let Υ_d denote the set of permutations on $[\![d]\!]$. Consider any vectors $\boldsymbol{x} = (x_1, \ldots, x_d) \in \mathbb{R}_+^d$ and any $\boldsymbol{b} = (b_1, \ldots, b_d) \in \mathbb{R}_+^d$.*

$$\sum_{\pi \in \Upsilon_d} \sum_{i=1}^{d} b_i \, x_{\pi(i)} = \frac{|\Upsilon_d|}{d} \sum_{i=1}^{d} x_i \sum_{i=1}^{d} b_i \ .$$

Proof. Fix any such $\boldsymbol{b}$ and $\boldsymbol{x}$. We simply exchange the summation order.

$$\begin{aligned}\sum_{\pi \in \Upsilon_d} \sum_{i=1}^{d} b_i \, x_{\pi(i)} &= \sum_{i=1}^{d} b_i \sum_{\pi \in \Upsilon_d} x_{\pi(i)} = \sum_{i=1}^{d} b_i \, \frac{|\Upsilon_d|}{d} \sum_{i=1}^{d} x_i \\ &= \frac{|\Upsilon_d|}{d} \sum_{i=1}^{d} x_i \sum_{i=1}^{d} b_i \ .\end{aligned}$$

□

Lemma F.2 (Kaufmann et al., 2016, with slight modification)**.** *Let ν and ν' be two collections of d probability distributions on $\mathbb{R}$, such that for all $a \in [\![d]\!]$, the distributions ν_a and $\nu_{a'}$ are mutually absolutely continuous. For any almost-surely finite stopping time τ with respect to the data collected before τ,*

$$\sup_{\mathcal{E} \in \mathcal{F}_\tau} |\mathbb{P}_\nu(\mathcal{E}) - \mathbb{P}_{\nu'}(\mathcal{E})| \leq 1 - \frac{1}{2} \exp\left\{ -\sum_{a=1}^{d} \mathbb{E}_\nu[N_a(\tau)] KL(\nu_a, \nu'_a) \right\}.$$

Lemma F.3. *Let $(x_i)_{i \in [\![d]\!]}$ denote a sequence of non-increasing numbers in $[0,1]$. We have*

$$\sum_{i=1}^{d} x_i \leq \log(2d) \max_{1 \leq s \leq d} \{s \, x_s\}. \tag{9}$$

Proof. Suppose for the sake of contradiction that, for any $s \in [\![d]\!]$, we have

$$s \log(2d) \, x_s < \sum_{i=1}^{d} x_i.$$

Dividing the above inequality by s and summing it over $s \in [\![d]\!]$, we deduce that

$$\log(2d) \sum_{s=1}^{d} x_s < \sum_{s=1}^{d} \frac{1}{s} \sum_{i=1}^{d} x_i,$$

which contradict the upper bound on the partial sum of harmonic series $H_d = \sum_{s=1}^{d} \frac{1}{s}$:

$$H_d \leq \log(d) + \gamma + \frac{1}{2d-1} \ .$$

□

The following lemma is a direct consequence of Chernoff's inequality applied to a binomial random variable.

Lemma F.4. *Let $X = \sum_{i=1}^{n} X_i$, where X_i are independent and follow the Bernoulli distribution with parameter p, and let $\mu = \mathbb{E}[X]$, we have for any $\kappa \geq 0$*

$$\mathbb{P}(X \geq (1+\kappa)\mu) \leq \frac{e^{\kappa\mu}}{(1+\kappa)^{(1+\kappa)\mu}}.$$

Besides, for any $\kappa \in (0,1)$, we have

$$\mathbb{P}(X \leq (1-\kappa)\mu) \leq \exp\left(-\frac{\kappa^2\mu}{2}\right)$$
$$\mathbb{P}(X \geq (1+\kappa)\mu) \leq \exp\left(-\frac{\kappa^2\mu}{3}\right)$$

Proof. We only show the first inequality, the other ones being classical. All of them are consequences of Chernoff inequality. For a variable Y, we denote $M_Y(t)$ its moment generating function. Recall that, for any $X_i \sim \text{Ber}(p)$, we have $M_{X_i}(t) \leq e^{p(e^t-1)}$. Moreover for $X = \sum_{i=1}^{n} X_i$, we have

$$\begin{aligned}\mathbb{P}(X \geq k) &\leq \min_{t>0} \frac{M_X(t)}{e^{(k)}} = \min_{t>0} \frac{\prod_{i=1}^{n} M_{X_i}(t)}{e^{tk}} \\ &\leq \min_{t>0} \frac{\left(e^{p(e^t-1)}\right)^n}{e^{tk}} = \min_{t>0} \frac{e^{\mu(e^t-1)}}{e^{tk}} \ .\end{aligned}$$

For $k = (1+\kappa)\mu$, we take $t = \log(1+\kappa)$ and we obtain

$$\mathbb{P}(X \geq (1+\kappa)\mu) \leq \frac{e^{\mu\left(e^{\log(1+\kappa)}-1\right)}}{e^{(1+\kappa)\mu\log(1+\kappa)}} = \frac{e^{\kappa\mu}}{(1+\kappa)^{(1+\kappa)\mu}} \ .$$

□