Verification Based Solution for Structured MAB Problems

Zohar Karnin
Yahoo Research
New York, NY 10036
`zkarnin@ymail.com`

Abstract

We consider the problem of finding the best arm in a stochastic Multi-armed Bandit (MAB) game and propose a general framework based on verification that applies to multiple well-motivated generalizations of the classic MAB problem. In these generalizations, additional structure is known in advance, causing the task of verifying the optimality of a candidate to be easier than discovering the best arm. Our results are focused on the scenario where the failure probability δ must be very low; we essentially show that in this high confidence regime, identifying the best arm is as easy as the task of verification. We demonstrate the effectiveness of our framework by applying it, and matching or improving the state-of-the art results in the problems of: Linear bandits, Dueling bandits with the Condorcet assumption, Copeland dueling bandits, Unimodal bandits and Graphical bandits.

1 Introduction

The Multi-Armed Bandit (MAB) game is one where in each round the player chooses an action, also referred to as an arm, from a pre-determined set. The player then gains a reward associated with the chosen arm and observes the reward while rewards associated with the other arms are not revealed. In the stochastic setting, each arm x has a fixed associated value $\mu(x)$ throughout all rounds, and the reward associated with the arm is a random variable, independent of the history, with an expected value of $\mu(x)$. In this paper we focus on the pure exploration task [9] in the stochastic setting where our objective is to identify the arm maximizing $\mu(x)$ with sufficiently high probability, while minimizing the required number of rounds, otherwise known as the query complexity. This task, as opposed to the classic task of maximizing the sum of accumulated rewards is motivated by numerous scenarios where exploration (i.e. trying multiple options) is only possible in an initial testing phase, and not throughout the running time of the game.

As an example consider a company testing several variations of a (physical) product, and then once realizing the best one, moving to a production phase where the product is massively produced and shipped to numerous vendors. It is very natural to require that the identified option is the best one with very high probability, as a mistake can be very costly. Generally speaking, the vast majority of uses-cases of a pure exploration requires the error probability δ to be very small, so much so that even a logarithmic dependence over δ is non-negligible. Another example to demonstrate this is that of explore-then-exploit type algorithms. There are many examples of papers providing a solution to a regret based MAB problem where the first phase consists of identifying the best arm with probability at least $1 - 1/T$, and then using it in the remainder of the rounds. Here, $\delta = 1/T$ is often assumed to be the only non-constant.

We do not focus on the classic MAB problem but rather on several extensions of it for settings where we are given as input some underlying structural properties of the reward function μ. We elaborate on the formal definitions and different scenarios in Section 2. Another extension we consider is that

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

of *Dueling Bandits* where, informally, we do not query a single arm but rather a pair, and rather than observing the reward of the arms we observe a hint as to the difference between their associated μ values. Each extension we discuss is motivated by different scenarios which we elaborate on in the upcoming sections. In all of the cases mentioned, we focus on the regime of high confidence meaning where the failure probability δ is very small.

Notice that due to the additional structure (that does not exist in the classic case), verifying a candidate arm is indeed the best arm can be a much easier task, at least conceptually, compared to that of discovering which arm is the best. This observation leads us to the following design: Explore the arms and obtain a candidate arm that is the best arm w.p. $1-\kappa$ for some constant κ, then verify it is indeed the best with confidence $1-\delta$. If the exploration procedure happened to be correct, the query complexity of the problem will be composed of a sum of two quantities. One is that of the exploration algorithm that is completely independent of δ, and the other is dependent of δ but is the query complexity of the easier verification task. The query complexity is either dominated by that of the verification task, or by that of the original task with a *constant* failure probability. Either way, for small values of δ the savings are potentially huge. As it turns out, as discussed in Section 3, a careful combination of an exploration and verification algorithm can achieve an expected query complexity of $H_{\text{explore}} + H_{\text{verify}}$ where H_{explore} is the exploration query complexity, independent of δ, and H_{verify} is the query complexity of the verification procedure with confidence $1-\delta$. Below, we design exploration and verification algorithms for the problems of: Dueling bandits §4, Linear bandits §5, Unimodal graphical bandits §6 and Graphical bandits[1] . In the corresponding sections we provide short reviews of each MAB problem, and analyze their exploration and verification algorithms. Our results improve upon the state-of-the-art results in each of these mentioned problem (See Table 1 for a detailed comparison).

Related Works: We are aware of one attempt to capture multiple (stochastic) bandit problems in a single frameworks, given in [20]. The focus there is mostly on problems where the observed random variables do not necessarily reflect the reward, such as the dueling bandit problem, rather than methods to exploit structure between the arms. For example, in the case of the dueling bandit problem with the Condorcet assumption their algorithm does not take advantage of the structural properties and the corresponding query complexity is larger than that obtained here (see Section 4.1). We review the previous literature of each specific problem in the corresponding sections.

2 Formulation of Bandit Problems

The pure exploration Multi-Armed Bandit (MAB) problem, in the stochastic setting, can be generally formalized as follows. Our input consists of a set $\mathcal{K}$ of arms, where each arm x is associated with some reward $\mu(x)$. In each round t we play an arm x_t and observe the outcome of a random variable whose expected value is $\mu(x_t)$. Other non-stochastic settings exist yet they are outside the scope of our paper; see [4] for a survey on bandit problems, including the stochastic and non-stochastic settings. The objective in the best arm identification problem is to identify the arm[2] $x^* = \arg\max \mu(x)$ while minimizing the expected number of queries to the reward values of the arms. Other than the classic MAB problem, where $\mathcal{K}$ is a finite set and μ is an arbitrary function there exist other frameworks where some structure is assumed regarding the behavior of μ over the arms of $\mathcal{K}$. An example for a common framework matching this formulation, that we will analyze in detail in Section 5, is that of the linear MAB. Here, $\mathcal{K}$ is a compact subset of $\mathbb{R}^d$, and the reward function μ is assumed to be linear. Unlike the classic MAB case, an algorithm can take advantage of the structure of μ and obtain a performance that is independent of the size of $\mathcal{K}$. Yet another example, discuss in Section 6, is that of unimodal bandits, where we are given a graph whose vertices are the arms, and it is guaranteed that the best arm is the unique arm having a maximal value among its neighbors in the graph.

The above general framework captures many variants of the MAB problem, yet does not capture the Dueling Multi Armed Bandit (DMAB) problem. Here, the input as before consists of a set of arms denoted by $\mathcal{K}$ yet we are not allowed to play a single arm in a round but rather a pair $x, y \in \mathcal{K}$. The general definition of the observation from playing the pair x, y is a random variable whose

---

[1] Do to space restrictions we defer the section of Graphical bandits [7] to the extended version.

[2] This objective is naturally extended in the PAC setting where we are interested in an arm that is approximately the best arm. For simplicity we restrict our focus to the best arm identification problem. We note that our general framework of exploration and verification can be easily expanded to handle the PAC setting as well.

expected value is $P(x,y)$ where $P : \mathcal{K} \times \mathcal{K} \to \mathbb{R}$. The original motivating example for the DMAB [22] problem is that of information retrieval, where a query to a pair of arms is a presentation of the interleaved results of two ranking algorithms. The output is the 0 or 1, depending on the choice of the user, i.e. whether she chose a result from one or ranker or the other. The μ score here can be thought of a quality score for a ranker, defined according to the P scores. We elaborate on the motivation for the MAB problem and the exact definition of the best arm in Section 4. In an extended version of this paper we discuss the problem of *graphical bandits* that is in some sense a generalization of the dueling bandit problem. There, we are not allowed to query any pair but rather pairs from some predefined set $E \subseteq \mathcal{K} \times \mathcal{K}$.

3 Boosting the Exploration Process with a Verification Policy

In what follows we present results for different variants of the MAB problem. We discuss two types of problems. The first is the well known pure exploration problem. Our input is the MAB instance, including the set of arms and possible structural information, and a confidence parameter κ. The objective is to find the best arm w.p. at least $1-\kappa$ while using a minimal number of queries. We often discuss variants of the exploration problem where in addition to finding the best arm, we wish to obtain some additional information about the problem such as an estimate of the gaps of the reward value of suboptimal arms from the optimal one, the identity of important arm pairs, etc. We refer to this additional information as an advice vector θ, and our objective is to minimize queries while obtaining a sufficiently accurate advice vector and the true optimal arm with probability at least $1-\kappa$. For each MAB problem we describe an algorithm referred to as *FindBestArm* with a query complexity of[3] $H_{\text{explore}} \cdot \log(1/\kappa)$ that obtains an advice vector θ that is sufficiently accurate[4] w.p. at least $1-\kappa$.

Definition 1. *Let* FindBestArm *be an algorithm that given the MAB problem and confidence parameter* $\kappa > 0$ *has the following guarantees. (1) with probability at least* $1-\kappa$ *it outputs a correct best arm and advice vector* θ*. (2) its expected query complexity is* $H_{\text{explore}} \cdot \log(1/\kappa)$*, where* H_{explore} *is some instance specific complexity (that is not required to be known).*

The second type of problem is that of verification. Here we are given as input not only the MAB problem and confidence parameter δ, but an advice vector θ, including the identity of a candidate optimal arm.

Definition 2. *Let* VerifyBestArm *be an algorithm that given the MAB problem, confidence parameter* $\delta > 0$ *and an advice vector* θ *including a proposed identity of the best arm, has the following guarantees. (1) if the candidate optimal arm is not the actual optimal arm, the output is 'fail' w.p. at least* $1-\delta$*. (2) if the advice vector is sufficiently accurate, and in particular the candidate is indeed the optimal arm, we should output 'success' w.p. at least* $1-\delta$*. (3) if the advice vector is sufficiently accurate the expected query complexity is* $H_{\text{verify}} \log(1/\delta)$*. Otherwise, it is* $H_{\text{explore}} \log(1/\delta)$*.*

It is very common that $H_{\text{verify}} \ll H_{\text{explore}}$ as it is clearly an easier problem to simply verify the identity of the optimal arm rather than discover it. Our main result is thus somewhat surprising as it essentially shows that in the regime of high confidence, the best arm identification problem is as easy as verifying the identity of a candidate. Specifically we provide a complexity that is additive in H_{explore} and $\log(1/\delta)$ rather than multiplicative. The formal result is as follows.

Algorithm 1 Explore-Verify Framework

```
Input: Best arm identification problem, Oracle access to FindBestArm and VerifyBestArm with
  failure probability tuning, failure probability parameter $\delta$, parameter $\kappa$.
  for all $r = 1 \ldots$ do
    Call FindBestArm with failure probability $\kappa$, denote by $\theta$ its output.
    Call VerifyBestArm with advice vector $\theta$, that includes a candidate best arm $\hat{x}$, and failure
    probability $\delta/2r^2$. If succeeded, return $\hat{x}$. Else, continue to the next iteration
  end for
```

---

[3] The general form of such algorithms is in fact $H_1 \log(1/\kappa) + H_0$. For simplicity we state our results for the form $H \log(1/\kappa)$; the general statements are an easy modification.

[4] The exact definition of *sufficiently accurate* is given per problem instance.

Theorem 3. *Assume that algorithm 1 is given oracle access to* FindBestArm *and* VerifyBestArm *with the above mentioned guarantees, and a confidence parameter $\delta < 1/3$. For any $\kappa < 1/3$, the algorithm identifies the best arm with probability $1-\delta$ while using an expected number of at most*

$$O\left(H_{explore}\log(1/\kappa) + (H_{verify} + \kappa \cdot H_{explore})\log(1/\delta)\right)$$

The following provides the guarantees for two suggested values of κ. The first may not be known to us but can very often be estimated beforehand. The second depends only on δ hence is always known in advance.

Corollary 4. *By setting $\kappa = \min\{1/3, H_{verify}/H_{explore}\}$, algorithm 1 has an expected number of at most*

$$O(H_{explore}\log(H_{explore}/H_{verify}) + H_{verify}\log(1/\delta))$$

queries. By setting $\kappa = \min\{1/3, 1/\log(1/\delta)\}$, algorithm 1 has an expected query complexity of at most

$$O(H_{explore}\log(\log(1/\delta)) + H_{verify}\log(1/\delta))$$

Notice that by setting κ to $\min\{1/3, 1/\log(1/\delta)\}$, for any practical use-case, the dependence on δ in the left summand is nonexistent. In particular, this default value for κ provides a multiplicative saving of either $H_{\text{explore}}/H_{\text{verify}}$, i.e. the ratio between the exploration and verification problem, or $\frac{\log(1/\delta)}{\log(\log(1/\delta))}$. Since $\log(1/\delta)$ is rarely a negligible term, and as we will see in what follows, neither is $H_{\text{explore}}/H_{\text{verify}}$, the savings are significant, hence the effectiveness of our result.

Proof of Theorem 3. In the analysis we often discuss the output of the sub-procedures in round $r > 1$, even if the algorithm terminated before round r. We note that these values are well-defined random variables regardless of the fact that we may not reach the round. To prove the correctness of the algorithm notice that since $\sum_{r=1}^{\infty} r^{-2} \le 2$ we have with probability at least $1-\delta$ that all runs of *VerifyBestArm* do not err. Since we halt only when *VerifyBestArm* outputs 'success' our algorithm indeed outputs the best arm w.p. at least $1-\delta$

We proceed to analyze the expected query complexity, and start with a simple observation. Let $\text{QC}_{\text{single}}(r)$ denote the expected query complexity in round r, and let Y_r be the indicator variable to whether the algorithm reached round r. Since Y_r is independent of the procedures running in round r and in particular of the number of queries required by them, we have that the total expected query complexity is

$$\mathbb{E}\left[\sum_{r=1}^{\infty} Y_r \text{QC}_{\text{single}}(r)\right] = \sum_{r=1}^{\infty} \mathbb{E}\left[Y_r\right] \cdot \mathbb{E}\left[\text{QC}_{\text{single}}(r)\right]$$

Hence, we proceed to analyze $\mathbb{E}\left[\text{QC}_{\text{single}}(r)\right]$ and $\mathbb{E}[Y_r]$ separately. For $\mathbb{E}\left[\text{QC}_{\text{single}}(r)\right]$ we have

$$\begin{aligned}\mathbb{E}\left[\text{QC}_{\text{single}}(r)\right] \le H_{\text{explore}}\log(1/\kappa) + &\\ \left((1-\kappa)\, H_{\text{verify}} + \kappa H_{\text{explore}}\right)\log\left(\frac{2r^2}{\delta}\right) &\le \\ H_{\text{explore}}\log(1/\kappa) + (\kappa H_{\text{explore}} + H_{\text{verify}})\log\left(\frac{2r^2}{\delta}\right)&\end{aligned}$$

To explain the first inequality, the first summand is the complexity of *FindBestArm* . The second summand is that of *VerifyBestArm* , that is decomposed to the complexity in the scenario where *FindBestArm* succeeded vs. the scenario where it failed. To compute $\mathbb{E}[Y_r]$, we notice that Y_r is an indicator function hence $\mathbb{E}[Y_r] = \Pr[Y_r = 1]$. In order for Y_r to take the value of 1 we must have that for all rounds $r' < r$ either *VerifyBestArm* or *FindBestArm* have failed. Since the failure or success of the algorithms at different rounds are independent we have

$$\Pr[Y_r = 1] \le \prod_{r' < r}\left(\kappa + \frac{\delta}{2(r')^2}\right) \le 2^{1-r} \,.$$

The last inequality is since $\delta, \kappa \le 1/3$. We get that the expected number of queries required by the algorithm is at most

$$2 \cdot \sum_{r=1}^{\infty} 2^{-r}\left(H_{\text{explore}}\log(1/\kappa) + (\kappa H_{\text{explore}} + H_{\text{verify}})\log\left(\frac{2r^2}{\delta}\right)\right) =$$

MAB task	cite	existing solution	our solution	improvement ratio
Dueling Bandits (Condorcet)	[16]	$\left(K^{1+\epsilon} \cdot \sum_{x \neq x^*} \min_{\substack{y: \\ p_{xy}<0}} p_{xy}^{-2}\right) +$ $\sum_{x \neq x^*} \min_{y, p_{xy}<0} p_{xy}^{-2} \log(1/\delta)$	$\sum_{\substack{x \neq x^* \\ y \neq x}} \min \left\{ p_{xy}^{-2}, \min_{\substack{y': \\ p_{xy'}<0}} p_{xy}^{-2} \right\} +$ $\sum_{x \neq x^*} \min_{y, p_{xy}<0} p_{xy}^{-2} \log(1/\delta)$	$\geq K^{\epsilon}$ for large δ
Linear Bandits	[19]	$\frac{d \log(K/\delta)}{\Delta_{\min}^2}$	$\frac{d \log\left(Kd/\Delta_{\min}^2\right)}{\Delta_{\min}^2} +$ $\rho^*(Y^*) \log(1/\delta)$	up to d for small δ
Unimodal Bandits (line graph) (line graph)	[6]	$\sum_{x \neq x^*} (\Delta_x^{\Gamma})^{-2} +$ $\sum_{x \in \Gamma(x^*)} \Delta_x^{-2} \log(1/\delta)$	$\sum_{x \neq x^*} (\Delta_x)^{-2} +$ $\sum_{x \in \Gamma(x^*)} \Delta_x^{-2} \log(1/\delta)$	can be $\Omega(K)$ in typical settings (large δ)
Graphical Bandits	[7]	$\frac{KD \log(K/\delta) \log^2(K)}{\Delta_{\min}^2}$	$\frac{KD \log^3(K)}{\Delta_{\min}^2} + \frac{KD \log(1/\delta)}{\Delta_{\min}^2}$	$\log^2(K)$

Table 1: Comparison between the results obtained by our techniques and the state-of-the-art results in several bandit problem. K represents the total number of arms, δ the failure probability; in the case of linear bandits, d is the dimension of the space in which the arms lie. The definitions the rest of the problem specific quantities are given in the corresponding sections. The ratio between the solutions, for a typical case is given in the last column.

$$2 \cdot \sum_{r=1}^{\infty} 2^{-r} \left(H_{\text{explore}} \log(1/\kappa) + (\kappa H_{\text{explore}} + H_{\text{verify}}) \log(1/\delta)\right) +$$

$$2 \cdot \sum_{r=1}^{\infty} 2^{-r} \log(2r^2) \left(\kappa H_{\text{explore}} + H_{\text{verify}}\right) = O\left(H_{\text{explore}} \log(1/\kappa) + (\kappa H_{\text{explore}} + H_{\text{verify}}) \log(1/\delta)\right)$$

□

In the following sections we provide algorithms for several bandit problems using the framework of Theorem 3. In Table 1 we provide a comparison between the state-of-the-art results prior to this paper and the results here.

4 Application to Dueling Bandits

The *dueling bandit problem*, introduced in [22], arises naturally in domains where feedback is more reliable when given as a pairwise preference (e.g., when it is provided by a human) and specifying real-valued feedback instead would be arbitrary or inefficient. Examples include *ranker evaluation* [14, 23, 12] in information retrieval, ad placement and recommender systems. As with other *preference learning* problems [10], feedback consists of a pairwise preference between a selected pair of arms, instead of scalar reward for a single selected arm, as in the K-armed bandit problem.

The formulation of the problem is the following. Given a set of arms $\mathcal{K}$, a query is to a pair $x, y \in \mathcal{K}$ and its output is a r.v. in $\{-1, 1\}$ with an expected reward of P_{ij}. It is assumed that P is anti-symmetric meaning[5] $P(x, y) = -P(y, x)$ and the μ values are determined by those of P. One common assumption regarding P is the existence of a Condorcet winner, meaning there exist some $x^* \in \mathcal{K}$ for which $P(x^*, y) \geq 0$ for all $y \in \mathcal{K}$. In this case, x^* is defined as the best arm and the reward associated with arm y is typically $P(x^*, y)$. A more general framework can be considered where a Condorcet winner is not assumed to exist. In the absence of a Condorcet winner there is no clear answer as to which arm is the best; several approaches are discussed in [20], [5], and recently in [8, 3], that use some of the notions proposed by social choice theorists, such as the Copeland score or the Borda score to measure the quality of each arm, or game theoretic concepts to determine the best worst-case strategy over arms; we do not elaborate on all of them as they are outside the scope of this paper. In Appendix B.2 we discuss one solution based on the Copeland score, where $\mu(x)$ is defined as the number of arm $y \neq x$ where $P(x, y) > 0$.

A general framework capturing both the MAB and DMAB scenarios is that of *partial monitoring games* introduced by [18]. In this framework, when playing an arm $\mathcal{K}$ one obtains a reward $\mu(x)$ yet observes a different function $h(x)$. Some connection between h and μ is known in advance and based on it, one can design a strategy to discover the best arm or minimize regret. As we do not present results regarding this framework we do not elaborate on it any further, but rather mention that our results, in terms of query complexity, cannot be matched by the existing results there.

---

[5] It is actually common to define the output of P as a number in $[0, 1]$ and have $P(x, y) = 1 - P(y, x)$, but both definitions are equivalent up to a linear shift of P.

4.1 Dueling Bandits with the Condorcet Assumption

The Condorcet assumption in the Dueling bandit setting asserts the existence of an arm x^* that beats all other arms. In this section we discuss a solution for finding this arm under the assumption of its existence. Recall that the observable input consists of a set of arms $\mathcal{K}$ of size K. There is assumed to exist some matrix P mapping each pair of arms $x, y \in \mathcal{K}$ to a number $p_{xy} \in [-1, 1]$; the matrix P has a zero diagonal, meaning $p_{xx} = 0$ and is anti-symmetric $p_{xy} = -p_{yx}$. A query to the pair (x, y) gives an observation to a random Bernoulli variable with expected value $(1 + p_{xy})/2$ and is considered as an outcome of a match between x, y. As we assume the existence of a Condorcet winner, there exists some $x^* \in \mathcal{K}$ with $p_{x^*y} > 0$ for all $y \neq x$.

The Condorcet dueling bandit problem, as stated here and without any additional assumptions was tackled in several papers [20, 26, 16]. The best guarantees to date are given by [16] that provide an asymptotically optimal regret bound for the problem, for the regime of a very large time horizon. This result can be transformed into a best-arm identification algorithm, and the corresponding guarantee is listed in Table 1. Loosely speaking, the result shows that it suffices to query each pair sufficiently many times to separate the corresponding $P_{x,y}$ from 0.5 with constant probability, and additionally only K pairs must be queried sufficiently many times in order to separate the corresponding $P_{x,y}$ from 0.5 with probability $1 - \delta$. We note that other improvements exist that achieve a better constant term (the additive term independent of δ) [25, 24] or an overall improved result via imposing additional assumptions about P such as an induced total order, stochastic triangle inequality etc. [22, 23, 1]. These types of results however fall outside the scope of our paper.

In Appendix B.1 we provide an exploration and verification algorithm for the problem. The exploration algorithm queries all pairs until finding, for each suboptimal arm x, an arm y with $p_{xy} < 0$; the exploration algorithm provides as output not only the identity of the optimal arm, but for each sub-optimal arm x, the identity of an arm $y(x)$ that (approximately) maximizes p_{yx} meaning it beats x by the largest gap. The verification procedure is now straightforward. Given the above advice the algorithm makes sure that for each allegedly sub-optimal x, the arm $y(x)$ indeed beats it meaning $p(yx) > 0$. We obtain the following formal result.

Theorem 5. *Algorithm 1, along with the exploration and verification algorithms given in Appendix B.1, finds the Condorcet winner w.p. at least* $1 - \delta$ *while using an expected amount of at most*

$$\tilde{O}\left(\sum_{y \neq x^*} p_{x^*y}^{-2} + \sum_{x \neq x^*} \sum_{y \neq x} \min\left\{p_{xy}^{-2}, \min_{y', p_{xy'} < 0} p_{xy}^{-2}\right\}\right) + O\left(\sum_{x \neq x^*} \min_{y, p_{xy} < 0} p_{xy}^{-2} \ln(K/\delta p_{xy}^2)\right)$$

queries, where x^* *is the Condorcet winner.*

5 Application to Linear Bandits

The linear bandit problem was originally introduced in [2]. It captures multiple problems where there is linear structure among the available options. Its pure exploration variant (as opposed to the regret setting) was recently discussed in [19]. Recall that in the linear bandit problem the set of arms $\mathcal{K}$ is a subset of $\mathbb{R}^d$. The reward function associated with an arm x is a random variable with expected value $\mu(x) = w^\top x$, for some unknown $w \in \mathbb{R}^d$. For simplicity we assume that all vectors w, and those of $\mathcal{K}$ lie inside the Euclidean unit ball, and that the noise is sub-gaussian with variance 1 (hence concentration bounds such as Hoeffding's inequality can be applied).

The results of [19] offer two approaches. The first is a static strategy that guarantees, for failure probability κ, a query complexity of $\frac{d \log(K/\kappa)}{\Delta_{\text{min}}^2}$ with x^* being the best arm, $\Delta_x = w^\top(x^* - x)$ for $x \neq x^*$ and $\Delta_{\text{min}} = \min_{x \neq x^*} \Delta_x$. The second is adaptive and provides better bounds in a specific case where the majority of the hardship of the problem is in separating the best arm from the second best arm.

The algorithms are based on tools from the area of *Optimal design of experiments* where the high level idea is the following: Consider our set of vectors (arms) $\mathcal{K}$ and an additional set of vecotrs Y. We are interested in querying a sequence of t arms from $\mathcal{K}$ that will minimize the maximum variance of the estimation of $w^\top y$, where the maximum is taken over all $y \in Y$. Recall that via the Azuma-Hoeffding inequality, one can show that by querying a set of points $x_1, \ldots, x_t$ and solving the Ordinary Least

Squares (OLS) problem, one obtains an unbiased estimator of w and the corresponding variance to a point y is

$$\rho_{x_1,\ldots,x_t}(y) \triangleq y^\top \left(\sum_{i=1}^{t} x_i x_i^\top\right)^{-1} y$$

Hence, our formal problem statement is to obtain a sequence $x_1, \ldots, x_t$ that minimizes $\rho_{x_1,\ldots,x_t}(Y)$ defined as $\rho_{x_1,\ldots,x_t}(Y) = \max_{y\in Y} \rho_{x_1,\ldots,x_t}(y)$. Tools from the area of *Optimal design of experiments* (see e.g. [21]) provide ways to obtain such sequences that achieve a multiplicative approximation of $1 + d(d+1)/t$ of the optimal sequence. In particular it is shown that as t tends to infinity, t times the ρ value of the optimal sequence of length t tends to

$$\rho^*(Y) \triangleq \min_{p} \max_{y\in Y} y^\top \left(\sum_{x\in\mathcal{K}} p_x x x^\top\right)^{-1} y$$

with p restricted to being a distribution over $\mathcal{K}$. We elaborate on these in the extended version of the paper.

[19] propose two and analyze two different choices of the set Y. The first is the set $Y = \mathcal{K}$; querying points of $\mathcal{K}$ in order to minimize $\rho_{x_1\ldots,x_t}(\mathcal{K})$ leads to a best arm identification algorithm with a query complexity of $d\log(K/\kappa)/\Delta^2_{\text{min}}$ for failure probability κ. We use essentially the same approach for the exploration procedure (given in the extended version), and with the same (asymptotic) query complexity we do not only obtain a candidate best arm $\hat{x}$ but also approximations of the different Δ_x for all $x \neq x^*$. These are required for the verification procedure.

The second interesting set Y is the set $Y = \left\{\frac{x^*-x}{\Delta_x} | x \in \mathcal{K}, x \neq x^*\right\}$. Clearly this set is not known to us in advance, but it helps in [19] to define a notion of the 'true' complexity of the problem. Indeed, one cannot discover the best arm without verifying that it is superior to the others, and the set Y provides the best strategy to do so. The authors show that[6]

$$\max_{y\in Y} \|y\|^2 \leq \rho^*(Y) \leq 4d/\Delta^2_{\text{min}}$$

and bring examples where each of the inequalities are tight. Notice that the multiplicative gap between the bounding expressions can be huge (at least linear in the dimension d), hence an algorithm with a query complexity depending on $\rho^*(Y)$ as opposed to d/Δ^2_{min} can potentially be much better than the above mentioned algorithm. The bound on $\rho^*(Y)$ proves in particular that indeed querying w.r.t. Y is a better strategy than querying w.r.t. $\mathcal{K}$. This immediately translates into a verification procedure. Given the advice from our exploration procedure, we have access to a candidate best arm, and approximate Δ values. Hence, we construct this set Y and query according to it. We show that given a correct advice, the query complexity for failure probability δ is at most $O\left(\rho^*(Y^*)\log(K\rho^*(Y^*)/\delta)\right)$. Combining the exploration and verification algorithms, we get the following result.

Theorem 6. *Algorithm 1, along with the exploration and verification algorithms described above (we give a the formal version only in the extended version of the paper), finds the best arm w.p. at least* $1-\delta$ *while using an expected query complexity of*

$$O\left(\frac{d\log\left(Kd/\Delta^2_{\text{min}}\right)}{\Delta^2_{\text{min}}} + \rho^*(Y^*)\log\left(1/\delta\right)\right)$$

6 Application to Unimodal Bandits

The unimodal bandit problem consists of a MAB problem given unimodality information. We focus on a graphical variant defined as follows: There exist some graph G whose vertex set is the set of arm $\mathcal{K}$ and an arbitrary edge set E. For every sub-optimal arm x there exist some neighbor y in the graph such that $\mu(x) < \mu(y)$. In other words, the best arm x^* is the unique arm having a superior reward compared to its immediate neighbors. The graphical unimodal bandit problem was introduced by[7] [13].

---

[6] Under the assumption that all vectors in $\mathcal{K}$ lie in the Euclidean unit sphere

[7] Other variants of the unimodal bandit problem exist, e.g. one where the arms are the scalars in the intervals $[0,1]$ yet we do not deal with them in this paper, as we focus on pure best arm identification problems and in that scenario the regret setting is more common, and only a PAC algorithm is possible, translating to a $T^{2/3}$ rather than $\sqrt{T}$ regret algorithm

Due to space constraints we limit the discussion here to a specific type of unimodal bandits in which the underlying graph is line. The motivation here comes from a scenario where the point set $\mathcal{K}$ represents an ϵ-net over the $[0,1]$ interval and the μ values come from some unimodal one-dimensional function. We discuss the more general graph scenario only in the extended version of the paper. To review the existing results we introduce some notations. For an arm x let $\Gamma(x)$ denote the set of its neighbors in the graph. For a suboptimal arm x we let $\Delta_x^\Gamma = \max_{y \in \Gamma(x)} \mu(y) - \mu(x)$ be the gap between the reward of x and its neighbors and let $\Delta_x = \mu(x^*) - \mu(x)$ be its gap from the best arm x^*. We denote by $\Delta_{\min}^\Gamma$ the minimal value of Δ_x^Γ and $\Delta_{\min}$ be the minimal value of Δ_x. Notice that in reasonable scenarios, for a typical arm x we have $\Delta_x^\Gamma \ll \Delta_x$ since many arms are far from being optimal but have a close value to those of their two neighbors.

The state-of-the-art results to date, as far as we are aware, for the problem at hand is by [6], where a method OSUB is proposed achieving an expected query complexity of (up to logarithmic terms independent of δ)[8]

$$O\left(\sum_{x \neq x^*} (\Delta_x^\Gamma)^{-2} + \sum_{x \in \Gamma(x^*)} \Delta_x^{-2} \log(1/\delta)\right)$$

They show that the summand with the logarithmic dependence over δ is optimal. In the context of a line graph we provide an algorithm whose exploration is a simple naive application of a best arm identification algorithm that ignores the structure of the problem, e.g. *Exponential Gap-Elimination* by [15]. The verification algorithm requires only the identity of the candidate best arm as advice. It simply applies a best arm identification algorithm over the candidate arm and its neighborhood. The following provides our formal results.

Theorem 7. *Algorithm 1, along with the exploration of* Exponential Gap-Elimination *and the verification algorithm of* Exponential Gap-Elimination*, applied to the neighborhood of the candidate best arm, finds the best arm w.p. at least $1-\delta$ while using an expected query complexity of*

$$O\left(\sum_{x \neq x^*} \Delta_x^{-2} \log\left(K/\Delta_{\min}\right) + \sum_{x \in \Gamma(x^*)} \Delta_x^{-2} \log\left(1/\delta\right)\right)$$

The improvement w.r.t. the results of [6] is in the constant term independent of δ. The replacement of Δ_x^Γ with Δ_x leads to a significant improvement in many reasonable submodular functions. For example, if the arms for an ϵ-net over the $[0,1]$ interval, and the function is $O(1)$-Lipchitz then $\sum_{x \neq x^*} (\Delta_x^\Gamma)^{-2} = \Omega(\epsilon^{-3})$ while $\sum_{x \neq x^*} (\Delta_x)^{-2}$ can potentially be $O(\epsilon^{-2})$. Perhaps for this reason, experiments in [6] showed that often, performing UCB on an ϵ-net is superior to other algorithms.

7 Conclusions

We presented a general framework for improving the performance of best-arm identification problems, for the regime of high confidence. Our framework is based on the fact that in MAB problems with structure, it is often easier to design an algorithm for verifying a candidate arm is the best one, rather than discovering the identity of the best arm. We demonstrated the effectiveness of our framework by improving the state-of-the-art results in several MAB problems.

References

[1] Nir Ailon, Zohar Karnin, and Thorsten Joachims. Reducing dueling bandits to cardinal bandits. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 856–864, 2014.

[2] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *The Journal of Machine Learning Research*, 3:397–422, 2003.

[3] Akshay Balsubramani, Zohar Karnin, Robert Schapire, and Masrour Zoghi. Instance-dependent regret bounds for dueling bandits. In *Proceedings of The 29th Conference on Learning Theory, COLT 2016*, 2016.

---

[8] The result of [6] is in fact tighter in the sense that it takes advantage of the variance of the estimators by using confidence bounds based on KL-divergence. In the case of uniform variance however, the stated results here are accurate. More importantly, the KL-divergence type techniques can be applied here to obtain the same type of guarantees, at the expense of a slightly more technical analysis. For this reason we present the results for the case of uniform variance.

[4] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Machine Learning*, 5(1):1–122, 2012.

[5] Róbert Busa-Fekete, Balázs Szörényi, and Eyke Hüllermeier. Pac rank elicitation through adaptive sampling of stochastic pairwise preferences. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI*, 2014.

[6] Richard Combes and Alexandre Proutiere. Unimodal bandits: Regret lower bounds and optimal algorithms. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 521–529, 2014.

[7] Dotan Di Castro, Claudio Gentile, and Shie Mannor. Bandits with an edge. *CoRR, abs/1109.2296*, 2011.

[8] Miroslav Dudík, Katja Hofmann, Robert E. Schapire, Aleksandrs Slivkins, and Masrour Zoghi. Contextual dueling bandits. In Grünwald et al. [11], pages 563–587.

[9] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *The Journal of Machine Learning Research*, 7:1079–1105, 2006.

[10] J. Fürnkranz and E. Hüllermeier, editors. *Preference Learning*. Springer-Verlag, 2010.

[11] Peter Grünwald, Elad Hazan, and Satyen Kale, editors. *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, volume 40 of *JMLR Proceedings*. JMLR.org, 2015.

[12] K. Hofmann, S. Whiteson, and M. de Rijke. Balancing exploration and exploitation in listwise and pairwise online learning to rank for information retrieval. *Information Retrieval*, 16(1):63–90, 2013.

[13] Y Yu Jia and Shie Mannor. Unimodal bandits. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 41–48, 2011.

[14] T. Joachims. Optimizing search engines using clickthrough data. In *KDD*, 2002.

[15] Zohar Karnin, Tomer Koren, and Oren Somekh. Almost optimal exploration in multi-armed bandits. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1238–1246, 2013.

[16] Junpei Komiyama, Junya Honda, Hisashi Kashima, and Hiroshi Nakagawa. Regret lower bound and optimal algorithm in dueling bandit problem. In Grünwald et al. [11], pages 1141–1154.

[17] Junpei Komiyama, Junya Honda, and Hiroshi Nakagawa. Copeland dueling bandit problem: Regret lower bound, optimal algorithm, and computationally efficient algorithm, 2016.

[18] A. Piccolboni and C. Schindelhauer. Discrete prediction games with arbitrary feedback and loss. In *Computational Learning Theory*, pages 208–223, 2001.

[19] Marta Soare, Alessandro Lazaric, and Rémi Munos. Best-arm identification in linear bandits. In *Advances in Neural Information Processing Systems*, pages 828–836, 2014.

[20] Tanguy Urvoy, Fabrice Clerot, Raphael Féraud, and Sami Naamane. Generic exploration and k-armed voting bandits. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 91–99, 2013.

[21] Kai Yu, Jinbo Bi, and Volker Tresp. Active learning via transductive experimental design. In *Proceedings of the 23rd international conference on Machine learning*, pages 1081–1088. ACM, 2006.

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

[23] Y. Yue and T. Joachims. Beat the mean bandit. In *ICML*, 2011.

[24] Masrour Zoghi, Zohar Karnin, Shimon Whiteson, and Maarten de Rijke. Copeland dueling bandits. In *Advances in Neural Information Processing Systems*, pages 307–315, 2015.

[25] Masrour Zoghi, Shimon Whiteson, and Maarten de Rijke. Mergerucb: A method for large-scale online ranker evaluation. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 17–26. ACM, 2015.

[26] Masrour Zoghi, Shimon Whiteson, Remi Munos, and Maarten D Rijke. Relative upper confidence bound for the k-armed dueling bandit problem. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 10–18, 2014.

Supplementary Material

A Auxiliary Algorithms

In what follows we present algorithms aimed for exploration and verification in several bandit setups. In both cases we would like to bound the expected query complexity of the algorithm, rather than (or in addition to) have a high probability bound on it. The algorithms we present can be directly designed to have a bounded expected query complexity, albeit at the cost of having more technical components in both the algorithm and their proofs. To ease the reading we design these algorithms to have high probability bounds on the query complexity and provide a meta-algorithm below to transform high probability bounds to bounds on the expectation.

Consider the following example to demonstrate the problem dealt here: Consider an algorithm that given some event of probability $1-\delta$ succeeds and has a query complexity of at most T. If this event does not occur the query complexity of the algorithm can be, say, $10^6 T/\delta$. The expected query complexity of the algorithm is much larger than T but given the techniques we provide below it can be reduced to $O(T \log\log(T))$ while keeping the error probability $O(\delta)$. We note that these tricks are rather standard, and for this reason they only appear in the appendix.

A.1 Some Notation and Observations

In what follows we require some technical notation w.r.t. abstract MAB tasks and algorithms for them. In our setting, a MAB task $\mathcal{T}$ is defined as a problem whose input is an oracle with black box access to the stochastic variables corresponding to the arms (or arm pairs, in the dueling style problems), parametrized by some unknown $\mu \in \mathcal{M}$. In addition to the oracle, the input may also contain some advice vector θ^{advice}. The possible outputs for a solver of a task is denoted by the set Θ. The output set Θ in the identification task is the set of arms $\mathcal{K}$. In some cases it will be a Cartesian product of multiple copies of $\mathcal{K}$ and $\mathbb{R}$, as we will require not only the best arm but various other arms, arm-pairs, and estimates of gaps. In verification tasks, the output set Θ is simply $\{0,1\}$.

For every characterization μ and advice vector θ^{advice} there exists a subset $\Theta^{\text{correct}} \subseteq \Theta$ of correct outputs. In these, the estimates are sufficiently accurate, and the arm (pairs) identities are as expected. In what follows, our exploration algorithms will be much simpler if designed to have a high probability bound over their query complexity, rather than a bound over their expected query complexity. This is formalized below.

Definition 8. *An algorithm $\mathcal{A}$ for a MAB task $\mathcal{T}$ is said to be a* high probability solver *when: given an input characterized by μ and a failure parameter κ it provides the following guarantee: There exists some random event $\mathcal{E}$, whose randomness originates from the randomness of the oracle outputs and possibly the internal randomness of the algorithm, that occurs with probability at least $1-\kappa$ such that given its occurrence, $\mathcal{A}$ provides a correct output $\theta \in \Theta^{correct}$. Also, the expected query complexity of $\mathcal{A}$, conditioned on the occurrence of the event $\mathcal{E}$ is at most $T_1 \log(1/\kappa) + T_0$ for some constants T_1, T_0 dependent only on the input's characterization μ.*

Our verification algorithms will be much simpler if designed to have a guaranteed bound on their query complexity, only if given a correct advice vector θ^{advice}. This is formally characterized as follows.

Definition 9. *An algorithm $\mathcal{A}$ for a MAB task $\mathcal{T}$ is said to be an* input-dependent high probability solver *when given an input characterized by μ, an advice vector θ^{advice} and a failure parameter δ it provides the following guarantee: There exists some random event $\mathcal{E}$, whose randomness originates from the randomness of the oracle outputs and possibly the internal randomness of the algorithm, that occurs with probability at least $1-\delta$ such that given its occurrence, $\mathcal{A}$ provides a correct output $\theta \in \Theta^{correct}$. Also, there exist some set Θ^{CI} such that if the advice vector $\theta^{advice} \in \Theta^{CI}$ then the expected query complexity of $\mathcal{A}$, conditioned on the occurrence of the event $\mathcal{E}$ is at most $T_1^V \log(1/\delta) + T_0^V$ for some constants T_1^V, T_0^V dependent only on the input's characterization μ.*

A.2 Auxiliary Algorithm for Exploration

We now provide a meta-algorithm to transform a *high probability solver* (Definition 8) into an algorithm with a bounded expected query complexity. In Algorithm 2 we invoke the input high probability solver $\mathcal{A}$ with a query-cap of T. This means that if the algorithm does not terminate after T queries it is forcefully stopped.

Algorithm 2 Controlling the Query Complexity

```
Input: Oracle access to a high probability solver $\mathcal{A}$ with, failure threshold $\kappa < 1/8$
  for all $r = 1, 2, 3, \dots$ do
    Run $\mathcal{A}$ with parameter $\kappa/2r^2$ and query-cap of $2^r$
    If $\mathcal{A}$ terminated before reaching the cap, halt and return its output.
  end for
```

Lemma 10. *Assume Algorithm 2 is given a high probability solver with guaranteed query complexity of* $T_1 \log(1/\kappa) + T_0$*, and failure parameter* $\kappa < 1/8$*. It provides a correct output w.p. at least* $1 - \kappa$ *and its expected query complexity is at most* $O\left(T_1 \log(\log(T_1 + T_0)/\kappa) + T_0\right)$

Proof. The claim regarding the correct output stems from a union bound over all invocations of $\mathcal{A}$. For the query complexity, let r_0 be the first value of r for which

$$2^r \geq T_1 \log(r^2/2\kappa) + T_0$$

Notice that for round $r \geq r_0$, given the event in which the expected query complexity of $\mathcal{A}$ is $T_1 \log(r^2/2\kappa) + T_0$, the probability of the algorithm not terminating in round r is, by Markov's inequality, at most

$$\frac{T_1 \log(r^2/2\kappa) + T_0}{2^r} \leq (1 + 2\log(r/r_0)) \cdot \frac{T_1 \log(r_0^2/2\kappa) + T_0}{2^r} \leq$$

$$(1 + 2\log(r/r_0))2^{r_0 - r}$$

Hence, the overall probability of the algorithm not terminating in round r is at most

$$(1-\kappa)(1 + 2\log(r/r_0))2^{r_0 - r} + \kappa$$

Assuming $r_0 \geq 5$ (otherwise, T_0, T_1 are constants and the result is trivial), for $r \geq r_0 + 3$ and $\kappa \leq 1/8$, this probability is at most $1/4$. We get that for $r \geq r_0 + 3$, the probability that the algorithm reached round $r + 1$ is at most $4^{r_0 + 2 - r}$ due to the independence of the algorithm invocations. The expected query complexity is thus bounded by

$$\sum_{r=1}^{r_0+2} 2^r + \sum_{r=r_0+3}^{\infty} 4^{r_0+2-r} 2^r =$$

$$\sum_{r=1}^{r_0+2} 2^r + 2^{r_0+3} \sum_{r=r_0+3}^{\infty} 2^{r_0+2-r} \leq 2 \cdot 2^{r_0+3} \leq$$

$$O\left(T_1 \log\left(\log\left(T_1 \log(1/\kappa) + T_0\right)/2\kappa\right) + T_0\right)$$

The last inequality is due to the minimality of r_0. ☐

A.3 Auxiliary Algorithm for Verification

In this section we deal with the verification algorithms that are designed to be *input-dependent high probability solver* (Definition 9). Whenever discussing a verification algorithm $\mathcal{A}_V$ it is always a case that we have an exploration algorithm, as discussed above. Given the previous section we formally assume that there exist some exploration algorithm $\mathcal{A}_E$ that does not require any advice and is guaranteed to (1) terminate with an expected query complexity of $T_1^E \log(1/\delta) + T_0^E$ and (2) produce a correct output w.p. at least $1 - \delta$. We may also assume, again by using the result of the above section, that the algorithm $\mathcal{A}_V$, given a correct advice vector $\theta^{\text{advice}} \in \Theta^{\text{CI}}$, has an expected query complexity of $O\left(T_1^V \log(\log(T_1^V + T_0^V)/\delta) + T_0^V\right)$.

The procedure we suggest here is straightforward. Run the verification algorithm $\mathcal{A}_V$ in parallel to $\mathcal{A}_E$. Once one algorithm terminates we can terminate and use its output. This will in the worst case, increase the query complexity by a factor of two.

Lemma 11. *Given an advice θ^{advice}, when running $\mathcal{A}_V$ and $\mathcal{A}_E$ in parallel with parameter δ, the output is correct with probability at least $1-\delta$. If the advice is correct ($\theta^{advice} \in \Theta^{CI}$), the expected query complexity is bounded by*

$$O\left(T_1^V \log(\log(T_1^V + T_0^V)/\delta) + T_0^V\right)$$

If the advice is incorrect, the expected query complexity is bounded by

$$2\left(T_1^E \log(1/\delta) + T_0^E\right)$$

B Dueling Bandits

In this section we provide exploration and verification algorithms for the Dueling Bandit problem, both for the Condorcet version and the Copeland version.

B.1 Dueling Bandits with the Condorcet Assumption

B.1.1 Exploration

Algorithm 3 given below provides the required guarantees for exploration.

Algorithm 3 Exploration in Condorcet Bandits

```
Input: set of arms $\mathcal{K}$, failure probability parameter $\kappa$.
  $Q \leftarrow \{(x, y) \mid x \neq y \in \mathcal{K}\}$, the set of ordered arm pairs.
  for all $t = 1 \ldots$ do
    query each pair in $\{\{x, y\} \mid (x, y) \in Q$ or $(y, x) \in Q\}$ once.
    let $\gamma_t = \sqrt{2ln(2t^2K^2/\kappa)/t}$, and let $\ell_{xy}, u_{xy}$ be the lower and upper bounds of $p_{xy}$ according
    to the confidence interval of radius $\gamma_t$.
    Remove $(x, y)$ from $Q$ if
      • $\ell_{xy} > 0$
      • $u_{xy} < 0$ and $2u_{xy} < \ell_{xy}$
      • $\ell_{xy} < 0$ and $\ell_{xy} > u_{xy'}$ for some $y \neq y'$
    stop when $Q$ is empty
  end for
  output $\hat{x}$ as the unique element $x$ for which $\ell_{xy} > 0$ for all $y$. For all $x \neq \hat{x}$, output $y(x) =$
  $\arg\min u_{xy}$
```

Lemma 12. *It holds with probability at least $1-\kappa$ that throughout time, for any pair x, y we have $\ell_{xy} \leq p_{xy} \leq u_{xy}$.*

Proof. According to Hoeffding's inequality, since the observed r.v.'s for each pair are independent and bounded in $[-1, 1]$ we have that for any specific pair x, y and any time step t, the required property holds w.p. at least $1-\kappa/t^2K^2$. The claims follows via union bound as there are $K(K-1)/2$ possible pairs.

□

Lemma 13. *Given the event of Lemma 12, when the algorithm halts we have*

- *The Condorcet winner x^* is the unique arm for which for all $y \neq x^*$ we have $\ell_{x^*y} > 0$*
- *For all arms $x \neq x^*$ let $y(x) = \arg\min u_{xy}$. We have $p_{xy(x)} \leq \frac{1}{2}\min_y p_{xy} < 0$*

Proof. For the Condorcet winner x^* we have $p_{x^*y} > 0$ for all $y \neq x^*$ hence the only way a pair (x^*, y) is eliminated from Q is when $\ell_{x^*y} > 0$. Also, for any non-winner x there is some y s.t. $p_{xy} < 0$ hence it cannot be the case that $\ell_{xy} > 0$. This proves the first item. Consider now some $x \neq x^*$.

Define y' as $\arg\min_y p_{xy}$, the arm that beats x by the most. We first show that y' could only have been eliminated according to the elimination rule in Algorithm 3 described in the second bullet. Since

$p_{xy'} < 0$ it must be the case that $\ell_{xy'} < 0$ at all times, hence it could not have been eliminated according to the first bullet. If y' was eliminated according to the third bullet we must have for some other y that

$$p_{xy'} > \ell_{xy'} \geq u_{xy} \geq p_{xy}$$

contradicting the definition of y'. It follows that indeed y' is eliminated according to the second bullet and at termination

$$u_{xy'} < 0, \;\; u_{xy'} < \frac{1}{2}\ell_{xy'}$$

Now, according to the definition of $y(x)$ as the minimizer of u_{xy} we have

$$p_{xy(x)} < u_{xy(x)} \leq u_{xy'} < \frac{1}{2}\ell_{xy'} < \frac{1}{2}p_{xy'}$$

as required. □

After proving the correctness of the algorithm we proceed to analyze its query complexity.

Lemma 14. *Given the event of Lemma 12, the algorithm will terminate after at most*

$$\tilde{O}\left(\sum_{y\neq x^*} p_{x^*y}^{-2} + \sum_{x\neq x^*}\sum_{y\neq x} \min\left\{p_{xy}^{-2}, \min_{y'\colon p_{xy'}<0} p_{xy}^{-2}\right\}\right)\ln(1/\kappa)$$

queries

Proof. Given the definition of γ_t, for pairs of the form x^*, y it holds that after $O\left(p_{x^*y}^{-2}\log(p_{x^*y}^{-2}K^2/\kappa)\right)$ many queries, $\ell_{x^*y} > 0$, taking care of the left summand in the expression of the Lemma. The same holds for any pair x, y with $p_{xy} > 0$, taking care of the components in the right summand where the dominant part of the min expression is p_{xy}^{-2}. For the remaining summands, consider an arm $x \neq x^*$. Let $\Delta_x = |\min_{y', p_{xy'}<0} p_{xy}|$. Consider a time point $t = O(\log(K/\kappa\Delta_x^2)/\Delta_x^2)$ where we are guaranteed that all of the confidence intervals have a radius of at most $\gamma = \Delta_x/6$. At this time point, for $y' = \arg\min p_{xy}$ we have

$$u_{xy'} < -5\Delta_x/6$$

Now, consider an arm y for which $p_{xy} > -\Delta_x/2$. We have that

$$\ell_{xy} > -\Delta_x/2 - \Delta_x/6 > u_{xy'}$$

hence the pair xy must be eliminated by time t. This complies with the minimum expression in the Lemma statement. The remaining type of pairs to discuss are those with $x \neq x^*$ and y with $p_{xy} \leq -\Delta_x/2$. For these we have

$$u_{xy} < -\Delta_x/3, \;\; \ell_{xy} > -2\Delta_x/3 > 2u_{xy}$$

and the pair xy is indeed eliminated by time t. The claim follows. □

B.1.2 Verification

Lemma 15. *Assume that $\hat{x} = x^*$ and for all $x \neq x^*$, $p_{xy(x)} \leq \frac{1}{2}\min_y p_{xy}$. Then w.p. at least $1-\delta$ the algorithm halts after*

$$O\left(\sum_{x\neq x^*} \min_{y, p_{xy}<0} p_{xy}^{-2}\ln(K/\delta p_{xy}^2)\right)$$

many queries, and outputs 'success'

Proof. We start with the observation that w.p. at least $1-\delta$ all of the confidence intervals contain the relevant p_{xy} throughout time (this is true due to Hoeffding's inequality and a simple union bound argument). For $x \neq x^*$ let $\Delta_x = \max_y p_{yx}$ and let $t_x = c\Delta_x^{-2}\ln(K/\delta\Delta_x^2)$ for some sufficiently large constant c. Due to the guarantee for $y(x)$, the guarantee about the correctness of the confidence

Algorithm 4 Condorcet Dueling Bandits Verification

```
Input: set of arms $\mathcal{K}$, failure probability parameter $\delta$, candidate Condorcet winner $\hat{x}$, candidate adversary $y(x)$ for each $x \neq \hat{x}$ in $\mathcal{K}$.
  $Q \leftarrow \{x \mid x \neq \hat{x} \in \mathcal{K}\}$
  for all $t = 1 \ldots$ do
    For every $x \in Q$, query the pair $(x, y(x))$ once
    let $\gamma_t = \sqrt{2ln\left(2t^2K^2/\delta\right)/t}$, and let $\ell_{xy}, u_{xy}$ be the lower and upper bounds of $p_{xy}$ according to the confidence interval of radius $\gamma_t$.
    Remove $x$ from $Q$ if $u_{xy(x)} < 0$
    if $\ell_{xy(x)} > 0$ for some $x$, terminate with an answer 'fail'
    if $Q$ becomes empty, terminate with an answer of 'success'
  end for
```

intervals, and their definition we have that after t_x rounds it must be the case that $u_{xy(x)} < 0$. It follows that the total number of queries is

$$\sum_{x \neq x^*} t_x = O\left(\sum_{x \neq x^*} \min_{y, p_{xy} < 0} p_{xy}^{-2} \ln(K/\delta p_{xy}^2)\right)$$

as required. To prove the correctness of the output, notice that since $p_{xy(x)} < 0$ it cannot be the case that $\ell_{xy(x)} > 0$ for any x hence the algorithm will only terminate when all arms are removed from Q hence its output will be 'success'. ☐

Lemma 16. *Assume that $\hat{x} \neq x^*$. Then w.p. at least $1 - \delta$, when the algorithm terminates it outputs 'fail'*

Proof. Recall that w.p. at least $1-\delta$ all of the confidence intervals contain the relevant p_{xy} throughout time. We assume this is indeed the case and show the output must be 'fail'. Since $\hat{x} \neq x^*$ it must be the case that some $x \neq \hat{x}$ is the Condorcet winner. For this arm, $p_{xy(x)} > 0$. Hence, throughout time we must have $u_{xy(x)} > 0$ and if the algorithm terminates it cannot be because of Q being empty, but only due to $\ell_{x'y(x')} > 0$ for some x', meaning the output will be 'fail'. ☐

Theorem 5 is now an immediate corollary of the above results, combined with those of Appendix A.

B.2 Copeland Dueling Bandits

In this section we analyze a natural extension of the Condorcet winner denoted as the Copeland winner. This approach was suggested in several papers, e.g. [20, 5, 24]. The Copeland score of an arm x is the number of arm it beats, i.e. $\sum_{y \neq x} \mathbf{1}[p_{xy} > 0]$. Notice that if the Condorcet winner exists then the arm with the maximal Copeland score is unique and is indeed the Condorcet winner. We assume throughout for simplicity that for all $x \neq y$, $p_{xy} \neq 0$ meaning there are no ties. Extending our results to a setting where ties can occur is purely technical and will cause complex guarantees, hence we do not discuss the issue any further.

The state-of-the-art identification procedure can be achieved via an easy modification of the regret-based results of [24]. We note mention that a very recent result by [17] improves the regret based Copeland dueling bandit paper of [24], yet it is not clear how to transform it into a high probability best arm identification algorithm. A loose description of the query complexity of the CCB algorithm given in [24] is $K^2 + K(s+1)\log(1/\delta)$ for failure probability δ, where s is the number of losses suffered by a Copeland winner. They provide convincing arguments as to why in practice, the value of s is constant despite the theoretical possibility of it scaling as K. The exact query complexity guarantee is (up to logarithmic terms)

$$O\left(\sum_{x \neq y}\left(|p_{yx}| + \max\left\{0, \max_{y' \neq x}^{s+1} p_{y'x}\right\}\right)^{-2} + \left(\sum_{x \in C, y \notin C} p_{xy}^{-2} + \sum_{x \notin C} \frac{s+1}{\left(\max_{y \neq x}^{s+1} p_{yx}\right)^2}\right)\log(1/\delta)\right)$$

Here, C is the set of Copeland winners, and we use $\max^k$ to denote an operator returning the k'th largest value from a set. We note that in the same paper an additional algorithm (SCB) is presented with an incomparable query complexity. The term that is related to δ is larger, but there is no quadratic dependence in K, assuming the quantities of $1/p_{xy}^2$ are large, specifically $p_{xy} \gg 1/\sqrt{K}$ for all x, y. For the purpose of brevity we do not elaborate on this result further.

Our methods can be used to improve upon the above result via rather simple algorithms. In what follows we present an exploration and verification algorithm. The exploration algorithm queries arm pairs uniformly and eliminates them once either it is clear whether $p_{xy} > 0$ or vice versa, or both x and y can be excluded from being a Copeland winner regardless of whether x beats y or vice versa. The advice that is eventually produced consists of (1) the identity of a Copeland winner $\hat{x}$ (2) the identity of $K-1-s$ arms $\hat{y}_1, \ldots, \hat{y}_{K-1-s}$ that $\hat{x}$ beats by the largest gap, and (3) for every arm $x \neq \hat{x}$, the identity of s arms $y_1^x, \ldots, y_s^x$ that beat x by the largest gaps. Also, the provided Copeland winner is the one for which the gaps in (2) are the largest in the sense that it requires the least queries to verify that it beats the respective $\hat{y}$'s with high confidence. We show (Lemma 18) that the query complexity required to achieve this advice correctly w.p. at least $1-\kappa$ is $O\left(\sum_{x \neq y} p_{xy}^{-2} \log\left(K/\kappa p_{xy}^2\right)\right)$. Given the above advice our verification algorithm verifies that indeed $\hat{x}$ is a Copeland winner. It queries the pairs x, y_i^x uniformly until it discovers, with high confidence, that $p_{y_i^x x} > 0$ for all $x \neq \hat{x}, i$ and queries the pairs $\hat{x}, \hat{y}_i$ until realizing $p_{\hat{x}\hat{y}_i} > 0$ for all i. If this is indeed the case then we have with high confidence that all arms other than $\hat{x}$ are beaten by at least s arms, and $\hat{x}$ is beaten by at most s arms, hence $\hat{x}$ is indeed a Copeland winner w.h.p. If any of the above inequalities do not hold, the algorithm outputs 'fail'. The query complexity, for failure probability δ, given a good advice vector is easily shown to be $H_{\text{cplnd}}^{\text{v}} \log(1/\delta)$ for

$$H_{\text{cplnd}}^{\text{v}} = \tilde{O}\left(\min_{x^* \in C} \min_{y_1,\ldots,y_{K-1-s} \,|p_{x^*y_i}>0} \sum_i p_{x^*y_i}^{-2} + \sum_{x \neq x^*} \min_{y_1,\ldots,y_s \,\mid\, p_{xy}<0} \sum_i p_{xy_i}^{-2}\right)$$

Theorem 17 describes the guarantees given by the described exploration and verification algorithms, when combined with the techniques of Algorithm 1. It is an easy task to verify that in the regime of small δ, the expression is strictly smaller than the guarantee obtained by [24]. The exact ratio depends on the structure of the different $p'_{xy}s$.

Theorem 17. *Algorithm 1, along with the exploration and verification algorithms given below, finds a Copeland winner w.p. at least $1-\delta$ while using an expected amount of at most*

$$O\left(\sum_{x \neq y} p_{xy}^{-2} \log\left(K/p_{xy}^2\right) + H_{\text{cplnd}}^{\text{v}} \log(1/\delta)\right)$$

queries.

B.2.1 Exploration

Lemma 18. *w.p. at least $1-\kappa$ we have*

1. *If there is more than one Copeland winner, let s denote the number of losses suffered by the Copeland winners. Otherwise, let s be such that $s+1$ is the smallest amount of losses suffered by a non-Copeland winner*
2. *Upon termination, $s_t = s$.*
3. *(minimality of x^*) If there is more than one Copeland winner then x^* is a Copeland winner with*

   $$\sum_{y \,|p_{x^*y}>0} p_{x^*y}^{-2} \le 4 \min_{x \in C} \sum_{y \,|p_{xy}>0} p_{xy}^{-2}$$

   with C being the set of Copeland winners.
4. *(minimality of y_i^*)*

   $$\sum_{i=1}^{K-1-s} p_{x^*y_i^*}^{-2} \le 4 \min_{y_1,\ldots,y_{K-1-s} \,|p_{xy_i}>0} \sum_i p_{x^*y_i}^{-2}$$

Algorithm 5 Copeland Bandits Exploration

```
Input: set of arms $\mathcal{K}$, failure probability parameter $\kappa$.
  $Q \leftarrow \{(x,y) \mid x \neq y\}$
  for all $t = 1 \dots$ do
    query each pair in $\{\{x,y\} \mid (x,y) \in Q$ or $(y,x) \in Q\}$ once
    let $\gamma_t = \sqrt{2ln\left(2t^2K^2/\kappa\right)/t}$, and let $\ell_{xy}, u_{xy}$ be the lower and upper bounds of $p_{xy}$ according to the confidence interval of radius $\gamma_t$.
    For arm $x$, let $L_\ell(x) = |\{y \mid u_{xy} < 0\}|$, $L_u(x) = |\{y \mid \ell_{xy} < 0\}|$ be the lower and upper bounds on the number of losses of $x$.
    Set $s_t = \min_{x \in \mathcal{K}} L_u(x)$
    Let $B = \{x \mid L_\ell(x) > s_t\}$ be the set of arms that can be excluded from being a Copeland winner.
    Remove $(x,y)$ from $Q$ if
      • $\ell_{xy} > 0$ and $2\ell_{xy} > u_{xy}$
      • $u_{xy} < 0$ and $2u_{xy} < \ell_{xy}$
      • $x \in B$ and $\ell_{xy} > 0$
      • $x \in B$, $\ell_{xy} < 0$ and $\ell_{xy} > \min_{y_1,\dots,y_{s_t}} \max_{i \in [s_t]} u_{xy_i}$
    If $Q$ is empty, terminate and:
      • Output $x^*$ as the arm not in $B$, minimizing $\min_{y_1,\dots,y_{K-1-s_t} \mid \ell_{x^*y_i}>0} \sum_{i=1}^{K-1-s_t} \ell_{x^*y_i}^{-2}$
      • Output $y_1^*, \dots, y_{K-1-s_t}^*$, the minimizers of the above expression
      • For every $x \neq x^*$, output $\{y_i(x)\}_{i=1}^{s_t}$, the minimizers of $\min_{y_1,\dots,y_{s_t}} \max_{i \in [s_t]} u_{xy_i}$
  end for
```

5. *(minimality of $y_i(x)$) For any $x \neq x^*$ we have*

$$\sum_{i=1}^{s} p_{xy_i(x)}^{-2} \leq 4 \min_{y_1,\dots,y_{s_t} \mid p_{xy}<0} \max_{i \in [s_t]} p_{xy_i}^{-2}$$

6. *The query complexity of the algorithm is at most*

$$O\left(\sum_{x \neq y} p_{xy}^{-2} \ln(K/\kappa p_{xy}^2)\right)$$

Proof. We prove the lemma based on the event that all confidence intervals contain p_{xy} for all x, y pairs, throughout time. It is an easy exercise to see that this event happens w.p. at least $1-\kappa$.

Given the event of the confidence intervals being accurate it is clear that $L_u(x), L_\ell(x)$ are indeed upper and lower bounds on the losses of an arm x. As a result, any arm $x \in B$ cannot be a Copeland winner since being in B interprets into having some arm $x' \neq x$ with $L_u(x') < L_\ell(x)$.

To prove item 2, it suffices to show that s_t is not too large upon termination. For s_t to be too large we must have that for all $x \in C$, $L_u(x)$ is strictly larger than the true number of losses of x. This implies that there exist some (x,y) pair with $x \in C$ such that $u_{xy} > 0$ and $\ell_{xy} < 0$. This pair could not have been eliminated according to the first and second bullets describing the elimination. Since $x \in C$ cannot be in the set B, the pair x, y cannot be eliminated according to the third and fourth bullets either. It follows that while $s_t > s$, the set Q cannot be empty and the algorithm will not terminate.

We proceed to prove items 3 and 4. A conclusion of the above discussion is that upon termination, we have for all pairs x, y where x is a Copeland winner that $\ell_{xy} > 2u_{xy} > 0$ or $u_{xy} < 0$. In case there is more than one Copeland winner we have that for x^*,

$$\sum_{y \mid p_{x^*y}>0} p_{x^*y}^{-2} \leq \sum_{y \mid p_{x^*y}>0} \ell_{x^*y}^{-2} =$$

$$\min_{x \in C} \sum_{y \mid p_{xy}>0} \ell_{xy}^{-2} \leq \min_{x \in C} \sum_{y \mid p_{xy}>0} 4u_{xy}^{-2} \leq$$

$$\min_{x \in C} \sum_{y \ | p_{xy} > 0} 4 p_{xy}^{-2}$$

Also,

$$\sum_{i=1}^{K-1-s} p_{x^* y_i^*}^{-2} \leq \sum_{i=1}^{K-1-s} \ell_{x^* y_i^*}^{-2} =$$

$$\min_{y_1, \ldots, y_{K-1-s} \ | p_{x^* y_i} > 0} \sum_i \ell_{x^* y_i}^{-2} \leq$$

$$4 \min_{y_1, \ldots, y_{K-1-s} \ | p_{x^* y_i} > 0} \sum_i u_{x^* y_i}^{-2} \leq$$

$$4 \min_{y_1, \ldots, y_{K-1-s} \ | p_{x^* y_i} > 0} \sum_i p_{x^* y_i}^{-2}$$

We proceed to analyze Item 5. Notice first that due to $s_t = s$, for all arms $x \neq x^*$ we have upon termination knowledge of at least s arms y that beat x, meaning $u_{xy} < 0$. If follows that

$$\sum_{i=1}^{s} p_{x y_i(x)}^{-2} \leq \sum_{i=1}^{s} u_{x y_i(x)}^{-2} = \min_{y_1, \ldots, y_s \ | \ u_{x y_i} < 0} \sum_{i=1}^{s} u_{x y_i}^{-2} \leq$$

$$\min_{y_1, \ldots, y_s \ | \ u_{x y_i} < 0} 4 \sum_{i=1}^{s} \ell_{x y_i}^{-2} \leq \min_{y_1, \ldots, y_s \ | \ u_{x y_i} < 0} 4 \sum_{i=1}^{s} p_{x y_i}^{-2}$$

Next, consider an arm y for which $p_{xy} < 0$ but upon termination it holds that $u_{xy} > 0$. It must be the case that for some arm $y'_x(y)$,

$$0 > p_{xy} > \ell_{xy} > u_{x y'_x(y)} > p_{x y'_x(y)}$$

and we have that

$$\min_{y_1, \ldots, y_s \ | \ u_{x y_i} < 0} 4 \sum_{i=1}^{s} p_{x y_i}^{-2} \leq \min_{y_1, \ldots, y_s \ | \ p_{x y_i} < 0} 4 \sum_{i=1}^{s} p_{x y_i}^{-2}$$

thus proving Item 5.

It remains to analyze the query complexity. Notice that once $0 < 0.5 u_{xy} < \ell_{xy}$ or $0 > 0.5 \ell_{xy} > u_{xy}$, both the (ordered) pairs $(x, y), (y, x)$ are eliminated from Q and hence the unordered pair x, y is not queried again. Since p_{xy} lies within the confidence region, and its radius at time t scales as $\sqrt{\log(Kt/\kappa)/t}$, we get that x, y can be queried no more than $O(p_{xy}^{-2} \log(K/\kappa p_{xy}^2))$ times before being eliminated from Q. $\square$

B.2.2 Verification

Algorithm 6 Copeland Bandits Verification

```
Input: set of arms $\mathcal{K}$, failure probability parameter $\delta$, candidate winner $\hat{x}$ and $K - s$ different arms
    $\hat{y}_1, \ldots, \hat{y}_{K-s}$, for every $x \neq \hat{x}$, $s$ arms $y_1(x), \ldots, y_s(x)$.
  $Q \leftarrow \{(\hat{x}, \hat{y}_i) \mid i \in [K-s]\} \cup \{(x, y_i(x)) \mid x \neq \hat{x},\ i \in [s]\}$
  for all $t = 1 \ldots$ do
    query each pair in $\{\{x, y\} \mid (x, y) \in Q$ or $(y, x) \in Q\}$ once
    let $\gamma_t = \sqrt{2ln\left(2t^2K^2/\delta\right)/t}$, and let $\ell_{xy}, u_{xy}$ be the lower and upper bounds of $p_{xy}$ according
      to the confidence interval of radius $\gamma_t$.
    If $u_{\hat{x}\hat{y}_i} < 0$ for some $i \in [K-s]$, terminate and output 'fail'
    If $\ell_{x y_i(x)} > 0$ for some $x \neq \hat{x}$, $i \in [s]$, terminate and output 'fail'
    Remove from $Q$ all $(x, y)$ for which the corresponding confidence region does not contain zero.
    If $Q$ is empty, terminate and output 'success'
  end for
```

Lemma 19. *If there is more than one Copeland winner, let s denote the number of losses suffered by the Copeland winners. Otherwise, let s be such that $s + 1$ is the smallest amount of losses suffered by a non-Copeland winner. Assume the advice is proper, meaning that:*

1. *$\hat{x}$ is the Copeland winner minimizing, up to a multiplicative term of 4, the expression*
$$\sum_{y \ |p_{xy}>0} p_{xy}^{-2}$$
2. $$\sum_{i=1}^{K-1-s} p_{\hat{x}\hat{y}_i}^{-2} \leq 4 \min_{y_1,\ldots,y_{K-1-s} \ |p_{\hat{x}y_i}>0} \sum_i p_{\hat{x}y_i}^{-2}$$
3. *For any $x \neq \hat{x}$ we have*
$$\sum_{i=1}^{s} p_{xy(x)}^{-2} \leq 4 \min_{y_1,\ldots,y_{s_t} \ | \ p_{xy}<0} \max_{i \in [s_t]} p_{xy(x)}^{-2}$$

Then, w.p. at least $1-\delta$ Algorithm 6 outputs 'success' and has a query complexity of at most

$$O\left(\min_{x^* \in C} \min_{y_1,\ldots,y_{K-1-s} \ |p_{x^*y_i}>0} \sum_i p_{x^*y_i}^{-2} \log(Kp_{x^*y_i}^{-2}/\delta) + \sum_{x \neq x^*} \sum_{y_1,\ldots,y_s \ | \ p_{xy}<0} p_{xy_i}^{-2} \log(Kp_{xy_i}^{-2}/\delta) \right)$$

Proof. We notice that w.p. at least $1-\delta$, for all pairs x, y and throughout time p_{xy} lies inside the corresponding confidence interval. Given this event, and the fact that the radius of the confidence intervals scale as $\sqrt{\log(t/\delta)/t}$ we get that after

$$O\left(\min_{x^* \in C} \min_{y_1,\ldots,y_{K-1-s} \ |p_{x^*y_i}>0} \sum_i p_{x^*y_i}^{-2} \log(Kp_{x^*y_i}^{-2}/\delta) + \sum_{x \neq x^*} \sum_{y_1,\ldots,y_s \ | \ p_{xy}<0} p_{xy_i}^{-2} \log(Kp_{xy_i}^{-2}/\delta) \right)$$

many queries, no confidence interval contains zero and the algorithm terminates. This proves the query complexity. The correctness follows from the fact that p_{xy} is always contained in the confidence intervals and that the advice is proper, as detailed in the claim. ☐

Lemma 20. *Given an advice with $\hat{x}$ that is not a Copeland winner, the probability of Algorithm 6 giving an output of 'success' is at most δ.*

Proof. Assume that for all $\hat{y}_1, \ldots, \hat{y}_{K-1-s}$ it holds that $p_{\hat{x}\hat{y}_i} > 0$. This means that $\hat{x}$ suffers at most s losses. Assume further that for all $x \neq \hat{x}$ and all $i \in [s]$, $p_{xy_i(x)} < 0$. This means that all $x \neq \hat{x}$ suffer at least s losses. It follows that these assumptions lead to $\hat{x}$ being a Copeland winner. Since the claim is for the opposite case, we must have $p_{\hat{x}\hat{y}_i} < 0$ for some i or $p_{xy_i(x)} > 0$ for some i.

Hence, if the confidence intervals are always correct, the corresponding pair will never be eliminated from Q and the only possible output for the algorithm is 'fail'. It follows that in order for the algorithm to output 'success' some confidence interval must be wrong and this event happens w.p. at most δ, as required. ☐

Theorem 17 is now an immediate corollary of the above results, combined with those of Appendix A.

C Linear Bandits

C.1 Auxilary Tools

In this section we analyze algorithms that query the unknown vector w at various points and obtain a confidence region for it. We make use of the following lemma providing a high probability confidence region given a set of queried points.

Lemma 21. *Let y be an arbitrary vector in the unit sphere and let $z_1, \ldots, z_t$ be an arbitrary sequence of vectors in the unit sphere. Let $r_1, \ldots, r_t$ be noisy outcomes of queries to $\{w^\top z_i\}_i$ where the noise has mean zero, absolute value of at most 1 almost surely, and is independent in each i. Let $A = \sum z_i z_i^\top$, $b = \sum z_i r_i$ and $\hat{w} = A^{-1}b$. It holds for any $\alpha > 0$ that*

$$\Pr\left[\left| (w-\hat{w})^\top y \right| > \sqrt{\alpha y^\top A^{-1} y} \right] \leq 2\exp\left(-\alpha/2\right)$$

Proof. Denote by Z the matrix whose i'th column is z_i. Denote by ϵ_i the noise in query i meaning $r_i = w^\top z_i + \epsilon_i$ and let ϵ be the vector of length t with the different ϵ_i values. According to the definition of $\hat{w}$ we see that

$$y^\top (w - \hat{w}) = y^\top Z^\dagger \epsilon = \sum (y^\top Z^\dagger)_i \epsilon_i$$

for $Z^\dagger$ being the pseudo-inverse[9] matrix of Z. It follows that the error associated with y is a martingale sum and according to the Azuma-Hoeffding inequality it can be bounded by

$$\Pr\left[\left| (w - \hat{w})^\top y \right| > \sqrt{\alpha} \| y^\top Z^\dagger \| \right] \le 2 \exp\left(-\alpha/2\right)$$

Since $Z^\dagger (Z^\dagger)^\top = A^{-1}$ the claim follows □

In both the exploration and verification algorithms, we use a fixed strategy for querying points. In particular we query a sequence of points $z_1, \ldots, z_t$ that aim to minimize

$$\rho(z_1, \ldots, z_t) = \max_{y \in Y} y^\top \left(\sum_i z_i z_i^\top \right)^{-1} y$$

For some fixed set Y. We follow a common approach in the area of *optimal design of experiments* and choose the z vectors greedily. At each time step we choose the point that minimizes the above expression. In [19], appendix C, a review of this method is given stating that for

$$\rho^*(Y) = \min_p \max_{y \in Y} y^\top \left(\sum_{x \in \mathcal{K}} p_x x x^\top \right)^{-1} y$$

where p is restricted to be a distribution over $\mathcal{K}$, it holds that for any t,

$$\rho^*(Y) \le t \cdot \rho(z_1, \ldots, z_t) \le \rho^*(Y) \cdot (1 + d(d+1)/t) \tag{1}$$

Additionally, in the case where $Y = \mathcal{K}$, it is also known that

$$\rho^*(\mathcal{K}) \le d \tag{2}$$

In the following section we will aim to find a sequence corresponding to a set Y while having access to a different set $Y' = \{\alpha(y) \cdot y \mid y \in Y\}$, where the $\alpha(y)$'s are arbitrary scalars in $[1/c, c]$ for some constant $c > 1$. The following observation provides a guarantee for a sequence w.r.t. Y given its guarantee w.r.t. Y'.

Lemma 22. *Let $Y \subseteq \mathbb{R}^d$, let $Y' = \{\alpha(y) \cdot y \mid y \in Y\}$, where the $\alpha(y)$'s are arbitrary scalars in $[1/c, c]$, with $c > 1$, and let $x_1, \ldots, x_t \in \mathcal{K}$. Assume that*

$$c\rho^*(Y') \ge t \max_{y \in Y'} y^\top \left(\sum_{i=1}^t x_i x_i^\top \right)^{-1} y$$

then we have that

$$c^5 \rho^*(Y) \ge t \max_{y \in Y} y^\top \left(\sum_{i=1}^t x_i x_i^\top \right)^{-1} y$$

Proof. Let

$$p_Y^* = \arg \min_{p \in \Delta_{|\mathcal{K}|}} \max_{y \in Y} y^\top \left(\sum_{x \in \mathcal{K}} p(x) x x^\top \right)^{-1} y$$

be the distribution over the possible set of points achieving $\rho^*(Y)$.

$$\rho^*(Y) = \max_{y \in Y} y^\top \left(\sum_{x \in \mathcal{K}} p_Y^*(x) x x^\top \right)^{-1} y \ge$$

---

[9] If the singular value decomposition of Z is $\sum \sigma_i v_i u_i^\top$ then $Z^\dagger = \sum \sigma_i^{-1} u_i v_i^\top$

$$\frac{1}{c^2}\max_{y\in Y'} y^\top \left(\sum_{x\in\mathcal{K}} p_Y^*(x)xx^\top\right)^{-1} y \geq \frac{1}{c^2}\rho^*(Y')$$

Hence, we get that

$$t\max_{y\in Y} y^\top \left(\sum_{i=1}^t x_i x_i^\top\right)^{-1} y \leq$$

$$c^2 t\max_{y\in Y'} y^\top \left(\sum_{i=1}^t x_i x_i^\top\right)^{-1} y \leq c^3\rho^*(Y') \leq c^5\rho^*(Y)$$

□

C.2 Exploration

Algorithm 7 Linear Bandits Exploration

```
Input: set of arms $\mathcal{K}$, failure probability parameter $\kappa$.
  $A_0 \leftarrow I$
  for all $t = 1, 2, 3, \ldots$ do
    Pick a point $z_t \in \mathcal{K}$ minimizing $\max_{x\in\mathcal{K}} x^\top(A_{t-1} + zz^\top)x$
    Update $A_t = A_{t-1} + z_t z_t^\top$, $b = \sum_{i=1}^t r_i z_i$, $\hat{w} = A_t^{-1}b$
    For $x \in \mathcal{K}$ let $\gamma_x = \sqrt{x^\top A_t^{-1} x \ln(K4t^2/\kappa)/2}$
    If there exist some $\hat{x} \in \mathcal{K}$ such that
        $3(\gamma_x + \gamma_{\hat{x}}) < \hat{w}^\top(\hat{x} - x)$
    for all $x \neq \hat{x}$, output $\hat{x}$ as the best arm and $\hat{\Delta}_x = \hat{w}^\top(\hat{x} - x)$ for all $x \neq \hat{x}$.
  end for
```

The following lemma is immediate given a union bound and Lemma 21.

Lemma 23. *w.p. at least $1-\kappa$ it holds that for all $t > 0$ and all $x \in \mathcal{K}$ that*

$$\left|(w-\hat{w})^\top x\right| < \gamma_x$$

Lemma 24. *Let x^* be the best arm, $\Delta_x = w^\top(x^* - x)$ and $\Delta_{\min} = \min_x \Delta_x$. Given the event of Lemma 23 the algorithm terminates after at most $O\left(\max\left\{d^2, d\ln(Kd/\Delta_{\min}\kappa)/\Delta_{\min}^2\right\}\right)$ queries. Also, $\hat{x}$ is the best arm and $0.5\Delta_x \leq \hat{\Delta}_x \leq 1.5\Delta_x$ for all suboptimal x.*

Proof. We begin with correctness of the algorithm: Since it always holds that

$$\left|(w-\hat{w})^\top x\right| < \gamma_x$$

for all x, upon termination we have that for any $x \neq \hat{x}$,

$$w^\top(\hat{x}-x) > \hat{w}^\top(\hat{x}-x) - \gamma_x - \gamma_{\hat{x}} > 2\gamma_x + 2\gamma_{\hat{x}} > 0$$

meaning that $\hat{x}$ is indeed the optimal arm. It follows that $\Delta_x > 2\gamma_x + 2\gamma_{\hat{x}}$. Since

$$\left|(\hat{w}-w)^\top(\hat{x}-x)\right| < \gamma_x + \gamma_{\hat{x}} < \Delta_x/2$$

the claim regarding $\hat{\Delta}_x$ holds.

We now turn to bounding the query complexity. For $t \geq d(d+1)$ we have for all $x \in \mathcal{K}$ by Equations (1) and (2) that

$$\gamma_x \leq \sqrt{\rho^*(\mathcal{K})\ln(K4t^2/\kappa)/t} \leq \sqrt{d\ln(K4t^2/\kappa)/t}$$

It follows that for $t \geq 64d\ln(K4t^2/\kappa)/\Delta_{\min}^2$ we have

$$\gamma_x < \Delta_{\min}/8$$

for all x. For such t,

$$\hat{w}^\top(x^*-x) \geq w^\top(x^*-x) - \gamma_x - \gamma_{x^*} >$$

$$3\Delta_{\min}/4 \geq 3(\gamma_x + \gamma_{x^*})$$

and the algorithm will terminate after $O\left(d\ln(Kd/\Delta_{\min}^2\kappa)/\Delta_{\min}^2\right)$ queries as required □

C.3 Verification

Algorithm 8 Linear Bandits Verification

```
Input: set of arms $\mathcal{K}$, failure probability parameter $\delta$, candidate winner $\hat{x}$, for any $x \neq \hat{x}$ a parameter $\hat{\Delta}_x > 0$.
  $A_0 \leftarrow I$
  $Y \leftarrow \{\frac{\hat{x}-x}{\hat{\Delta}_x} \mid x \in \mathcal{K},\ x \neq \hat{x}\}$
  for all $t = 1, 2, 3, \ldots$ do
    Pick a point $z_t \in \mathcal{K}$ minimizing $\max_{y \in Y} y^\top (A_{t-1} + zz^\top) y$
    Update $A_t = A_{t-1} + z_t z_t^\top$, $b = \sum_{i=1}^t r_i z_i$, $\hat{w} = A_t^{-1} b$
    For $x \in \mathcal{K}$ let
        $$\gamma_x = \sqrt{(\hat{x}-x)^\top A_t^{-1} (\hat{x}-x) \ln(K4t^2/\delta)/2}$$
    If
        $$\gamma_x < \hat{w}^\top (\hat{x} - x)$$
    for all $x \neq \hat{x}$, output 'success'
    if there exist $x \neq \hat{x}$ for which
        $$\gamma_x < \hat{w}^\top (x - \hat{x})$$
    output 'fail'
  end for
```

As mentioned in the main paper, the query complexity of the algorithm will be expressed in terms of $\rho^*(Y^*)$ defined as

$$\rho^*(Y^*) = \min_p \max_{x \in \mathcal{K}, x \neq x^*} \frac{(x^* - x)^\top \left(\sum_{x \in \mathcal{K}} p_x x x^\top\right)(x^* - x)}{\Delta_x^2}$$

where the minimum is taken over distributions over $\mathcal{K}$

The following lemma is immediate given a union bound and Lemma 21.

Lemma 25. *w.p. at least $1 - \delta$ it holds that for all $t > 0$ and all $x \in \mathcal{K}$ that*

$$\left|(w - \hat{w})^\top (\hat{x} - x)\right| < \gamma_x$$

Lemma 26. *If $\hat{x}$ is the best arm and $\Delta_x/2 \leq \hat{\Delta}_x \leq 1.5\Delta_x$ for all $x \neq \hat{x}$ then given the event of Lemma 25 the algorithm will output 'success' and will terminate after at most $t = O\left(\max\left\{d^2, \rho^*(Y^*)\ln(K\rho^*(Y^*)t/\delta)\right\}\right)$ queries.*

Proof. In order for the algorithm to output 'fail' it must be that

$$\hat{w}^\top (x - \hat{x}) > \gamma_x > w^\top (x - \hat{x}) + \gamma_x$$

which is assumed not to happen, hence the algorithm must output 'success'.

To analyze the query complexity, For $t \geq d(d+1)$ we have for all $x \in \mathcal{K}$ by Equation 1 that

$$\gamma_x < \sqrt{\rho^*(Y)\Delta_x^2 \ln(K4t^2/\delta)/t}$$

hence if $t \geq \sqrt{\rho^*(Y)\ln(K4t^2/\delta)}$ we get that $\gamma_x < \Delta_x$, meaning that for all $x \neq \hat{x}$,

$$\hat{w}^\top (\hat{x} - x) > w^\top (\hat{x} - x) - \gamma_x > 0$$

and the algorithm will terminate. The claim follows from noticing that since $\Delta_x/2 \leq \hat{\Delta}_x \leq 1.5\Delta_x$ it must be the case that $\rho^*(Y) \leq 32\rho^*(Y^*)$ (Lemma 22). ☐

Lemma 27. *If $\hat{x}$ is not the best arm then w.p. at least $1 - \delta$ the algorithm will output 'fail'*

Proof. We prove the lemma conditioned on the occurrence of the event of Lemma 25. Since $\hat{x}$ is suboptimal there must be some $x \in \mathcal{K}$ for which $w^\top(\hat{x} - x) < 0$. For the algorithm to output 'success' it must be the case that

$$\hat{w}^\top(\hat{x} - x) > \gamma_x > w^\top(\hat{x} - x) + \gamma_x \ .$$

But, according to our assumption this can never happen, hence the algorithm must output 'fail' when it terminates. $\square$

Theorem 6 is now an immediate corollary of the above results, combined with those of Appendix A.

D Unimodal Bandits

D.1 Unimodal Bandits for General Graphs

In this section we present the existing results, as well as our own for the unimodal bandit problem with general graphs. To review the existing results we introduce some notations. We denote by d the maximum degree of the graph. For an arm x let $\Gamma(x)$ be the set of arms y in the immediate neighborhood of x in the graph. For a suboptimal arm x we let $\Delta_x^\Gamma = \max_{y \in \Gamma(x)} \mu(y) - \mu(x)$ be the gap between the reward of x and its neighbors and let $\Delta_x = \mu(x^*) - \mu(x)$ be its gap from the best arm x^*. We denote by $\Delta_{\text{min}}^\Gamma$ the minimal value of Δ_x^Γ and Δ_{min} be the minimal value of Δ_x. Notice that $\Delta_{\text{min}}^\Gamma \leq \Delta_{\text{min}}$ and that the ratio between the two is potentially unbounded. Furthermore, it is very often the case that for most x, $\Delta_x^\Gamma \ll \Delta_x$ as x may be a clear bad choice compared to the optimal arm but still have a very close value to those of its immediate neighbors. Consider a subset of the edges forming a spanning tree T over the graph. We say that T is traversable if it preserves the unimodality property, meaning that every $x \neq x^*$ there exist some neighbor in T with a superior reward. Denote by $D(T)$ the diameter, i.e. the longest shortest path between a pair of vertices, in T, and denote by D the maximum value of $D(T)$ with the maximum taken over all possible traversable trees T.

The method GLSE in [13], though aimed for the regret setting can be proved to achieve, for failure probability κ an expected query complexity of $O\left(Dd(\Delta_{\text{min}}^\Gamma)^{-2}\log(1/\kappa) + D\log(D)\right)$ for the special case where the graph is a tree, in which case D is simply the diameter of the tree. In [6], a method OSUB is proposed achieving an expected query complexity of (up to logarithmic terms independent of κ)[10]

$$O\left(\sum_{x \neq x^*} (\Delta_x^\Gamma)^{-2} + \sum_{x \in \Gamma(x^*)} \Delta_x^{-2}\log(1/\kappa)\right)$$

for general graphs. The latter result has a better dependency over κ, and they in fact prove that it is asymptotically optimal when κ tends to 0. However, when compared to GLSE while discussing trees, in some cases the size of the graph could be as large as $K = d^{\Omega(D)}$ in which case the linear dependence over $\mathcal{K}$ can lead to inferior results compared to those of [13].

Our methods lead to two algorithms, differing only in the exploration strategy, each improving a different result of those mentioned above. In the first setting we use the idea of [13] and jump from one vertex to the next, while always increasing the reward of the visited arm. The algorithm is detailed in Appendix D, and achieves an expected query complexity of $O\left(Dd\log\left(Dd/\kappa\Delta_{\text{min}}^\Gamma\right)/(\Delta_{\text{min}}^\Gamma)^2\right)$. The second exploration algorithm is a simple naive application of a best arm identification algorithm that ignores the structure of the problem, e.g. *Exponential Gap-Elimination* [15] that achieves an expected query complexity of

$$O\left(\sum_{x \neq x^*} \Delta_x^{-2}\log\left(\log(1/\Delta_x)/\kappa\right)\right) \leq O\left(K\Delta_{\text{min}}^{-2}\log\left(\log(1/\Delta_{\text{min}})/\kappa\right)\right)$$

---

[10] The result of [6] is in fact tighter in the sense that it takes advantage of the variance of the estimators by using confidence bounds based on KL-divergence. In the case of uniform variance however, the stated results here are accurate. More importantly, the KL-divergence type techniques can be applied here to obtain the same type of guarantees, at the expense of a slightly more technical analysis. For this reason we present the results for the case of uniform variance.

In both cases, the verification algorithm requires only the identity of the candidate best arm as advice. It simply applies a best arm identification algorithm over the candidate arm and its neighborhood. Its expected query complexity, given a correct advice is $O\left(\sum_{x\in\Gamma(x^*)} \Delta_x^{-2}\log\left(\log(1/\Delta_x)/\delta\right)\right)$. The following provides our formal results.

Theorem 28. *Algorithm 1, along with the exploration algorithm detailed in Appendix D.2 and the verification algorithm of* Exponential Gap-Elimination*, applied to the neighborhood of the candidate best arm, finds the best arm w.p. at least $1-\delta$ while using an expected query complexity of*

$$O\left(Dd\log\left(Dd/\Delta_{\min}^{\Gamma}\right)/(\Delta_{\min}^{\Gamma})^2 + \sum_{x\in\Gamma(x^*)} \Delta_x^{-2}\log\left(1/\delta\right)\right)$$

queries. When applied with the exploration algorithm of Exponential Gap-Elimination*, it achieves an expected query complexity of*

$$O\left(\sum_{x\neq x^*} \Delta_x^{-2}\log\left(K/\Delta_{\min}\right) + \sum_{x\in\Gamma(x^*)} \Delta_x^{-2}\log\left(1/\delta\right)\right)$$

Notice that the first result strictly improves that of [13], while being applicable to both trees and general graphs, and the second result improves, in the terms independent of δ, the result of [6], as $\Delta_{\min} \geq \Delta_{\min}^{\Gamma}$.

D.2 Exploration Algorithm

We proceed to provide an exploration algorithm for the graphical unimodal bandit problem. We begin by presenting a sub-procedure that visits a single node in the graph in Algorithm 9.

Algorithm 9 Node Visit in Graphical Unimodal Bandits

```
Input: set of arms $\Gamma$, additional arm $x$ and confidence parameter $\kappa$.
  $Q \leftarrow \Gamma \cup \{x\}$
  for all $t = 1, 2, 3, \dots$ do
    Query each arm in $Q$ once
    Let $\hat{\mu}(y)$ be the empirical reward of arm $y \in Q$.
    Set
        $\gamma = \sqrt{2\ln\left(2(|\Gamma|+1)t^2/\kappa\right)/t}$
    Eliminate all arms $y \in Q$ for which $\hat{\mu}(x) - \hat{\mu}(y) > \gamma$
    If there exists an $y \in Q$ for which $\hat{\mu}(y) - \hat{\mu}(x) > \gamma$, output $y$
    If $Q$ is empty, output 'x'
  end for
```

To provide the analysis of the algorithm we introduce some notations. For an arm x and set Γ, let $\Delta = \max\{\max_{y\in\Gamma}\mu(y) - \mu(x), \min_{y\in\Gamma}\mu(x) - \mu(y)\}$.

Lemma 29. *With probability at least $1-\kappa$ it holds that Algorithm 9 (1) terminates within*

$$O\left(|\Gamma|\log\left(|\Gamma|/\kappa\Delta^2\right)/\Delta^2\right)$$

queries, and (2) if x has the maximal value among $\Gamma\cup\{x\}$ then the output is x, otherwise the output is some y with $\mu(y) > \mu(x)$.

Proof. Recall that our model assumption dictates that the random variables of $\mu(y)$ are in the region $[0,1]$, hence according to Hoeffding's inequality we have that for any arm y and any time t, w.p. at least $1-\kappa/2(|\Gamma|+1)t^2$, $|\hat{\mu}(y)-\mu(y)| < \gamma/2$. It follows via union bound that for all arms and all time steps $|\hat{\mu}(y)-\mu(y)| < \gamma/2$. The claim immediately follows. □

Our exploration algorithm, given in Algorithm 10, consists of applying the above procedure in order to traverse the graph while increasing the μ value until reaching x^*.

Algorithm 10 Graphical Unimodal Bandits Exploration

```
Input: set of arms $\mathcal{K}$, confidence parameter $\kappa$.
  Pick an arbitrary arm $x_1$.
  for all $r = 1, 2, 3, \ldots$ do
    If $r = 1$, set $\Gamma = \Gamma(x_1)$, else set $\Gamma = \Gamma(x_r) \setminus \{x_{r-1}\}$.
    Invoke Algorithm 9 with input $x_r, \Gamma, \kappa' = \kappa/2r^2$
    If the output is $x_r$, halt and output $x_r$ as the best arm. Otherwise, for an output $y$ proceed to the
    next iteration with $x_{r+1} = y$.
  end for
```

The following is immediate given the definitions of D and $\Delta^\Gamma_{\min}$ given in the previous section and the Lemma 29.

Lemma 30. *With probability at least $1 - \kappa$ it holds that Algorithm 10 (1) terminates within*

$$O\left(Dd \log\left(Dd/\kappa\Delta^\Gamma_{\min}\right)/(\Delta^\Gamma_{\min})^2\right)$$

queries, and (2) outputs the best arm x^*.

Theorem 28 is now an immediate corollary of the above results, combined with those of Appendix A.

E Application to Graphical Bandits

The *Graphical Bandits* problem is a variant of the dueling bandit problem presented in [7]. As in the dueling bandit scenario, in each round the user plays a pair of arms (x, y). The difference is an additional restriction where not all pairs are valid but only a subset of the pairs denoted by E. In their paper, the authors assume that each arm has an associated reward $\mu(x) \in [0, 1]$ and that the outcome of a query to (x, y) is a random variable in $[-1, 1]$ with an expected value of $\mu(x) - \mu(y)$. While other variations may be worth considering, we restrict ourselves to the same assumptions here for simplicity.

In [7], the complexity of the problem is tied to the diameter of the graph, meaning the largest distance in edges in the graph $G(\mathcal{K}, E)$ between a pair of arms. Denoting the diameter as D, they provide a best arm identification problem that fails w.p. at most κ and requires an expected query complexity of

$$\frac{KD\log(K/\kappa)\log^2(K)}{\Delta^2_{\min}}$$

with $\Delta_{\min}$ being the gap between the best and second best arm. We use the exact same algorithm for the exploration, as the verification process requires only the identity of a candidate best arm. Given this candidate $\hat{x}$, our verification procedure is defined as follows. It first computes a shortest path (in edges) tree originating from $\hat{x}$, denoted by T. Next, it performs an elimination tournament where at each round the edges leading to surviving arms are queried. That is, once an arm x is known w.h.p. to be beaten by $\hat{x}$, x is removed from the set of surviving arms and any edges in T that do not lead to any other surviving arm, will not be queried again. The formal algorithm and its analysis are given in Appendix E.1. To present its guarantee we introduce a few notations. Let x^* be the best arm and let $\Delta_x = \mu(x^*) - \mu(x)$ be the gap for a suboptimal arm. Let T be the shortest path tree originating from x^* produced by the BFS (breadth first search) algorithm given and let $d(x)$ be the distance from x^* to x in the graph. Let $E(T)$ be the set of edges in T and for an edge e in $E(T)$ let $h_e = \max_x d(x)/\Delta^2_x$ where the maximum is taken over arms x whose path towards them in T contains the edge e. Our verification algorithm in Appendix E.1, given a correct advice and tuned for failure probability δ, achieves an expected query complexity of

$$O\left(\sum_{e\in E(T)} h_e \log\left(Kh_e/\delta\right)\right) \leq O\left(\frac{KD\ln(K/\delta)}{\Delta^2_{\min}}\right)$$

As a corollary, given the techniques of Theorem 3 we get:

Theorem 31. *Algorithm 1, along with the exploration algorithm of [7] and the verification algorithms given in Appendix E.1, finds the best arm w.p. at least* $1-\delta$ *while using an expected amount of at most*

$$O\left(\frac{KD\log\left(KD/\Delta_{\min}\right)\log^2(K)}{\Delta_{\min}^2}+\sum_{e\in E(T)}h_e\log\left(Kh_e/\delta\right)\right)\leq$$

$$O\left(\frac{KD\left(\log\left(KD/\Delta_{\min}\right)\log^2(K)+\log(1/\delta)\right)}{\Delta_{\min}^2}\right)$$

queries.

E.1 Verification in Graphical Bandits

Algorithm 11 is our verification algorithm. For $x\in\mathcal{K}$, denote by $\tilde{\Delta}_x=\mu(\hat{x})-\mu(x)$ the gap between rewards of the candidate arm $\hat{x}$ and that of x. In the algorithm we keep a confidence interval around the empirical estimation $\hat{\Delta}_x$ of $\tilde{\Delta}_x$, of radius γ_x. We use an adaptation of Proposition 1 in [7] to our terminology to prove the required a high probability confidence bound.

Algorithm 11 Graphical Bandits Verification

```
Input: set of arms $\mathcal{K}$, a graph structure $G=(\mathcal{K},E)$, and a candidate winner $\hat{x}$.
  compute the shortest path tree (in edges) originating from $\hat{x}$
  $Q\leftarrow\mathcal{K}\setminus\{\hat{x}\}$
  For every edge $e$ in the tree, keep $n_e\leftarrow 0$,$\hat{\mu}_e\leftarrow 0$.
  for all $t=1,2,3,\ldots$ do
    For arm $x$ denote by $\pi(x)$ the path from $\hat{x}$ to $x$ in the tree.
    Let $E'=\cup_{x\in Q}\pi(x)$
    query each $e\in E'$ once
    Let $\hat{\mu}_e$ be the empirical estimation of the expected value returned by a query to an edge $e$. Set
        $\hat{\Delta}_x=\sum_{e\in\pi(x)}\hat{\mu}_e$ ,
        $\gamma_x=\sqrt{2|\pi(x)|\ln(4Kt^2/\delta)/t}$
    If $\hat{\Delta}_x+\gamma_x<0$ for any $x$, output 'fail'
    Remove all $x$ meeting $\hat{\Delta}_x-\gamma_x>0$ from $Q$
    If $Q$ is empty, output 'success'
  end for
```

Lemma 32. *For any* $x\in\mathcal{K}$ *and any time step* t *we have w.p. at least* $1-\delta/2Kt^2$ *that*

$$\left|\tilde{\Delta}_x-\hat{\Delta}_x\right|<\gamma_x$$

Proof. Due to the definition of $\tilde{\Delta}_x$ we have that for our estimator $\hat{\Delta}_x$ it holds that

$$\hat{\Delta}=\tilde{\Delta}+\frac{1}{t}\sum_{i=1}^{|\pi(x)|t}\epsilon_i$$

with ϵ_i being independent zero mean noise terms in $[-1,1]$. By Hoeffding's inequality we have that for any $\gamma>0$,

$$\Pr\left[\frac{1}{|\pi(x)|}\left|\tilde{\Delta}_x-\hat{\Delta}_x\right|<\gamma\right]\leq 2\exp(-\gamma^2|\pi(x)|t/2)$$

In particular, for γ_x we get

$$\Pr\left[\left|\tilde{\Delta}_x-\hat{\Delta}_x\right|<\gamma_x\right]\leq 2\exp(-\gamma_x^2 t/2|\pi(x)|)=2\exp(-\ln(4Kt^2/\delta))=\frac{\delta}{2Kt^2}$$

□

As a corollary, via a simple union bound argument we obtain

Lemma 33. *With probability at least $1-\delta$ it holds that for all t and all x*

$$\left|\tilde{\Delta}_x - \hat{\Delta}_x\right| < \gamma_x$$

Lemma 34. *If $\hat{x}$ is not the optimal arm then w.p. at least $1-\delta$ the algorithm will output 'fail'*

Proof. Since $\hat{x}$ is not the optimal arm there is some arm x s.t. $\tilde{\Delta}_x < 0$. If the algorithm provided an output of 'success' then x must have been eliminated from Q. At that point we must have had

$$\hat{\Delta}_x > \gamma_x > \tilde{\Delta}_x + \gamma_x$$

and this event can occur w.p. of at most δ ☐

Lemma 35. *Assume that $\hat{x}$ is the best arm. Let e be an edge in T, the shortest path tree originating from x^*, computed by our algorithm. Let $\mathcal{K}(e)$ be the set of arms for which $e \in \pi(x)$, and let $h_e = \max_{x \in \mathcal{K}(e)} |\pi(x)|\Delta_x^{-2}$. Then w.p. at least $1-\delta$ the algorithm will output 'success' and use at most $O\left(\sum_{e \in E(T)} h_e \ln(Kh_e/\delta)\right)$ queries.*

Proof. We analyze the algorithm given the occurrence of the event of Lemma 33, that happens w.p. at least $1-\delta$. We begin with the proof of correctness. If the algorithm provided an output of 'fail' then for some x must have had

$$\hat{\Delta}_x < -\gamma_x < \tilde{\Delta}_x - \gamma_x$$

where the last inequality holds since $\hat{x}$ is the best arm hence $\tilde{\Delta}_x = \Delta_x > 0$. This is a contradiction.

We continue to analyze the query complexity. Consider an arbitrary sub-optimal arm x and let t be such that

$$t \geq 8\Delta_x^{-2}|\pi(x)|\ln(4Kt^2/\delta)$$

We have that $\gamma_x \leq \Delta_x/2$ and

$$\hat{\Delta}_x > \Delta_x - \gamma_x \geq \gamma_x$$

meaning that x is eliminated from Q at that time. The claim regarding the query complexity immediately follows. ☐