Quantum Exploration Algorithms for Multi-Armed Bandits

Daochen Wang,[*1,2] **Xuchen You,**[*1,3] **Tongyang Li,**[†1,3,4] **Andrew M. Childs**[1,3]

[1]Joint Center for Quantum Information and Computer Science, University of Maryland
[2]Department of Mathematics, University of Maryland
[3]Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland
[4]Center for Theoretical Physics, Massachusetts Institute of Technology
{daochen,xyou,amchilds}@umd.edu, tongyang@mit.edu

Abstract

Identifying the best arm of a multi-armed bandit is a central problem in bandit optimization. We study a quantum computational version of this problem with coherent oracle access to states encoding the reward probabilities of each arm as quantum amplitudes. Specifically, we provide an algorithm to find the best arm with fixed confidence based on variable-time amplitude amplification and estimation. This algorithm gives a quadratic speedup compared to the best possible classical result in terms of query complexity. We also prove a matching quantum lower bound (up to poly-logarithmic factors).

Introduction

The multi-armed bandit (MAB) model is one of the most fundamental settings in reinforcement learning. This simple scenario captures crucial issues such as the tradeoff between exploration and exploitation. Furthermore, it has wide applications to areas including operations research, mechanism design, and statistics.

A basic challenge about multi-armed bandits is the problem of *best-arm identification*, where the goal is to efficiently identify the arm with the largest expected reward. This problem captures a common difficulty in practical scenarios, where at unit cost, only partial information about the system of interest can be obtained. A real-world example is a recommendation system, where the goal is to find appealing items for users. For each recommendation, only feedback on the recommended item is obtained. In the context of machine learning, best-arm identification can be viewed as a high-level abstraction and core component of active learning, where the goal is to minimize the uncertainty of an underlying concept, and each step only reveals the label of the data point being queried.

Quantum computing is a promising technology with potential applications to diverse areas including cryptanalysis, optimization, and simulation of quantum physics. Quantum computing devices have recently been demonstrated to experimentally outperform classical computers on a specific sampling task (Arute et al. 2019). While noise limits the current practical usefulness of quantum computers, they can in principle be made fault tolerant and thus capable of executing a wide variety of algorithms. It is therefore of significant interest to understand quantum algorithms from a theoretical perspective to anticipate future applications. In particular, there has been increasing interest in *quantum machine learning* (see for example the surveys by Biamonte et al. 2017; Schuld, Sinayskiy, and Petruccione 2015; Arunachalam and de Wolf 2017; Dunjko and Briegel 2018). In this paper, we study best-arm identification in multi-armed bandits, establishing quantum speedup.

Problem setup. We work in a standard multi-armed bandit setting (Even-Dar, Mannor, and Mansour 2002) in which the MAB has n arms, where arm $i \in [n] \coloneqq \{1, \ldots, n\}$ is a Bernoulli random variable taking value 1 with probability p_i and value 0 with probability $1 - p_i$. Each arm can therefore be regarded as a coin with *bias* p_i. As our algorithms and lower bounds are symmetric with respect to the arms, we assume without loss of generality that $p_1 \geq \cdots \geq p_n$, and denote $\Delta_i \coloneqq p_1 - p_i$ for all $i \in \{2, \ldots, n\}$. We further assume that $p_1 > p_2$, i.e., the best arm is unique. Given a parameter $\delta \in (0, 1)$, our goal is to use as few queries as possible to determine the best arm with probability $\geq 1 - \delta$. This is known as the *fixed-confidence setting*. We primarily characterize complexity in terms of the parameter

$$H \coloneqq \sum_{i=2}^{n} \frac{1}{\Delta_i^2} \tag{1}$$

which arises in the analysis of classical MAB algorithms (as discussed below).

We consider a quantum version of best-arm identification in which we can access the arms *coherently*. This means we have access to a quantum oracle $\mathcal{O}$ that acts as

$$\begin{aligned} \mathcal{O}\colon\ & |i\rangle_I |0\rangle_B |0\rangle_J \\ & \mapsto |i\rangle_I \left(\sqrt{p_i}\,|1\rangle_B |v_i\rangle_J + \sqrt{1-p_i}\,|0\rangle_B |u_i\rangle_J\right), \end{aligned} \tag{2}$$

where $|v_i\rangle$ and $|u_i\rangle$ are arbitrary states, for all $i \in [n]$. We have used standard Dirac notation which we review in the Preliminaries section. Register I is the "index" register with n states that correspond to the n arms. Register

[*]Equal contribution. Full version of the paper (including the supplementary material) is at https://arxiv.org/abs/2007.07049

[†]Corresponding author. Email: tongyang@mit.edu

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

B is the single-qubit "bandit" register with two states, $|1\rangle$ corresponding to a reward and $|0\rangle$ corresponding to no reward. Register J is a multi-qubit "junk" register. For convenience, we omit register labels when this causes no confusion. Compared to pulling an arm classically—which can be implemented by measuring the bandit register—the quantum oracle allows access to different arms in superposition, a necessary feature for quantum speedup. In real-world applications, we usually have junk when instantiating our oracle (see below). When deriving our results however, we will assume there is no junk (i.e., we set $|v_i\rangle = |u_i\rangle = 1$ for all $i \in [n]$ in (2)). This is without loss of generality as the algorithm we construct is insensitive to junk.

Previous work on quantum algorithms for clustering (Kerenidis et al. 2019; Wiebe, Kapoor, and Svore 2015) and reinforcement learning (Dunjko, Taylor, and Briegel 2016; Dunjko and Briegel 2018) has discussed how to instantiate $\mathcal{O}$. In clustering, $\mathcal{O}$ is created using the SWAP test where for each i, p_i encodes the distance between some fixed vector and the i^{th} vector in some collection. Our algorithm can be used to speed up the algorithms of Kerenidis et al. (2019) and Wiebe, Kapoor, and Svore (2015). In reinforcement learning, $\mathcal{O}$ naturally appears in stochastic agent environments; for instance, $\mathcal{O}$ can be viewed as a special case of the oracle in Dunjko, Taylor, and Briegel (2016) for a Markov decision problem (MDP) of epoch length 1 and state set $\{0, 1\}$, where the goal of the agent is to reach the state 1.

As a concrete example, consider a classical Monte Carlo strategy[1]: at a given position, evaluate the quality of a next move i by uniformly randomly playing out games $x \in X(i)$, where $X(i)$ is the set of valid games from move i onwards, and querying a computer program f that computes a bit $f(i, x) \in \{0, 1\}$ indicating if game x is won (1) or lost (0). In the classical case, we obtain one sample of win or loss using one query to f. In the quantum case, we can also instantiate one query to the quantum oracle in Eq. (2) using just one query to f. To do this, we apply the circuit for f, made reversible in the usual way (Nielsen and Chuang 2000, Sec. 1.4.1), on the quantum state corresponding to uniformly random play as follows:

$$\begin{aligned} &|i\rangle\,|0\rangle\,\frac{1}{\sqrt{|X(i)|}}\sum_{x\in X(i)}|x\rangle \\ &\overset{f}{\mapsto}|i\rangle\sum_{x\in X(i)}\frac{1}{\sqrt{|X(i)|}}\,|f(i,x)\rangle\,|x\rangle \\ &=|i\rangle\,(\sqrt{p_i}\,|1\rangle\,|u_i\rangle+\sqrt{1-p_i}\,|0\rangle\,|v_i\rangle), \end{aligned} \tag{3}$$

where $|u_i\rangle$ and $|v_i\rangle$ are some states, and p_i is the empirical probability that move i leads to a win. Our quantum algorithm then uses quadratically fewer calls to f compared with classical Monte Carlo search to find the best next move.

We stress that we do not need to know the p_is to instantiate the quantum oracle above. We also remark that our algorithm does not apply to every MAB situation. For example, in clinical trials to identify the best drug, we cannot instantiate the quantum oracle because human participants, unlike computer programs, cannot be queried in superposition.

[1]This is Monte Carlo tree search without tree expansion.

Our algorithm can also be adapted to work when the reward distributions are promised to have bounded variance (for example, if they are sub-Gaussian). The adaptation essentially follows by replacing amplitude estimation (introduced in the Preliminaries section) with quantum mean estimation (Montanaro 2015), which works on any distribution with bounded variance. We remark that the situation is different for the other main type of bandits: adversarial bandits. Studies on adversarial bandits are mainly focused on regret minimization and a quantum analogue first requires a proper notion of regret which we are unsure how to even define.

Contributions. In this paper, we give a comprehensive study of best-arm identification using quantum algorithms. Specifically, we obtain the following main result:

Theorem 1. *Given a multi-armed bandit oracle $\mathcal{O}$ and confidence parameter $\delta \in (0, 1)$, there exists a quantum algorithm that, with probability $\geq 1 - \delta$, outputs the best arm using $\tilde{O}\big(\sqrt{H}\big)$ queries to $\mathcal{O}$. Moreover, this query complexity is optimal up to poly-logarithmic factors in n, δ, and Δ_2.*

This represents a quadratic quantum speedup over what is possible classically. The speedup essentially derives from Grover's search algorithm (Grover 1996), where a marker oracle is used to approximately "rotate" a uniform initial state to the marked state. One way to understand the quadratic speedup is to observe that each rotation step, making one query to the oracle, increases the amplitude of the marked state by $\Omega(1/\sqrt{n})$. This is possible since quantum computation linearly manipulates amplitudes, which are square roots of probabilities.

However, to establish Theorem 1 we use more sophisticated machinery that extends Grover's algorithm, namely variable-time amplitude amplification (VTAA) (Ambainis 2010b; Childs, Kothari, and Somma 2017) and estimation (VTAE) (Chakraborty, Gilyén, and Jeffery 2019). We apply VTAA and VTAE on a variable-time quantum algorithm $\mathcal{A}$ that we construct. $\mathcal{A}$ outputs a state with labeled "good" and "bad" parts. Using that label, VTAA removes the bad part so that only the good part remains, and VTAE estimates the proportion of the good part. In our application, the good part is eventually the best-arm state.

We emphasize that our quantum algorithm, like classical ones (Even-Dar, Mannor, and Mansour 2002; Gabillon, Ghavamzadeh, and Lazaric 2012; Jamieson et al. 2014; Karnin, Koren, and Somekh 2013; Mannor and Tsitsiklis 2004), does not require any prior knowledge about the p_is.

Given knowledge of p_1 and p_2, our quantum algorithm is conceptually related to the classical successive elimination (SE) algorithm (Even-Dar, Mannor, and Mansour 2002). Namely, we use that knowledge to help eliminate sub-optimal arms i by checking whether $p_i < (p_1 + p_2)/2$, say. The quantum quadratic speedup arises because we can check this "in superposition" across the different arms. For intuition only, checking in superposition can be thought of as a form of checking in parallel. We stress however that while it does not make sense to compare the parallel (classical) sample complexity of best-arm identification with its usual

(classical) sample complexity, it does makes sense to compare the latter with the quantum query complexity. We also stress that the similarity of our quantum algorithm to SE, given knowledge of p_1 and p_2, ends at the conceptual level. Technically, our algorithm makes the SE concept work by first marking all sub-optimal arms and then rotating towards the unmarked best arm in quantum state space via a careful application of VTAA. This has no classical analogue.

It is classically easy to remove any assumed knowledge of p_1 and p_2 because classical samples from a multi-armed bandit contain information about their values. Quantumly however, we cannot simply ask our quantum multi-armed bandit to supply *classical* samples as that would prevent interference, eliminating any quantum speedup. Therefore, we need to do something conceptually different in the quantum case. We construct another quantum algorithm whose goal is to estimate both p_1 and p_2 to precision $\Theta(\Delta_2)$ using $\tilde{O}(\sqrt{H})$ quantum queries. For a given test point l, VTAE (roughly) gives us the ability to *count* the number of arms i with $p_i > l$, and thus allows us to perform binary search to find p_1 and p_2.

Related work. Classically, a naive algorithm for best-arm identification is to simply sample each arm the same number of times and output the arm with the best empirical bias (Even-Dar, Mannor, and Mansour 2002). This algorithm has complexity $O(\frac{n}{\Delta_2^2}\log(\frac{n}{\delta}))$ but is sub-optimal for most multi-armed bandit instances. Therefore, classical research on best-arm identification (Even-Dar, Mannor, and Mansour 2002; Gabillon, Ghavamzadeh, and Lazaric 2012; Jamieson et al. 2014; Karnin, Koren, and Somekh 2013; Mannor and Tsitsiklis 2004) has primarily focused on proving bounds of the form $\tilde{O}(H)$ (recall that $H \coloneqq \sum_{i=2}^{n}\frac{1}{\Delta_i^2}$), which can be shown to be almost tight for every instance. The first work to provide an algorithm with such complexity is Even-Dar, Mannor, and Mansour (2002), giving $O(H\log(\frac{n}{\delta}) + \sum_{i=2}^{n}\Delta_i^{-2}\log(\Delta_i^{-1}))$. This was further improved to $O\big(H\log(\frac{1}{\delta}) + \sum_{i=2}^{n}\Delta_i^{-2}\log\log(\Delta_i^{-1})\big)$ by Gabillon, Ghavamzadeh, and Lazaric (2012); Jamieson et al. (2014); Karnin, Koren, and Somekh (2013), which is almost optimal except for the additive term of $\sum_{i=2}^{n}\Delta_i^{-2}\log\log(\Delta_i^{-1})$ (Mannor and Tsitsiklis 2004). More recent work (Chen and Li 2015; Chen, Li, and Qiao 2017) has focused on bringing down even this additive term by tightening both the upper and lower bounds, leaving behind a gap only of the order $\sum_{i=2}^{n}\Delta_i^{-2}\log\log(\min\{n,\Delta_i^{-1}\})$.

Prior work on quantum machine learning has focused primarily on supervised (Lloyd, Mohseni, and Rebentrost 2014, 2013; Rebentrost, Mohseni, and Lloyd 2014; Li, Chakrabarti, and Wu 2019) and unsupervised learning (Lloyd, Mohseni, and Rebentrost 2013; Wiebe, Kapoor, and Svore 2015; Amin et al. 2018; Kerenidis et al. 2019). Dunjko, Taylor, and Briegel (2017); Dunjko et al. (2017); Jerbi et al. (2019) gave quantum algorithms for general reinforcement learning with provable guarantees, but do not consider the best-arm identification problem. The only directly comparable previous work on quantum algorithms for best-arm identification that we are aware of are Casalé et al. (2020) and Wiebe, Kapoor, and Svore (2015).[2] By applying Grover's algorithm, Casalé et al. (2020) shows that quantum computers can find the best arm with confidence $p_1/\sum_{i=1}^{n}p_i$ quadratically faster than classical ones. However, Casalé et al. (2020) does not show how to find the best arm with a given *fixed* confidence, which is the standard requirement. In fact, there is a relatively simple quantum algorithm, analogous to the naive classical algorithm, that can achieve arbitrary confidence with quadratic speedup in terms of n/Δ_2^2. This algorithm, which appears in Fig. 3 of Wiebe, Kapoor, and Svore (2015), works by using the quantum minimum finding of Dürr and Høyer (1996) on top of quantum amplitude estimation (Brassard et al. 2002). As in the classical case, we show that this simple quantum algorithm is suboptimal for most multi-armed bandit instances. Specifically, we show that a quantum algorithm can achieve quadratic speedup in terms of the parameter H.

Preliminaries

Definitions and notations. Quantum computing is naturally formulated in terms of linear algebra. An n-dimensional *quantum state* is a unit vector in the complex Hilbert space $\mathbb{C}^n$, i.e., $\vec{x} = (x_1,\ldots,x_n)^\top$ such that $\sum_{i=1}^{n}|x_i|^2 = 1$. Such a column vector $\vec{x}$ is written in *Dirac notation* as $|x\rangle$ and called a "ket". The complex conjugate transpose of $|x\rangle$ is written $\langle x|$ and called a "bra", i.e., $\langle x| \coloneqq \vec{x}^\dagger$. The reason for the names is because the combination of a bra and a ket is a inner product bracket: $\langle x|y\rangle \coloneqq \langle x|\,|y\rangle = \vec{x}^\dagger\vec{y} = \langle x,y\rangle \in \mathbb{C}$.

The *computational basis* of $\mathbb{C}^n$ is the set of vectors $\{\vec{e}_1,\ldots,\vec{e}_n\}$, where $\vec{e}_i = (0,\ldots,1,\ldots,0)^\top$ is a one-hot column vector with 1 in the i^{th} coordinate. In Dirac notation, it is common to reserve symbols $|i\rangle \coloneqq \vec{e}_i$ and $\langle i| \coloneqq \vec{e}_i^\dagger = \vec{e}_i^\top$. Then, for example, $|x\rangle = \sum_{i=1}^{n}x_i\,|i\rangle$ and $\langle x| = \sum_{i=1}^{n}x_i^*\,\langle i|$.

The *tensor product* of quantum states is their Kronecker product: if $|x\rangle \in \mathbb{C}^{n_1}$ and $|y\rangle \in \mathbb{C}^{n_2}$, then

$$|x\rangle\,|y\rangle \coloneqq |x\rangle \otimes |y\rangle \tag{4}$$

$$\coloneqq (x_1y_1, x_1y_2, \ldots, x_{n_1}y_{n_2})^\top \in \mathbb{C}^{n_1}\otimes\mathbb{C}^{n_2}. \tag{5}$$

A quantum algorithm is a sequence of unitary matrices, i.e., a linear transformation U such that $U^\dagger = U^{-1}$.

For any $p \in [0,1]$, we define the *coin state* in $\mathbb{C}^2$ as

$$|\mathrm{coin}\,p\rangle \coloneqq \sqrt{p}\,|1\rangle + \sqrt{1-p}\,|0\rangle = (\sqrt{1-p},\sqrt{p})^\top. \tag{6}$$

Measuring $|\mathrm{coin}\,p\rangle$ in the computational basis gives 1 with probability p, hence the name.

Quantum multi-arm bandit oracle. Recall the quantum multi-armed bandit oracle defined in (2). The arms are accessed in *superposition* by applying the unitary oracle $\mathcal{O}$ on

[2] Wiebe, Kapoor, and Svore (2015) is not framed as solving best-arm identification, but is partly concerned with this problem.

a state $|x\rangle_I |0\rangle_B$ in the joint register of I and B. This results in the output quantum state

$$\mathcal{O}|x\rangle_I |0\rangle_B = \sum_{i=1}^{n} x_i |i\rangle_I |\mathrm{coin}\, p_i\rangle_B \tag{7}$$

(recall that we assume there is no junk). A classical pull of the i-th arm can be simulated by choosing $|x\rangle_I = |i\rangle_I$ with $|i\rangle_I |\mathrm{coin}\, p_i\rangle_B$ as the output, and then measuring register B to observe 1 with probability p_i.

In this paper, we mainly focus on *quantum query complexity*, which is defined as the total number of oracle queries. If we have an efficient quantum algorithm for an explicit computational problem in the query complexity setting, then if we are given an explicit circuit realizing the black-box transformation, we will have an efficient quantum algorithm for the problem.

Amplitude amplification and estimation. Our quantum speed-up can be traced back to *amplitude amplification and estimation* (Brassard et al. 2002). For a classical randomized algorithm for a search problem that returns a correct solution y with probability p_{succ}, the success probability can be amplified to a constant by $O(1/p_{\mathrm{succ}})$ repetitions. Let $\mathcal{A}$ be a quantum procedure that outputs a quantum state $\sqrt{p_{\mathrm{succ}}}|1\rangle|y\rangle + \sqrt{1-p_{\mathrm{succ}}}|0\rangle|y'\rangle$ for some arbitrary quantum state $|y'\rangle$. Measuring the output state yields the solution y with probability p_{succ} just like a classical randomized algorithm. Brassard et al. (2002) provided an amplitude amplification procedure that amplifies the amplitude of $|1\rangle|y\rangle$ to a constant with $O(1/\sqrt{p_{\mathrm{succ}}})$ queries to the quantum procedure $\mathcal{A}$. This effectively provides a randomized algorithm with constant success probability with query complexity $O(t/\sqrt{p_{\mathrm{succ}}})$ if $\mathcal{A}$ makes t queries to the oracle. The same speed-up can be achieved for the closely related task of estimating p_{succ} with *amplitude estimation*.

Amplitude amplification and estimation originates from *Grover's search algorithm.* (Grover 1996). The formal statements of Grover's algorithm and amplitude amplification and estimation are postponed to the start of the appendix. We refer the interested reader to the book Nielsen and Chuang (2000) on quantum computing for a detailed introduction to basic definitions (Section 3), Grover's algorithm and amplitude amplification (Section 6), and related topics.

Variable-time amplitude amplification and estimation. *Variable-time amplitude amplification* (VTAA) and *estimation* (VTAE) are procedures that apply on top of so-called variable-time quantum algorithms that may stop at different (variable) time steps with certain probabilities. More precisely, for $t = (t_1, t_2, \cdots, t_m) \in \mathbb{R}^m$ and $w = (w_1, w_2, \cdots, w_m) \in \mathbb{R}^m$, a (t, w)-variable-time algorithm $\mathcal{A}$ is one that can be divided into m steps (i.e., $\mathcal{A} = \mathcal{A}_m \cdots \mathcal{A}_1$) where t_j is the query complexity of $\mathcal{A}_j \cdots \mathcal{A}_1$ and w_j is the probability of stopping at step j. We have:

Theorem 2 (Informal: Variable-time amplitude amplification and estimation–Ambainis 2010b; Childs, Kothari, and Somma 2017; Chakraborty, Gilyén, and Jeffery 2019)**.** *Given a (t, w)-variable-time quantum algorithm $\mathcal{A} = \mathcal{A}_m \cdots \mathcal{A}_1$ with success probability p_{succ}, there exists a quantum algorithm $\mathcal{A}'$ that uses $O(Q)$ queries to output the solution with probability $\geq \frac{1}{2}$, where*

$$Q \coloneqq t_m \log(t_m) + \frac{t_{\mathrm{avg}}}{\sqrt{p_{\mathrm{succ}}}} \log(t_m). \tag{8}$$

with $t_{\mathrm{avg}} \coloneqq \sqrt{\sum_{j=1}^{m} w_j t_j^2}$ being the root-mean-square average query complexity of $\mathcal{A}$.

There also exists a quantum algorithm that uses $O(\frac{Q}{\epsilon} \log^2(t_m) \log\log(\frac{t_m}{\delta}))$ queries to estimate p_{succ} with multiplicative error ϵ with probability $\geq 1 - \delta$.

For comparison, recall that applying amplitude amplification and estimation procedures on general quantum algorithms requires $O(t_m/\sqrt{p_{\mathrm{succ}}})$ queries. See the first section of the appendix for a rigorous definition of variable-time algorithms and formal statements of the query complexities of variable-time amplitude amplification and estimation.

Fast Quantum Algorithm For Best-arm Identification

In this section, we construct a quantum algorithm for best-arm identification and analyze its performance. Specifically:

Theorem 3. *Given a multi-armed bandit oracle $\mathcal{O}$ and confidence parameter $\delta \in (0, 1)$, there exists a quantum algorithm that outputs the best arm with probability $\geq 1 - \delta$ using $\tilde{O}(\sqrt{H})$ queries to $\mathcal{O}$.*

Throughout this section, the oracle $\mathcal{O}$ is fixed, so we may omit explicit reference to it. All logs have base 2.

There are essentially two steps in our construction. In the first step, we construct two subroutines Amplify and Estimate using VTAA and VTAE, respectively, on a variable-time quantum algorithm $\mathcal{A}$. Roughly speaking, given $l \in [0, 1]$, Amplify outputs an arm index i randomly chosen from those i with $p_i > l$ while Estimate counts the number of such is. This means that if we knew the values of p_1 and p_2, we could take l to be $(p_1 + p_2)/2$, then Amplify would output the best arm. But we can use Estimate in a binary search procedure to estimate p_1 and p_2. This is exactly what we do in the second step and so we are done.

We now discuss the construction more precisely. Amplify and Estimate actually use two thresholds l_2, $l_1 \in [0, 1]$ with $l_2 < l_1$ instead of a single threshold l. In the first step, we construct a variable-time quantum algorithm denoted $\mathcal{A}$ (Algorithm 1) that is initialized in a uniform superposition state $|u\rangle \coloneqq \frac{1}{\sqrt{n}} \sum_{i \in [n]} |i\rangle$ (since initially we have no information about which arm is the best). Given an input interval $I = [l_2, l_1]$, $\mathcal{A}$ "flags" arm indices in $S'_{\mathrm{right}} \coloneqq \{i \in [n] : p_i \geq l_1\}$ with a bit $f = 1$ and those in $S'_{\mathrm{left}} \coloneqq \{i \in [n] : p_i \leq l_2\}$ with a bit $f = 0$. The flag bit f is written to a separate flag register F, so that the state (approximately) becomes $\frac{1}{\sqrt{n}}\big(\sum_{i \in S'_{\mathrm{right}}} |i\rangle |1\rangle_F + \sum_{i \in S'_{\mathrm{left}}} |i\rangle |0\rangle_F + \sum_{i \in S'_{\mathrm{middle}}} |i\rangle |\psi_i\rangle_F\big)$ for some states $|\psi_i\rangle \in \mathbb{C}^2$, where $S'_{\mathrm{middle}} \coloneqq [n] - (S'_{\mathrm{left}} \cup S'_{\mathrm{right}}) = \{i \in [n] : l_2 < p_i < l_1\}$. The flag bit f stored in the F

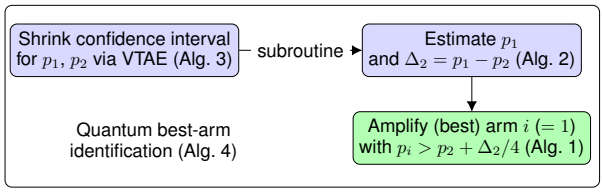

Figure 1: Overview of our best-arm identification algorithm.

register indicates whether VTAA (resp. VTAE), when applied on $\mathcal{A}$, should ($f = 1$) or should not ($f = 0$) amplify (resp. estimate) that part of the state. We then apply VTAA and VTAE on $\mathcal{A}$ to construct Amplify and Estimate, respectively. Amplify produces a uniform superposition of all those is with F register in $|1\rangle$, i.e., it amplifies such is relative to the others. Estimate counts the number of such is. More precisely, Estimate (approximately) counts the number of indices in S'_{right}, as their F register is in $|1\rangle$, plus some (unknown) fraction of indices in S'_{middle} as dictated by the fraction of $|1\rangle$ in the (unknown) states $|\psi_i\rangle$.

In the second step, we use Estimate as a subroutine in Locate (Algorithm 2) to find a interval $[l_2, l_1]$ such that $p_2 < l_2 < l_1 < p_1$ and that $|l_1 - l_2| \geq \Delta_2/4$. Then, running Amplify with these l_2, l_1 in BestArm (Algorithm 4) gives the state $|1\rangle$ containing the best-arm index because only p_1 is to the right of l_2. Locate is a type of binary search that counts the number of indices in S'_{right} using Estimate. There is a technical difficulty here because Estimate actually counts the number of indices in S'_{right} plus some fraction of indices in S'_{middle}. Trying to fix this by simply setting $l_2 = l_1$, so that $S'_{\text{middle}} = \emptyset$, does not work as it would increase the cost of Estimate. We overcome this difficulty via the Shrink subroutine (Algorithm 3) of Locate, which employs a technique from recent work on quantum ground state preparation (Lin and Tong 2020). See Figure 1 for an illustration of the overall structure of the algorithm.

Amplify **and** Estimate

We first construct a variable-time quantum algorithm (Algorithm 1) that we call $\mathcal{A}$ throughout. $\mathcal{A}$ uses the following registers: input register I; bandit register B; clock register $C = (C_1, \ldots, C_{m+1})$, where each C_i is a qubit; ancillary amplitude estimation register $P = (P_1, \ldots, P_m)$, where each P_i has $O(m)$ qubits; and flag register F. We set $m \coloneqq \lceil \log(1/(l_1 - l_2)) \rceil + 2$ as assigned in Algorithm 1.

$\mathcal{A}$ is indeed a variable-time quantum algorithm according to Definition 1. This is because we can write $\mathcal{A} = \mathcal{A}_{m+1}\mathcal{A}_m \cdots \mathcal{A}_1\mathcal{A}_0$ as a product of $m + 2$ sub-algorithms, where $\mathcal{A}_0$ is the initialization step (Line 4), $\mathcal{A}_j$ consists of the operations in iteration j of the for loop (Lines 6–9) for $j \in [m]$, and $\mathcal{A}_{m+1}$ is the termination step (Lines 10–11). The state spaces $\mathcal{H}_C$ and $\mathcal{H}_A$ in Definition 1 correspond to the state spaces of the C register and the remaining registers of $\mathcal{A}$, respectively. $\mathcal{A}_{m+1}$ ensures that Condition 4 of Definition 1 is satisfied.

With $\Delta \coloneqq l_1 - l_2$ being the length of $[l_2, l_1]$, we define

Algorithm 1: $\mathcal{A}(\mathcal{O}, l_2, l_1, \alpha)$

Input: Oracle $\mathcal{O}$ as in (2); $0 < l_2 < l_1 < 1$; approximation parameter $0 < \alpha < 1$.

1 $\Delta \leftarrow l_1 - l_2$
2 $m \leftarrow \lceil \log \frac{1}{\Delta} \rceil + 2$
3 $a \leftarrow \frac{\alpha}{2mn^{3/2}}$
4 Initialize state to $\frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle_I |\text{coin}\, p_i\rangle_B |0\rangle_C |0\rangle_P |1\rangle_F$
5 **for** $j = 1, \ldots, m$ **do**
6 $\quad \epsilon_j \leftarrow 2^{-j}$
7 $\quad$ **if** *register I is in state $|i\rangle$ and registers $C_1, \ldots, C_{j-1}$ are in state $|0\rangle$* **then**
8 $\quad\quad$ Apply $\mathsf{GAE}(\epsilon_j, a; l_1)$ with $\mathcal{O}_{p_i}$ on registers B, C_j, and P_j
9 $\quad$ Apply controlled-NOT gate with control on register C_j and target on register F
10 **if** *registers $C_1, \ldots, C_m$ are in state $|0\rangle$* **then**
11 $\quad$ Flip the bit stored in register C_{m+1}

the following three sets that partition $[n]$:

$$S_{\text{left}} \coloneqq \{i \in [n] : p_i < l_1 - \Delta/2\}, \tag{9}$$

$$S_{\text{middle}} \coloneqq \{i \in [n] : l_1 - \Delta/2 \leq p_i < l_1 - \Delta/8\}, \tag{10}$$

$$S_{\text{right}} \coloneqq \{i \in [n] : p_i \geq l_1 - \Delta/8\}. \tag{11}$$

These sets play the roles of aforementioned S'_{left}, S'_{middle}, and S'_{right}. They can be regarded as functions of (the input to) $\mathcal{A}$. For later convenience, we also define $S_{\text{lm}} \coloneqq S_{\text{left}} \cup S_{\text{middle}}$ and $S_{\text{mr}} \coloneqq S_{\text{middle}} \cup S_{\text{right}}$.

Lemma 1 (Correctness of $\mathcal{A}$)**.** *Let p_{succ} denote the success probability $\mathcal{A}$. Then $|p_{\text{succ}} - p'_{\text{succ}}| \leq \frac{2\alpha}{n}$ where $p'_{\text{succ}} = \frac{1}{n}\big(|S_{\text{right}}| + \sum_{i \in S_{\text{middle}}} |\beta_{i,1}|^2\big)$ for some $|\beta_{i,1}|^2 \in [0, 1]$.*

At a high level, at iteration j, Line 8 approximately identifies those $i \in S_{\text{left}}$ with $p_i \in [l_1 - 2\epsilon_j, l_1 - \epsilon_j)$ and stops computation on these is by setting their associated C registers to $|1\rangle$. Line 9 then flags these is by setting their associated F registers to $|0\rangle$, indicating failure. We defer the detailed proof to the supplementary material which is mainly concerned with bounding the error in the aforementioned approximation, as well as the lemma as follows.

Lemma 2 (Complexity of $\mathcal{A}$)**.** *With $\Delta = l_1 - l_2$ being the length of the interval, we have:*

1. *The j^{th} stopping time t_j of $\mathcal{A}_j\mathcal{A}_{j-1}\cdots\mathcal{A}_0$ is of order $\sum_{k=1}^{j} \frac{1}{\epsilon_k} \log \frac{1}{a} \leq 2^{j+1} \log \frac{1}{a}$. In particular, $t_{m+1} = O(\frac{1}{\Delta} \log \frac{1}{a})$.*
2. *The average stopping time squared, t^2_{avg}, is of order*

$$\frac{1}{n}\left(\frac{|S_{\text{right}}|}{\Delta^2} + \sum_{i \in S_{\text{lm}}} \frac{1}{(l_1 - p_i)^2}\right) \log^2\Big(\frac{1}{a}\Big). \tag{12}$$

Now we fix algorithm $\mathcal{A}$ and its input parameters. We always assume that $|S_{\text{right}}| \geq 1$, which we need for some of the following results to hold. This is without loss of generality as we can always add an artificial arm 0 with bias $p_0 = 1$ to the bandit oracle $\mathcal{O}$, as we do in Line 3 of Algorithm 3.

Algorithm 2: $\mathsf{Locate}(\mathcal{O}, \delta)$

Input: Oracle $\mathcal{O}$ as in (2); confidence parameter $0 < \delta < 1$.

$I_1, I_2 \leftarrow [0, 1]$
$\delta \leftarrow \delta/8$
while $\min I_1 - \max I_2 < 2\,|I_1|$ **do**
  $I_1 \leftarrow \mathsf{Shrink}(\mathcal{O}, 1, I_1, \delta)$
  $I_2 \leftarrow \mathsf{Shrink}(\mathcal{O}, 2, I_2, \delta)$
  $\delta \leftarrow \delta/2$
return I_1, I_2

We apply VTAA and VTAE (Theorem 2)[3] on our variable-time quantum algorithm $\mathcal{A}$ to prepare the state $|\psi_{\text{succ}}\rangle$ and to estimate the probability p_{succ}, respectively. This gives two new algorithms Amplify and Estimate with the following performance guarantees.

Lemma 3 (Correctness and complexity of $\mathsf{Amplify}(\mathcal{A}, \delta)$, $\mathsf{Estimate}(\mathcal{A}, \epsilon, \delta)$)**.** *Let $\mathcal{A} = \mathcal{A}(\mathcal{O}, l_2, l_1, 0.01\delta)$. Then $\mathsf{Amplify}(\mathcal{A}, \delta)$ uses $O(Q)$ queries to output an index $i \in S_{\text{mr}}$ with probability $\geq 1 - \delta$, and $\mathsf{Estimate}(\mathcal{A}, \epsilon, \delta)$ uses $O(Q/\epsilon)$ queries to output an estimate r of p'_{succ} (defined in Lemma 1) such that*

$$(1-\epsilon)\Big(p'_{\text{succ}} - \frac{0.1}{n}\Big) < r < (1+\epsilon)\Big(p'_{\text{succ}} + \frac{0.1}{n}\Big) \quad (13)$$

with probability $\geq 1 - \delta$, where Q is

$$\left(\frac{1}{\Delta^2} + \frac{1}{|S_{\text{right}}|}\sum_{S_{\text{lm}}} \frac{1}{(l_1 - p_i)^2}\right) \text{poly}\Big(\log\Big(\frac{n}{\delta\Delta}\Big)\Big), \quad (14)$$

where $\Delta = l_1 - l_2$.

This lemma follows by applying Lemma 1 and Lemma 2 to Theorem 2. The proof detail is given in the appendices.

Quantum Algorithm for Best-arm Identification

In this subsection, we use Amplify and Estimate to construct three algorithms (Algorithms 2–4) that work together to identify the best arm following the outline that we described at the beginning of this section.

We state the correctness and complexities of Amplify and Estimate as follows:

Lemma 4 (Correctness and complexity of Algorithm 2)**.** *Fix a confidence parameter $0 < \delta < 1$. Then the event $E = \{p_1 \in I_1 \text{ and } p_2 \in I_2 \text{ in all iterations of the while loop}\}$ holds with probability $\geq 1 - \delta$. When E holds, Algorithm 2 also satisfies the following for both $k \in \{1, 2\}$:*

1. *its while loop (Line 3) breaks at or before the end of iteration $\lceil \log_{5/3}(\frac{1}{\Delta_2}) \rceil + 3$ and then returns I_k with $p_k \in I_k$ and $\min I_1 - \max I_2 \geq 2\,|I_1|$; during the while loop, we always have $|I_1| = |I_2| \geq \Delta_2/8$; and*
2. *it uses $O\big(\sqrt{H}\ \text{poly}\big(\log\big(\frac{n}{\delta\,\Delta_2}\big)\big)\big)$ queries.*

[3]The state spaces $\mathcal{H}_C$, $\mathcal{H}_F$, and $\mathcal{H}_W$ correspond to the state spaces of the C, F, and remaining registers of $\mathcal{A}$, respectively.

Algorithm 3: $\mathsf{Shrink}(\mathcal{O}, k, I, \delta)$

Input: Oracle $\mathcal{O}$ as in (2); $k \in \{1, 2\}$; interval $I = [a, b]$; confidence parameter $0 < \delta < 1$.

$\epsilon \leftarrow (b - a)/5$
$\delta \leftarrow \delta/2$
Append arm $i = 0$ with bias $p_0 = 1$ to $\mathcal{O}$; call the resulting oracle $\mathcal{O}'$
Construct variable-time quantum algorithms $\mathcal{A}_1, \mathcal{A}_2$:
  $\mathcal{A}_1 \leftarrow \mathcal{A}(\mathcal{O}', l_2 = a + \epsilon, l_1 = a + 3\epsilon, 0.01\delta)$
  $\mathcal{A}_2 \leftarrow \mathcal{A}(\mathcal{O}', l_2 = a + 2\epsilon, l_1 = a + 4\epsilon, 0.01\delta)$
$r_1 \leftarrow \mathsf{Estimate}(\mathcal{A}_1, \epsilon = 0.1, \delta)$
$r_2 \leftarrow \mathsf{Estimate}(\mathcal{A}_2, \epsilon = 0.1, \delta)$
$B_1 \leftarrow \mathbb{1}(r_1 > \frac{k+0.5}{n+1})$; $B_2 \leftarrow \mathbb{1}(r_2 > \frac{k+0.5}{n+1})$
switch (B_1, B_2) **do**
  case $(0, 0) : I \leftarrow [a, a + 3\epsilon]$
  case $(0, 1) : I \leftarrow [a + \epsilon, a + 4\epsilon]$
  case $(1, 0) : I \leftarrow [a + \epsilon, a + 4\epsilon]$
  case $(1, 1) : I \leftarrow [a + 2\epsilon, a + 5\epsilon = b]$
return I

Algorithm 4: $\mathsf{BestArm}(\mathcal{O}, \delta)$

Input: Oracle $\mathcal{O}$ as in (2); confidence parameter $0 < \delta < 1$.

$\delta \leftarrow \delta/2$
$I_1, I_2 \leftarrow \mathsf{Locate}(\mathcal{O}, \delta)$
$l_1 \leftarrow \min I_1$ (left endpoint of I_1)
$l_2 \leftarrow \max I_2$ (right endpoint of I_2)
Construct variable-time quantum algorithm $\mathcal{A}$:
  $\mathcal{A} \leftarrow \mathcal{A}(\mathcal{O}, l_2, l_1, 0.01\delta)$
$i \leftarrow \mathsf{Amplify}(\mathcal{A}, \delta)$
return i

Lemma 5 (Correctness and complexity of Algorithm 3)**.** *Fix $k \in \{1, 2\}$, an interval $I = [a, b]$, and a confidence parameter $0 < \delta < 1$. Suppose that $p_k \in I$ and $|I| \geq \Delta_2/8$. Then Algorithm 3*

1. *outputs an interval J with $|J| = \frac{3}{5}\,|I|$ such that $p_k \in J$ with probability $\geq 1 - \delta$, and*
2. *uses $O\big(\sqrt{H}\ \text{poly}\big(\log\big(\frac{n}{\delta\,\Delta_2}\big)\big)\big)$ queries.*

The proofs of Lemma 4 and Lemma 5 appear in the supplementary material.

The following theorem is equivalent to Theorem 3.

Theorem 4 (Correctness and complexity of Algorithm 4)**.** *Fix a confidence parameter $0 < \delta < 1$. Then, with probability $\geq 1 - \delta$, Algorithm 4*

1. *outputs the best arm, and*
2. *uses $O\big(\sqrt{H}\ \text{poly}\big(\log\big(\frac{n}{\delta\,\Delta_2}\big)\big)\big)$ queries.*

Proof. Note that δ is halved at the beginning, on Line 1. For the first claim, we know from the first claim of Lemma 4 that, with probability $\geq 1 - \delta/2$, the two intervals I_k assigned in Line 2 have $\min I_1 - \max I_2 \geq 2\,|I_1| \geq \Delta_2/4$ and $p_k \in I_k$. Assuming this holds, we have $p_2 < l_2 < l_2 + \Delta_2/4 \leq l_1 < p_1$ for the endpoints l_k assigned in

Lines 3 and 4. This means that the variable-time quantum algorithm $\mathcal{A}$ defined in Line 6 has $S_{\text{right}} \cup S_{\text{middle}} = \{1\}$, so $\mathsf{Amplify}(\mathcal{A}, \delta/2)$ returns index 1 with probability $\geq 1-\delta/2$. Therefore, the overall probability of Algorithm 4 returning the best arm is at least $1-\delta$.

The second claim follows immediately from adding the complexity of $\mathsf{Locate}(\mathcal{O}, \delta/2)$ (Lemma 4) and $\mathsf{Amplify}(\mathcal{A}, \delta/2)$ (Lemma 3, using $l_1 - l_2 \geq \Delta_2/4$). □

By establishing Theorem 4, we have established Theorem 3, our main claim. As discussed previously, the main complexity measure of interest in the classical case is H, and we see that we get a quadratic speedup in terms of this parameter.

We can see that the poly-logarithmic factor has degree about 6 from (38), (40), and (42). It would be interesting to reduce this degree. A more fundamental challenge is to remove the variable n that appears in our log factors. In the classical case, n was already removed from log factors in early work (Even-Dar, Mannor, and Mansour 2002) by a procedure called "median elimination". However, quantizing the median elimination framework is nontrivial, as the query complexity for outputting the $n/2$ smallest items among n elements is $\Theta(n)$ (Ambainis 2010a, Theorem 1), exceeding our budget of $O(\sqrt{n})$.

As corollaries of our main results in the fixed-confidence setting, we provide results on best-arm identification in the PAC (Probably Approximately Correct) and fixed-budget settings. In the (ϵ, δ)-PAC setting, the goal is to identify an arm i with $p_i \geq p_1 - \epsilon$ with probability $\geq 1-\delta$. Our best-arm identification algorithm can be modified to work in this setting as well. More precisely, we can modify Locate (Algorithm 2) by adding a breaking condition to the while loop when $|I_1|$ (or equivalently $|I_2|$) is smaller than ϵ. This gives the following result:

Corollary 1. *There is a quantum algorithm that finds an ϵ-optimal arm with query complexity $O\big(\sqrt{\min\{\frac{n}{\epsilon^2}, H\}} \cdot \operatorname{poly}\big(\log\big(\frac{n}{\delta\,\Delta_2}\big)\big)\big)$.*

Note that our modification means that the Amplify step in Algorithm 4 takes an input interval I with $|I| = l_1 - l_2 \in [\epsilon/2, \epsilon]$. The correctness and complexity follow directly from Lemma 1 and Lemma 3. For comparison, Even-Dar, Mannor, and Mansour (2002) gave a classical PAC algorithm with complexity $O\big(\frac{n}{\epsilon^2}\log\big(\frac{n}{\delta}\big)\big)$, which was later improved to $O\big(\sum_{i=1}^{n} \min\{\epsilon^{-2}, \Delta_i^{-2}\} \cdot \log\big(\frac{n}{\delta\Delta_2}\big)\big)$ by Gabillon, Ghavamzadeh, and Lazaric (2012).

In the supplementary material, we also show how to identify the best arm with high probability for a fixed number of total queries (the fixed-budget setting) given knowledge of H.

Quantum Lower Bound

In this section, we describe a lower bound for the quantum best-arm identification problem. Our lower bound shows that the algorithm of Theorem 3 is optimal up to poly-logarithmic factors.

Theorem 5. *Let $p \in (0, 1/2)$. For any biases $p_i \in [p, 1-p]$, any quantum algorithm that identifies the best arm requires $\Omega(\sqrt{H})$ queries to the multi-armed bandit oracle $\mathcal{O}$.*

To prove this lower bound, we use the quantum adversary method to show quantum hardness of distinguishing n oracles $\mathcal{O}_x$, $x \in [n]$, corresponding to the following n bandits. In the 1st bandit, we assign bias p_i to arm i for all i. In the x^{th} bandit for $x \in \{2, \ldots, n\}$, we assign bias $p_1 + \eta$ to arm x and p_i to arm i for all $i \neq x$, where η is an appropriately chosen parameter. This hard set of bandits is inspired by the proof of a corresponding classical lower bound (Mannor and Tsitsiklis 2004, Theorem 5).

More precisely, for a positive integer T, consider an arbitrary T-query quantum algorithm that distinguishes the oracles $\mathcal{O}_x$. The main idea of the adversary method is to keep track of certain quantities $s_k \in \mathbb{R}$ where $k \in \{0, 1, \ldots, T\}$. For each k, s_k quantifies how close the states of the quantum algorithm are when it operates using k queries to the different $\mathcal{O}_x$. At the start, when $k = 0$, s_0 must be large because when no queries have been made, the states must be close. At the end, when $k = T$, s_T must be small because the states are distinguishable by assumption.

The key point is that we can also bound how much s_k can change in one query, that is we can bound the quantities $|s_{k+1} - s_k|$ for each k. Of course, this bound immediately gives a lower bound on T, the number of queries it takes to go from s_0 (large) to s_T (small). To bound $|s_{k+1} - s_k|$, the key point is to bound the distance between oracles, i.e. matrices, $\mathcal{O}_x$ and $\mathcal{O}_y$ for different $x, y \in [n]$.

We defer the full proof and full description of the quantum adversary method to the supplementary material.

Conclusions

In this paper, we propose a quantum algorithm for identifying the best arm of a multi-armed bandit, which gives a quadratic speedup compared to the best possible classical result. We also prove a matching quantum lower bound (up to poly-logarithmic factors).

This work leaves several natural open questions:

- Can we give fast quantum algorithms for the exploitation of multi-armed bandits? In particular, can we give online algorithms with favorable regret? The quantum hedging algorithm (Hamoudi et al. 2020) and the quantum boosting algorithm (Arunachalam and Maity 2020) might be relevant to this challenge.
- Can we give fast quantum algorithms for other types of multi-armed bandits, such as contextual bandits or adversarial bandits (e.g., Beygelzimer et al. 2011; Agarwal et al. 2014; Auer et al. 2002)?
- Can we give fast quantum algorithms for finding a near-optimal policy of a Markov decision process (MDP)? MDPs are a natural generalization of MABs, where the goal is to maximize the expected reward over sequences of decisions. Even-Dar, Mannor, and Mansour (2002) gave a reduction from this problem to best-arm identification by viewing the Q-function of each state as a multi-armed bandit.

Acknowledgements

DW thanks Robin Kothari, Jin-Peng Liu, Yuan Su, and Aarthi Sundaram for helpful discussions. This work received support from the Army Research Office (W911NF-17-1-0433 and W911NF-20-1-0015); the National Science Foundation (CCF-1755800, CCF-1813814, and PHY-1818914); and the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Quantum Algorithms Teams and Accelerated Research in Quantum Computing programs. DW and TL were also supported by QISE-NET Triplet Awards (NSF grant DMR-1747426) and TL by an IBM PhD Fellowship.

Ethics Statement

This work is purely theoretical. Researchers working on theoretical aspects of bandits and quantum computing may immediately benefit from our results. In the long term, once fault-tolerant quantum computers have been built, our results may find practical applications in multi-armed bandit scenarios arising in the real world. As far as we are aware, our work does not have negative ethical impact.

References

Agarwal, A.; Hsu, D.; Kale, S.; Langford, J.; Li, L.; and Schapire, R. 2014. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, 1638–1646.

Ambainis, A. 2010a. A new quantum lower bound method, with an application to a strong direct product theorem for quantum search. *Theory of Computing* 6(1): 1–25. arXiv:quant-ph/0508200

Ambainis, A. 2010b. Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations. arXiv:1010.4458

Amin, M. H.; Andriyash, E.; Rolfe, J.; Kulchytskyy, B.; and Melko, R. 2018. Quantum Boltzmann machine. *Physical Review X* 8(2): 021050. arXiv:1601.02036

Arunachalam, S.; and de Wolf, R. 2017. Guest column: a survey of quantum learning theory. *ACM SIGACT News* 48(2): 41–67. arXiv:1701.06806

Arunachalam, S.; and Maity, R. 2020. Quantum Boosting. In *To appear in the Thirty-seventh International Conference on Machine Learning*. arXiv:2002.05056

Arute et al., F. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574(7779): 505–510. arXiv:1910.11333

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 2002. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing* 32(1): 48–77.

Beygelzimer, A.; Langford, J.; Li, L.; Reyzin, L.; and Schapire, R. 2011. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 19–26.

Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; and Lloyd, S. 2017. Quantum machine learning. *Nature* 549(7671): 195. arXiv:1611.09347

Brassard, G.; Høyer, P.; Mosca, M.; and Tapp, A. 2002. Quantum amplitude amplification and estimation. *Contemporary Mathematics* 305: 53–74. arXiv:quant-ph/0005055

Casalé, B.; Di Molfetta, G.; Kadri, H.; and Ralaivola, L. 2020. Quantum Bandits. arXiv:2002.06395

Chakraborty, S.; Gilyén, A.; and Jeffery, S. 2019. The Power of Block-Encoded Matrix Powers: Improved Regression Techniques via Faster Hamiltonian Simulation. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 33:1–33:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. arXiv:1804.01973

Chen, L.; and Li, J. 2015. On the optimal sample complexity for best arm identification. arXiv:1511.03774

Chen, L.; Li, J.; and Qiao, M. 2017. Towards Instance Optimal Bounds for Best Arm Identification. In *Conference on Learning Theory*, 535–592. arXiv:1608.06031

Childs, A. M.; Kothari, R.; and Somma, R. D. 2017. Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision. *SIAM Journal on Computing* 46(6): 1920–1950. arXiv:1511.02306

Dunjko, V.; and Briegel, H. J. 2018. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics* 81(7): 074001. arXiv:1709.02779

Dunjko, V.; Liu, Y.-K.; Wu, X.; and Taylor, J. M. 2017. Exponential improvements for quantum-accessible reinforcement learning. arXiv:1710.11160

Dunjko, V.; Taylor, J. M.; and Briegel, H. J. 2016. Quantum-enhanced machine learning. *Physical Review Letters* 117(13): 130501. arXiv:1610.08251

Dunjko, V.; Taylor, J. M.; and Briegel, H. J. 2017. Advances in quantum reinforcement learning. In *Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics*, 282–287. IEEE. arXiv:1811.08676

Dürr, C.; and Høyer, P. 1996. A quantum algorithm for finding the minimum. arXiv:quant-ph/9607014

Even-Dar, E.; Mannor, S.; and Mansour, Y. 2002. PAC Bounds for Multi-armed Bandit and Markov Decision Processes. In Kivinen, J.; and Sloan, R. H., eds., *Computational Learning Theory*, 255–270. Berlin, Heidelberg: Springer Berlin Heidelberg.

Gabillon, V.; Ghavamzadeh, M.; and Lazaric, A. 2012. Best arm identification: A unified approach to fixed budget and fixed confidence. In *Advances in Neural Information Processing Systems*, 3212–3220.

Grover, L. K. 1996. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, 212–219. ACM. arXiv:quant-ph/9605043

Hamoudi, Y.; Ray, M.; Rebentrost, P.; Santha, M.; Wang, X.; and Yang, S. 2020. Quantum algorithms for hedging and the Sparsitron. arXiv:2002.06003

Jamieson, K.; Malloy, M.; Nowak, R.; and Bubeck, S. 2014. lil'UCB: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, 423–439. arXiv:1312.7308

Jerbi, S.; Nautrup, H. P.; Trenkwalder, L. M.; Briegel, H. J.; and Dunjko, V. 2019. A framework for deep energy-based reinforcement learning with quantum speed-up. arXiv:1910.12760

Karnin, Z.; Koren, T.; and Somekh, O. 2013. Almost optimal exploration in multi-armed bandits. In *International Conference on Machine Learning*, 1238–1246.

Kerenidis, I.; Landman, J.; Luongo, A.; and Prakash, A. 2019. q-means: A quantum algorithm for unsupervised machine learning. In *Advances in Neural Information Processing Systems*, 4136–4146. arXiv:1812.03584

Li, T.; Chakrabarti, S.; and Wu, X. 2019. Sublinear quantum algorithms for training linear and kernel-based classifiers. In *International Conference on Machine Learning*, 3815–3824. arXiv:1904.02276

Lin, L.; and Tong, Y. 2020. Near-optimal ground state preparation. arXiv:2002.12508

Lloyd, S.; Mohseni, M.; and Rebentrost, P. 2013. Quantum algorithms for supervised and unsupervised machine learning. arXiv:1307.0411

Lloyd, S.; Mohseni, M.; and Rebentrost, P. 2014. Quantum principal component analysis. *Nature Physics* 10(9): 631. arXiv:1307.0401

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

Montanaro, A. 2015. Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A* 471(2181): 20150301.

Nielsen, M. A.; and Chuang, I. L. 2000. *Quantum computation and quantum information*. Cambridge University Press.

Rebentrost, P.; Mohseni, M.; and Lloyd, S. 2014. Quantum support vector machine for big data classification. *Physical Review Letters* 113(13): 130503. arXiv:1307.0471

Schuld, M.; Sinayskiy, I.; and Petruccione, F. 2015. An introduction to quantum machine learning. *Contemporary Physics* 56(2): 172–185. arXiv:1409.3097

Wiebe, N.; Kapoor, A.; and Svore, K. M. 2015. Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *Quantum Information & Computation* 15(3-4): 316–356. arXiv:1401.2142