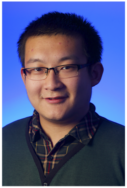
Quantum exploration algorithms for multi-armed bandits

Daochen Wang
University of Maryland

arXiv: 2006.12760 (in AAAI 2021)



Xuchen You (Maryland)

Tongyang Li (MIT)

Andrew Childs (Maryland)

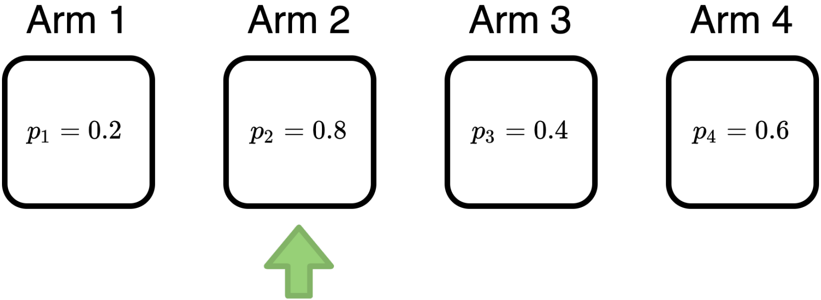
2nd March 2021, GMU seminar

Outline

Exploring multi-armed bandits

You are in a casino...

...faced with n slot machines each with an *unknown* probability p_i of giving unit reward when its arm is pulled.



The exploration problem (or best-arm identification)

How many arm pulls (aka queries) are necessary and sufficient to find the arm with the largest p_i (best arm) with high probability?

- Classically, one query is one sample from one of the machines, i.e., a sample from a Bernoulli(p_i) random variable.
- Quantumly, one query is one application of the *quantum bandit oracle*:

$$\mathcal{O}: \; |i\rangle |0\rangle |0\rangle \mapsto |i\rangle \left(\sqrt{p_i}\,|1\rangle |u_i\rangle + \sqrt{1-p_i}\,|0\rangle |v_i\rangle\right), \qquad (1)$$

for some arbitrary states $|u_i\rangle$ and $|v_i\rangle$.

Example application: finding the best move in a game

You have n candidate moves, where move i can lead to one in a set $X(i)$ of possible subsequent games.

- Assume you have computer code f that, for move i and game $x \in X(i)$, computes $f(i,x) = 1$ if you win and $= 0$ if you lose.
- We can instantiate one query to the quantum bandit oracle using one call to f:

$$\begin{aligned} &|i\rangle\,|0\rangle\,\frac{1}{\sqrt{|X(i)|}}\sum_{x\in X(i)}|x\rangle \\ \overset{f}{\mapsto}\,&|i\rangle \sum_{x\in X(i)}\frac{1}{\sqrt{|X(i)|}}\,|f(i,x)\rangle\,|x\rangle \\ =\,&|i\rangle\left(\sqrt{p_i}\,|1\rangle\,|u_i\rangle + \sqrt{1-p_i}\,|0\rangle\,|v_i\rangle\right), \end{aligned} \tag{2}$$

where $|u_i\rangle$ and $|v_i\rangle$ are some states and p_i equals the probability that move i leads to your win.

Quantum exploration algorithms

Quadratic quantum speedup in query and time complexity

Suppose that $p_1 > p_2 \geq p_3 \geq \cdots \geq p_n$.

- Classically: necessary and sufficient to use order[1]

$$H := \sum_{i=2}^{n} \frac{1}{(p_1 - p_i)^2} \tag{3}$$

reward samples to identify the best arm.

- Quantumly (our result): necessary and sufficient to use order

$$\sqrt{\sum_{i=2}^{n} \frac{1}{(p_1 - p_i)^2}} = \sqrt{H} \tag{4}$$

queries to the quantum bandit oracle to identify the best arm. This scaling also holds for time complexity.

[1] In this talk, "order" also means "order up to log factors".

Fast quantum algorithm (overview)

- **Case 1: know both p_1 and p_2.** Mark arms i with p_i smaller than $(p_1 + p_2)/2$ using about $t_i := 1/(p_1 - p_i)$ queries by amplitude estimation. Then use variable time amplitude amplification[2], on top of the marking algorithm, to amplify the *unmarked* arm, i.e., arm $i = 1$, so that it is output with high probability. Uses order $\sqrt{t_2^2 + t_3^2 + \cdots + t_n^2} = \sqrt{H}$ queries.

- **Case 2: know neither p_1 nor p_2.** For a given probability p, can count how many arms i have $p_i > p$ using variable time amplitude estimation[3]. Therefore, can locate p_1 and p_2 by binary search. Uses order $\sqrt{H}$ queries. Then back to the first case.

[2] Ambainis (2012).

[3] Chakraborty, Gilyén, and Jeffery (2019).

Variable time quantum algorithms (1/2)

First example: variable time quantum search by Ambainis (2006).

- Like in Grover search, the goal is to find a marked item among n different items.
- The problem is generalized such that a query cost of t_i is associated with checking if item i is marked.
- Result: an overall query complexity of $O\left(\sqrt{t_1^2 + \cdots + t_n^2}\right)$ is necessary and sufficient to find the marked item. In the Grover case, all $t_j = O(1)$, so recover $O(\sqrt{n})$ scaling.

Variable time quantum algorithms (2/2)

Variable time amplitude amplification (VTAA) and estimation (VTAE) generalize variable time quantum search.

- Suppose $\mathcal{A}$ is a quantum algorithm such that

$$\mathcal{A}\left|0^m\right\rangle = \sqrt{p}\left|\psi_1\right\rangle\left|1\right\rangle + \sqrt{1-p}\left|\psi_0\right\rangle\left|0\right\rangle. \tag{5}$$

- Suppose further that $\mathcal{A}$ is a *variable time algorithm*. That is, $\mathcal{A}$ can be written as a product $\mathcal{A} := \mathcal{A}_m\mathcal{A}_{m-1}\dots\mathcal{A}_0$. Suppose further that after each step $j \in \{1, \dots, n\}$ there is some probability ω_j of the algorithm stopping and that the query complexity up to that step is t_j.

- Then can roughly obtain $\left|\psi_1\right\rangle$ and p using roughly $O(t_{\text{avg}}/\sqrt{p})$ queries, where $t_{\text{avg}}^2 := \sum_{j=1}^m \omega_j t_j^2$, by VTAA and VTAE applied to $\mathcal{A}$ respectively.

Constructing a variable time quantum algorithm $\mathcal{A}$

For given $0 < \ell_2 < \ell_1 < 1$, we construct a variable time quantum algorithm $\mathcal{A}$, inspired by classical successive elimination, such that

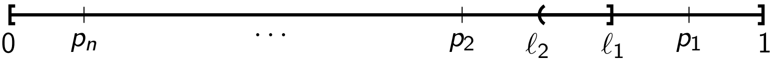
$$\mathcal{A}\,|0^m\rangle = \sqrt{\frac{|S_{\mathrm{right}}|}{n}}\;|\psi_1\rangle\,|1\rangle + \sqrt{\frac{|S_{\mathrm{left}}|}{n}}\;|\psi_0\rangle\,|0\rangle + \lambda\;|\psi_{\mathrm{junk}}\rangle\,, \quad (6)$$

where $S_{\mathrm{right}} := \{i : p_i > \ell_1\}$ and $S_{\mathrm{left}} := \{i : p_i \leq \ell_2\}$, $|\psi_1\rangle$ contains an equal superposition of indices in S_{right}, and

$$t_{\mathrm{avg}}^2 = \frac{1}{n}\Big(\frac{|S_{\mathrm{right}}|}{(\ell_1-\ell_2)^2} + \sum_{i\in S_{\mathrm{left}}\cup S_{\mathrm{middle}}} \frac{1}{(\ell_1-p_i)^2}\Big), \quad (7)$$

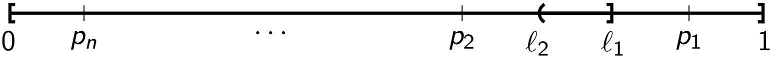
where $S_{\mathrm{middle}} := \{i : \ell_2 < p_i \leq \ell_1\}$.

Illustration of $S_{\mathrm{left}} = \{2,\ldots,n\}$, $S_{\mathrm{middle}} = \emptyset$, and $S_{\mathrm{right}} = \{1\}$:



Case 1: know both p_1 and p_2 – just apply VTAA to $\mathcal{A}$

Set $\ell_1 = p_1 - (p_1 - p_2)/3$ and $\ell_2 = p_2 + (p_1 - p_2)/3$, say. Then we have the same picture as before:



and so again $S_{\mathrm{left}} = \{2, \ldots, n\}$, $S_{\mathrm{middle}} = \emptyset$, and $S_{\mathrm{right}} = \{1\}$. We can simplify the previous expressions:

$$\begin{aligned} \mathcal{A}\,|0^m\rangle &= \sqrt{1/n}\,|\psi_1\rangle\,|1\rangle + \sqrt{(n-1)/n}\,|\psi_0\rangle\,|0\rangle\,, \\ t_{\mathrm{avg}}^2 &= O\Big(\frac{1}{n}\sum_{i=2}^{n}\frac{1}{(p_1 - p_i)^2}\Big). \end{aligned} \tag{8}$$

Applying VTAA to $\mathcal{A}$ costs $O(t_{\mathrm{avg}}/\sqrt{p}) = O(\sqrt{H})$ queries and yields $|\psi_1\rangle$ which now just contains the best-arm index state $|1\rangle$.

Case 2: know neither p_1 nor p_2 – use VTAE first (1/2)

Recall

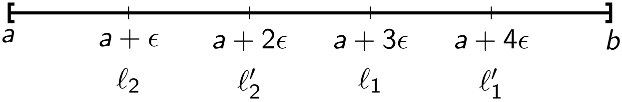
$$\mathcal{A}|0^m\rangle = \sqrt{\frac{|S_{\mathrm{right}}|}{n}}\,|\psi_1\rangle|1\rangle + \sqrt{\frac{|S_{\mathrm{left}}|}{n}}\,|\psi_0\rangle|0\rangle + \lambda\,|\psi_{\mathrm{junk}}\rangle\,. \quad (9)$$

- *If* we could set $\ell_2 = \ell_1$ in the definition of $\mathcal{A}$ then $S_{\mathrm{middle}} = \emptyset$, so $|S_{\mathrm{right}}| + |S_{\mathrm{left}}| = n$, and so λ must be 0. Therefore, VTAE on $\mathcal{A}$ gives us an estimate of $|S_{\mathrm{right}}|/n$. So by binary search, we can estimate each of p_1 and p_2 very cheaply.
- But the cost of $\mathcal{A}$ scales with $1/(\ell_1 - \ell_2)^2$, so the above doesn't work. In fact, a similar problem shows up in the problem of *quantum ground state preparation*. That problem was only recently resolved by a clever trick introduced by Lin and Tong (2020) in their paper "Near-optimal ground state preparation". We use a modified version of their trick.

Case 2: know neither p_1 nor p_2 – use VTAE first (2/2)

The main idea is to use *two* choices for the pair (ℓ_1, ℓ_2) at each binary search step.

- Suppose it is currently known that $p_1 \in [a, b]$, we apply VTAE to $\mathcal{A}$ defined with (ℓ_2, ℓ_1) first set to $(a + \epsilon, a + 3\epsilon)$ and then to $(a + 2\epsilon, a + 4\epsilon)$, where $\epsilon = (b - a)/5$.



- Depending on the output of the VTAE algorithm, we can always *shrink* the interval in which we are confident p_1 belongs to one of $[a, a + 3\epsilon]$, $[a + \epsilon, a + 4\epsilon]$, and $[a + 2\epsilon, a + 5\epsilon]$.
- These intervals have length $3/5$ that of the original $[a, b]$. Repeatly applying this procedure is sort of like binary searching for p_1. Same procedure also works for p_2.

Brief description of $\mathcal{A}$

Our best-arm identification algorithm applies VTAA and VTAE to a variable time algorithm $\mathcal{A}$. But what is $\mathcal{A}$?

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 $0 < \alpha < 1$.

$\Delta \leftarrow l_1 - l_2$

$m \leftarrow \lceil \log \frac{1}{\Delta} \rceil + 2$

$a \leftarrow \frac{\alpha}{2mn^{3/2}}$

Initialize state to
$\frac{1}{\sqrt{n}} \sum_{i=1}^{n} |i\rangle_I \, |\text{coin}\, p_i\rangle_B \, |0\rangle_C \, |0\rangle_P \, |1\rangle_F$

for $j = 1, \ldots, m$ **do**

- $\epsilon_j \leftarrow 2^{-j}$
- **if** *register I is in state $|i\rangle$ and registers $C_1, \ldots, C_{j-1}$ are in state $|0\rangle$* **then**
  - Apply GAE$(\epsilon_j, a; l_1)$ with $\mathcal{O}_{p_i}$ on registers B, C_j, and P_j
- Apply controlled-NOT gate with control on register C_j and target on register F

if *registers $C_1, \ldots, C_m$ are in state $|0\rangle$* **then**

- Flip the bit stored in register C_{m+1}

Variants: PAC, fixed budget, and non-Bernoulli

By slight modifications, our quantum algorithm can be adapted to work in the following settings.

- PAC. If our goal is only to output an ϵ-optimal arm i with $p_1 - p_i < \epsilon$, our algorithm can be adapted to have smaller query complexity that is of order $\sqrt{\min\{n/\epsilon^2, H\}}$.
- Fixed budget. If H is known in advance, for any sufficiently large T, our algorithm can be adapted to use T queries to output the best arm with probability at least $1 - \exp(-\Omega(T/\sqrt{H}))$.
- Non-Bernoulli. Our algorithm can be adapted to work even if the arm distributions are only guaranteed to have bounded variance, in particular, if they are sub-Gaussian. The modification goes via the quantum mean estimation algorithm of Montanaro (2015).

Quantum lower bound

Quantum lower bound proof (1/2)

Let $\eta \approx p_1 - p_2$. Use the quantum adversary method[4] to prove that the following set of n multi-armed bandit oracles require $\Omega(\sqrt{H})$ queries to distinguish:

$$
\begin{array}{lllllll}
1 & p_1, & p_2, & p_3, & \ldots & p_n \\
2 & p_1, & p_1 + \eta, & p_3, & \ldots & p_n \\
 & \ldots & & & & \\
n & p_1, & p_2, & p_3, & \ldots & p_1 + \eta
\end{array}
$$

But our quantum algorithm can distinguish them using $O(\sqrt{H})$ queries, so it is tight (up to log factors).

[4] Ambainis (2000).

Quantum lower bound proof (2/2)

- The standard adversary method applies only to oracles U_x encoding Boolean bitstrings $x \in \{0,1\}^n$ ($U_x : |i\rangle |b\rangle \mapsto |i\rangle |b \oplus x_i\rangle$).
- The quantum bandit oracle encode probabilities instead. Therefore, we cannot make use of ready-made adversary method lower bounds.
- Instead we use the *idea* of the adversary method to derive our lower bound from scratch. Mathematically, this comes down to bounding the entries of the matrix

$$\begin{pmatrix} \sqrt{1-p_i} & \sqrt{p_i} \\ \sqrt{p_i} & -\sqrt{1-p_i} \end{pmatrix}^\dagger \begin{pmatrix} \sqrt{1-p_1'} & \sqrt{p_1'} \\ \sqrt{p_1'} & -\sqrt{1-p_1'} \end{pmatrix} - \mathbb{I}, \quad (10)$$

where $i > 1$ and $p_1' := p_1 + \eta$.

Conclusion

Conclusion

We have constructed an asymptotically optimal quantum algorithm that offers a quadratic speedup for finding the best arm in a multi-armed bandit.

Open problems and future directions:

- Can we give quantum algorithms for exploration in the fixed budget setting with improved success probability?
- Can we give quantum algorithms for the *exploitation* of multi-armed bandits with favorable regret?
- Can we give fast quantum algorithms for finding a near-optimal policy of a Markov decision process (MDP)?

Thank you for your attention!