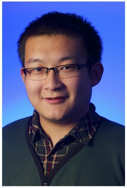
Quantum exploration algorithms for multi-armed bandits

Daochen Wang
University of Maryland
arXiv: 2006.12760

Xuchen You



Tongyang Li

Andrew Childs

10th November, QTML 2020

Outline

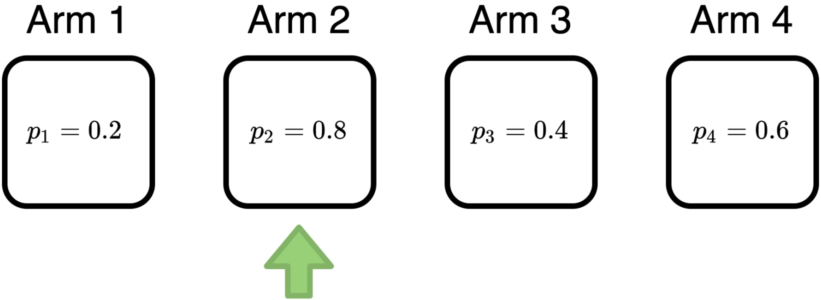
Exploring multi-armed bandits

Quantum exploration algorithms

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 (aka best-arm identification)

How many arm pulls (aka queries) are necessary and sufficient to find the arm with highest p_i (aka 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:

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

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

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

  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.

Fast quantum algorithm

- **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[1], 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[2]. Therefore, can locate p_1 and p_2 by binary search. Uses order $\sqrt{H}$ queries. Then back to the first case.

[1] Ambainis (2012).

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

Quantum lower bound proof

Let $\eta \approx p_1 - p_2$. Use the quantum adversary method[3] to prove that the following set of n multi-armed bandit oracles require $\Omega(\sqrt{H})$ queries to distinguish:

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

[3] Ambainis (2000).

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!