Quantum exploration algorithms for multi-armed bandits

Daochen Wang
University of Maryland

Joint work with
Xuchen You, Tongyang Li, and Andrew M. Childs
arXiv: 2006.12760

MSR MLO Lunch (short talk)
22nd July 2020

Outline

Basics of quantum algorithms

Multi-armed bandits and our results

Open questions

Basics of quantum algorithms

Information stored in qubits instead of prbits or bits

- Deterministic algorithms use bits. n bits can be in *one of* 2^n different configurations:

$$(b_0, b_1, \ldots, b_{2^n-1}) \tag{1}$$

where there is a unique i with $b_i = 1$ and $b_j = 0$ for all $j \neq i$.

- Randomized algorithms use probabilistic bits (prbits). n prbits can be in a *probabilistic mixture* of 2^n different configurations:

$$(p_0, p_1, \ldots, p_{2^n-1}) \tag{2}$$

where $p_i \in \mathbb{R}$, $p_i \geq 0$, and $\sum_{i=0}^{2^n-1} p_i = 1$

- Quantum algorithms use quantum bits (qubits). n qubits can be in a *quantum superposition* of 2^n different configurations:

$$(\alpha_0, \alpha_1, \ldots, \alpha_{2^n-1}) \leftrightarrow \sum_{i=0}^{2^n-1} \alpha_i \left|i\right\rangle \tag{3}$$

where $\alpha_i \in \mathbb{C}$ and $\sum_{i=0}^{2^n-1} |\alpha|^2 = 1$. α_i are called "amplitudes".

Computation by unitary matrices instead of stochastic or permutation matrices

- Deterministic algorithms (made reversible) on n bits compute using permutation matrices $P \in \{0,1\}^{2^n \times 2^n}$. For example, on a single bit, the NOT gate corresponds to the matrix

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{4}$$

- Randomized algorithms on n prbits compute using stochastic matrices $S \in \mathbb{R}^{2^n \times 2^n}$ where the columns sum to 1 and all entries are ≥ 0.
- Quantum algorithms on n qubits compute using unitary matrices $U \in \mathbb{C}^{2^n \times 2^n}$ where $U^\dagger U = I = UU^\dagger$.

Output with probabilities equal to norm squared of the amplitudes

At the end of the computation:

- Deterministic algorithms are in a state $(b_0, \ldots, b_{2^n-1})$ and they output the bitstring (corresponding to) $i \in \{0, \ldots, 2^n - 1\}$ with $b_i = 1$. (There is a unique such i.)
- Randomized algorithms are in a state $(p_0, p_1, \ldots, p_{2^n-1})$ and they output a bitstring $i \in \{0, \ldots, 2^n - 1\}$ with probability p_i.
- Quantum algorithms are in a state $(\alpha_0, \alpha_1, \ldots, \alpha_{2^n-1})$, equivalently

$$\sum_{i=0}^{2^n-1} \alpha_i |i\rangle , \tag{5}$$

  and they output a bitstring $i \in \{0, \ldots, 2^n - 1\}$ with probability $|\alpha_i|^2$.

Grover's quantum search algorithm

Problem: given "query access" to an unknown n-bit string $x \in \{0,1\}^n$ with exactly one i such that $x_i = 1$; how many queries is necessary and sufficient to find i with high probability?

- Classically (deterministic or randomized), queries are of the form $i \mapsto x_i$, and it can be seen that at least $\Omega(n)$ such queries are necessary and sufficient to solve the problem.
- Quantumly, queries are to the unitary matrix $O_x \in \mathbb{C}^{2^{2n} \times 2^{2n}}$:

$$\begin{aligned} \mathcal{O}_x : \mathbb{C}^n \otimes \mathbb{C}^2 &\to \mathbb{C}^n \otimes \mathbb{C}^2 \\ |i\rangle \otimes |b\rangle &\mapsto |i\rangle \otimes |b \oplus x_i\rangle \,, \end{aligned} \tag{6}$$

  where $\otimes$ denotes vector (space) tensor product. This means we can query x_i in superposition over positions i. Grover's algorithm uses $O(\sqrt{n})$ queries to O_x to solve the problem. Matches lower bound of $\Omega(\sqrt{n})$.

More on querying in superposition

From the previous slide:

$$\begin{aligned} \mathcal{O}_x : \mathbb{C}^n \otimes \mathbb{C}^2 &\to \mathbb{C}^n \otimes \mathbb{C}^2 \\ |i\rangle \otimes |b\rangle &\mapsto |i\rangle \otimes |b \oplus x_i\rangle \,, \end{aligned} \tag{7}$$

usually the $\otimes$ is omitted. Can do the following:

- Query in superposition. Create the state $\sum_{i=1}^{n} \alpha_i |i\rangle |0\rangle$ without queries to O_x, and then query O_x to map $\sum_{i=1}^{n} \alpha_i |i\rangle |0\rangle \xmapsto{O_x} \sum_{i=1}^{n} \alpha_i |i\rangle |x_i\rangle$.
- If we set $\alpha_j = 1$ for some j and $\alpha_i = 0$ for all $i \neq j$, then the above map is $|j\rangle |0\rangle \mapsto |j\rangle |x_j\rangle$, i.e. same as a classical query!

Multi-armed bandits and our results

The best-arm identification problem in multi-armed bandits

Setting: Bernoulli multi-armed bandit with n arms where arm i has probability p_i of giving a reward of 1 and probability $1 - p_i$ of giving no reward (reward of 0).

Problem: given query access to the multi-armed bandit, how many queries is necessary and sufficient to find the arm with highest p_i (aka best arm) with high probability?

- Classically, queries are reward samples from the arms.
- Quantumly, queries are to the *quantum bandit oracle*:

$$\mathcal{O} : \mathbb{C}^n \otimes \mathbb{C}^2 \otimes \mathbb{C}^m \to \mathbb{C}^n \otimes \mathbb{C}^2 \otimes \mathbb{C}^m$$
$$|i\rangle |0\rangle |0\rangle \mapsto |i\rangle \left(\sqrt{p_i} |1\rangle |v_i\rangle + \sqrt{1 - p_i} |0\rangle |u_i\rangle\right). \tag{8}$$

This means we can query the multi-armed bandit in superposition over arms.

Result: quantum gives quadratic speedup in query complexity

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

- Classically: necessary and sufficient to use on the order of about

$$H := \sum_{i=2}^{n} \frac{1}{(p_1 - p_i)^2} \tag{9}$$

reward samples to identify the best arm.

- Quantumly (our result): necessary and sufficient to use on the order of about

$$\sqrt{\sum_{i=2}^{n} \frac{1}{(p_1 - p_i)^2}} = \sqrt{H} \tag{10}$$

queries to the quantum bandit oracle to identify the best arm.

Brief overview of techniques

Quantum algorithm. In the case that we know p_1, we can mark those is with p_i smaller than p_1 using about $t_i := 1/(p_1 - p_i)$ queries by a well-known quantum technique called amplitude estimation. We can then use another quantum technique, called variable time amplitude amplification, on top of the marking algorithm, to amplify the *unmarked* i, i.e. $i = 1$, so that it is output with high probability. This takes $\sqrt{t_2^2 + t_3^2 + \cdots + t_n^2}$ queries[1]. If we don't know p_1, we first locate it by binary search.

Quantum lower bound. For $\eta \approx p_1 - p_2$, can show the following MAB instances require $\Omega(\sqrt{H})$ queries to distinguish

$$p_1, \quad p_2, \quad p_3, \quad \ldots, p_n \tag{11}$$

$$p_1, \quad p_1 + \eta, \quad p_3, \quad \ldots, p_n \tag{12}$$

$$\ldots \tag{13}$$

$$p_1, \quad p_2, \quad p_3, \quad \ldots, p_1 + \eta \tag{14}$$

[1] Ambainis 2012.

Open questions

Open questions

Thank you for your attention, here are our open problems.

1. Can we improve the efficiency of our quantum algorithm. In particular, can we remove a factor of n from inside the logs?
2. Can we construct quantum algorithms with favorable regret? Actually, we have found it difficult to formulate this problem in the quantum setting.
3. Can we construct fast quantum algorithms for Markov decision processes?