Quantum algorithms for reinforcement learning with a generative model[1]

Daochen Wang (University of Maryland)

QTML 2021



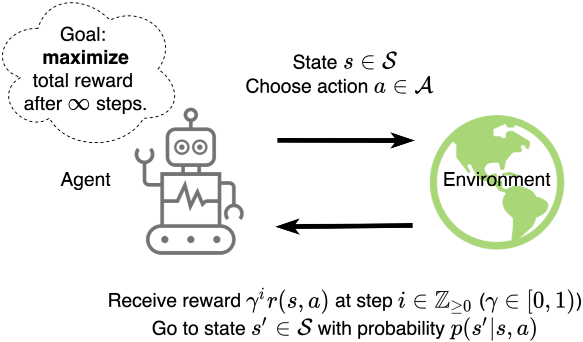

Aarthi Sundaram | Robin Kothari | Ashish Kapoor | Martin Roetteler

(Microsoft Research)

[1]Full paper appears in ICML 2021: http://proceedings.mlr.press/v139/wang21w.html

Reinforcement Learning (RL)

Main question of RL: how should an agent interact with its environment to maximize its total reward?



An RL algorithm is typically required to output (i) an *optimal policy* $\pi^* : \mathcal{S} \to \mathcal{A}$, (ii) the *optimal value function* $v^* : \mathcal{S} \to \mathbb{R}$, and (iii) the *optimal Q-value function* $q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

RL has many applications in robotics, engineering, gaming, natural language processing, finance, and so on

	Playing Go	Self-driving cars
States ($\mathcal{S}$)	Board positions	Cells of a finite 2D grid
Actions ($\mathcal{A}$)	Valid moves	$\{$up, down, left, right, stay$\}$
Rewards	win: $+1$ draw: 0 lose: -1	destination cell reached: 1 destination cell not reached: -1

Image credit: Nature

Classical and quantum generative models (1/2)

- Classical RL often assumes we have query access to an oracle $\mathcal{C}$ that can, for any $(s, a) \in \mathcal{S} \times \mathcal{A}$ of our choice, sample $s' \in \mathcal{S}$ with probability $p(s'|s, a)$.
- $\mathcal{C}$ is known as a (classical) generative model.
- If we have the circuit for $\mathcal{C}$, then we can systematically and efficiently construct a quantum oracle $\mathcal{Q}$ such that

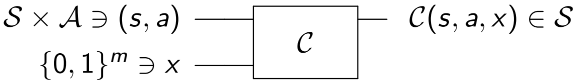
$$\mathcal{Q} : |s\rangle |a\rangle |0\rangle |0\rangle \mapsto |s\rangle |a\rangle \sum_{s'} \sqrt{p(s'|s, a)} \left|s'\right\rangle \left|\psi_{s',s,a}\right\rangle , \quad (1)$$

  where $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ and $\{\left|\psi_{s',s,a}\right\rangle\}_{s',s,a}$ are some quantum states.
- We call $\mathcal{Q}$ a *quantum generative model*.

Classical and quantum generative models (2/2)

Why $\mathcal{Q} : |s\rangle |a\rangle |0\rangle |0\rangle \mapsto |s\rangle |a\rangle \sum_{s'} \sqrt{p(s'|s,a)} |s'\rangle \left|\psi_{s',s,a}\right\rangle$?

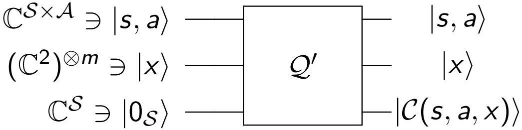
- The circuit $\mathcal{C}$ must be a *deterministic* circuit taking two inputs and producing one output:



(2)

such that $\mathrm{Pr}_{x \sim_U \{0,1\}^m}(\mathcal{C}(s,a,x) = s') = p(s'|s,a)$.

- We can quantize $\mathcal{C}$ as per usual to give a quantum circuit $\mathcal{Q}'$:



(3)

- Appending one Hadamard gate to each of the m qubits in the $|x\rangle$ register before running $\mathcal{Q}'$ and changing the input in the second register to ket of the all-zeros bitstring gives $\mathcal{Q}$.

Quantum algorithms for RL

Summary of quantum speedups

Notation: $S := |\mathcal{S}|$, $A := |\mathcal{A}|$, $\Gamma := (1-\gamma)^{-1}$, $\epsilon :=$ max error; q^*, v^*, $\pi^* :=$ optimal (Q-value function, value function, policy).

Goal: output an ϵ-accurate estimate of	Classical query complexity[2]	Quantum query complexity (our work)	
	Upper and lower bound	Upper bound	Lower bound
q^*	$\frac{SA\Gamma^3}{\epsilon^2}$	$\frac{SA\Gamma^{1.5}}{\epsilon}$	$\frac{SA\Gamma^{1.5}}{\epsilon}$
v^*, π^*	$\frac{SA\Gamma^3}{\epsilon^2}$	$\min\{\frac{SA\Gamma^{1.5}}{\epsilon}, \frac{S\sqrt{A}\Gamma^3}{\epsilon}\}$	$\frac{S\sqrt{A}\Gamma^{1.5}}{\epsilon}$

[2] Sidford et. al. (2018) and Azar et. al. (2013)

Quantum speedups from applying quantum mean estimation and maximum finding to value iteration

- Quantum mean estimation (Montanaro, 2015) estimates $\mathbb{E}[X]$ to error ϵ using $\tilde{O}(\sqrt{\mathrm{Var}[X]}/\epsilon)$ quantum queries to X.
- Quantum maximum finding (Dürr and Høyer, 1996) finds the maximum of a size-n list using $\tilde{O}(\sqrt{n})$ quantum queries to it.
- They can speed up, e.g., the value iteration algorithm for v^*:

  $v \leftarrow \mathbf{0} \in \mathbb{R}^A$
  for $\ell = 1, 2, \ldots, L = \tilde{O}(\Gamma)$ **do**
      for $s \in \mathcal{S}$ **do**
          $v(s) \leftarrow \max_{a \in \mathcal{A}} \{r(s,a) + \gamma \mathbb{E}[v(s') \mid s' \sim p(\cdot|s,a)]\}$
      end
  end

- But this value iteration turns out to be highly sub-optimal so we speed up a modern variant (Sidford et. al., 2018) instead which gives us the (near-)optimal bounds in the summary.

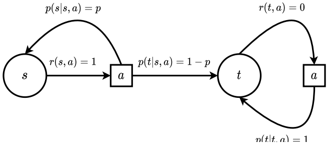
The total variance technique: a technical challenge

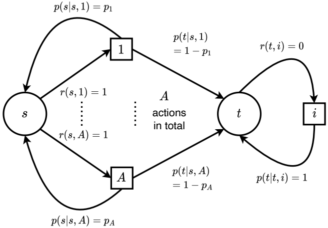
Consider n random variables $Y_1, \ldots, Y_n$ such that we know an upper bound B on their *total* standard deviation. Suppose we have (appropriate) query access to the Y_is and wish to estimate *each* of their means such that the *total* error is $\leq \epsilon$.

1. Chebyshev's inequality easily shows that $\tilde{O}(B^2/\epsilon^2)$ queries suffice for this task classically. Quantumly we would like to have $\tilde{O}(B/\epsilon)$, a quadratic speedup.
2. Problem: quantum algorithms that try to emulate Chebyshev's inequality (Montanaro, 2015; Hamoudi and Magniez, 2018) require an upper bound on the variance of *each* Y_i to work and simply using B^2 for each leads to a bound of $\tilde{O}(nB/\epsilon)$ which is highly sub-optimal.
3. We solve this problem by showing that getting a total error of $\epsilon + \eta$ can be achieved using $\tilde{O}(B/\epsilon)$ quantum queries, where $\eta > 0$ is some small additional error. We then show η can be dealt with by other parts of our algorithm.

Quantum lower bounds proved by a reduction from the computation of certain Boolean functions

- We have quantum query lower bounds on computing Boolean functions {PARITY, OR, approximate counting}. By quantum composition theorems (Reichardt, 2011), we also have quantum query lower bounds on compositions of these functions.
- We reduce the computation of such compositions to the computation of q^*, v^*, and π^* on certain hard RL instances that we construct. This implies our quantum lower bounds.





Open problems

Here are some open problems:

1. Can we circumvent our quantum lower bounds by asking for particular entries of q^*, v^*, or π^*, or maybe these quantities encoded in a quantum state?
2. Can we close the gap between the upper and lower bounds for computing v^* and π^*?
3. Our quantum algorithms quantize model-free classical algorithms. Can we quantize model-based ones?

Thank you for your attention!