Quantum Algorithms for Reinforcement Learning with a Generative Model

Daochen Wang (University of Maryland)

ICML 2021

Aarthi Sundaram Robin Kothari Ashish Kapoor Martin Roetteler

(Microsoft Research)

Quantum computing basics

- Classical computers use bits, quantum computers use qubits.
- A single qubit can be in a state that is a superposition of 0 and 1. n qubits can be in a state that is a superposition over all 2^n bitstrings of length n.
- Quantum computers can efficiently manipulate these quantum states to solve certain problems much faster than classical computers.

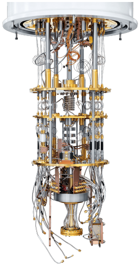

Image credit:
Rigetti Computing

RL in the quantum generative model

- We consider a γ-discounted Markov Decision Process (MDP) given a *classical* generative model G for sampling state transitions.
- Fact: *assuming* G is given as a classical circuit with N gates, then we can efficiently convert G to a quantum circuit $\mathcal{G}$ with $O(N)$ gates that can sample state transitions in superposition.
- We develop quantum algorithms that use $\mathcal{G}$ to solve the MDP. We call $\mathcal{G}$ the *quantum generative model* and the number of times an algorithm uses $\mathcal{G}$ its quantum sample complexity.

Summary of quantum speedups

Notation: $S =$ size of state space, $A =$ size of action space, $\Gamma = 1/(1-\gamma)$, $\epsilon =$ accuracy

Goal: output an ϵ-accurate estimate of	Classical sample complexity	Quantum sample complexity[1]	
	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^*	$\frac{SA\Gamma^3}{\epsilon^2}$	$\min\{\frac{SA\Gamma^{1.5}}{\epsilon}, \frac{S\sqrt{A}\Gamma^3}{\epsilon}\}$	$\frac{S\sqrt{A}\Gamma^{1.5}}{\epsilon}$

[1] This equals the quantum *time* complexity up to log factors assuming access to quantum random access memory (QRAM).

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

- Quantum mean estimation (Montanaro, 2015) estimates $\mathbb{E}[X]$ to accuracy ϵ using $\tilde{O}(\sqrt{\mathrm{Var}[X]}/\epsilon)$ quantum samples of 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 $\ell = 1, 2, \ldots, \tilde{O}(\frac{1}{1-\gamma})$ **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 above value iteration is highly sub-optimal so we speed up a modern version (Sidford et. al., 2018; Wainwright, 2019) which gives us the (near-)optimal bounds in the summary.

Thank you!