Introduction to quantum computing (Part II)

Anthony Leverrier, Inria Paris

Quantum computing and scientific research:
state of the art and potential impact in nuclear physics

Outline of the course

course 1: Basic quantum mechanical principles of quantum computing

- motivation for studying quantum algorithms
- the various models for quantum computing
- the circuit model
- a simple quantum algorithm: Deutsch-Josza

course 2: More advanced quantum algorithms

- Grover's algorithm for search
- linear systems (HHL) and machine learning
- quantum supremacy
- the NISQ era

Grover's algorithm for search

The search problem

The problem

Input: function $f : \{0,1\}^n \to \{0,1\}$. Find x such that $f(x) = 1$ or output no solution if no such x.

Complexity

- randomized classical algorithm: $\Theta(2^n)$ queries if single correct value
- Grover's algorithm: $O(\sqrt{2^n})$ queries and $O(n\sqrt{2^n})$ other gates

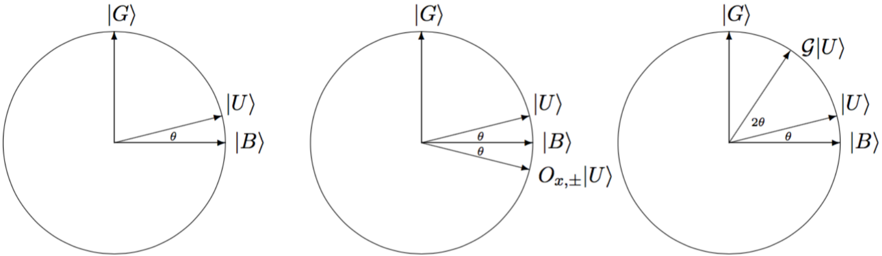
$\Longrightarrow$ quadratic speedup

Idea of the algorithm

Start with uniform superposition (via Hadamard: $H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_x |x\rangle$):

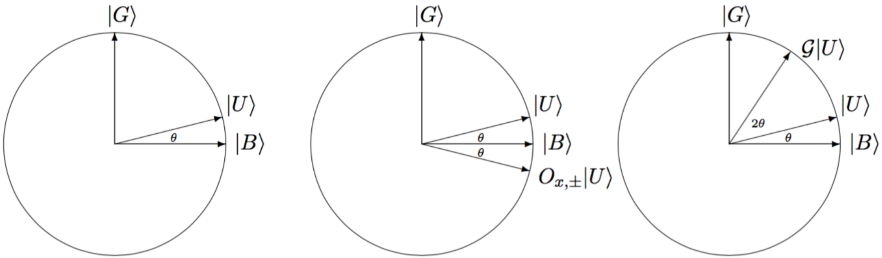
$$|U\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle = \sin\theta|G\rangle + \cos\theta|B\rangle$$

- $\sin\theta = \sqrt{t/2^n}$ and $t = \#\{x \,|\, f(x) = 1\}$
- *good state* $|G\rangle = \frac{1}{\sqrt{t}} \sum_{x \text{ s.t. } f(x)=1} |x\rangle$, *bad state* $|B\rangle = \frac{1}{\sqrt{2^n - t}} \sum_{x \text{ s.t. } f(x)=0} |x\rangle$



goal: rotate in the $\{|B\rangle, |G\rangle\}$ plane to reach $|G\rangle$

How to implement rotation



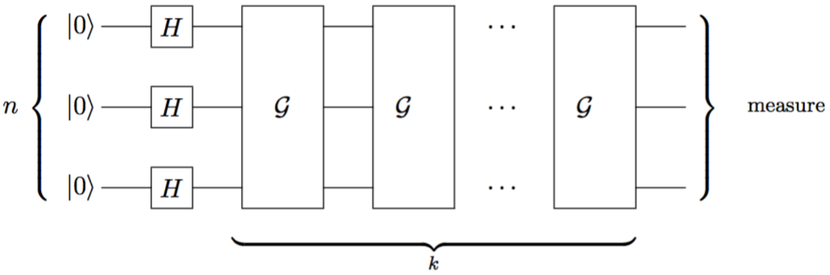
perform two reflections:

- through $|B\rangle$ by calling the oracle $O_{f,\pm} : |x\rangle \mapsto (-1)^{f(x)}|x\rangle$
- through $|U\rangle$ by $H^{\otimes n} R H^{\otimes n} = 2|U\rangle\langle U| - \mathbb{1}$, where $R : |x\rangle \to (-1)^{[x \neq 0^n]}|x\rangle$

define $\mathcal{G} = H^{\otimes n} R H^{\otimes n} O_{f,\pm} \implies$ *rotation of angle 2θ*

Grover's algorithm

assuming we know the fraction of solutions $t/2^n = \sin^2\theta \approx \theta^2$



1. start with $|U\rangle = H^{\otimes n}|0\rangle$
2. repeat $k \approx \frac{\pi/2}{2\theta} = O(1/\sqrt{t/2^n}) = O(1/\sqrt{p_{good}})$ times the rotation $\mathcal{G}$ of angle 2θ
3. measure and check that the outcome is a solution

Generalizations of Grover's algorithm

- the algorithm is very general and search problems occur everywhere
- there are many variants
  - amplitude amplification
  - amplitude estimation
  - quantum walks (generalize random walks)
- for all these problems, *quadratic improvement* in complexity compared to classical algorithms

linear systems (HHL) and machine learning

Until a few years ago, the only main algorithms were:

- Grover
- Shor for factorization (exploits the quantum Fourier transform)
- some variants / generalization (e.g. quantum walks)

  (also quantum chemistry ...)

then came HHL and the prospect of *quantum machine learning*

HHL = Harrow, Hassidim, Lloyd (2009)

The HHL algorithm, "exponential speedup"

the problem

Given an $n \times n$ matrix A and a vector $b \in \mathbb{C}^n$, solve the linear system:

$$Ax = b$$

Complexity

- classically, complexity at least n^2
- quantumly, $O(\log n)$!!

How is that even possible??

by cheating a little bit... Read the fine prints!

The HHL algorithm, "exponential speedup"

the problem

Given an $n \times n$ matrix A and a vector $b \in \mathbb{C}^n$, solve the linear system:

$$Ax = b$$

the algorithm (Harrow, Hassidim, Lloyd 2009)

Given access to a matrix A and a vector b such that

1. b can be efficiently loaded in a quantum memory: Quantum Random Access Memory (QRAM)
2. A has sparsity s per row and condition number κ

there exists a Q algo that

- outputs $|x\rangle = \sum_{i=1}^{n} x_i |i\rangle$ where $Ax = b$
- runs in time polynomial in (s, κ) and *logarithmic in the dimension*

Is this algorithm useful?? (does not return the complete answer)

Description of the HHL algorithm

- without loss of generality, A is Hermitian, otherwise consider $\begin{pmatrix} 0 & \mathrm{A} \\ \mathrm{A}^\dagger & 0 \end{pmatrix}$
- spectral decomposition of A:
$$\mathrm{A} = \sum \lambda_i |\mathrm{E}_i\rangle\langle \mathrm{E}_i|$$
$$\text{with} \qquad \Lambda \leq |\lambda_i| \leq \|\mathrm{A}\| \qquad \forall i$$
- $|b\rangle$ is given by $|b\rangle = \sum b_i |\mathrm{E}_i\rangle$. This is a $\log_2(n)$-qubit state.
- we want to prepare the state
$$|\mathrm{A}^{-1}b\rangle \propto \sum_i \frac{b_i}{\lambda_i} |\mathrm{E}_i\rangle$$

Description of the HHL algorithm

- we want to prepare the state

$$|A^{-1}b\rangle \propto \sum_i \frac{b_i}{\lambda_i}|E_i\rangle$$

- start with $|b\rangle = \sum b_i|E_i\rangle$

quantum phase estimation

Given a unitary U and a state $|\psi\rangle$ such that $U|\psi\rangle = e^{i\theta}|\psi\rangle$, QPE returns an approximation of θ within error ε using $O(1/\varepsilon)$ controlled-U operations.

- apply *quantum phase estimation* under e^{-iA} with state $|b\rangle$

$$\sum b_i|E_i\rangle \xrightarrow{\text{QPE}} \sum b_i|E_i\rangle|\lambda_i\rangle$$

- add an ancillary qubit $|0\rangle$ and rotate it through an angle $\arcsin(\Lambda/\lambda_i)$

$$\sum b_i|E_i\rangle|0\rangle|\lambda_i\rangle \xrightarrow{\text{controlled-rotation}} \sum b_i|E_i\rangle\left(\frac{\Lambda}{\lambda_i}|1\rangle + \sqrt{1-\frac{\Lambda^2}{\lambda_i^2}}|0\rangle\right)|\lambda_i\rangle$$

Description of the HHL algorithm

- we want to prepare the state

$$|A^{-1}b\rangle \propto \sum_i \frac{b_i}{\lambda_i}|E_i\rangle$$

uncomputing

Given a circuit $|x\rangle|0\rangle \to |x\rangle|f(x)\rangle$, one can uncompute it to obtain

$$|x\rangle|f(x)\rangle \to |x\rangle|0\rangle$$

- *uncompute* the quantum phase estimation

$$\sum b_i|E_i\rangle\left(\frac{\Lambda}{\lambda_i}|1\rangle + \sqrt{1-\frac{\Lambda^2}{\lambda_i^2}}|0\rangle\right)|\lambda_i\rangle \quad \xrightarrow{\mathrm{QPE}^{-1}} \quad \sum b_i|E_i\rangle\left(\frac{\Lambda}{\lambda_i}|1\rangle + \sqrt{1-\frac{\Lambda^2}{\lambda_i^2}}|0\rangle\right)|0\rangle$$

Description of the HHL algorithm

- we want to prepare the state $|A^{-1}b\rangle \propto \sum_i \frac{b_i}{\lambda_i}|E_i\rangle$
- we have $\sum b_i|E_i\rangle \left(\frac{\Lambda}{\lambda_i}|1\rangle + \sqrt{1-\frac{\Lambda^2}{\lambda_i^2}}|0\rangle\right)$
- measure the second qubit:
  - with probability $\frac{\Lambda^2}{|\lambda_i|^2} \geq \frac{\Lambda^2}{\|A\|^2}$, the outcome is 1 and we get the state

$$\propto \sum_i \frac{b_i}{\lambda_i}|E_i\rangle = |A^{-1}b\rangle,$$

    as expected

- using amplitude amplification (similar to Grover's algorithm), sufficient to repeat the procedure $O(\|A\|/\Lambda) = O(\kappa)$ times

Caveats of the HHL algorithm

- finding the full answer $\vec{x}$ from $|x\rangle$ requires $O(n)$ repetitions
- HHL is useful if we only need some features of $\vec{x}$, such as moments or expectation values $\vec{x}^\dagger B\vec{x}$ for some sparse matrix B
- the state $|b\rangle$ should be prepared on a quantum computer, or with QRAM. Might be expensive
- A should be well conditioned, i.e. $\frac{\max_i |\lambda_i|}{\min_i |\lambda_i|}$ not too large
- e^{-iA} should be efficiently simulatable

To get an exponential algorithm, one needs a scenario where all these points are addressed...

Quantum machine learning

HHL initiated the field of *quantum machine learning* (subroutine for many algos).

Least Square Fitting [Wiebe, Braun, Lloyd 12]

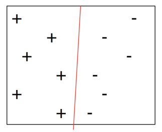
- input: N labelled points (x_i, y_i)
- output: A fit function $f(x, \lambda) = \sum_j f_j(x)\lambda_j$ that minimizes

$$\mathrm{err} = \sum_{i=1}^{N} |f(x_i, \lambda) - y_i|^2$$

Using HHL, the algorithm returns $|\lambda\rangle \propto \sum \lambda_i |i\rangle$.

Support Vector Machine [Lloyd, Mohseni, Rebentrost 13]

- input: M labelled N-dimensional points (x_i, y_i), $x_i \in \mathbb{R}^N, y_i \in \{-1, 1\}$
- output: A maximum margin hyperplane that separates the classes



SVM can be recast as a system of linear equations
$\implies$ the quantum algorithm returns $|w\rangle$, the normal vector to the hyperplane
Application: classify data by estimating the inner product with w (l_2-norm)

Quantum machine learning

Singular Value Estimation

- Sampling eigenvalues/vectors [Lloyd, Mohseni, Rebentrost 13]
  For a psd matrix with trace 1, quantum algorithm that efficiently samples an eigenvector with corresponding eigenvalue
- Singular Value Estimation [LMR 13, Prakash 15]
  Given a matrix and a singular vector, outputs an estimate of the singular value

Not clear whether these algorithms offer a true speedup

- to obtain a good complexity, the data should be "nice", b should be efficiently accessible. Maybe there are efficient classical algorithms in such cases?

$\Longrightarrow$ HHL should be seen as a template for a generic algorithm: one needs to provide a setup where

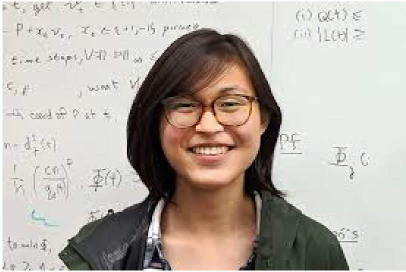
- b can be loaded efficiently in the quantum memory
- the data is sufficiently nice
- we're not interested in x but in quantities efficiently computable from $|x\rangle$

Quantum machine learning: speedup for a real-life problem

Quantum recommendation systems [Kerenidis, Prakash 16]

- input: a hidden preference matrix T with $T_{i,j} \in \{0,1\}$, depending on whether product j is "good" for user i
- output: a high value of row i (i.e. a recommendation for user i)
- idea: the preference matrix is approximately *low rank*
  - step 1 (offline): construction of a low rank approximation of T
  - step 2: new customer reveals some preference and the system outputs a recommendation (good whp) $\implies$ speedup
- crucial that the algorithm doesn't try to reconstruct the full matrix!
- complexity: polylog(mn) $\implies$ *exponential speedup for a real-life problem!*

Big surprise last year: Ewin Tang



- *dequantization* of the algorithm
  $\implies$ classical algorithm with complexity $\mathrm{polylog}(mn)$!!
- but the dependence in the rank of the matrix is terrible
  $\implies$ still an important polynomial speedup for the quantum computer

Last 6 months

- Many quantum algorithms have been dequantized!

1. arXiv:1811.04909 [pdf, other] cs.DS quant-ph

Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension

Authors: András Gilyén, Seth Lloyd, Ewin Tang

Abstract: We construct an efficient classical analogue of the quantum matrix inversion algorithm (HHL) for low-rank matrices. Inspired by recent work of Tang, assuming length-square sampling access to input data, we implement the pseudoinverse of a low-rank matrix and sample from the solution to the problem $Ax = b$ using fast sampling techniques. We implement the pseudo-inverse by finding an approximate sing... ▽ More

Submitted 12 November, 2018; **originally announced** November 2018.

Comments: 10 pages

2. arXiv:1811.00414 [pdf, ps, other] cs.DS cs.IR cs.LG quant-ph

Quantum-inspired classical algorithms for principal component analysis and supervised clustering

Authors: Ewin Tang

Abstract: We describe classical analogues to quantum algorithms for principal component analysis and nearest-centroid clustering. Given sampling assumptions, our classical algorithms run in time polylogarithmic in input, matching the runtime of the quantum algorithms with only polynomial slowdown. These algorithms are evidence that their corresponding problems do not yield exponential quantum speedups. To b... ▽ More

Submitted 30 October, 2018; **originally announced** November 2018.

Comments: 5 pages

3. arXiv:1807.04271 [pdf, ps, other] cs.IR cs.DS cs.LG quant-ph doi 10.1145/3313276.3316310

A quantum-inspired classical algorithm for recommendation systems

Authors: Ewin Tang

Abstract: We give a classical analogue to Kerenidis and Prakash's quantum recommendation system, previously believed to be one of the strongest candidates for provably exponential speedups in quantum machine learning. Our main result is an algorithm that, given an $m \times n$ matrix in a data structure supporting certain ℓ^2-norm sampling operations, outputs an ℓ^2-norm sample from a rank-k app... ▽ More

Submitted 9 May, 2019; **v1** submitted 10 July, 2018; **originally announced** July 2018.

Comments: 32 pages; revised structure of document, improved runtime, simplified algorithm and notation

- situation far from settled but problems with sparse matrices but *large rank* probably can't be dequantized

quantum supremacy

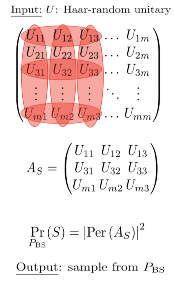
What is quantum supremacy?

- we would like a *convincing experimental demonstration of the fact that quantum computers are much more efficient/faster* than classical ones
- Shor's algorithm? exponential speedup?
  - not ready to outperform classical algorithms (would need many protected qubits-
  - no proof that factoring cannot be done efficiently with a classical computer
- simulation of physical system?
  - already been done
  - no proof that there doesn't exist a classical algorithm
- a solution: *sampling problems!*
  $\implies$ a quantum computer can efficiently sample from a distribution hard to emulate classically
- current race to be the first to demonstrate quantum supremacy (Google)

Boson sampling

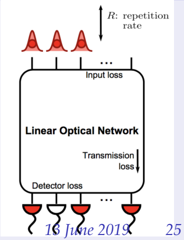
The problem

- given input unitary U, sample from specific probability distribution
- Aaronson, Arkhipov (2013): approximate efficient classical algo for Boson sampling would imply collapse of the polynomial hierarchy to 2nd level



Quantum solution

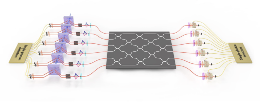
- efficient quantum algorithm of a "simple" quantum computer (no need for quantum error correction)



Boson sampling: quantum experiments vs classical simulation

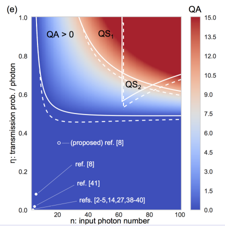
Experiments aren't that easy

- need to prepare many indistinguishable single photons, without much loss, and good detectors
- state of the art: 5 photons in 12 optical modes:
  A. Zhong & al. PRL 121, 250505 (2018)



Progress on the classical side

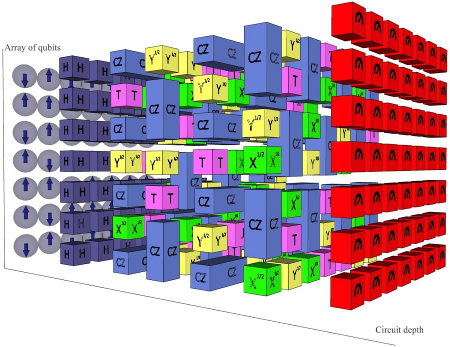
- better classical algorithms for simulation:
  A. Neville & al. Nature Physics, 13(12) 2017



$\Longrightarrow$ maybe not the best approach to quantum supremacy

Random circuit sampling

- similar to Boson sampling, but with random circuit



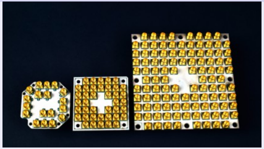
(from Google)

- again good theoretical arguments to show that efficient classical simulation is impossible, but quantum simulation is indeed efficient
- advantage: can use the same superconducting circuits as for quantum computers

Boson sampling: quantum experiments vs classical simulation

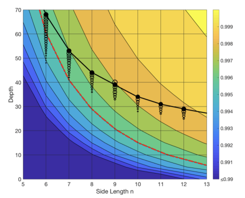
Competition between experimental platforms

- IBM 50-qubit, Intel 49-qubit
- Google Britlescone 72-qubit processor



Progress on the classical side

- better classical algorithms
  e.g. with Alibaba supercomputers J. Chen & al. arXiv:1805.01450
- simulation of circuit of depth 40 for grid of 9×9 qubits
- (right) required 2-qubit gate fidelity to achieve 5% output fidelity



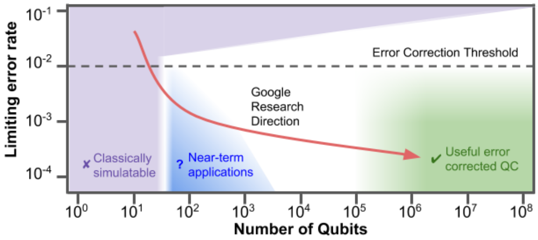
$\Longrightarrow$ very fruitful competition between experiments and classical simulation

The NISQ era

Noisy Intermediate Scale-Quantum technology

What's next?

- from Google



- what to do with circuits without fault tolerance?
- *Noisy Intermediate-Scale Quantum* (NISQ) technology

  $\Longrightarrow$ excellent survey addressed to non specialist by John Preskill "Quantum Computing in the NISQ era and beyond"
  `https://quantum-journal.org/papers/q-2018-08-06-79`

Optimization problems

- formula with m constraints on n bits
- find an *assignment satisfying as many constraints* as possible
- what's the *maximum number of constraints that can be satisfied* at the same time?
- finding the exact solution (or a good approximate one) is often *NP-hard*
- but often a *large gap* between best classical algo and barrier of NP-hardness

$\Longrightarrow$ room for quantum algorithms?

Quantum optimizers

hybrid classical-quantum algorithms

- quantum processor prepares an n-qubit state
- measure all the qubits and process classical outcomes with classical optimizer
- optimizer instructs how to modify quantum state preparation
- repeat until convergence

e.g.

- *Quantum Approximate Optimization Algorithm (QAOA):* application to classical combinatorial optimization pb
- *Variational Quantum Eigensolver (VQE):* finding low-energy states of many-particle quantum systems (large molecules)

Quantum annealing

- What about D-Wave? claims to have a *2000-qubit machine*
- not a circuit-based quantum computer, but a *quantum annealer*
- solves optimization problems, often correctly but no clear speedup
- really a noisy version (with poor quality qubits) of adiabatic quantum computing
- no theoretical argument that it can provide a speedup
- so far, mostly applied to cases where the annealing is *stoquastic* (= also easy for classical computer)
- we'll see what future non-stochastic quantum annealers can do
- also potential applications to quantum simulation?

Recap

Recap

- historical quantum algorithms were mostly variations of Grover and Shor
- many recent algorithms based on matrix inversion (HHL)
  - very relevant for *quantum machine learning*, but unclear which algorithms offer a true speedup (dequantization)
- *short-term challenges:*
  - *quantum supremacy:* speedup for "useless tasks"
  - *Noisy Intermediate-Scale Quantum (NISQ) technology:* useful tasks with small and noisy quantum computers
  - analog quantum simulation

Thanks!