The Church-Turing thesis in a quantum world

Ashley Montanaro

Centre for Quantum Information and Foundations,
Department of Applied Mathematics and Theoretical Physics,
University of Cambridge

April 17, 2012

EPSRC Engineering and Physical Sciences Research Council

Introduction

Quantum complexity theory [Bernstein and Vazirani '97]

Just as the theory of computability has its foundations in the Church-Turing thesis, computational complexity rests on a modern strengthening of this thesis, which asserts that any "reasonable" model of computation can be efficiently simulated on a probabilistic Turing machine...

Introduction

Quantum complexity theory [Bernstein and Vazirani '97]

Just as the theory of computability has its foundations in the Church-Turing thesis, computational complexity rests on a modern strengthening of this thesis, which asserts that any "reasonable" model of computation can be efficiently simulated on a probabilistic Turing machine...

However, the Turing Machine fails to capture all physically realizable computing devices for a fundamental reason: the Turing Machine is based on a classical physics model of the Universe, whereas current physical theory asserts that the Universe is quantum physical.

Introduction

Quantum complexity theory [Bernstein and Vazirani '97]

Just as the theory of computability has its foundations in the Church-Turing thesis, computational complexity rests on a modern strengthening of this thesis, which asserts that any "reasonable" model of computation can be efficiently simulated on a probabilistic Turing machine...

However, the Turing Machine fails to capture all physically realizable computing devices for a fundamental reason: the Turing Machine is based on a classical physics model of the Universe, whereas current physical theory asserts that the Universe is quantum physical.

What does this imply for the Church-Turing thesis?

Introduction

Quantum computers can be simulated by classical computers (with exponential slowdown).

- In fact, in terms of complexity theory, we even have BQP$\subseteq$PSPACE: quantum computers can be simulated space-efficiently by classical computers.
- So the "original" (aka weak) Church-Turing thesis is not affected by quantum computation.

Introduction

Quantum computers can be simulated by classical computers (with exponential slowdown).

- In fact, in terms of complexity theory, we even have BQP⊆PSPACE: quantum computers can be simulated space-efficiently by classical computers.
- So the "original" (aka weak) Church-Turing thesis is not affected by quantum computation.

However, there are certain quantum computations which we don't know how to simulate classically without exponential slowdown.

- The canonical example is factoring: Shor's quantum algorithm factorises an n-digit integer in time poly(n), but the best known classical algorithm takes time super-polynomial in n.
- So quantum computers pose a significant challenge to the strong Church-Turing thesis.

This talk

I will briefly discuss several aspects of this challenge:

This talk

I will briefly discuss several aspects of this challenge:

- The ability of quantum computers to simulate physical systems which we don't know how to simulate efficiently classically;

This talk

I will briefly discuss several aspects of this challenge:

- The ability of quantum computers to simulate physical systems which we don't know how to simulate efficiently classically;

- Models of computation where quantum computers provably outperform classical computers;

This talk

I will briefly discuss several aspects of this challenge:

- The ability of quantum computers to simulate physical systems which we don't know how to simulate efficiently classically;

- Models of computation where quantum computers provably outperform classical computers;

- How quantum computation helps us understand classical complexity theory.

Simulating physical systems

- There are quantum systems for which no efficient classical simulation is known, but which we can simulate on a universal quantum computer.

Simulating physical systems

- There are quantum systems for which no efficient classical simulation is known, but which we can simulate on a universal quantum computer.
- What does it mean to "simulate" a physical system?
- According to the OED, simulation is "the technique of imitating the behaviour of some situation or process (whether economic, military, mechanical, etc.) by means of a suitably analogous situation or apparatus".

Simulating physical systems

- There are quantum systems for which no efficient classical simulation is known, but which we can simulate on a universal quantum computer.
- What does it mean to "simulate" a physical system?
- According to the OED, simulation is "the technique of imitating the behaviour of some situation or process (whether economic, military, mechanical, etc.) by means of a suitably analogous situation or apparatus".
- What we will take simulation to mean here is approximating the dynamics of a physical system.

Simulating physical systems

- There are quantum systems for which no efficient classical simulation is known, but which we can simulate on a universal quantum computer.

- What does it mean to "simulate" a physical system?

- According to the OED, simulation is "the technique of imitating the behaviour of some situation or process (whether economic, military, mechanical, etc.) by means of a suitably analogous situation or apparatus".

- What we will take simulation to mean here is approximating the dynamics of a physical system.

- We are given a description of a system, and would like to determine something about its state at time t.

Simulating physical systems

- According to the laws of quantum mechanics, time evolution of the state $|\psi\rangle$ of a quantum system is governed by Schrödinger's equation,

$$i\hbar \frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle,$$

where $H(t)$ is a linear operator known as the Hamiltonian of the system and $\hbar$ is a constant (which we will absorb into $H(t)$).

Simulating physical systems

- According to the laws of quantum mechanics, time evolution of the state $|\psi\rangle$ of a quantum system is governed by Schrödinger's equation,

$$i\hbar \frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle,$$

  where $H(t)$ is a linear operator known as the Hamiltonian of the system and $\hbar$ is a constant (which we will absorb into $H(t)$).
- In the time-independent setting where $H(t) = H$,

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle.$$

Simulating physical systems

- According to the laws of quantum mechanics, time evolution of the state $|\psi\rangle$ of a quantum system is governed by Schrödinger's equation,

$$i\hbar \frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle,$$

  where $H(t)$ is a linear operator known as the Hamiltonian of the system and $\hbar$ is a constant (which we will absorb into $H(t)$).

- In the time-independent setting where $H(t) = H$,

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle.$$

- Given H specifying a physical system, we would like to approximate the operator

$$U(t) = e^{-iHt}.$$

Simulating physical systems

Why can't we do this classically just by calculating $U(t)$?

Simulating physical systems

Why can't we do this classically just by calculating $U(t)$?

- In general, H is too big to write down explicitly. If H describes a system of n particles (atoms, photons, ...), it has dimension exponential in n.

Simulating physical systems

Why can't we do this classically just by calculating $U(t)$?

- In general, H is too big to write down explicitly. If H describes a system of n particles (atoms, photons, ...), it has dimension exponential in n.

- However, with a quantum computer we can approximate $U(t)$ for the physically meaningful class of k-local Hamiltonians.

- These are Hamiltonians which are given by a sum of terms describing interactions between at most $k = O(1)$ particles. So H is described by a set of $O(1)$-dimensional matrices.

The quantum simulation algorithm (sketch)

Assume we would like to simulate a Hamiltonian $H = \sum_j H_j$.

1. Prepare the desired initial state $|\psi\rangle$.

The quantum simulation algorithm (sketch)

Assume we would like to simulate a Hamiltonian $H = \sum_j H_j$.

1. Prepare the desired initial state $|\psi\rangle$.
2. Write
$$e^{-iHt} \approx \prod_j e^{-iH_j t}$$
(accurate for small enough t). As each H_j only acts non-trivially on $O(1)$ particles, $e^{-iH_j t}$ can be implemented efficiently on a quantum computer.

The quantum simulation algorithm (sketch)

Assume we would like to simulate a Hamiltonian $H = \sum_j H_j$.

1. Prepare the desired initial state $|\psi\rangle$.
2. Write
$$e^{-iHt} \approx \prod_j e^{-iH_j t}$$
(accurate for small enough t). As each H_j only acts non-trivially on $O(1)$ particles, $e^{-iH_j t}$ can be implemented efficiently on a quantum computer.
3. Concatenate the approximations to produce a state
$$|\widetilde{\psi(t)}\rangle \approx e^{-iHt}|\psi\rangle.$$

The quantum simulation algorithm (sketch)

Assume we would like to simulate a Hamiltonian $H = \sum_j H_j$.

1. Prepare the desired initial state $|\psi\rangle$.
2. Write
$$e^{-iHt} \approx \prod_j e^{-iH_j t}$$
(accurate for small enough t). As each H_j only acts non-trivially on $O(1)$ particles, $e^{-iH_j t}$ can be implemented efficiently on a quantum computer.
3. Concatenate the approximations to produce a state
$$|\widetilde{\psi(t)}\rangle \approx e^{-iHt}|\psi\rangle.$$
4. Perform a measurement to extract information from $|\widetilde{\psi(t)}\rangle$.

Provable separations

- In the setting of time complexity, we conjecture that quantum computers are more powerful than classical computers, but have no proof.

Provable separations

- In the setting of time complexity, we conjecture that quantum computers are more powerful than classical computers, but have no proof.
- One model in which separations are provable is the model of query complexity.
- In this model, we want to compute a known function $f(x)$ using the smallest possible worst-case number of queries to the unknown input $x \in \{0,1\}^n$.

Provable separations

- In the setting of time complexity, we conjecture that quantum computers are more powerful than classical computers, but have no proof.
- One model in which separations are provable is the model of query complexity.
- In this model, we want to compute a known function $f(x)$ using the smallest possible worst-case number of queries to the unknown input $x \in \{0,1\}^n$.
- We have access to x via an oracle which, given input i, returns the bit x_i. We allow the use of randomness and some probability of failure (e.g. up to 1/3).

Provable separations

- In the setting of time complexity, we conjecture that quantum computers are more powerful than classical computers, but have no proof.
- One model in which separations are provable is the model of query complexity.
- In this model, we want to compute a known function $f(x)$ using the smallest possible worst-case number of queries to the unknown input $x \in \{0, 1\}^n$.
- We have access to x via an oracle which, given input i, returns the bit x_i. We allow the use of randomness and some probability of failure (e.g. up to $1/3$).
- For some functions f, clever strategies can allow us to compute $f(x)$ using far fewer than n queries.

Query complexity

- In the quantum version of the model, we can query the bits of x in superposition (i.e. in some sense we can query more than one bit at once).

Query complexity

- In the quantum version of the model, we can query the bits of x in superposition (i.e. in some sense we can query more than one bit at once).
- For many functions f, this allows $f(x)$ to be computed more quickly than is possible classically.

Query complexity

- In the quantum version of the model, we can query the bits of x in superposition (i.e. in some sense we can query more than one bit at once).

- For many functions f, this allows $f(x)$ to be computed more quickly than is possible classically.

- For example, the OR function ($f(x) = 1 \Leftrightarrow x \neq 0$) can be computed using $O(\sqrt{n})$ quantum queries using Grover's algorithm [Grover '97].

Query complexity

- In the quantum version of the model, we can query the bits of x in superposition (i.e. in some sense we can query more than one bit at once).

- For many functions f, this allows $f(x)$ to be computed more quickly than is possible classically.

- For example, the OR function ($f(x) = 1 \Leftrightarrow x \neq 0$) can be computed using $O(\sqrt{n})$ quantum queries using Grover's algorithm [Grover '97].

- However, it is easy to see that any classical algorithm requires $\Omega(n)$ queries.

Knowns and unknowns

We know that:

- If f is a partial function (i.e. the algorithm is allowed to fail on certain inputs x), quantum query complexity can be exponentially smaller than classical query complexity (e.g. [Simon '94]).

Knowns and unknowns

We know that:

- If f is a partial function (i.e. the algorithm is allowed to fail on certain inputs x), quantum query complexity can be exponentially smaller than classical query complexity (e.g. [Simon '94]).
- If f is a total function, there can only be at most a polynomial (6th power) separation [Beals et al '01].

Knowns and unknowns

We know that:

- If f is a partial function (i.e. the algorithm is allowed to fail on certain inputs x), quantum query complexity can be exponentially smaller than classical query complexity (e.g. [Simon '94]).
- If f is a total function, there can only be at most a polynomial (6th power) separation [Beals et al '01].

But there are still many open questions, such as:

- Can we achieve better than a quadratic separation for total functions?

Knowns and unknowns

We know that:

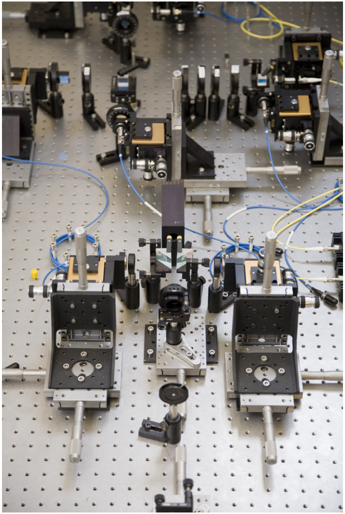
- If f is a partial function (i.e. the algorithm is allowed to fail on certain inputs x), quantum query complexity can be exponentially smaller than classical query complexity (e.g. [Simon '94]).
- If f is a total function, there can only be at most a polynomial (6th power) separation [Beals et al '01].

But there are still many open questions, such as:

- Can we achieve better than a quadratic separation for total functions?
- If the algorithm must succeed with certainty on all inputs, can we achieve better than a constant factor separation? (see [AM, Jozsa and Mitchison '11] for some examples of such separations).

A world without quantum computers?

- Small-scale quantum computers already exist in the lab.
- But what if we never manage to build large-scale quantum computers?
- Or what if quantum computers turn out to be easy to simulate classically?
- Studying quantum computing nevertheless has implications for the rest of computer science.



A computational hardness result

- Let T be a 3-index tensor, i.e. a $d \times d \times d$ array of complex numbers, such that $\sum_{i,j,k} |T_{ijk}|^2 = 1$.
- The injective tensor norm of T is defined as

$$\|T\|_{\text{inj}} := \max_{\substack{x,y,z, \\ \|x\|=\|y\|=\|z\|=1}} \left| \sum_{i,j,k=1}^{d} T_{ijk} x_i y_j z_k \right| .$$

A computational hardness result

- Let T be a 3-index tensor, i.e. a $d \times d \times d$ array of complex numbers, such that $\sum_{i,j,k} |T_{ijk}|^2 = 1$.
- The injective tensor norm of T is defined as

$$\|T\|_{\text{inj}} := \max_{\substack{x,y,z,\\ \|x\|=\|y\|=\|z\|=1}} \left| \sum_{i,j,k=1}^{d} T_{ijk} x_i y_j z_k \right| .$$

Theorem [Harrow & AM '11]

Assume that the (NP-complete) problem 3-SAT on n clauses can't be solved in time subexponential in n. Then there are universal constants $0 < s < c < 1$ such that distinguishing between $\|T\|_{\text{inj}} \leqslant s$ and $\|T\|_{\text{inj}} \geqslant c$ can't be done in time $\text{poly}(d)$.

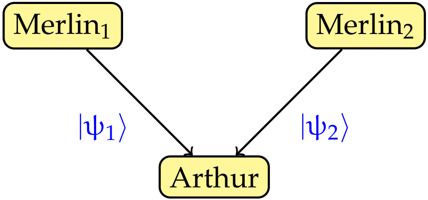
Many other problems in tensor optimisation reduce to computing injective tensor norms.

The proof strategy

Surprisingly, the proof is based on quantum computing – specifically, the framework of quantum Merlin-Arthur games.

The proof strategy

Surprisingly, the proof is based on quantum computing – specifically, the framework of quantum Merlin-Arthur games.



- Arthur has a hard decision problem to solve and has access to two separate provers ("Merlins"), who are all-powerful but cannot be trusted.
- The Merlins want to convince Arthur that the answer to the problem is "yes". Each of them sends Arthur a quantum state ("proof"). He then runs a quantum algorithm to check the proofs.

The proof strategy

- Unlike the situation classically, two Merlins may be more powerful than one: the lack of entanglement helps Arthur tell when the Merlins are cheating.

The proof strategy

- Unlike the situation classically, two Merlins may be more powerful than one: the lack of entanglement helps Arthur tell when the Merlins are cheating.

- Indeed, 3-SAT on n clauses can be solved by a 2-prover protocol with constant probability of error using proofs of length $O(\sqrt{n}\,\mathrm{polylog}(n))$ qubits [Harrow and AM '11].

The proof strategy

- Unlike the situation classically, two Merlins may be more powerful than one: the lack of entanglement helps Arthur tell when the Merlins are cheating.
- Indeed, 3-SAT on n clauses can be solved by a 2-prover protocol with constant probability of error using proofs of length $O(\sqrt{n}\,\mathrm{polylog}(n))$ qubits [Harrow and AM '11].
- And it turns out that the maximal probability with which the Merlins can convince Arthur to output "yes" is given by the injective tensor norm of a tensor T.

The proof strategy

- Unlike the situation classically, two Merlins may be more powerful than one: the lack of entanglement helps Arthur tell when the Merlins are cheating.

- Indeed, 3-SAT on n clauses can be solved by a 2-prover protocol with constant probability of error using proofs of length $O(\sqrt{n}\,\mathrm{polylog}(n))$ qubits [Harrow and AM '11].

- And it turns out that the maximal probability with which the Merlins can convince Arthur to output "yes" is given by the injective tensor norm of a tensor T.

- So, if we could compute $\|T\|_{\mathrm{inj}}$ up to an additive constant in time $\mathrm{poly}(d)$, we would have a subexponential-time algorithm for 3-SAT!

Other classical results with quantum proofs

Some other purely classical problems have quantum solutions.

- Classical communication complexity of the inner product function
- Lower bounds on locally decodable codes
- Rigidity of Hadamard matrices
- Finding low-degree approximating polynomials
- Closure properties of complexity classes
- …

For many more, see the survey "Quantum proofs for classical theorems" [Drucker and de Wolf '09].

Conclusions

- Efficiently simulating the physical world around us appears to require us to use quantum computers.

Conclusions

- Efficiently simulating the physical world around us appears to require us to use quantum computers.
- There are (arguably unrealistic?) models of computation in which quantum computers provably outperform classical computers.

Conclusions

- Efficiently simulating the physical world around us appears to require us to use quantum computers.
- There are (arguably unrealistic?) models of computation in which quantum computers provably outperform classical computers.
- The quantum model can be used to obtain new results in complexity theory without needing to actually build quantum computers.

Conclusions

- Efficiently simulating the physical world around us appears to require us to use quantum computers.
- There are (arguably unrealistic?) models of computation in which quantum computers provably outperform classical computers.
- The quantum model can be used to obtain new results in complexity theory without needing to actually build quantum computers.

Thanks!