Overview of Quantum Algorithmic Tools

András Gilyén

Institute for Quantum Information and Matter

Caltech

Quantum Cryptanalysis of Post-Quantum Cryptography
Berkeley, 22nd February 2020

Block-encodings and

Quantum Singular Value Transformation

Block-encoding

A way to represent large matrices on a quantum computer efficiently

$$U = \begin{bmatrix} A & \cdot \\ \cdot & \cdot \end{bmatrix} \iff A = (\langle 0|^a \otimes I)U\big(|0\rangle^b \otimes I\big).$$

Block-encoding

A way to represent large matrices on a quantum computer efficiently

$$U = \begin{bmatrix} A & \cdot \\ \cdot & \cdot \end{bmatrix} \quad \Longleftrightarrow \quad A = (\langle 0|^a \otimes I)U\big(|0\rangle^b \otimes I\big).$$

One can efficiently construct block-encodings of

Block-encoding

A way to represent large matrices on a quantum computer efficiently

$$U = \begin{bmatrix} A & \cdot \\ \cdot & \cdot \end{bmatrix} \quad \Longleftrightarrow \quad A = (\langle 0|^a \otimes I)U\big(|0\rangle^b \otimes I\big).$$

One can efficiently construct block-encodings of

- an efficiently implementable unitary U,

Block-encoding

A way to represent large matrices on a quantum computer efficiently

$$U = \begin{bmatrix} A & \cdot \\ \cdot & \cdot \end{bmatrix} \iff A = (\langle 0|^a \otimes I)U\big(|0\rangle^b \otimes I\big).$$

One can efficiently construct block-encodings of

- an efficiently implementable unitary U,
- a sparse matrix with efficiently computable elements,

Block-encoding

A way to represent large matrices on a quantum computer efficiently

$$U = \begin{bmatrix} A & \cdot \\ \cdot & \cdot \end{bmatrix} \quad \Longleftrightarrow \quad A = \left(\langle 0|^a \otimes I\right) U \Big(|0\rangle^b \otimes I\Big).$$

One can efficiently construct block-encodings of

- an efficiently implementable unitary U,
- a sparse matrix with efficiently computable elements,
- a matrix stored in a clever data-structure in a QRAM,

Block-encoding

A way to represent large matrices on a quantum computer efficiently

$$U = \begin{bmatrix} A & \cdot \\ \cdot & \cdot \end{bmatrix} \quad \Longleftrightarrow \quad A = (\langle 0|^a \otimes I)U\big(|0\rangle^b \otimes I\big).$$

One can efficiently construct block-encodings of

- an efficiently implementable unitary U,
- a sparse matrix with efficiently computable elements,
- a matrix stored in a clever data-structure in a QRAM,
- a density operator ρ given a unitary preparing its purification.

Block-encoding

A way to represent large matrices on a quantum computer efficiently

$$U = \begin{bmatrix} A & \cdot \\ \cdot & \cdot \end{bmatrix} \quad \Longleftrightarrow \quad A = (\langle 0|^a \otimes I)U\big(|0\rangle^b \otimes I\big).$$

One can efficiently construct block-encodings of

- an efficiently implementable unitary U,
- a sparse matrix with efficiently computable elements,
- a matrix stored in a clever data-structure in a QRAM,
- a density operator ρ given a unitary preparing its purification.
- a POVM operator M given we can sample from the rand.var.: $\mathrm{Tr}(\rho M)$,

Block-encoding

A way to represent large matrices on a quantum computer efficiently

$$U = \begin{bmatrix} A & \cdot \\ \cdot & \cdot \end{bmatrix} \quad \Longleftrightarrow \quad A = (\langle 0|^a \otimes I)U\big(|0\rangle^b \otimes I\big).$$

One can efficiently construct block-encodings of

- an efficiently implementable unitary U,
- a sparse matrix with efficiently computable elements,
- a matrix stored in a clever data-structure in a QRAM,
- a density operator ρ given a unitary preparing its purification.
- a POVM operator M given we can sample from the rand.var.: $\mathrm{Tr}(\rho M)$,

Implementing arithmetic operations on block-encoded matrices

- Given block-encodings A_j we can implement convex combinations.
- Given block-encodings A, B we can implement block-encoding of AB.

Quantum Singular Value Transformation (QSVT)

Main theorem about QSVT (G, Su, Low, Wiebe 2018)

Let $P\colon [-1,1] \to [-1,1]$ be a degree-d odd polynomial map.

Quantum Singular Value Transformation (QSVT)

Main theorem about QSVT (G, Su, Low, Wiebe 2018)

Let $P\colon [-1,1] \to [-1,1]$ be a degree-d odd polynomial map. Suppose that

$$U = \begin{bmatrix} A & . \\ . & . \end{bmatrix} = \begin{bmatrix} \sum_i \varsigma_i |w_i\rangle\langle v_i| & . \\ . & . \end{bmatrix}$$

Quantum Singular Value Transformation (QSVT)

Main theorem about QSVT (G, Su, Low, Wiebe 2018)

Let $P\colon [-1,1] \to [-1,1]$ be a degree-d odd polynomial map. Suppose that

$$U = \begin{bmatrix} A & . \\ . & . \end{bmatrix} = \begin{bmatrix} \sum_i \varsigma_i |w_i\rangle\langle v_i| & . \\ . & . \end{bmatrix} \Longrightarrow U_\Phi = \begin{bmatrix} \sum_i P(\varsigma_i) |w_i\rangle\langle v_i| & . \\ . & . \end{bmatrix},$$

Quantum Singular Value Transformation (QSVT)

Main theorem about QSVT (G, Su, Low, Wiebe 2018)

Let $P\colon [-1,1] \to [-1,1]$ be a degree-d odd polynomial map. Suppose that

$$U = \begin{bmatrix} A & . \\ . & . \end{bmatrix} = \begin{bmatrix} \sum_i \varsigma_i |w_i\rangle\langle v_i| & . \\ . & . \end{bmatrix} \Longrightarrow U_\Phi = \begin{bmatrix} \sum_i P(\varsigma_i) |w_i\rangle\langle v_i| & . \\ . & . \end{bmatrix},$$

where $\Phi(P) \in \mathbb{R}^d$ is efficiently computable and U_Φ is the following circuit:

Quantum Singular Value Transformation (QSVT)

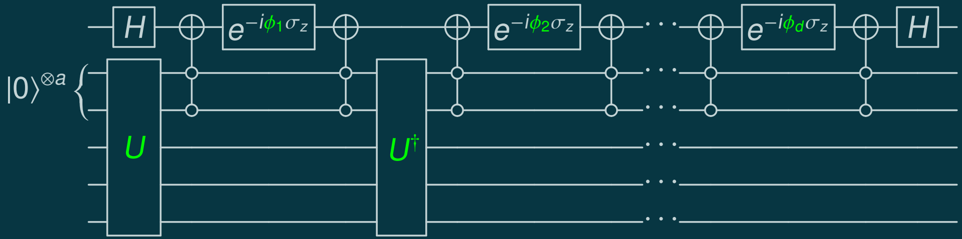
Main theorem about QSVT (G, Su, Low, Wiebe 2018)

Let $P\colon [-1,1] \to [-1,1]$ be a degree-d odd polynomial map. Suppose that

$$U = \begin{bmatrix} A & . \\ . & . \end{bmatrix} = \begin{bmatrix} \sum_i \varsigma_i |w_i\rangle\langle v_i| & . \\ . & . \end{bmatrix} \Longrightarrow U_\Phi = \begin{bmatrix} \sum_i P(\varsigma_i) |w_i\rangle\langle v_i| & . \\ . & . \end{bmatrix},$$

where $\Phi(P) \in \mathbb{R}^d$ is efficiently computable and U_Φ is the following circuit:

Alternating phase modulation sequence $U_\Phi :=$



Quantum Singular Value Transformation (QSVT)

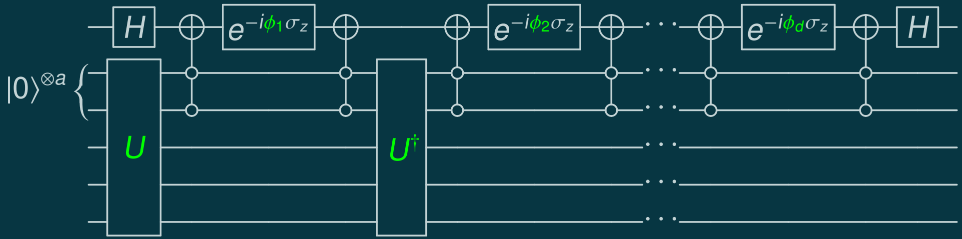
Main theorem about QSVT (G, Su, Low, Wiebe 2018)

Let $P\colon [-1,1] \to [-1,1]$ be a degree-d odd polynomial map. Suppose that

$$U = \begin{bmatrix} A & . \\ . & . \end{bmatrix} = \begin{bmatrix} \sum_i \varsigma_i |w_i\rangle\langle v_i| & . \\ . & . \end{bmatrix} \Longrightarrow U_\Phi = \begin{bmatrix} \sum_i P(\varsigma_i) |w_i\rangle\langle v_i| & . \\ . & . \end{bmatrix},$$

where $\Phi(P) \in \mathbb{R}^d$ is efficiently computable and U_Φ is the following circuit:

Alternating phase modulation sequence $U_\Phi :=$



Simmilar result holds for even polynomials.

Amplitude amplification and estimation

Fixed-point amplitude ampl. (Yoder, Low, Chuang 2014)

Amplitude amplification problem: Given U such that

$$U|\bar{0}\rangle = \sqrt{p}|0\rangle|\psi_{\text{good}}\rangle + \sqrt{1-p}|1\rangle|\psi_{\text{bad}}\rangle, \qquad \text{prepare } |\psi_{\text{good}}\rangle.$$

Amplitude amplification and estimation

Fixed-point amplitude ampl. (Yoder, Low, Chuang 2014)

Amplitude amplification problem: Given U such that

$$U|\bar{0}\rangle = \sqrt{p}|0\rangle|\psi_{\mathrm{good}}\rangle + \sqrt{1-p}|1\rangle|\psi_{\mathrm{bad}}\rangle, \qquad \text{prepare } |\psi_{\mathrm{good}}\rangle.$$

$$\begin{bmatrix} \sqrt{p}\psi_{\mathrm{good}} & \cdot & \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \implies \begin{bmatrix} \psi_{\mathrm{good}} & \cdot & \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

Amplitude amplification and estimation

Fixed-point amplitude ampl. (Yoder, Low, Chuang 2014)

Amplitude amplification problem: Given U such that

$$U|\bar{0}\rangle = \sqrt{p}|0\rangle|\psi_{\text{good}}\rangle + \sqrt{1-p}|1\rangle|\psi_{\text{bad}}\rangle, \qquad \text{prepare } |\psi_{\text{good}}\rangle.$$

$$\begin{bmatrix} \rotatebox{90}{$\sqrt{p}\psi_{\text{good}}$} & \cdot & \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \implies \begin{bmatrix} \rotatebox{90}{ψ_{good}} & \cdot & \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

Note that $(|0\rangle\langle 0| \otimes I)U(|\bar{0}\rangle\langle\bar{0}|) = \sqrt{p}|0, \psi_{\text{good}}\rangle\langle\bar{0}|$; we can apply QSVT.

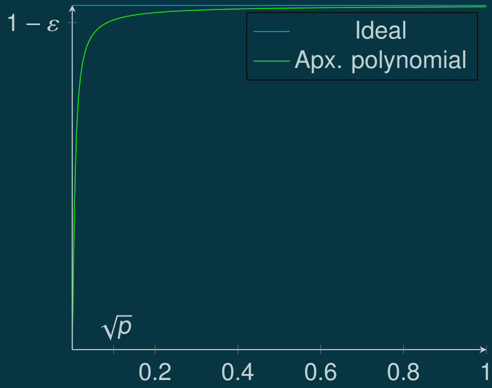
Amplitude amplification and estimation

Fixed-point amplitude ampl. (Yoder, Low, Chuang 2014)

Amplitude amplification problem: Given U such that

$$U|\bar{0}\rangle = \sqrt{p}|0\rangle|\psi_{\text{good}}\rangle + \sqrt{1-p}|1\rangle|\psi_{\text{bad}}\rangle, \quad \text{prepare } |\psi_{\text{good}}\rangle.$$

Amplification using QSVT (degree $\approx \log(1/\varepsilon)/\sqrt{p}$)



Detecting a bias in a quantum sampler

Suppose we are given U such that

$U|\bar{0}\rangle = \sqrt{p'}|0\rangle|\psi_{\text{good}}\rangle + \sqrt{1-p'}|1\rangle|\psi_{\text{bad}}\rangle$, distinguish $p' \leq p - \delta$ vs. $p + \delta \leq p'$.

Detecting a bias in a quantum sampler

Suppose we are given U such that

$$U|\bar{0}\rangle = \sqrt{p'}|0\rangle|\psi_{\text{good}}\rangle + \sqrt{1-p'}|1\rangle|\psi_{\text{bad}}\rangle, \text{ distinguish } p' \leq p - \delta \text{ vs. } p + \delta \leq p'.$$

Can be solved by QSVT (or using amplitude estimation).

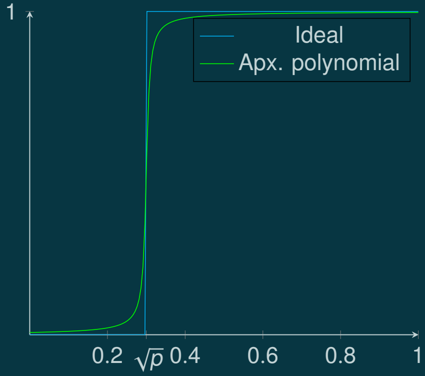
Detecting a bias in a quantum sampler

Suppose we are given U such that

$$U|\bar{0}\rangle = \sqrt{p'}|0\rangle|\psi_{\text{good}}\rangle + \sqrt{1-p'}|1\rangle|\psi_{\text{bad}}\rangle, \text{ distinguish } p' \leq p-\delta \text{ vs. } p+\delta \leq p'.$$

Can be solved by QSVT (or using amplitude estimation).

Bias detection using QSVT (degree $\approx \log(1/\varepsilon)/\delta$)



Speeding up Monte Carlo methods (Montanaro 2015)

Sampling algorithm

Suppose we have a sampling algorithm sampling from a random variable X.

Speeding up Monte Carlo methods (Montanaro 2015)

Sampling algorithm

Suppose we have a sampling algorithm sampling from a random variable X.

Classical complexity of estimation

We can estimate $\mathbb{E}[X]$ with ε precision using $\approx \frac{\sigma^2}{\varepsilon^2}$ samples.

Speeding up Monte Carlo methods (Montanaro 2015)

Sampling algorithm

Suppose we have a sampling algorithm sampling from a random variable X.

Classical complexity of estimation

We can estimate $\mathbb{E}[X]$ with ε precision using $\approx \frac{\sigma^2}{\varepsilon^2}$ samples.

Quantum complexity of estimation

We can estimate $\mathbb{E}[X]$ with ε precision using $\approx \frac{\sigma}{\varepsilon}$ samples.

Speeding up Monte Carlo methods (Montanaro 2015)

Sampling algorithm

Suppose we have a sampling algorithm sampling from a random variable X.

Classical complexity of estimation

We can estimate $\mathbb{E}[X]$ with ε precision using $\approx \frac{\sigma^2}{\varepsilon^2}$ samples.

Quantum complexity of estimation

We can estimate $\mathbb{E}[X]$ with ε precision using $\approx \frac{\sigma}{\varepsilon}$ samples.

- Implement sampler as a quantum circuit
- Replace random input seed with Hadamard gates
- Apply amplitude estimation + combine with other tricks

Quantum walks

Discrete-time random / quantum walks

Discrete-time random walk on a weighted graph

Let $G = (V, E)$ be a finite simple graph, with non-negative edge-weights $w\colon E \to \mathbb{R}_+$.
Transition probability in one step (stochastic matrix)

$$P_{vu} = \Pr(\text{step to } v \,|\, \text{being at } u) = \frac{w_{vu}}{\sum_{v' \in U} w_{v'u}}$$

Discrete-time random / quantum walks

Discrete-time random walk on a weighted graph

Let $G = (V, E)$ be a finite simple graph, with non-negative edge-weights $w\colon E \to \mathbb{R}_+$.
Transition probability in one step (stochastic matrix)

$$P_{vu} = \Pr(\text{step to } v \,|\, \text{being at } u) = \frac{w_{vu}}{\sum_{v' \in U} w_{v'u}}$$

Basic primitives – classical

$$\begin{aligned} \text{(Setup) } S &: \text{ sample } v \text{ with probability } \sigma_v \\ \text{(Update) } U &: \text{ given } u \text{ sample } v \text{ with probability } P_{vu} \\ \text{(Check) } C &: \text{ given } v \text{ check if it is marked, i.e., } v \in M? \end{aligned}$$

Discrete-time random / quantum walks

Discrete-time random walk on a weighted graph

Let $G = (V, E)$ be a finite simple graph, with non-negative edge-weights $w\colon E \to \mathbb{R}_+$.
Transition probability in one step (stochastic matrix)

$$P_{vu} = \Pr(\text{step to } v \mid \text{being at } u) = \frac{w_{vu}}{\sum_{v' \in U} w_{v'u}}$$

Basic primitives – quantum

$$\text{(Setup) } S\colon |0\rangle \mapsto \sum_{v \in V} \sqrt{\sigma_v}|v\rangle$$

$$\text{(Update) } U\colon |0\rangle|u\rangle \mapsto \sum_{v \in V} \sqrt{P_{vu}}|v\rangle|u\rangle$$

$$\text{(Check) } C\colon |0\rangle|v\rangle \mapsto |v \in M\rangle|v\rangle$$

Discrete-time random / quantum walks

Discrete-time random walk on a weighted graph

Let $G = (V, E)$ be a finite simple graph, with non-negative edge-weights $w\colon E \to \mathbb{R}_+$.
Transition probability in one step (stochastic matrix)

$$P_{vu} = \Pr(\text{step to } v \mid \text{being at } u) = \frac{w_{vu}}{\sum_{v' \in U} w_{v'u}}$$

Basic primitives – quantum

$$\text{(Setup) } S\colon |0\rangle \mapsto \sum_{v \in V} \sqrt{\sigma_v}|v\rangle$$

$$\text{(Update) } U\colon |0\rangle|u\rangle \mapsto \sum_{v \in V} \sqrt{P_{vu}}|v\rangle|u\rangle$$

$$\text{(Check) } C\colon |0\rangle|v\rangle \mapsto |v \in M\rangle|v\rangle$$

$$\text{(Walk operator) } W\colon U^\dagger \cdot \mathrm{SWAP} \cdot U = \begin{bmatrix} P & \cdot \\ \cdot & \cdot \end{bmatrix}$$

High-level explanation of quadratic speed-ups

Quantum fast-forwarding (Apers & Sarlette 2018)

We can implement a unitary V such that

$$(\langle 0| \otimes I) V (|0\rangle \otimes I) \overset{\varepsilon}{\approx} P^t$$

with using only $\mathcal{O}\left(\sqrt{t \log(1/\varepsilon)}\right)$ quantum walk steps.

High-level explanation of quadratic speed-ups

Quantum fast-forwarding (Apers & Sarlette 2018)

We can implement a unitary V such that

$$(\langle 0| \otimes I) V (|0\rangle \otimes I) \overset{\varepsilon}{\approx} P^t$$

with using only $\mathcal{O}\left(\sqrt{t \log(1/\varepsilon)}\right)$ quantum walk steps.

Proof:

$$x^t \approx \sum_{k-\sqrt{t}}^{\sqrt{t}} \binom{2t}{t+k} T_k(x)$$

Szegedy quantum walk based search

Suppose we have some unknown marked vertices $M \subset V$.

Quadratically faster hitting

Hitting time: expected time to hit a marked vertex starting from the stationary distr.
Starting from the quantum state $\sum_{v\in V} \sqrt{\pi_v}|v\rangle$ we can

- detect the presence of marked vertices ($M \neq 0$) in time $\mathcal{O}\left(\sqrt{HT}\right)$ (Szegedy 2004)
- find a marked vertex in time $\mathcal{O}\left(\frac{1}{\sqrt{\delta\varepsilon}}\right)$ (Magniez, Nayak, Roland, Sántha 2006)
- find a marked vertex in time $\widetilde{\mathcal{O}}\left(\sqrt{HT}\right)$ (Ambainis, **G**, Jeffery, Kokainis 2019)

Szegedy quantum walk based search

Suppose we have some unknown marked vertices $M \subset V$.

Quadratically faster hitting

Hitting time: expected time to hit a marked vertex starting from the stationary distr.
Starting from the quantum state $\sum_{v \in V} \sqrt{\pi_v}|v\rangle$ we can

- detect the presence of marked vertices ($M \neq 0$) in time $\mathcal{O}\left(\sqrt{HT}\right)$ (Szegedy 2004)
- find a marked vertex in time $\mathcal{O}\left(\frac{1}{\sqrt{\delta\varepsilon}}\right)$ (Magniez, Nayak, Roland, Sántha 2006)
- find a marked vertex in time $\widetilde{\mathcal{O}}\left(\sqrt{HT}\right)$ (Ambainis, **G**, Jeffery, Kokainis 2019)

Starting from arbitrary distributions

Starting from distribution σ on some vertices we can

- detect marked vertices in square-root commute time $\mathcal{O}\left(\sqrt{C_{\sigma,M}}\right)$ (Belovs 2013)
- find a marked vertex in time $\widetilde{\mathcal{O}}\left(\sqrt{C_{\sigma,M}}\right)$ (Piddock; Apers, **G**, Jeffery 2019)

Walks on the Johnson graph (Sántha `arXiv:0808.0059`)

Vertices: $\{S \subset N \colon |S| = K\}$; Edges: $\{(S, S') \colon |S \triangle S'| = 2\}$

Walks on the Johnson graph (Sántha `arXiv:0808.0059`)

Vertices: $\{S \subset N \colon |S| = K\}$; Edges: $\{(S, S') \colon |S \triangle S'| = 2\}$

Element Distinctness

- Black box: Computes f on inputs corresponding to elements of $[n]$
- Question: Are there any $i \neq j \in [n] \times [n]$ such that $f(i) = f(j)$?
- Query complexity: $\mathcal{O}(n^{2/3})$ (Ambainis 2003) $\Omega(n^{2/3})$ (Aaronson & Shi 2001)

Walks on the Johnson graph (Sántha `arXiv:0808.0059`)

Vertices: $\{S \subset N \colon |S| = K\}$; Edges: $\{(S, S') \colon |S \triangle S'| = 2\}$

Element Distinctness

- Black box: Computes f on inputs corresponding to elements of $[n]$
- Question: Are there any $i \neq j \in [n] \times [n]$ such that $f(i) = f(j)$?
- Query complexity: $\mathcal{O}(n^{2/3})$ (Ambainis 2003) $\Omega(n^{2/3})$ (Aaronson & Shi 2001)

Triangle Finding [(2014) non-walk algorithm by Le Gall: $\widetilde{\mathcal{O}}(n^{5/4})$]

- Black box: For any pair $u, v \in V \times V$ tells whether there is an edge uv
- Question: Is there any triangle in G?
- Query complexity: $\mathcal{O}(n^{13/10})$ (Magniez, Sántha, Szegedy 2003)

Walks on the Johnson graph (Sántha `arXiv:0808.0059`)

Vertices: $\{S \subset N : |S| = K\}$; Edges: $\{(S, S') : |S \triangle S'| = 2\}$

Element Distinctness

- Black box: Computes f on inputs corresponding to elements of $[n]$
- Question: Are there any $i \neq j \in [n] \times [n]$ such that $f(i) = f(j)$?
- Query complexity: $\mathcal{O}(n^{2/3})$ (Ambainis 2003) $\Omega(n^{2/3})$ (Aaronson & Shi 2001)

Triangle Finding [(2014) non-walk algorithm by Le Gall: $\widetilde{\mathcal{O}}(n^{5/4})$]

- Black box: For any pair $u, v \in V \times V$ tells whether there is an edge uv
- Question: Is there any triangle in G?
- Query complexity: $\mathcal{O}(n^{13/10})$ (Magniez, Sántha, Szegedy 2003)

Matrix Product Verification

- Black box: Tells any entry of the $n \times n$ matrices A, B or C.
- Question: Does $AB = C$ hold?
- Query complexity: $\mathcal{O}(n^{5/3})$ (Buhrman, Špalek 2004)

k-distinctness

Are there k distinct elements mapped to the same image?

k-distinctness

Are there k distinct elements mapped to the same image?

Quantum algorithms (using QRAM)

- Ambainis (2003) – Johnson graph based quantum walk

$$\mathcal{O}\left(n^{k/(k+1)}\right)$$

k-distinctness

Are there k distinct elements mapped to the same image?

Quantum algorithms (using QRAM)

- Ambainis (2003) – Johnson graph based quantum walk

$$\mathcal{O}\left(n^{k/(k+1)}\right)$$

- Belovs et al. (2013) – 3-distinctness – Electric network based analysis

$$\mathcal{O}\left(n^{5/7}\right)$$

k-distinctness

Are there k distinct elements mapped to the same image?

Quantum algorithms (using QRAM)

- Ambainis (2003) – Johnson graph based quantum walk

$$\mathcal{O}\left(n^{k/(k+1)}\right)$$

- Belovs et al. (2013) – 3-distinctness – Electric network based analysis

$$\mathcal{O}\left(n^{5/7}\right)$$

- Belovs (2012) – Learning graph based algorithm (time-complexity???)

$$\mathcal{O}\left(n^{1-2^{k-2}/(2^k-1)}\right)$$

k-distinctness

Are there k distinct elements mapped to the same image?

Quantum algorithms (using QRAM)

- Ambainis (2003) – Johnson graph based quantum walk

$$\mathcal{O}\left(n^{k/(k+1)}\right)$$

- Belovs et al. (2013) – 3-distinctness – Electric network based analysis

$$\mathcal{O}\left(n^{5/7}\right)$$

- Belovs (2012) – Learning graph based algorithm (time-complexity???)

$$\mathcal{O}\left(n^{1-2^{k-2}/(2^k-1)}\right)$$

Quantum time-space trade-offs?

- Hamoudi & Magniez (arXiv: yesterday) Progress towards conjectured

$$T^2 S \geq \widetilde{\Omega}(n^2)$$

Quantum linear equation solving (HHL)

Direct implementation of the pseudoinverse (HHL)

$Ax = b$: solve the regression problem in a quantum sense – output $|x\rangle \propto A^+|b\rangle$

Direct implementation of the pseudoinverse (HHL)

$Ax = b$: solve the regression problem in a quantum sense – output $|x\rangle \propto A^{+}|b\rangle$

Singular value decomposition and pseudoinverse

Suppose $A = \sum_i \varsigma_i |w_i\rangle\langle v_i| \in \mathbb{C}^{n\times m}$ is a singular value decomposition.
Then the pseudoinverse of A is $A^{+} = \sum_i 1/\varsigma_i |v_i\rangle\langle w_i|$

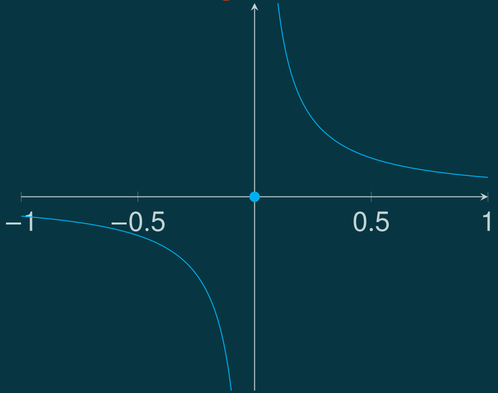
Direct implementation of the pseudoinverse (HHL)

$Ax = b$: solve the regression problem in a quantum sense – output $|x\rangle \propto A^+|b\rangle$

Singular value decomposition and pseudoinverse

Suppose $A = \sum_i \varsigma_i |w_i\rangle\langle v_i| \in \mathbb{C}^{n\times m}$ is a singular value decomposition.
Then the pseudoinverse of A is $A^+ = \sum_i 1/\varsigma_i |v_i\rangle\langle w_i|$ (note $A^\dagger = \sum_i \varsigma_i |v_i\rangle\langle w_i|$).

Implementing the pseudoinverse using QSVT



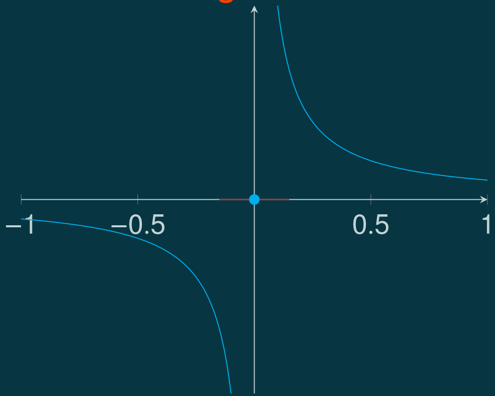
Direct implementation of the pseudoinverse (HHL)

$Ax = b$: solve the regression problem in a quantum sense – output $|x\rangle \propto A^+|b\rangle$

Singular value decomposition and pseudoinverse

Suppose $A = \sum_i \varsigma_i |w_i\rangle\langle v_i| \in \mathbb{C}^{n\times m}$ is a singular value decomposition.
Then the pseudoinverse of A is $A^+ = \sum_i 1/\varsigma_i |v_i\rangle\langle w_i|$ (note $A^\dagger = \sum_i \varsigma_i |v_i\rangle\langle w_i|$).

Implementing the pseudoinverse using QSVT



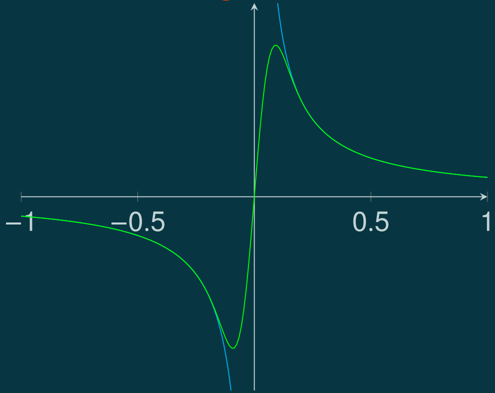
Direct implementation of the pseudoinverse (HHL)

$Ax = b:$ solve the regression problem in a quantum sense – output $|x\rangle \propto A^{+}|b\rangle$

Singular value decomposition and pseudoinverse

Suppose $A = \sum_i \varsigma_i |w_i\rangle\langle v_i| \in \mathbb{C}^{n\times m}$ is a singular value decomposition.
Then the pseudoinverse of A is $A^{+} = \sum_i 1/\varsigma_i |v_i\rangle\langle w_i|$ (note $A^{\dagger} = \sum_i \varsigma_i |v_i\rangle\langle w_i|$).

Implementing the pseudoinverse using QSVT



Direct implementation of the pseudoinverse (HHL)

$Ax = b$: solve the regression problem in a quantum sense – output $|x\rangle \propto A^+|b\rangle$

Singular value decomposition and pseudoinverse

Suppose $A = \sum_i \varsigma_i |w_i\rangle\langle v_i| \in \mathbb{C}^{n\times m}$ is a singular value decomposition.
Then the pseudoinverse of A is $A^+ = \sum_i 1/\varsigma_i |v_i\rangle\langle w_i|$ (note $A^\dagger = \sum_i \varsigma_i |v_i\rangle\langle w_i|$).

Implementation cost (s = sparsity, κ =condition number, ε = precision)

$$O(T \cdot s \cdot \kappa \cdot \mathrm{polylog}(nm\kappa/\varepsilon)),$$

assuming in time T we can

- prepare $|b\rangle$,
- find and compute non-zero elements in a row / column of A.

Potentially exponential speed-ups?

Application for Boolean equations – Chen & Gao 2017

f_i multi-linear polynomials in $\mathbb{Z}_2[x_1, \ldots, x_n]$, solve the system $f_i(x_1, \ldots, x_n) = 0 \colon i \in [c]$

Application for Boolean equations – Chen & Gao 2017

f_i multi-linear polynomials in $\mathbb{Z}_2[x_1, \ldots, x_n]$, solve the system $f_i(x_1, \ldots, x_n) = 0 \colon i \in [c]$

Convert to a complex linear equation system (rough sketch)

- Set $D > \max_i \deg(f_i)$, and denote by m_D be the set of degree $\leq D$ monomials

Application for Boolean equations – Chen & Gao 2017

f_i multi-linear polynomials in $\mathbb{Z}_2[x_1, \ldots, x_n]$, solve the system $f_i(x_1, \ldots, x_n) = 0 \colon i \in [c]$

Convert to a complex linear equation system (rough sketch)

- Set $D > \max_i \deg(f_i)$, and denote by m_D be the set of degree $\leq D$ monomials
- Add trivial constraints $x_i^2 - x_i = 0$ & handle mod 2 freedom by $f \leftarrow f + \sum_j 2^j y_j^{(f)}$

Application for Boolean equations – Chen & Gao 2017

f_i multi-linear polynomials in $\mathbb{Z}_2[x_1, \ldots, x_n]$, solve the system $f_i(x_1, \ldots, x_n) = 0\colon i \in [c]$

Convert to a complex linear equation system (rough sketch)

- Set $D > \max_i \deg(f_i)$, and denote by m_D be the set of degree $\leq D$ monomials
- Add trivial constraints $x_i^2 - x_i = 0$ & handle mod 2 freedom by $f \leftarrow f + \sum_j 2^j y_j^{(f)}$
- For every constraint f include all degree $\leq D$ polynomial from $f \cdot m_D$

Application for Boolean equations – Chen & Gao 2017

f_i multi-linear polynomials in $\mathbb{Z}_2[x_1,\ldots,x_n]$, solve the system $f_i(x_1,\ldots,x_n)=0\colon i\in[c]$

Convert to a complex linear equation system (rough sketch)

- Set $D > \max_i \deg(f_i)$, and denote by m_D be the set of degree $\leq D$ monomials
- Add trivial constraints $x_i^2 - x_i = 0$ & handle mod 2 freedom by $f \leftarrow f + \sum_j 2^j y_j^{(f)}$
- For every constraint f include all degree $\leq D$ polynomial from $f \cdot m_D$
- Treat each monomial in m_D as a variable and solve the linear system over $\mathbb{C}$

Application for Boolean equations – Chen & Gao 2017

f_i multi-linear polynomials in $\mathbb{Z}_2[x_1, \ldots, x_n]$, solve the system $f_i(x_1, \ldots, x_n) = 0 \colon i \in [c]$

Convert to a complex linear equation system (rough sketch)

- Set $D > \max_i \deg(f_i)$, and denote by m_D be the set of degree $\leq D$ monomials
- Add trivial constraints $x_i^2 - x_i = 0$ & handle mod 2 freedom by $f \leftarrow f + \sum_j 2^j y_j^{(f)}$
- For every constraint f include all degree $\leq D$ polynomial from $f \cdot m_D$
- Treat each monomial in m_D as a variable and solve the linear system over $\mathbb{C}$

Suppose there is a unique Boolean solution – good news

- For large enough D there is a unique solution over $\mathbb{C}$

Application for Boolean equations – Chen & Gao 2017

f_i multi-linear polynomials in $\mathbb{Z}_2[x_1, \ldots, x_n]$, solve the system $f_i(x_1, \ldots, x_n) = 0 \colon i \in [c]$

Convert to a complex linear equation system (rough sketch)

- Set $D > \max_i \deg(f_i)$, and denote by m_D be the set of degree $\leq D$ monomials
- Add trivial constraints $x_i^2 - x_i = 0$ & handle mod 2 freedom by $f \leftarrow f + \sum_j 2^j y_j^{(f)}$
- For every constraint f include all degree $\leq D$ polynomial from $f \cdot m_D$
- Treat each monomial in m_D as a variable and solve the linear system over $\mathbb{C}$

Suppose there is a unique Boolean solution – good news

- For large enough D there is a unique solution over $\mathbb{C}$
- b contains at most c non-zero (1) elements (coming from constants in original f_i-s)

Application for Boolean equations – Chen & Gao 2017

f_i multi-linear polynomials in $\mathbb{Z}_2[x_1, \ldots, x_n]$, solve the system $f_i(x_1, \ldots, x_n) = 0 \colon i \in [c]$

Convert to a complex linear equation system (rough sketch)

- Set $D > \max_i \deg(f_i)$, and denote by m_D be the set of degree $\leq D$ monomials
- Add trivial constraints $x_i^2 - x_i = 0$ & handle mod 2 freedom by $f \leftarrow f + \sum_j 2^j y_j^{(f)}$
- For every constraint f include all degree $\leq D$ polynomial from $f \cdot m_D$
- Treat each monomial in m_D as a variable and solve the linear system over $\mathbb{C}$

Suppose there is a unique Boolean solution – good news

- For large enough D there is a unique solution over $\mathbb{C}$
- b contains at most c non-zero (1) elements (coming from constants in original f_i-s)
- $x = A^+b$ is the indicator of monomials evaluating to 1 for the Boolean solution

Application for Boolean equations – Chen & Gao 2017

f_i multi-linear polynomials in $\mathbb{Z}_2[x_1,\ldots,x_n]$, solve the system $f_i(x_1,\ldots,x_n)=0\colon i\in[c]$

Convert to a complex linear equation system (rough sketch)

- Set $D>\max_i \deg(f_i)$, and denote by m_D be the set of degree $\leq D$ monomials
- Add trivial constraints $x_i^2-x_i=0$ & handle mod 2 freedom by $f\leftarrow f+\sum_j 2^j y_j^{(f)}$
- For every constraint f include all degree $\leq D$ polynomial from $f\cdot m_D$
- Treat each monomial in m_D as a variable and solve the linear system over $\mathbb{C}$

Suppose there is a unique Boolean solution – good news

- For large enough D there is a unique solution over $\mathbb{C}$
- b contains at most c non-zero (1) elements (coming from constants in original f_i-s)
- $x=A^+b$ is the indicator of monomials evaluating to 1 for the Boolean solution

Bad news: $\kappa^2\geq\|x\|^2/\|b\|^2\geq\frac{\#\{\text{non-zero variables}\}}{c}=\frac{\binom{k+D-1}{k-1}}{c}\approx\left(\frac{eD}{k}\right)^k$

Classical brute-force: $\sum_{j=0}^{k}\binom{n}{k}\approx\left(\frac{en}{k}\right)^k\implies$ Does not seem to be useful if $D\geq n$

Summary of some relevant quantum speed-ups

Optimization

In general we want to find the best solution $\min_{x \in X} f(x)$

- Unstructured: can be solved with $\mathcal{O}\left(\sqrt{|X|}\right)$ queries (Dürr & Høyer 1996)

Optimization

In general we want to find the best solution $\min_{x \in X} f(x)$

- Unstructured: can be solved with $\mathcal{O}\left(\sqrt{|X|}\right)$ queries (Dürr & Høyer 1996)

Discrete structures:

- Finding the shortest path in a graph
  $\mathcal{O}\left(n^2\right)$ (Dijkstra 1956); quantum $\widetilde{\mathcal{O}}\left(n^{3/2}\right)$ (Dürr, Heiligman, Høyer, Mhalla 2004)
- Matching and flow problems:
  Polynomial speed-ups, typically based on Grover search

Optimization

In general we want to find the best solution $\min_{x \in X} f(x)$

- Unstructured: can be solved with $\mathcal{O}\left(\sqrt{|X|}\right)$ queries (Dürr & Høyer 1996)

Discrete structures:

- Finding the shortest path in a graph
  $\mathcal{O}\left(n^2\right)$ (Dijkstra 1956); quantum $\widetilde{\mathcal{O}}\left(n^{3/2}\right)$ (Dürr, Heiligman, Høyer, Mhalla 2004)
- Matching and flow problems:
  Polynomial speed-ups, typically based on Grover search
- NP–hard problems:
  Quadratic speed-ups for Schöning's algorithm for 3-SAT (Ampl. ampl.)
  Quadratic speed-ups for backtracking (Montanaro '15, Ambainis & Kokainis '17)
  Polynomial speed-ups for dynamical programming, e.g., TSP $2^n \to 1.73^n$
  (Ambainis, Balodis, Iraids, Kokainis, Prūsis, Vihrovs 2018)

Continuous optimization

Convex optimization

- Regression / Linear equation solving (HHL, with quantum output):
  $\mathcal{O}(\kappa s \,\mathrm{polylog}(nm\kappa/\varepsilon))$ (Harrow et al.; Ambainis; Childs et al.; Chakraborty et al.)

Continuous optimization

Convex optimization

- Regression / Linear equation solving (HHL, with quantum output):
  $\mathcal{O}(\kappa s \operatorname{polylog}(nm\kappa/\varepsilon))$ (Harrow et al.; Ambainis; Childs et al.; Chakraborty et al.)
- Linear programs, semidefinite programs
  SDPs: $\widetilde{\mathcal{O}}\big((\sqrt{n}+\sqrt{m})s\gamma^5\big)$ (Brandão et al., van Apeldoorn et al. 2016-18)
  LPs: $\widetilde{\mathcal{O}}\big((\sqrt{n}+\sqrt{m})\gamma^3\big)$, $\widetilde{\mathcal{O}}\big(s\gamma^{3.5}\big)$ (van Apeldoorn & G 2019)
  Zero-sum games: $\widetilde{\mathcal{O}}\big((\sqrt{n}+\sqrt{m})/\varepsilon^3\big)$, $\widetilde{\mathcal{O}}\big(s/\varepsilon^{3.5}\big)$ (van Apeldoorn & G 2019)
- Quantum interior point method (Kerenidis & Prakash 2018)

Continuous optimization

Convex optimization

- Regression / Linear equation solving (HHL, with quantum output):
  $\mathcal{O}(\kappa s \operatorname{polylog}(nm\kappa/\varepsilon))$ (Harrow et al.; Ambainis; Childs et al.; Chakraborty et al.)
- Linear programs, semidefinite programs
  SDPs: $\widetilde{\mathcal{O}}\big((\sqrt{n}+\sqrt{m})s\gamma^5\big)$ (Brandão et al., van Apeldoorn et al. 2016-18)
  LPs: $\widetilde{\mathcal{O}}\big((\sqrt{n}+\sqrt{m})\gamma^3\big)$, $\widetilde{\mathcal{O}}\big(s\gamma^{3.5}\big)$ (van Apeldoorn & G 2019)
  Zero-sum games: $\widetilde{\mathcal{O}}\big((\sqrt{n}+\sqrt{m})/\varepsilon^3\big)$, $\widetilde{\mathcal{O}}\big(s/\varepsilon^{3.5}\big)$ (van Apeldoorn & G 2019)
- Quantum interior point method (Kerenidis & Prakash 2018)
- Exponential speed-up for implementing separation oracles of convex bodies
  (van Apeldoorn et al. 2018; Chakrabarti et al. 2018)

Continuous optimization

Convex optimization

- Regression / Linear equation solving (HHL, with quantum output):
  $O(\kappa s \operatorname{polylog}(nm\kappa/\varepsilon))$ (Harrow et al.; Ambainis; Childs et al.; Chakraborty et al.)
- Linear programs, semidefinite programs
  SDPs: $\widetilde{O}\big((\sqrt{n}+\sqrt{m})s\gamma^5\big)$ (Brandão et al., van Apeldoorn et al. 2016-18)
  LPs: $\widetilde{O}\big((\sqrt{n}+\sqrt{m})\gamma^3\big)$, $\widetilde{O}\big(s\gamma^{3.5}\big)$ (van Apeldoorn & G 2019)
  Zero-sum games: $\widetilde{O}\big((\sqrt{n}+\sqrt{m})/\varepsilon^3\big)$, $\widetilde{O}\big(s/\varepsilon^{3.5}\big)$ (van Apeldoorn & G 2019)
- Quantum interior point method (Kerenidis & Prakash 2018)
- Exponential speed-up for implementing separation oracles of convex bodies
  (van Apeldoorn et al. 2018; Chakrabarti et al. 2018)
- Polynomial speed-up for estimating volumes of convex bodies
  (Chakrabarti, Childs, Hung, Li, Wang, Wu 2019)

Statistics and stochastic estimation algorithms

- Quadratic speed-up for Monte-Carlo methods $\mathcal{O}\left(\frac{\sigma}{\varepsilon}\right)$ (Montanaro 2015)
  Generalizes amplitude estimation (Brassard, Høyer, Mosca, Tapp 1998)

Statistics and stochastic estimation algorithms

- Quadratic speed-up for Monte-Carlo methods $\mathcal{O}\left(\frac{\sigma}{\varepsilon}\right)$ (Montanaro 2015)
  Generalizes amplitude estimation (Brassard, Høyer, Mosca, Tapp 1998)
- Testing equality of a distribution on $[n]$ (with query complexity)
  To an unknown distribution $\widetilde{\mathcal{O}}\left(n^{1/2}\right)$ (Bravyi, Hassidim, Harrow 2009; **G**, Li 2019)
  To a known distribution $\widetilde{\mathcal{O}}\left(n^{1/3}\right)$ (Chakraborty, Fischer, Matsliah, de Wolf 2010)
- Estimating the (Shannon / von Neumann) entropy of a distribution on $[n]$
  Classical distribution: query complexity $\widetilde{\mathcal{O}}\left(n^{1/2}\right)$ (Li & Wu 2017)
  Density operator: query complexity $\widetilde{\mathcal{O}}(n)$ (**G** & Li 2019)

Statistics and stochastic estimation algorithms

- Quadratic speed-up for Monte-Carlo methods $\mathcal{O}\left(\frac{\sigma}{\varepsilon}\right)$ (Montanaro 2015)
  Generalizes amplitude estimation (Brassard, Høyer, Mosca, Tapp 1998)
- Testing equality of a distribution on $[n]$ (with query complexity)
  To an unknown distribution $\widetilde{\mathcal{O}}\left(n^{1/2}\right)$ (Bravyi, Hassidim, Harrow 2009; **G**, Li 2019)
  To a known distribution $\widetilde{\mathcal{O}}\left(n^{1/3}\right)$ (Chakraborty, Fischer, Matsliah, de Wolf 2010)
- Estimating the (Shannon / von Neumann) entropy of a distribution on $[n]$
  Classical distribution: query complexity $\widetilde{\mathcal{O}}\left(n^{1/2}\right)$ (Li & Wu 2017)
  Density operator: query complexity $\widetilde{\mathcal{O}}(n)$ (**G** & Li 2019)
- Estimating the histogram to ε-precision
  Query and time complexity $\widetilde{\mathcal{O}}(1/\varepsilon)$ (Apeldoorn 2020)