Quantum Singular Value Transformation & Its Algorithmic Applications

András Gilyén

Institute for Quantum Information and Matter

Caltech

The Quantum Wave in Computing Boot Camp
Berkeley, 28th January 2020

Quantum walks

Continuous-time quantum / random walks

Laplacian of a weighted graph

Let $G = (V, E)$ be a finite simple graph, with non-negative edge-weights $w \colon E \to \mathbb{R}_+$. The Laplacian is defined as

$$u \neq v\colon\ L_{uv} = w_{uv}, \text{ and } L_{uu} = -\sum_v w_{uv}.$$

Continuous-time quantum / random walks

Laplacian of a weighted graph

Let $G = (V, E)$ be a finite simple graph, with non-negative edge-weights $w\colon E \to \mathbb{R}_+$. The Laplacian is defined as

$$u \neq v\colon\ L_{uv} = w_{uv},\ \text{and}\ L_{uu} = -\sum_{v} w_{uv}.$$

Continuous-time walks

Evolution of the state:

$$\frac{d}{dt} p_u(t) = \sum_{v \in V} L_{uv} p_v(t) \qquad \Longrightarrow \qquad p(t) = e^{tL} p(0)$$

Continuous-time quantum / random walks

Laplacian of a weighted graph

Let $G = (V, E)$ be a finite simple graph, with non-negative edge-weights $w\colon E \to \mathbb{R}_+$. The Laplacian is defined as

$$u \neq v\colon\ L_{uv} = w_{uv},\ \text{and}\ L_{uu} = -\sum_v w_{uv}.$$

Continuous-time walks

Evolution of the state:

$$\frac{d}{dt} p_u(t) = \sum_{v \in V} L_{uv} p_v(t) \qquad \Longrightarrow \qquad p(t) = e^{tL} p(0)$$

$$i\frac{d}{dt} \psi_u(t) = \sum_{v \in V} L_{uv} \psi_v(t) \qquad \Longrightarrow \qquad \psi(t) = e^{-itL} \psi(0)$$

Discrete-time quantum / random walks

Discrete-time Markov-chain on a weighted graph

Transition probability in one step (stochastic matrix)

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

Discrete-time quantum / random walks

Discrete-time Markov-chain on a weighted graph

Transition probability in one step (stochastic matrix)

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

A unitary implementing the update

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

Discrete-time quantum / random walks

Discrete-time Markov-chain on a weighted graph

Transition probability in one step (stochastic matrix)

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

A unitary implementing the update

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

How to erase history? The Szegedy quantum walk operator:

$$W' := U^\dagger \cdot \mathrm{SWAP} \cdot U$$

$$W := U^\dagger \cdot \mathrm{SWAP} \cdot U((2|0\rangle\langle 0| \otimes I) - I)$$

Understanding Szegedy's quantum walk operator

For simplicity let us assume $P_{uv} = P_{vu}$, i.e., the total weight of vertices is constant.

Understanding Szegedy's quantum walk operator

For simplicity let us assume $P_{uv} = P_{vu}$, i.e., the total weight of vertices is constant.

A block-encoding of the Markov chain: $(\langle 0| \otimes I)W'(|0\rangle \otimes I) = P$

Understanding Szegedy's quantum walk operator

For simplicity let us assume $P_{uv} = P_{vu}$, i.e., the total weight of vertices is constant.

A block-encoding of the Markov chain: $(\langle 0| \otimes I) W' (|0\rangle \otimes I) = P$

Proof:

$$\langle 0|\langle u|W'|0\rangle|v\rangle = \langle 0|\langle u|U^\dagger \cdot \mathrm{SWAP} \cdot U|0\rangle|v\rangle = \left(\sum_{v'\in V} \sqrt{P_{v'u}}|v'\rangle|u\rangle\right)^\dagger \mathrm{SWAP} \left(\sum_{u'\in V} \sqrt{P_{u'v}}|u'\rangle|v\rangle\right)$$

Understanding Szegedy's quantum walk operator

For simplicity let us assume $P_{uv} = P_{vu}$, i.e., the total weight of vertices is constant.

A block-encoding of the Markov chain: $(\langle 0| \otimes I) W' (|0\rangle \otimes I) = P$

Proof:

$$\langle 0|\langle u|W'|0\rangle|v\rangle = \langle 0|\langle u|U^\dagger \cdot \mathrm{SWAP} \cdot U|0\rangle|v\rangle = \left(\sum_{v'\in V} \sqrt{P_{v'u}}|v'\rangle|u\rangle\right)^\dagger \mathrm{SWAP}\left(\sum_{u'\in V} \sqrt{P_{u'v}}|u'\rangle|v\rangle\right)$$

Multiple steps of the quantum walk: $(\langle 0| \otimes I) W^k (|0\rangle \otimes I) = T_k(P)$

[$T_k(x) = \cos(k \arccos(x))$ Chebyshev polynomials: $T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$]

Understanding Szegedy's quantum walk operator

For simplicity let us assume $P_{uv} = P_{vu}$, i.e., the total weight of vertices is constant.

A block-encoding of the Markov chain: $(\langle 0| \otimes I)W'(|0\rangle \otimes I) = P$

Proof:

$$\langle 0|\langle u|W'|0\rangle|v\rangle = \langle 0|\langle u|U^\dagger \cdot \mathrm{SWAP} \cdot U|0\rangle|v\rangle = \left(\sum_{v'\in V} \sqrt{P_{v'u}}|v'\rangle|u\rangle\right)^\dagger \mathrm{SWAP}\left(\sum_{u'\in V} \sqrt{P_{u'v}}|u'\rangle|v\rangle\right)$$

Multiple steps of the quantum walk: $(\langle 0| \otimes I)W^k(|0\rangle \otimes I) = T_k(P)$

[$T_k(x) = \cos(k \arccos(x))$ Chebyshev polynomials: $T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$]

Proof: Proceed by induction, observe $T_0(P) = I \checkmark$, $T_1(P) = P \checkmark$

Understanding Szegedy's quantum walk operator

For simplicity let us assume $P_{uv} = P_{vu}$, i.e., the total weight of vertices is constant.

A block-encoding of the Markov chain: $(\langle 0| \otimes I) W' (|0\rangle \otimes I) = P$

Proof:

$$\langle 0|\langle u|W'|0\rangle|v\rangle = \langle 0|\langle u|U^\dagger \cdot \mathrm{SWAP} \cdot U|0\rangle|v\rangle = \left(\sum_{v'\in V} \sqrt{P_{v'u}}|v'\rangle|u\rangle\right)^\dagger \mathrm{SWAP} \left(\sum_{u'\in V} \sqrt{P_{u'v}}|u'\rangle|v\rangle\right)$$

Multiple steps of the quantum walk: $(\langle 0| \otimes I) W^k (|0\rangle \otimes I) = T_k(P)$

[$T_k(x) = \cos(k \arccos(x))$ Chebyshev polynomials: $T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$]

Proof: Proceed by induction, observe $T_0(P) = I\,\checkmark$, $T_1(P) = P\,\checkmark$

$$\begin{aligned}(\langle 0| \otimes I) W^{k+1} (|0\rangle \otimes I) &= (\langle 0| \otimes I) W' ((2|0\rangle\langle 0| \otimes I) - I) W^k (|0\rangle \otimes I) = \\ &= \underbrace{(\langle 0| \otimes I) W' (2|0\rangle}_{2P} \underbrace{\langle 0| \otimes I) W^k (|0\rangle \otimes I)}_{T_k(P)} - \underbrace{(\langle 0| \otimes I) W^{k-1} (|0\rangle \otimes I)}_{T_{k-1}(P)}\end{aligned}$$

Understanding Szegedy's quantum walk operator

For simplicity let us assume $P_{uv} = P_{vu}$, i.e., the total weight of vertices is constant.

A block-encoding of the Markov chain: $(\langle 0| \otimes I) W' (|0\rangle \otimes I) = P$

Proof:

$$\langle 0|\langle u|W'|0\rangle|v\rangle = \langle 0|\langle u|U^\dagger \cdot \mathrm{SWAP} \cdot U|0\rangle|v\rangle = \left(\sum_{v'\in V} \sqrt{P_{v'u}}|v'\rangle|u\rangle\right)^\dagger \mathrm{SWAP} \left(\sum_{u'\in V} \sqrt{P_{u'v}}|u'\rangle|v\rangle\right)$$

Multiple steps of the quantum walk: $(\langle 0| \otimes I) W^k (|0\rangle \otimes I) = T_k(P)$

[$T_k(x) = \cos(k \arccos(x))$ Chebyshev polynomials: $T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$]

Proof: Proceed by induction, observe $T_0(P) = I\,\checkmark$, $T_1(P) = P\,\checkmark$

$$(\langle 0| \otimes I) W^{k+1} (|0\rangle \otimes I) = (\langle 0| \otimes I) W' ((2|0\rangle\langle 0| \otimes I) - I) W^k (|0\rangle \otimes I) =$$

$$= \underbrace{(\langle 0| \otimes I) W' (2|0\rangle}_{2P} \underbrace{\langle 0| \otimes I) W^k (|0\rangle \otimes I)}_{T_k(P)} - \underbrace{(\langle 0| \otimes I) W^{k-1} (|0\rangle \otimes I)}_{T_{k-1}(P)}$$

Are we happy with Chebyshev polynomials?

Linear combination of (non-)unitary mat. [Childs & Wiebe '12, Berry et al. '15]

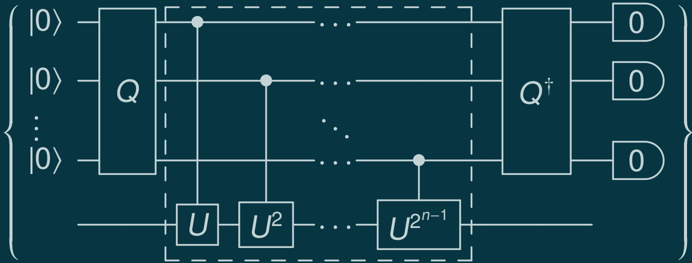
Suppose that $V = \sum_k |k\rangle\langle k| \otimes U^k$, and $Q : |0\rangle \mapsto \sum_i \sqrt{q_i}|i\rangle$ for $q_i \in [0, 1]$, then

Are we happy with Chebyshev polynomials?

Linear combination of (non-)unitary mat. [Childs & Wiebe '12, Berry et al. '15]

Suppose that $V = \sum_k |k\rangle\langle k| \otimes U^k$, and $Q : |0\rangle \mapsto \sum_i \sqrt{q_i}|i\rangle$ for $q_i \in [0,1]$, then

$$(\langle 0|Q^\dagger \otimes I)V(Q|0\rangle \otimes I) =$$



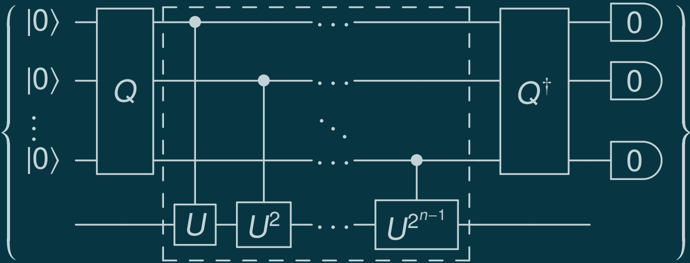
$$= \sum_k q_k U^k$$

Are we happy with Chebyshev polynomials?

Linear combination of (non-)unitary mat. [Childs & Wiebe ’12, Berry et al. ’15]

Suppose that $V = \sum_k |k\rangle\langle k| \otimes U^k$, and $Q : |0\rangle \mapsto \sum_i \sqrt{q_i}|i\rangle$ for $q_i \in [0,1]$, then

$$(\langle 0|Q^\dagger \otimes I)V(Q|0\rangle \otimes I) =$$



$$= \sum_k q_k U^k$$

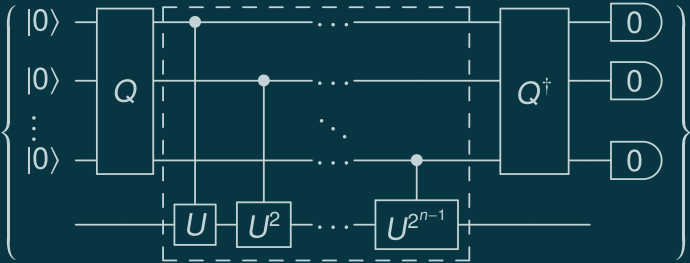
In particular the top-left corner of $\sum_k q_k W^k$ is $\sum_i q_k T_k(P)$.

Are we happy with Chebyshev polynomials?

Linear combination of (non-)unitary mat. [Childs & Wiebe '12, Berry et al. '15]

Suppose that $V = \sum_k |k\rangle\langle k| \otimes U^k$, and $Q : |0\rangle \mapsto \sum_i \sqrt{q_i}|i\rangle$ for $q_i \in [0, 1]$, then

$$(\langle 0|Q^\dagger \otimes I)V(Q|0\rangle \otimes I) =$$



$$= \sum_k q_k U^k$$

In particular the top-left corner of $\sum_k q_k W^k$ is $\sum_i q_k T_k(P)$.

Corollary: Quantum fast-forwarding (Apers & Sarlette 2018)

We can implement a unitary V such that

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

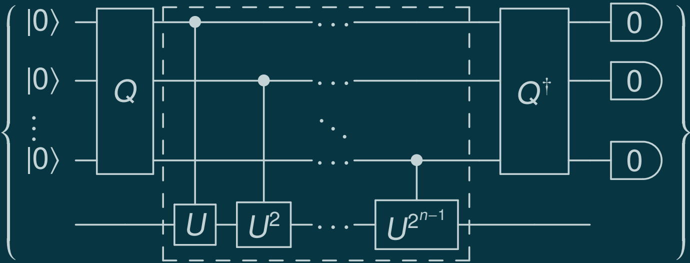
with using only $\mathcal{O}\left(\sqrt{t \log(1/\varepsilon)}\right)$ quantum walk steps.

Are we happy with Chebyshev polynomials?

Linear combination of (non-)unitary mat. [Childs & Wiebe '12, Berry et al. '15]

Suppose that $V = \sum_k |k\rangle\langle k| \otimes U^k$, and $Q : |0\rangle \mapsto \sum_i \sqrt{q_i}|i\rangle$ for $q_i \in [0,1]$, then

$$(\langle 0|Q^\dagger \otimes I)V(Q|0\rangle \otimes I) =$$



$$= \sum_k q_k U^k$$

In particular the top-left corner of $\sum_k q_k W^k$ is $\sum_i q_k T_k(P)$.

Corollary: 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 $O\left(\sqrt{t\log(1/\varepsilon)}\right)$ quantum walk steps. (Proof: $x^t \approx \sum_{k=0}^{\approx\sqrt{t}} T_k(x)$)

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 $\mathcal{O}\big(\sqrt{HT}\big)$ (Szegedy 2004)
- find a marked vertex in time $\mathcal{O}\big(\frac{1}{\sqrt{\delta\varepsilon}}\big)$ (Magniez, Nayak, Roland, Sántha 2006)
- find a marked vertex in time $\widetilde{\mathcal{O}}\big(\sqrt{HT}\big)$ (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)

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

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

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

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

One can efficiently construct block-encodings of

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

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,

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

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

One can efficiently construct block-encodings of

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

Example: Block-encoding sparse matrices

Suppose that A is s-sparse and $|A_{ij}| \leq 1$ for all i, j indices.

Example: Block-encoding sparse matrices

Suppose that A is s-sparse and $|A_{ij}| \leq 1$ for all i, j indices. Given ”sparse-access “ we can efficiently implement unitaries preparing ”rows“

Example: Block-encoding sparse matrices

Suppose that A is s-sparse and $|A_{ij}| \leq 1$ for all i, j indices. Given "sparse-access " we can efficiently implement unitaries preparing "rows"

$$R: |0\rangle|0\rangle|i\rangle \to |0\rangle \sum_k \frac{(\sqrt{A_{ik}})^*}{\sqrt{s}}|i\rangle|k\rangle + |1\rangle|i\rangle|\text{garbage}\rangle,$$

Example: Block-encoding sparse matrices

Suppose that A is s-sparse and $|A_{ij}| \leq 1$ for all i, j indices. Given ”sparse-access “ we can efficiently implement unitaries preparing ”rows“

$$R: |0\rangle|0\rangle|i\rangle \to |0\rangle \sum_k \frac{(\sqrt{A_{ik}})^*}{\sqrt{s}}|i\rangle|k\rangle + |1\rangle|i\rangle|\text{garbage}\rangle,$$

and ”columns“

$$C: |0\rangle|0\rangle|j\rangle \to |0\rangle \sum_\ell \frac{\sqrt{A_{\ell j}}}{\sqrt{s}}|\ell\rangle|j\rangle + |2\rangle|j\rangle|\text{garbage}\rangle,$$

Example: Block-encoding sparse matrices

Suppose that A is s-sparse and $|A_{ij}| \leq 1$ for all i, j indices. Given ”sparse-access “ we can efficiently implement unitaries preparing ”rows“

$$R\colon |0\rangle|0\rangle|i\rangle \to |0\rangle \sum_k \frac{(\sqrt{A_{ik}})^*}{\sqrt{s}}|i\rangle|k\rangle + |1\rangle|i\rangle|\text{garbage}\rangle,$$

and ”columns“

$$C\colon |0\rangle|0\rangle|j\rangle \to |0\rangle \sum_\ell \frac{\sqrt{A_{\ell j}}}{\sqrt{s}}|\ell\rangle|j\rangle + |2\rangle|j\rangle|\text{garbage}\rangle,$$

They form a block-encoding of A/s:

$$\langle 0|\langle 0|\langle i|R^\dagger C|0\rangle|0\rangle|j\rangle$$

Example: Block-encoding sparse matrices

Suppose that A is s-sparse and $|A_{ij}| \leq 1$ for all i, j indices. Given ”sparse-access “ we can efficiently implement unitaries preparing ”rows“

$$R: |0\rangle|0\rangle|i\rangle \to |0\rangle \sum_k \frac{(\sqrt{A_{ik}})^*}{\sqrt{s}}|i\rangle|k\rangle + |1\rangle|i\rangle|\text{garbage}\rangle,$$

and ”columns“

$$C: |0\rangle|0\rangle|j\rangle \to |0\rangle \sum_\ell \frac{\sqrt{A_{\ell j}}}{\sqrt{s}}|\ell\rangle|j\rangle + |2\rangle|j\rangle|\text{garbage}\rangle,$$

They form a block-encoding of A/s:

$$\langle 0|\langle 0|\langle i|R^\dagger C|0\rangle|0\rangle|j\rangle = (R|0\rangle|0\rangle|i\rangle)^\dagger \cdot (C|0\rangle|0\rangle|j\rangle)$$

Example: Block-encoding sparse matrices

Suppose that A is s-sparse and $|A_{ij}| \leq 1$ for all i, j indices. Given "sparse-access " we can efficiently implement unitaries preparing "rows"

$$R: |0\rangle|0\rangle|i\rangle \to |0\rangle \sum_k \frac{(\sqrt{A_{ik}})^*}{\sqrt{s}}|i\rangle|k\rangle + |1\rangle|i\rangle|\text{garbage}\rangle,$$

and "columns"

$$C: |0\rangle|0\rangle|j\rangle \to |0\rangle \sum_\ell \frac{\sqrt{A_{\ell j}}}{\sqrt{s}}|\ell\rangle|j\rangle + |2\rangle|j\rangle|\text{garbage}\rangle,$$

They form a block-encoding of A/s:

$$\langle 0|\langle 0|\langle i|R^\dagger C|0\rangle|0\rangle|j\rangle = (R|0\rangle|0\rangle|i\rangle)^\dagger \cdot (C|0\rangle|0\rangle|j\rangle) = \left(\sum_k \frac{(\sqrt{A_{ik}})^*}{\sqrt{s}}|i\rangle|k\rangle\right)^\dagger \left(\sum_\ell \frac{\sqrt{A_{\ell j}}}{\sqrt{s}}|\ell\rangle|j\rangle\right)$$

Example: Block-encoding sparse matrices

Suppose that A is s-sparse and $|A_{ij}| \leq 1$ for all i, j indices. Given ”sparse-access “ we can efficiently implement unitaries preparing ”rows“

$$R: |0\rangle|0\rangle|i\rangle \rightarrow |0\rangle \sum_k \frac{(\sqrt{A_{ik}})^*}{\sqrt{s}}|i\rangle|k\rangle + |1\rangle|i\rangle|\text{garbage}\rangle,$$

and ”columns“

$$C: |0\rangle|0\rangle|j\rangle \rightarrow |0\rangle \sum_\ell \frac{\sqrt{A_{\ell j}}}{\sqrt{s}}|\ell\rangle|j\rangle + |2\rangle|j\rangle|\text{garbage}\rangle,$$

They form a block-encoding of A/s:

$$\langle 0|\langle 0|\langle i|R^\dagger C|0\rangle|0\rangle|j\rangle = (R|0\rangle|0\rangle|i\rangle)^\dagger \cdot (C|0\rangle|0\rangle|j\rangle) = \left(\sum_k \frac{(\sqrt{A_{ik}})^*}{\sqrt{s}}|i\rangle|k\rangle\right)^\dagger \left(\sum_\ell \frac{\sqrt{A_{\ell j}}}{\sqrt{s}}|\ell\rangle|j\rangle\right) = \frac{A_{ij}}{s}$$

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)

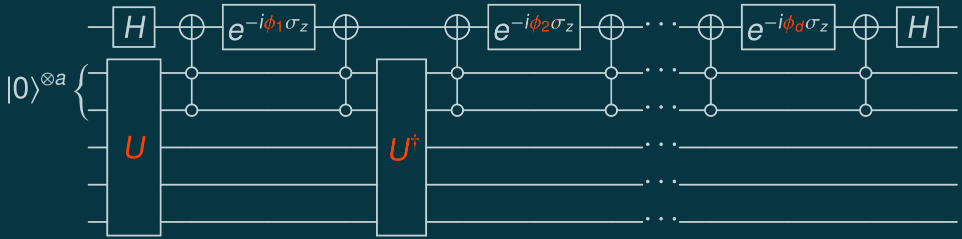
Main theorem about QSVT (G, Su, Low, Wiebe 2018)

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

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

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)

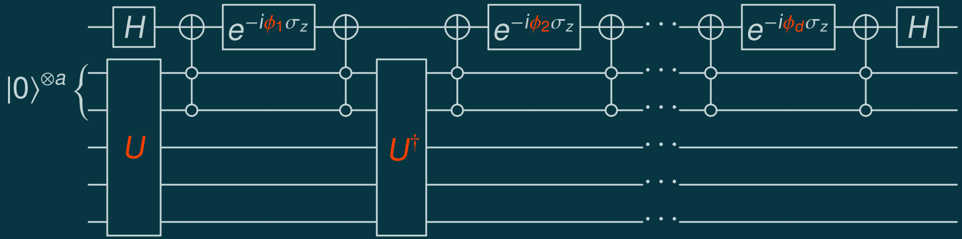
Main theorem about QSVT (G, Su, Low, Wiebe 2018)

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.

Singular vector transformation and projection

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

Amplitude amplification problem: Given U such that

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

Singular vector transformation and projection

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

Amplitude amplification problem: Given U such that

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

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.

Singular vector transformation and projection

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

Amplitude amplification problem: Given U such that

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

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.

Singular vector transformation [Oblivious ampl. ampl. (Berry et al. 2013)]

Given a unitary U, such that

$$A = (\langle 0|^{\otimes a} \otimes I)U(|0\rangle^{\otimes b} \otimes I) = \sum_{i=1}^{k} \varsigma_i |\phi_i\rangle\langle\psi_i|$$

is a singular value decomposition.

Singular vector transformation and projection

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

Amplitude amplification problem: Given U such that

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

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.

Singular vector transformation [Oblivious ampl. ampl. (Berry et al. 2013)]

Given a unitary U, such that

$$A = (\langle 0|^{\otimes a} \otimes I)U(|0\rangle^{\otimes b} \otimes I) = \sum_{i=1}^{k} \varsigma_i |\phi_i\rangle\langle\psi_i|$$

is a singular value decomposition. Transform one copy of a quantum state

$$|\psi\rangle = \sum_{i=i}^{k} \alpha_i |\psi_i\rangle \quad \text{to} \quad |\phi\rangle = \sum_{i=i}^{k} \alpha_i |\phi_i\rangle.$$

Singular vector transformation and projection

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

Amplitude amplification problem: Given U such that

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

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.

Singular vector transformation [Oblivious ampl. ampl. (Berry et al. 2013)]

Given a unitary U, such that

$$A = (\langle 0|^{\otimes a} \otimes I)U(|0\rangle^{\otimes b} \otimes I) = \sum_{i=1}^{k} \varsigma_i |\phi_i\rangle\langle\psi_i|$$

is a singular value decomposition. Transform one copy of a quantum state

$$|\psi\rangle = \sum_{i=i}^{k} \alpha_i |\psi_i\rangle \quad \text{to} \quad |\phi\rangle = \sum_{i=i}^{k} \alpha_i |\phi_i\rangle.$$

If $\varsigma_i \geq \delta$ for all $0 \neq \alpha_i$, we can ε-apx. using QSVT with compl. $\mathcal{O}\left(\frac{1}{\delta}\log\left(\frac{1}{\varepsilon}\right)\right)$.

Direct implementation of the pseudoinverse (HHL)

Singular value decomposition and pseudoinverse

Suppose $A = \sum_i \varsigma_i |w_i\rangle\langle v_i|$ 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)

Singular value decomposition and pseudoinverse

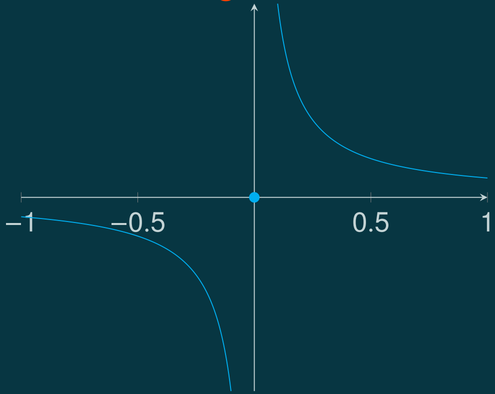
Suppose $A = \sum_i \varsigma_i |w_i\rangle\langle v_i|$ is a singular value decomposition.
Then the pseudoinverse of A is $A^+ = \sum_i 1/\varsigma_i |v_i\rangle\langle w_i|$ (note $A^\dagger = \sum_i \varsigma_i |v_i\rangle\langle w_i|$).

Direct implementation of the pseudoinverse (HHL)

Singular value decomposition and pseudoinverse

Suppose $A = \sum_i \varsigma_i |w_i\rangle\langle v_i|$ 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

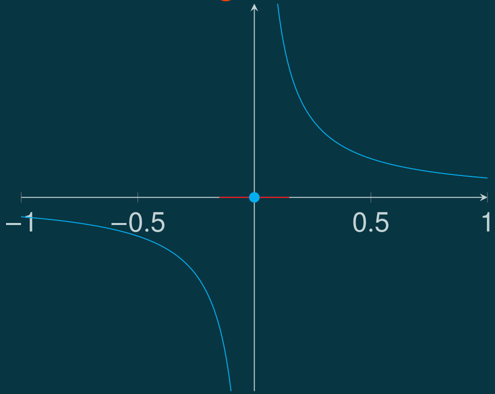


Direct implementation of the pseudoinverse (HHL)

Singular value decomposition and pseudoinverse

Suppose $A = \sum_i \varsigma_i |w_i\rangle\langle v_i|$ 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

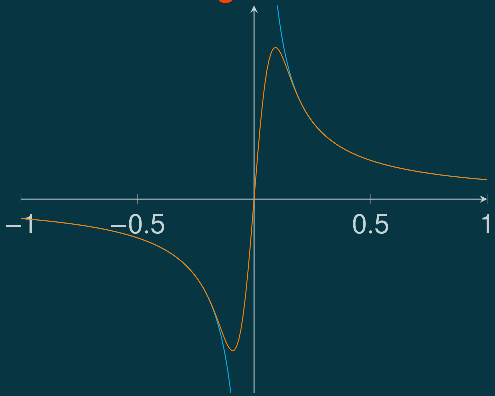


Direct implementation of the pseudoinverse (HHL)

Singular value decomposition and pseudoinverse

Suppose $A = \sum_i \varsigma_i |w_i\rangle\langle v_i|$ 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)

Singular value decomposition and pseudoinverse

Suppose $A = \sum_i \varsigma_i |w_i\rangle\langle v_i|$ 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

Suppose that U is an a-qubit block-encoding of A, and $\left\|A^+\right\| \leq \kappa$.

Direct implementation of the pseudoinverse (HHL)

Singular value decomposition and pseudoinverse

Suppose $A = \sum_i \varsigma_i |w_i\rangle\langle v_i|$ 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

Suppose that U is an a-qubit block-encoding of A, and $\left\|A^+\right\| \leq \kappa$.
By QSVT we can implement an ε-approximate block-encoding of

$$\frac{1}{2\kappa} A^+,$$

using $\mathcal{O}\!\left(\kappa \log\!\left(\frac{1}{\varepsilon}\right)\right)$ queries to U. Finally amplify the result ($\mathcal{O}(\kappa)$ times).

Direct implementation of the pseudoinverse (HHL)

Singular value decomposition and pseudoinverse

Suppose $A = \sum_i \varsigma_i |w_i\rangle\langle v_i|$ 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

Suppose that U is an a-qubit block-encoding of A, and $\left\|A^+\right\| \leq \kappa$.
By QSVT we can implement an ε-approximate block-encoding of

$$\frac{1}{2\kappa} A^+,$$

using $\mathcal{O}\left(\kappa \log\left(\frac{1}{\varepsilon}\right)\right)$ queries to U. Finally amplify the result ($\mathcal{O}(\kappa)$ times).

- Complexity can be improved to $\widetilde{\mathcal{O}}(\kappa)$ using variable-time amplitude-amplification.
- Other variants are possible, such as weighted and generalized least-squares.

The special case of Hermitian matrices

Singular value transf. = eigenvalue transf. [Low & Chuang (2017)]

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

The special case of Hermitian matrices

Singular value transf. = eigenvalue transf. [Low & Chuang (2017)]

Let $P\colon [-1,1] \to [-1,1]$ be a degree-d even/odd polynomial map.
If H is Hermitian, then $P(H)$ coincides with the singular value transform.

The special case of Hermitian matrices

Singular value transf. = eigenvalue transf. [Low & Chuang (2017)]

Let $P\colon [-1,1] \to [-1,1]$ be a degree-d even/odd polynomial map.
If H is Hermitian, then $P(H)$ coincides with the singular value transform.
(Direct corollary: Fast-Forwarding Markov Chains.)

The special case of Hermitian matrices

Singular value transf. = eigenvalue transf. [Low & Chuang (2017)]

Let $P\colon [-1,1] \to [-1,1]$ be a degree-d even/odd polynomial map.
If H is Hermitian, then $P(H)$ coincides with the singular value transform.
(Direct corollary: Fast-Forwarding Markov Chains.)

Removing parity constraint for Hermitian matrices

Let $P\colon [-1,1] \to [-\frac{1}{2}, \frac{1}{2}]$ be a degree-d polynomial map. Suppose that U is an a-qubit block-encoding of a Hermitian matrix H.

The special case of Hermitian matrices

Singular value transf. = eigenvalue transf. [Low & Chuang (2017)]

Let $P\colon [-1,1] \to [-1,1]$ be a degree-d even/odd polynomial map.
If H is Hermitian, then $P(H)$ coincides with the singular value transform.
(Direct corollary: Fast-Forwarding Markov Chains.)

Removing parity constraint for Hermitian matrices

Let $P\colon [-1,1] \to [-\frac{1}{2}, \frac{1}{2}]$ be a degree-d polynomial map. Suppose that U is an a-qubit block-encoding of a Hermitian matrix H. We can implement

$$U' = \begin{bmatrix} P(H) & . \\ . & . \end{bmatrix},$$

using d times U and $U^\dagger$, 1 controlled U, and $\mathcal{O}(ad)$ extra two-qubit gates.

Optimal block-Hamiltonian simulation

Suppose that H is given as an a-qubit block-encoding, i.e., $U = \begin{bmatrix} H & \cdot \\ \cdot & \cdot \end{bmatrix}$.

Optimal block-Hamiltonian simulation

Suppose that H is given as an a-qubit block-encoding, i.e., $U = \begin{bmatrix} H & \cdot \\ \cdot & \cdot \end{bmatrix}$.

Complexity of block-Hamiltonians simulation [Low & Chuang (2016)]

Given $t, \varepsilon > 0$, implement a unitary U', which is ε close to e^{itH}. Can be achieved with query complexity

$$\mathcal{O}(t + \log(1/\varepsilon)).$$

Gate complexity is $\mathcal{O}(a)$ times the above.

Optimal block-Hamiltonian simulation

Suppose that H is given as an a-qubit block-encoding, i.e., $U = \begin{bmatrix} H & \cdot \\ \cdot & \cdot \end{bmatrix}$.

Complexity of block-Hamiltonians simulation [Low & Chuang (2016)]

Given $t, \varepsilon > 0$, implement a unitary U', which is ε close to e^{itH}. Can be achieved with query complexity

$$\mathcal{O}(t + \log(1/\varepsilon)).$$

Gate complexity is $\mathcal{O}(a)$ times the above.

Proof sketch

Approximate to ε-precision $\sin(tx)$ and $\cos(tx)$ with polynomials of degree as above. Then use QSVT and combine even/odd parts.

Gibbs sampling

Suppose that H is given as $U = \begin{bmatrix} H & \cdot \\ \cdot & \cdot \end{bmatrix}$. The goal is to prepare $\rho \propto e^{-\beta H}$.

Gibbs sampling

Suppose that H is given as $U = \begin{bmatrix} H & \cdot \\ \cdot & \cdot \end{bmatrix}$. The goal is to prepare $\rho \propto e^{-\beta H}$.

The basic algorithm (Inspired by Poulin & Wocjan 2009)

- Start with a maximally mixed state I/N, and apply the map $e^{-\frac{\beta}{2}H}$.

Gibbs sampling

Suppose that H is given as $U = \begin{bmatrix} H & \cdot \\ \cdot & \cdot \end{bmatrix}$. The goal is to prepare $\rho \propto e^{-\beta H}$.

The basic algorithm (Inspired by Poulin & Wocjan 2009)

- Start with a maximally mixed state I/N, and apply the map $e^{-\frac{\beta}{2}H}$.
- The new state is proportional to $e^{-\frac{\beta}{2}H}\left(\frac{I}{N}\right)e^{-\frac{\beta}{2}H} = e^{-\beta H}$.

Gibbs sampling

Suppose that H is given as $U = \begin{bmatrix} H & \cdot \\ \cdot & \cdot \end{bmatrix}$. The goal is to prepare $\rho \propto e^{-\beta H}$.

The basic algorithm (Inspired by Poulin & Wocjan 2009)

- Start with a maximally mixed state I/N, and apply the map $e^{-\frac{\beta}{2}H}$.
- The new state is proportional to $e^{-\frac{\beta}{2}H}\left(\frac{I}{N}\right)e^{-\frac{\beta}{2}H} = e^{-\beta H}$.
- Finally, apply amplitude amplification. Probability of success?

Gibbs sampling

Suppose that H is given as $U = \begin{bmatrix} H & \cdot \\ \cdot & \cdot \end{bmatrix}$. The goal is to prepare $\rho \propto e^{-\beta H}$.

The basic algorithm (Inspired by Poulin & Wocjan 2009)

- Start with a maximally mixed state I/N, and apply the map $e^{-\frac{\beta}{2}H}$.
- The new state is proportional to $e^{-\frac{\beta}{2}H}\left(\frac{I}{N}\right)e^{-\frac{\beta}{2}H} = e^{-\beta H}$.
- Finally, apply amplitude amplification. Probability of success?

Final algorithm

- Use minimum finding (Dürr & Høyer 1996; van Apeldoorn, **G**, Gribling, de Wolf 2017) to find an approximation of the ground state energy H_0.

Gibbs sampling

Suppose that H is given as $U = \begin{bmatrix} H & \cdot \\ \cdot & \cdot \end{bmatrix}$. The goal is to prepare $\rho \propto e^{-\beta H}$.

The basic algorithm (Inspired by Poulin & Wocjan 2009)

- Start with a maximally mixed state I/N, and apply the map $e^{-\frac{\beta}{2}H}$.
- The new state is proportional to $e^{-\frac{\beta}{2}H}\left(\frac{I}{N}\right)e^{-\frac{\beta}{2}H} = e^{-\beta H}$.
- Finally, apply amplitude amplification. Probability of success?

Final algorithm

- Use minimum finding (Dürr & Høyer 1996; van Apeldoorn, **G**, Gribling, de Wolf 2017) to find an approximation of the ground state energy H_0.
- Use the previous procedure but with the map $e^{-\frac{\beta}{2}(H - H_0 I)}$
- The final complexity is $\widetilde{O}\left(\beta\sqrt{N}\right)$

Summarizing the various speed-ups by QSVT

Speed-up	Source of speed-up	Examples of algorithms
Exponential	Dimensionality of the Hilbert space	Hamiltonian simulation
	Precise polynomial approximations	Improved HHL algorithm
Quadratic	Singular value = square root of probability	Grover search
	Singular values are easier to distinguish	Amplitude estimation
	Close-to-1 singular values are more flexible	Quantum walks

Summarizing the various speed-ups by QSVT

Speed-up	Source of speed-up	Examples of algorithms
Exponential	Dimensionality of the Hilbert space	Hamiltonian simulation
	Precise polynomial approximations	Improved HHL algorithm
Quadratic	Singular value = square root of probability	Grover search
	Singular values are easier to distinguish	Amplitude estimation
	Close-to-1 singular values are more flexible	Quantum walks

Some other applications

- Fast QMA amplification, fast quantum OR lemma
- Quantum Machine learning: PCA, principal component regression
- "Non-commutative measurements" (for ground state preparation)
- Fractional queries
- ⋮

Quantum algorithms for optimization

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 2015)
  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

- 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)
  Zero-sum games: $\widetilde{O}\big((\sqrt{n}+\sqrt{m})/\varepsilon^3\big)$, $\widetilde{O}\big(s/\varepsilon^{3.5}\big)$ (Apeldoorn & **G** 2019)
- Quantum interior point method (Kerenidis & Prakash 2018)
- Polynomial speed-up for estimating volumes of convex bodies
  (Chakrabarti, Childs, Hung, Li, Wang, Wu 2019)

Continuous optimization

Convex optimization

- 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)
  Zero-sum games: $\widetilde{O}\big((\sqrt{n}+\sqrt{m})/\varepsilon^3\big)$, $\widetilde{O}\big(s/\varepsilon^{3.5}\big)$ (Apeldoorn & **G** 2019)
- Quantum interior point method (Kerenidis & Prakash 2018)
- Polynomial speed-up for estimating volumes of convex bodies
  (Chakrabarti, Childs, Hung, Li, Wang, Wu 2019)

Application of the SDP solver: Shadow tomography (Aaronson 2017)

Given n-dimensional quantum state ρ, estimate the probability of acceptance of each the two-outcome measurements $E_1, \ldots, E_m$, to within additive error ε.

- $$\widetilde{O}\left(\frac{\log^4(m)\log(n)}{\varepsilon^4}\right) \text{ copies suffice}$$

Zero-sum games

Pay-off matrix of Alice is $A \in \mathbb{R}^{m \times n}$. Expected pay-off for strategies x, y: $x^T A y$

Zero-sum games

Pay-off matrix of Alice is $A \in \mathbb{R}^{m \times n}$. Expected pay-off for strategies x, y: $x^T Ay$

Algorithm for approximate Nash-equilibrium (Grigoriadis & Khachiyan 1995)

Start with $x^{(0)} \leftarrow 0 \in \mathbb{R}^n$ and $y^{(0)} \leftarrow 0 \in \mathbb{R}^m$

Zero-sum games

Pay-off matrix of Alice is $A \in \mathbb{R}^{m \times n}$. Expected pay-off for strategies x, y: $x^T A y$

Algorithm for approximate Nash-equilibrium (Grigoriadis & Khachiyan 1995)

Start with $x^{(0)} \leftarrow 0 \in \mathbb{R}^n$ and $y^{(0)} \leftarrow 0 \in \mathbb{R}^m$
for $t = 1, 2, \ldots, \widetilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}\right)$ **do**

- $P^{(t)} \leftarrow e^{-A^T x^{(t)}}$ and $Q^{(t)} \leftarrow e^{A y^{(t)}}$
- $p^{(t)} \leftarrow P^{(t)} / \left\|P^{(t)}\right\|_1$ and $q^{(t)} \leftarrow Q^{(t)} / \left\|Q^{(t)}\right\|_1$
- Sample $a \sim p^{(t)}$ and $b \sim q^{(t)}$
- $y^{(t+1)} = y^{(t)} + \frac{\varepsilon}{4} e_a$ and $x^{(t+1)} = x^{(t)} + \frac{\varepsilon}{4} e_b$

Zero-sum games

Pay-off matrix of Alice is $A \in \mathbb{R}^{m \times n}$. Expected pay-off for strategies x, y: $x^T A y$

Algorithm for approximate Nash-equilibrium (Grigoriadis & Khachiyan 1995)

Start with $x^{(0)} \leftarrow 0 \in \mathbb{R}^n$ and $y^{(0)} \leftarrow 0 \in \mathbb{R}^m$
for $t = 1, 2, \ldots, \widetilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}\right)$ **do**

- $P^{(t)} \leftarrow e^{-A^T x^{(t)}}$ and $Q^{(t)} \leftarrow e^{A y^{(t)}}$
- $p^{(t)} \leftarrow P^{(t)} / \left\|P^{(t)}\right\|_1$ and $q^{(t)} \leftarrow Q^{(t)} / \left\|Q^{(t)}\right\|_1$
- Sample $a \sim p^{(t)}$ and $b \sim q^{(t)}$
- $y^{(t+1)} = y^{(t)} + \frac{\varepsilon}{4} e_a$ and $x^{(t+1)} = x^{(t)} + \frac{\varepsilon}{4} e_b$

The main taks is Gibbs sampling from a linear-combination of vectors.

Zero-sum games

Pay-off matrix of Alice is $A \in \mathbb{R}^{m \times n}$. Expected pay-off for strategies x, y: $x^T A y$

Algorithm for approximate Nash-equilibrium (Grigoriadis & Khachiyan 1995)

Start with $x^{(0)} \leftarrow 0 \in \mathbb{R}^n$ and $y^{(0)} \leftarrow 0 \in \mathbb{R}^m$
for $t = 1, 2, \ldots, \widetilde{O}\left(\frac{1}{\varepsilon^2}\right)$ **do**

- $P^{(t)} \leftarrow e^{-A^T x^{(t)}}$ and $Q^{(t)} \leftarrow e^{A y^{(t)}}$
- $p^{(t)} \leftarrow P^{(t)} / \left\| P^{(t)} \right\|_1$ and $q^{(t)} \leftarrow Q^{(t)} / \left\| Q^{(t)} \right\|_1$
- Sample $a \sim p^{(t)}$ and $b \sim q^{(t)}$
- $y^{(t+1)} = y^{(t)} + \frac{\varepsilon}{4} e_a$ and $x^{(t+1)} = x^{(t)} + \frac{\varepsilon}{4} e_b$

The main taks is Gibbs sampling from a linear-combination of vectors.
For SDPs: we need to Gibbs sample from a linear-combination of matrices.

Statistics, estimation and stochastic algorithms

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

Statistics, estimation and stochastic algorithms

- Quadratic speed-up for Monte-Carlo methods $\mathcal{O}\left(\frac{\sigma}{\varepsilon}\right)$ (Montanaro 2015)
  Generalizes approximate counting (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, estimation and stochastic algorithms

- Quadratic speed-up for Monte-Carlo methods $\mathcal{O}\left(\frac{\sigma}{\varepsilon}\right)$ (Montanaro 2015)
  Generalizes approximate counting (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)

Recommendation systems – Netflix challange

	Inside Out	Good Will Hunting	Mean Girls	Terminator	Titanic	Warrior
Tina Fey	3	1	5	1	?	1
Helen Mirren	2	?	?	2	5	1
Sylvester Stallone	1	3	1	4	2	5
Tom Hanks	?	3	1	?	4	3
George Clooney	2	2	1	3	1	4

Image source: https://towardsdatascience.com

The assumed structure of preference matrix:

Movies: a linear combination of a small number of features
User taste: a linear weighing of the features

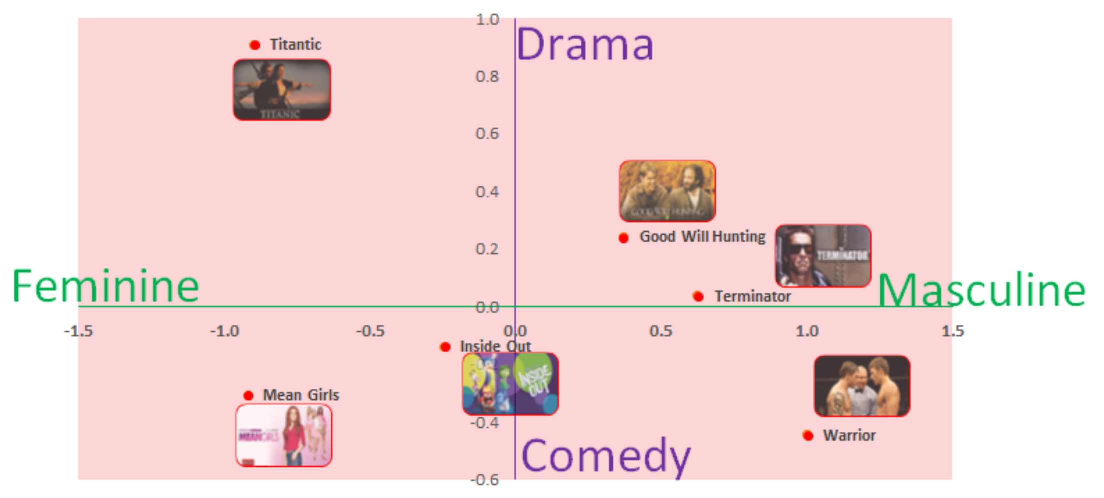


Image source: https://towardsdatascience.com

Major difficulty: how to input the data?

Data conversion: classical to quantum

- Given $b \in \mathbb{R}^m$ prepare

$$|b\rangle = \sum_{i=1}^{m} \frac{b_i}{||b||}$$

Major difficulty: how to input the data?

Data conversion: classical to quantum

- Given $b \in \mathbb{R}^m$ prepare

$$|b\rangle = \sum_{i=1}^{m} \frac{b_i}{||b||}$$

- Given $A \in \mathbb{R}^{m\times n}$ construct quantum circuit (block-encoding)

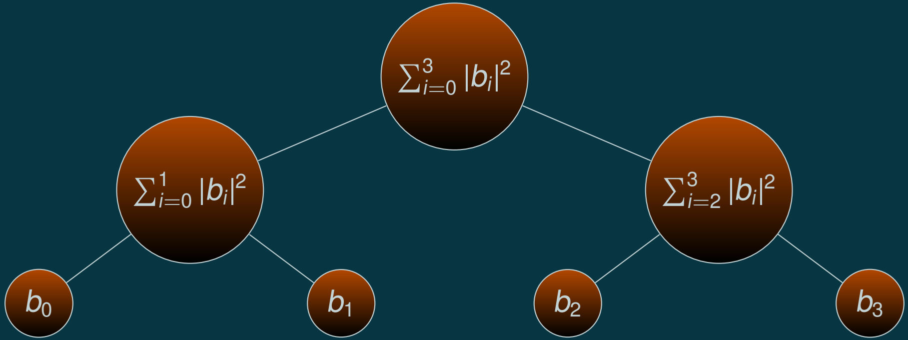
$$U = \begin{pmatrix} A/||A||_F & \cdot \\ \cdot & \cdot \end{pmatrix}.$$

How to preserve the exponential advantage?

Solution: assume QRAM (readable in superposition)

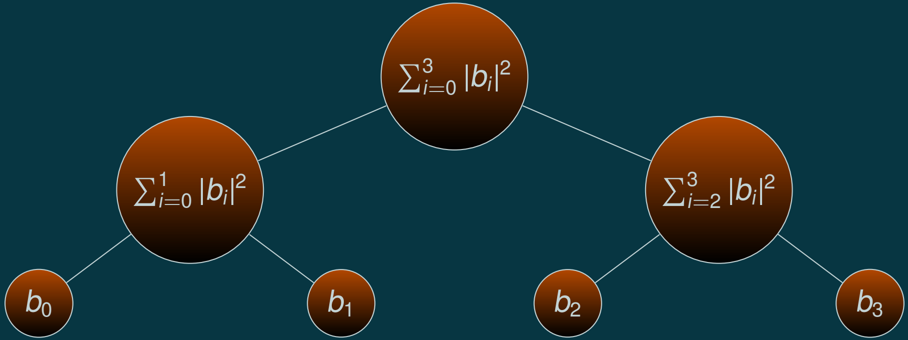
Solution: assume QRAM (readable in superposition)

Data structure



Solution: assume QRAM (readable in superposition)

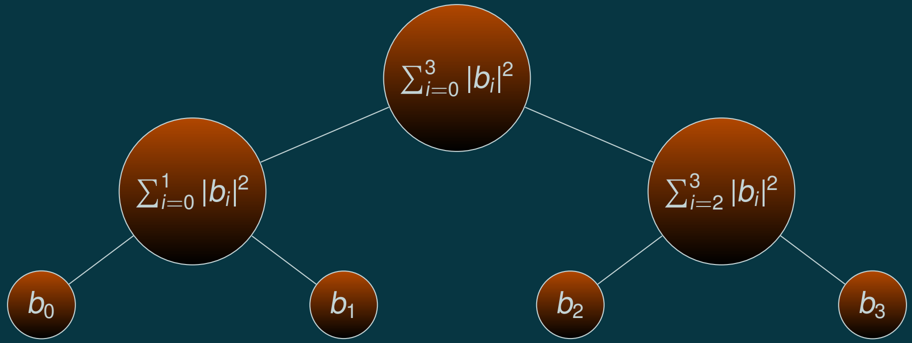
Data structure



Firs prepare: $\sqrt{\sum_{i=0}^{1} |b_i|^2}|0\rangle + \sqrt{\sum_{i=2}^{3} |b_i|^2}|1\rangle$

Solution: assume QRAM (readable in superposition)

Data structure

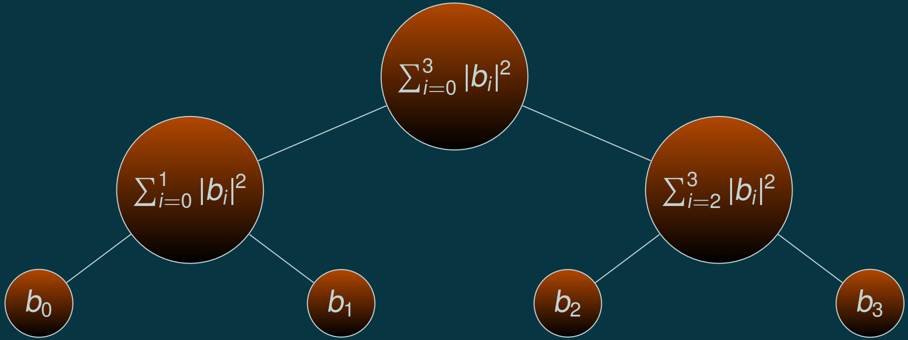


Firs prepare: $\sqrt{\sum_{i=0}^{1} |b_i|^2}|0\rangle + \sqrt{\sum_{i=2}^{3} |b_i|^2}|1\rangle$

Map $\sqrt{\sum_{i=0}^{1} |b_i|^2}|0\rangle \mapsto |b_0||00\rangle + |b_1||01\rangle$ and

Solution: assume QRAM (readable in superposition)

Data structure

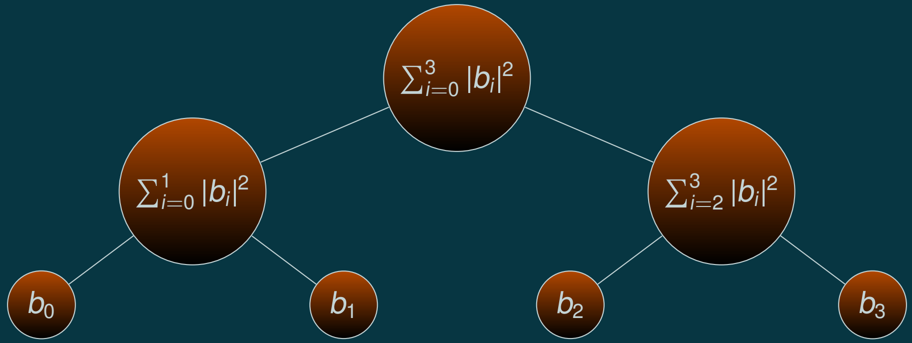


Firs prepare: $\sqrt{\sum_{i=0}^{1} |b_i|^2}|0\rangle + \sqrt{\sum_{i=2}^{3} |b_i|^2}|1\rangle$

Map $\sqrt{\sum_{i=0}^{1} |b_i|^2}|0\rangle \mapsto |b_0||00\rangle + |b_1||01\rangle$ and $\sqrt{\sum_{i=2}^{3} |b_i|^2}|10\rangle \mapsto |b_2||11\rangle + |b_3||01\rangle$

Solution: assume QRAM (readable in superposition)

Data structure



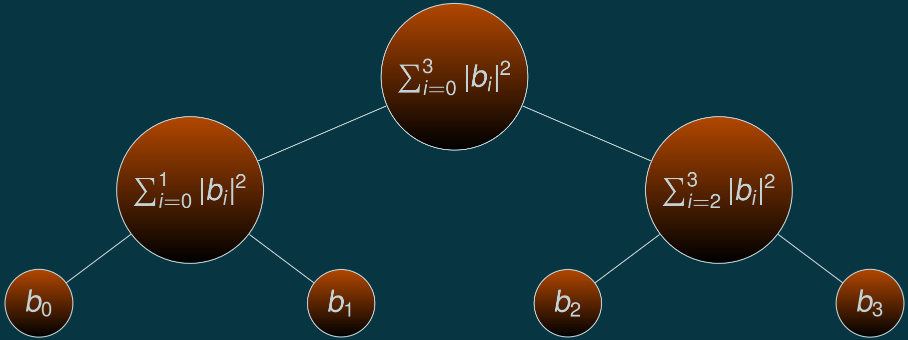
Firs prepare: $\sqrt{\sum_{i=0}^{1} |b_i|^2}|0\rangle + \sqrt{\sum_{i=2}^{3} |b_i|^2}|1\rangle$

Map $\sqrt{\sum_{i=0}^{1} |b_i|^2}|0\rangle \mapsto |b_0||00\rangle + |b_1||01\rangle$ and $\sqrt{\sum_{i=2}^{3} |b_i|^2}|10\rangle \mapsto |b_2||11\rangle + |b_3||01\rangle$

Add phases to get $b_0|00\rangle + b_1|01\rangle + b_2|10\rangle + b_3|11\rangle$

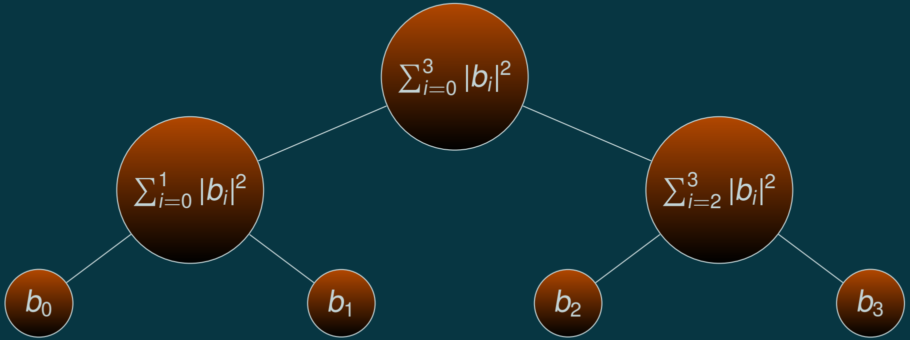
On-line updates to the data structure

Data structure



On-line updates to the data structure

Data structure



Cost is about the depth: $\log(\text{dimension})$

More about this in Ewin Tang's afternoon talk