Quantum algorithms for linear regression

Changpeng Shao

School of Mathematics, University of Bristol, UK
arXiv:2301.06107

Bath Numerical Analysis Seminar
17 February 2023

Background

Quantum computing

A quantum computer is a machine designed to use quantum mechanics to do things.

Quantum computing

A quantum computer is a machine designed to use quantum mechanics to do things.

Quantum computers are believed to be able to quickly solve certain problems that no classical computer could solve in any feasible amount of time.

Quantum computing

A quantum computer is a machine designed to use quantum mechanics to do things.

Quantum computers are believed to be able to quickly solve certain problems that no classical computer could solve in any feasible amount of time.
e.g., integral factorization, simulation of quantum systems, etc.

Quantum computing

A quantum computer is a machine designed to use quantum mechanics to do things.

Quantum computers are believed to be able to quickly solve certain problems that no classical computer could solve in any feasible amount of time.
e.g., integral factorization, simulation of quantum systems, etc.

Solving these problems are based on quantum algorithms.

Quantum linear algebra

One plausible problem domain where quantum computers could be applied is linear algebra.

[1] Kitaev '95
[2] Harrow, Hassidim, Lloyd '08
[3] Gilyén et al '18
[4] Lloyd et al '20

Quantum linear algebra

One plausible problem domain where quantum computers could be applied is linear algebra. For example

- Computing eigenvalues [1]
- Solving linear systems [2]
- Singular value decomposition [3]
- Polar decomposition [4]
- ......

[1] Kitaev '95
[2] Harrow, Hassidim, Lloyd '08
[3] Gilyén et al '18
[4] Lloyd et al '20

Quantum linear algebra

One plausible problem domain where quantum computers could be applied is linear algebra. For example

- Computing eigenvalues [1]
- Solving linear systems [2]
- Singular value decomposition [3]
- Polar decomposition [4]
- ......

In this talk, I will mainly focus on the linear system problem.

[1] Kitaev '95
[2] Harrow, Hassidim, Lloyd '08
[3] Gilyén et al '18
[4] Lloyd et al '20

Quantum state: a basic object of quantum computing

Quantum state: $|\psi\rangle = \sum_{i=0}^{n-1} a_i |i\rangle$, where $\sum_i |a_i|^2 = 1$. Each $|i\rangle$ stands for a classical state, and $|\psi\rangle$ is a superposition of these states. Each state has a certain amplitude.

Quantum state: a basic object of quantum computing

Quantum state: $|\psi\rangle = \sum_{i=0}^{n-1} a_i|i\rangle$, where $\sum_i |a_i|^2 = 1$. Each $|i\rangle$ stands for a classical state, and $|\psi\rangle$ is a superposition of these states. Each state has a certain amplitude.

Mathematically, $|\psi\rangle$ can be viewed as a vector

$$\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}.$$

Quantum state: a basic object of quantum computing

Quantum state: $|\psi\rangle = \sum_{i=0}^{n-1} a_i|i\rangle$, where $\sum_i |a_i|^2 = 1$. Each $|i\rangle$ stands for a classical state, and $|\psi\rangle$ is a superposition of these states. Each state has a certain amplitude.

Mathematically, $|\psi\rangle$ can be viewed as a vector

$$\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}.$$

There are two things we can do with a quantum state:

Quantum state: a basic object of quantum computing

Quantum state: $|\psi\rangle = \sum_{i=0}^{n-1} a_i |i\rangle$, where $\sum_i |a_i|^2 = 1$. Each $|i\rangle$ stands for a classical state, and $|\psi\rangle$ is a superposition of these states. Each state has a certain amplitude.

Mathematically, $|\psi\rangle$ can be viewed as a vector

$$\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}.$$

There are two things we can do with a quantum state:

- measure it: see $|i\rangle$ with probability $|a_i|^2 \rightarrow$ sampling. This is one way that we get useful information from $|\psi\rangle$.

Quantum state: a basic object of quantum computing

Quantum state: $|\psi\rangle = \sum_{i=0}^{n-1} a_i|i\rangle$, where $\sum_i |a_i|^2 = 1$. Each $|i\rangle$ stands for a classical state, and $|\psi\rangle$ is a superposition of these states. Each state has a certain amplitude.

Mathematically, $|\psi\rangle$ can be viewed as a vector

$$\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix}.$$

There are two things we can do with a quantum state:

- measure it: see $|i\rangle$ with probability $|a_i|^2 \to$ sampling. This is one way that we get useful information from $|\psi\rangle$.
- let it evolve unitarily without measuring it: $U|\psi\rangle$, where U is unitary.

Solving linear systems

Solving linear systems on a classical/quantum computer

Definition (Classical case)

Given access to a matrix $A \in \mathbb{R}^{n\times d}$*, and a vector* $\mathbf{b} \in \mathbb{R}^n$*, output* $A^+\mathbf{b}$ *up to error* ε*, where* A^+ *is the pseudoinverse.*

Solving linear systems on a classical/quantum computer

Definition (Classical case)

Given access to a matrix $A \in \mathbb{R}^{n\times d}$, and a vector $\mathbf{b} \in \mathbb{R}^n$, output $A^+\mathbf{b}$ up to error ε, where A^+ is the pseudoinverse.

One "quantum" way of thinking about this problem:

Definition (Quantum case)

Given the same access to a matrix $A \in \mathbb{R}^{n\times d}$, and the ability to prepare $|\mathbf{b}\rangle$, output $|A^+\mathbf{b}\rangle$ up to error ε.

Solving linear systems on a classical/quantum computer

Definition (Classical case)

Given access to a matrix $A \in \mathbb{R}^{n \times d}$, and a vector $\mathbf{b} \in \mathbb{R}^n$, output $A^+\mathbf{b}$ up to error ε, where A^+ is the pseudoinverse.

One "quantum" way of thinking about this problem:

Definition (Quantum case)

Given the same access to a matrix $A \in \mathbb{R}^{n \times d}$, and the ability to prepare $|\mathbf{b}\rangle$, output $|A^+\mathbf{b}\rangle$ up to error ε.

Theorem (Harrow, Hassidim, Lloyd 2008 + many others)

There is a quantum algorithm that solves this problem in cost

$$\kappa \cdot \mathrm{poly}(\log(nd), \log(1/\varepsilon)),$$

where κ is the condition number.

A comparison

Classical	Quantum
$\kappa nd \log(1/\varepsilon)$	$\kappa \cdot \mathrm{poly}(\log(nd), \log(1/\varepsilon))$

A comparison

Classical	Quantum
$\kappa nd \log(1/\varepsilon)$	$\kappa \cdot \mathrm{poly}(\log(nd), \log(1/\varepsilon))$

So, quantum computers are exponentially faster!?

A comparison

Classical	Quantum
$\kappa n d \log(1/\varepsilon)$	$\kappa \cdot \mathrm{poly}(\log(nd), \log(1/\varepsilon))$

So, quantum computers are exponentially faster!?

This comparison is not fair since the outputs are different.

A comparison

Classical	Quantum
$\kappa nd \log(1/\varepsilon)$	$\kappa \cdot \mathrm{poly}(\log(nd), \log(1/\varepsilon))$

So, quantum computers are exponentially faster!?

This comparison is not fair since the outputs are different.

To allow for a fair comparison, we need the quantum and classical computers to have similar outputs.

A comparison

Classical	Quantum
$\kappa nd \log(1/\varepsilon)$	$\kappa \cdot \mathrm{poly}(\log(nd), \log(1/\varepsilon))$

So, quantum computers are exponentially faster!?

This comparison is not fair since the outputs are different.

To allow for a fair comparison, we need the quantum and classical computers to have similar outputs. So we can either

- require a classical computer to output an analogue of a quantum state (quantum-inspired classical algorithms), or

A comparison

Classical	Quantum
$\kappa nd \log(1/\varepsilon)$	$\kappa \cdot \mathrm{poly}(\log(nd), \log(1/\varepsilon))$

So, quantum computers are exponentially faster!?

This comparison is not fair since the outputs are different.

To allow for a fair comparison, we need the quantum and classical computers to have similar outputs. So we can either

- require a classical computer to output an analogue of a quantum state (quantum-inspired classical algorithms), or
- require a quantum computer to output a vector solution

Main focus of this talk

Design a new quantum algorithm that can output vector solutions.

[5] Drineas+Mahoney '10; Mahoney '11; Drineas et al. '12

Main focus of this talk

Design a new quantum algorithm that can output vector solutions.

The quantum speedup is limited, however, these algorithms are useful for practical applications.

[5] Drineas+Mahoney '10; Mahoney '11; Drineas et al. '12

Main focus of this talk

Design a new quantum algorithm that can output vector solutions.

The quantum speedup is limited, however, these algorithms are useful for practical applications.

Remark (Not discussed in this talk)

For this problem, a naive quantum algorithm is

1. *apply a quantum linear solver to obtain* $|A^{+}\mathbf{b}\rangle$.
2. *apply quantum tomography to obtain* $A^{+}\mathbf{b}$.

[5] Drineas+Mahoney '10; Mahoney '11; Drineas et al. '12

Main focus of this talk

Design a new quantum algorithm that can output vector solutions.

The quantum speedup is limited, however, these algorithms are useful for practical applications.

Remark (Not discussed in this talk)

For this problem, a naive quantum algorithm is

1. *apply a quantum linear solver to obtain* $|A^{+}\mathbf{b}\rangle$.
2. *apply quantum tomography to obtain* $A^{+}\mathbf{b}$.

Main technique: a quantum speedup of leverage score sampling,

[5] Drineas+Mahoney '10; Mahoney '11; Drineas et al. '12

Main focus of this talk

Design a new quantum algorithm that can output vector solutions.

The quantum speedup is limited, however, these algorithms are useful for practical applications.

Remark (Not discussed in this talk)

For this problem, a naive quantum algorithm is

1. *apply a quantum linear solver to obtain* $|A^+\mathbf{b}\rangle$.
2. *apply quantum tomography to obtain* $A^+\mathbf{b}$.

Main technique: a quantum speedup of leverage score sampling, an important sampling and sketching tool in randomised numerical linear algebra.[5]

[5] Drineas+Mahoney '10; Mahoney '11; Drineas et al '12

Sampling and Sketching Method

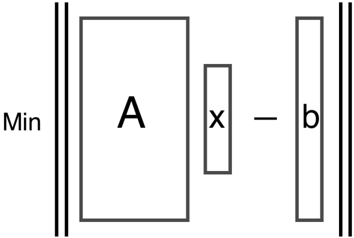
Sampling and Sketching

Main goal: reduce a large-scale problem to a small-scale problem, from which we can recover the solution to the original problem.

Sampling and Sketching

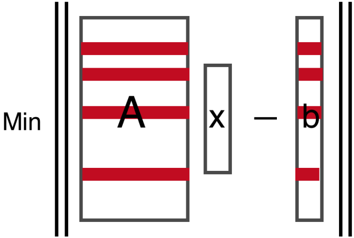
Main goal: reduce a large-scale problem to a small-scale problem, from which we can recover the solution to the original problem.

We are given a linear regression



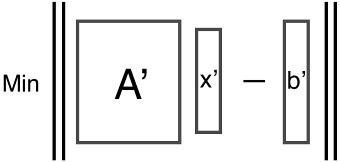
Sampling and Sketching

First, we sample some rows of $A, \mathbf{b}$ according to some distribution.



Sampling and Sketching

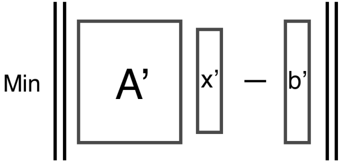
Then, we solve a reduced linear regression



[6] Clarkson+Woodruff '17

Sampling and Sketching

Then, we solve a reduced linear regression



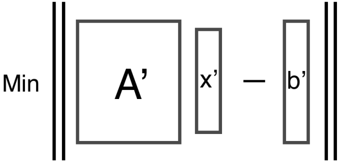
- Let $\mathbf{x}'_{\mathrm{opt}}$ be the optimal solution of the reduced problem and $\mathbf{x}_{\mathrm{opt}}$ be the optimal solution of the original problem. Then under certain conditions,

$$\|A\mathbf{x}'_{\mathrm{opt}} - \mathbf{b}\| \leq (1+\varepsilon)\|A\mathbf{x}_{\mathrm{opt}} - \mathbf{b}\|.$$

[6] Clarkson+Woodruff ’17

Sampling and Sketching

Then, we solve a reduced linear regression



- Let $\mathbf{x}'_{\text{opt}}$ be the optimal solution of the reduced problem and $\mathbf{x}_{\text{opt}}$ be the optimal solution of the original problem. Then under certain conditions,

$$\|A\mathbf{x}'_{\text{opt}} - \mathbf{b}\| \leq (1+\varepsilon)\|A\mathbf{x}_{\text{opt}} - \mathbf{b}\|.$$

- The complexity can be $O(\text{nnz}(A))$, the number of nonzero entries of A.[6]

[6] Clarkson+Woodruff '17

Leverage score sampling

Question: How to do the sampling?

Leverage score sampling

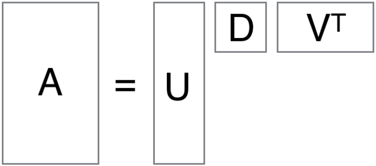
Question: How to do the sampling?
A popular one is leverage score sampling.

Leverage score sampling

Question: How to do the sampling?
A popular one is leverage score sampling.

Definition (Leverage scores)

Assume $A \in \mathbb{R}^{n \times d}$ *has rank* r *and SVD*



Then the i*-th row leverage score of* A *is*

$$\mathcal{L}_i = \|U_i\|^2,$$

where U_i *is the* i*-th row of* U*.*

Some properties

1. For $\arg\min \|A\mathbf{x} - \mathbf{b}\|$, a solution is $\mathbf{x}_{\text{opt}} = (A^T A)^{-1} A^T \mathbf{b}$. Thus $\mathbf{b} \approx A\mathbf{x}_{\text{opt}} = H\mathbf{b}$, where $H = A(A^T A)^{-1} A^T$ is the so-called hat matrix. Then

$$H_{ii} = \mathcal{L}_i.$$

Some properties

1. For $\arg\min \|A\mathbf{x} - \mathbf{b}\|$, a solution is $\mathbf{x}_{\text{opt}} = (A^T A)^{-1} A^T \mathbf{b}$. Thus $\mathbf{b} \approx A\mathbf{x}_{\text{opt}} = H\mathbf{b}$, where $H = A(A^T A)^{-1} A^T$ is the so-called hat matrix. Then

$$H_{ii} = \mathcal{L}_i.$$

2. $\sum_{i=1}^{n} \mathcal{L}_i = r$, so it defines a probability distribution

$$\{p_i = \mathcal{L}_i / r : \quad i = 1, \ldots, n\}.$$

Computing all leverage scores: randomised algorithms

In [Drineas et al: arXiv:1109.3843], a randomised algorithm was proposed to approximate all leverage scores up to relative error ε in cost

$$\widetilde{O}(nd/\varepsilon^2 + d^3/\varepsilon^2).$$

Computing all leverage scores: randomised algorithms

In [Drineas et al: arXiv:1109.3843], a randomised algorithm was proposed to approximate all leverage scores up to relative error ε in cost

$$\widetilde{O}(nd/\varepsilon^2 + d^3/\varepsilon^2).$$

This was later improved to

$$\widetilde{O}(\mathrm{nnz}(A) + \mathrm{poly}(r/\varepsilon))$$

by Clarkson and Woodruff [arXiv:1207.6365], where

- $\mathrm{nnz}(A)$ = number of nonzero entries of A.
- r = rank of A.

Quantum Speedup of Leverage Score Sampling

Quantum speedup of leverage score sampling

In the quantum case, one way to use leverage score sampling is by preparing

$$|\mathcal{L}_A\rangle := \frac{1}{\sqrt{r}} \sum_{i=1}^{n} \sqrt{\mathcal{L}_i}\, |i\rangle.$$

Quantum speedup of leverage score sampling

In the quantum case, one way to use leverage score sampling is by preparing

$$|\mathcal{L}_A\rangle := \frac{1}{\sqrt{r}} \sum_{i=1}^{n} \sqrt{\mathcal{L}_i}\, |i\rangle.$$

- Measuring this state $\Leftrightarrow$ Sampling from $\{p_i = \mathcal{L}_i / r\}_{i=1}^{n}$.

Quantum speedup of leverage score sampling

In the quantum case, one way to use leverage score sampling is by preparing

$$|\mathcal{L}_A\rangle := \frac{1}{\sqrt{r}} \sum_{i=1}^{n} \sqrt{\mathcal{L}_i}\,|i\rangle.$$

- Measuring this state $\Leftrightarrow$ Sampling from $\{p_i = \mathcal{L}_i/r\}_{i=1}^n$.
- Approximate $\mathcal{L}_i$ using amplitude estimation technique.

Quantum speedup of leverage score sampling

In the quantum case, one way to use leverage score sampling is by preparing

$$|\mathcal{L}_A\rangle := \frac{1}{\sqrt{r}} \sum_{i=1}^{n} \sqrt{\mathcal{L}_i}\,|i\rangle.$$

- Measuring this state $\Leftrightarrow$ Sampling from $\{p_i = \mathcal{L}_i/r\}_{i=1}^{n}$.
- Approximate $\mathcal{L}_i$ using amplitude estimation technique.
- We can prepare $|\mathcal{L}_A\rangle$ without computing any $\mathcal{L}_i$.

Quantum speedup of leverage score sampling

Theorem

Assume that A has rank r, and U is an α-block-encoding of A that is implemented in time $O(T)$. Then there is a quantum algorithm that prepares

$$|\mathcal{L}_A\rangle = \frac{1}{\sqrt{r}} \sum_{i=1}^{n} \sqrt{\mathcal{L}_i}\, |i\rangle$$

in time

$$\widetilde{O}\left(\mathcal{K}\sqrt{\min(n,d)/r}\right),$$

where $\mathcal{K} = T\alpha/\sigma_{\min}$ and $\sigma_{\min}$ is a lower bound of the minimal nonzero singular value of A.

Quantum speedup of leverage score sampling

Theorem

Assume that A has rank r, and U is an α-block-encoding of A that is implemented in time $O(T)$. Then there is a quantum algorithm that prepares

$$|\mathcal{L}_A\rangle = \frac{1}{\sqrt{r}} \sum_{i=1}^{n} \sqrt{\mathcal{L}_i}\, |i\rangle$$

in time

$$\widetilde{O}\left(\mathcal{K}\sqrt{\min(n,d)/r}\right),$$

where $\mathcal{K} = T\alpha/\sigma_{\min}$ and $\sigma_{\min}$ is a lower bound of the minimal nonzero singular value of A.

For example

- if A is s-sparse, then $\mathcal{K} = \widetilde{O}(s\|A\|_{\max}/\sigma_{\min})$.

Quantum speedup of leverage score sampling

Theorem

Assume that A has rank r, and U is an α-block-encoding of A that is implemented in time $O(T)$. Then there is a quantum algorithm that prepares

$$|\mathcal{L}_A\rangle = \frac{1}{\sqrt{r}} \sum_{i=1}^{n} \sqrt{\mathcal{L}_i}\, |i\rangle$$

in time

$$\widetilde{O}\left(\mathcal{K}\sqrt{\min(n,d)/r}\right),$$

where $\mathcal{K} = T\alpha/\sigma_{\min}$ and $\sigma_{\min}$ is a lower bound of the minimal nonzero singular value of A.

For example

- if A is s-sparse, then $\mathcal{K} = \widetilde{O}(s\|A\|_{\max}/\sigma_{\min})$.
- if A is dense, then $\mathcal{K} = \widetilde{O}(\sqrt{nd}\|A\|_{\max}/\sigma_{\min})$.

Quantum speedup of leverage score sampling

Theorem

Make the same assumptions as above, then there is a quantum algorithm that for each j returns $\widetilde{\mathcal{L}}_j$ in time

$$\widetilde{O}(\mathcal{K}/\varepsilon)$$

such that

$$\left|\widetilde{\mathcal{L}}_j - \mathcal{L}_j\right| \leq \varepsilon\sqrt{\mathcal{L}_j}.$$

Proof sketch: block-encoding

When solving matrix-related problems, we need a way to encode the data into a quantum computer. Block-encoding is one efficient way to do this.[7]

[7] Chakraborty, Gilyén, Jeffery '18

Proof sketch: block-encoding

When solving matrix-related problems, we need a way to encode the data into a quantum computer. Block-encoding is one efficient way to do this.[7]

Definition (α-block-encoding)

Let $A \in \mathbb{R}^{n\times d}$ be a matrix, $\alpha \in \mathbb{R}^+$. An α-block-encoding of A is a unitary matrix of the form $U = \begin{pmatrix} A/\alpha & * \\ * & * \end{pmatrix}$.

[7] Chakraborty, Gilyén, Jeffery '18

Proof sketch: block-encoding

When solving matrix-related problems, we need a way to encode the data into a quantum computer. Block-encoding is one efficient way to do this.[7]

Definition (α-block-encoding)

Let $A \in \mathbb{R}^{n\times d}$ be a matrix, $\alpha \in \mathbb{R}^+$. An α-block-encoding of A is a unitary matrix of the form $U = \begin{pmatrix} A/\alpha & * \\ * & * \end{pmatrix}$.

In the quantum case, we are more interested in U which has an efficient circuit implementation: $\exists$ a decomposition into a product of $T = \mathrm{polylog}(nd)$ simple unitaries.

[7] Chakraborty, Gilyén, Jeffery '18

Proof sketch: block-encoding

When solving matrix-related problems, we need a way to encode the data into a quantum computer. Block-encoding is one efficient way to do this.[7]

Definition (α-block-encoding)

Let $A \in \mathbb{R}^{n \times d}$ be a matrix, $\alpha \in \mathbb{R}^+$. An α-block-encoding of A is a unitary matrix of the form $U = \begin{pmatrix} A/\alpha & * \\ * & * \end{pmatrix}$.

In the quantum case, we are more interested in U which has an efficient circuit implementation: $\exists$ a decomposition into a product of $T = \mathrm{polylog}(nd)$ simple unitaries.

This is not always possible. It is possible for some structured matrices, e.g., sparse matrices.

[7] Chakraborty, Gilyén, Jeffery '18

Proof sketch: quantum singular value transform[8]

- Given $U = \begin{pmatrix} A/\alpha & * \\ * & * \end{pmatrix}$.

[8] Gilyén, Su, Low, Wiebe, Quantum singular value transformation, STOC'19

Proof sketch: quantum singular value transform[8]

- Given $U = \begin{pmatrix} A/\alpha & * \\ * & * \end{pmatrix}$.
- Given a bounded polynomial f of degree d.

[8] Gilyén, Su, Low, Wiebe, Quantum singular value transformation, STOC'19

Proof sketch: quantum singular value transform[8]

- Given $U = \begin{pmatrix} A/\alpha & * \\ * & * \end{pmatrix}$.
- Given a bounded polynomial f of degree d.
- Then there is a quantum circuit that implements the unitary

$$\widetilde{U} = \begin{pmatrix} f^{(\mathsf{SV})}(A/\alpha) & * \\ * & * \end{pmatrix},$$

where

$$f^{(\mathsf{SV})}(UDV^T) = Uf(D)V^T.$$

[8] Gilyén, Su, Low, Wiebe, Quantum singular value transformation, STOC'19

Proof sketch: quantum singular value transform[8]

- Given $U = \begin{pmatrix} A/\alpha & * \\ * & * \end{pmatrix}$.
- Given a bounded polynomial f of degree d.
- Then there is a quantum circuit that implements the unitary

$$\widetilde{U} = \begin{pmatrix} f^{(\mathsf{SV})}(A/\alpha) & * \\ * & * \end{pmatrix},$$

  where

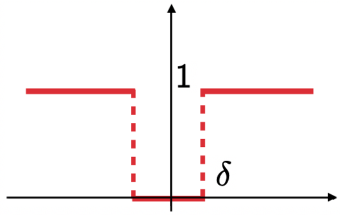
$$f^{(\mathsf{SV})}(UDV^T) = Uf(D)V^T.$$

- It uses $O(d)$ applications of U, U^T.

[8]Gilyén, Su, Low, Wiebe, Quantum singular value transformation, STOC'19

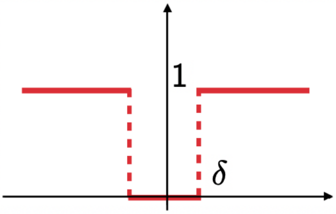
Proof sketch: An example

Suppose f is close to the rectangular function



Proof sketch: An example

Suppose f is close to the rectangular function



Suppose $\delta < \sigma_{\min}$, then

$$f^{(\mathsf{SV})}(UDV^T) \approx UV^T.$$

This is useful in polar decomposition.

Proof sketch: continued

On a quantum computer, by QSVT we obtain a unitary

$$W := \begin{pmatrix} VU^T & * \\ * & * \end{pmatrix}.$$

Proof sketch: continued

On a quantum computer, by QSVT we obtain a unitary

$$W := \begin{pmatrix} VU^T & * \\ * & * \end{pmatrix}.$$

Now consider $(I \otimes W)|\phi\rangle$, where (recall $\mathbf{e}_i = |i\rangle$)

$$|\phi\rangle = \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle \otimes \boxed{|0\rangle \otimes |i\rangle} \qquad \begin{pmatrix} |i\rangle \\ 0 \end{pmatrix}$$

Then we obtain

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle \otimes VU^T|i\rangle + |\perp\rangle.$$

Proof sketch: continued

On a quantum computer, by QSVT we obtain a unitary

$$W := \begin{pmatrix} VU^T & * \\ * & * \end{pmatrix}.$$

Now consider $(I \otimes W)|\phi\rangle$, where (recall $\mathbf{e}_i = |i\rangle$)

$$|\phi\rangle = \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle \otimes \boxed{|0\rangle \otimes |i\rangle} \rightarrow \begin{pmatrix} |i\rangle \\ 0 \end{pmatrix}$$

Then we obtain

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} |i\rangle \otimes VU^T|i\rangle + |\perp\rangle.$$

Since $\mathcal{L}_i = \|VU^T|i\rangle\|^2$, the first part is “equivalent” to

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} \sqrt{\mathcal{L}_i}\, |i\rangle$$

if we only care about sampling.

An application to solve linear regressions

Solving linear regressions: classical

A randomised algorithm for solving linear regressions.[9]

[9] Drinea et al: 0710.1435

[10] Clarkson, Woodruff: 1207.6365

Solving linear regressions: classical

A randomised algorithm for solving linear regressions.[9]

Main idea: solve a reduced problem $\arg\min_{\mathbf{x}} \|SA\mathbf{x} - S\mathbf{b}\|$ instead.

[9] Drinea et al: 0710.1435

[10] Clarkson, Woodruff: 1207.6365

Solving linear regressions: classical

A randomised algorithm for solving linear regressions.[9]

Main idea: solve a reduced problem $\arg\min_{\mathbf{x}} \|SA\mathbf{x} - S\mathbf{b}\|$ instead.

Question: How to find a good sampling matrix S?

[9] Drinea et al: 0710.1435

[10] Clarkson, Woodruff: 1207.6365

Solving linear regressions: classical

A randomised algorithm for solving linear regressions.[9]

Main idea: solve a reduced problem $\arg\min_{\mathbf{x}} \|SA\mathbf{x} - S\mathbf{b}\|$ instead.

Question: How to find a good sampling matrix S?

1. Initialize $S = 0 \in \mathbb{R}^{q\times n}$, where $q = O(r/\varepsilon)$, $r = \mathrm{rank}(A)$.

[9] Drinea et al: 0710.1435

[10] Clarkson, Woodruff: 1207.6365

Solving linear regressions: classical

A randomised algorithm for solving linear regressions.[9]

Main idea: solve a reduced problem $\arg\min_{\mathbf{x}} \|SA\mathbf{x} - S\mathbf{b}\|$ instead.

Question: How to find a good sampling matrix S?

1. Initialize $S = 0 \in \mathbb{R}^{q\times n}$, where $q = O(r/\varepsilon)$, $r = \mathrm{rank}(A)$.
2. **Approximation:**
   Approximate all the row leverage-scores $\{\mathcal{L}_i : i \in [n]\}$ of A.

[9] Drinea et al: 0710.1435
[10] Clarkson, Woodruff: 1207.6365

Solving linear regressions: classical

A randomised algorithm for solving linear regressions.[9]

Main idea: solve a reduced problem $\arg\min_{\mathbf{x}} \|SA\mathbf{x} - S\mathbf{b}\|$ instead.

Question: How to find a good sampling matrix S?

1. Initialize $S = 0 \in \mathbb{R}^{q\times n}$, where $q = O(r/\varepsilon)$, $r = \mathrm{rank}(A)$.
2. **Approximation:**
   Approximate all the row leverage-scores $\{\mathcal{L}_i : i \in [n]\}$ of A.
3. **Sampling:**
   Sampling according to $\{\mathcal{L}_i/r : i \in [n]\}$, if i is sampled, then set the i-th row of S as $\mathbf{e}_i^T/\sqrt{qp_i}$, where $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ is the standard basis of $\mathbb{R}^n$.

[9] Drinea et al: 0710.1435

[10] Clarkson, Woodruff: 1207.6365

Solving linear regressions: classical

A randomised algorithm for solving linear regressions.[9]

Main idea: solve a reduced problem $\arg\min_{\mathbf{x}} \|SA\mathbf{x} - S\mathbf{b}\|$ instead.

Question: How to find a good sampling matrix S?

1. Initialize $S = 0 \in \mathbb{R}^{q\times n}$, where $q = O(r/\varepsilon)$, $r = \mathrm{rank}(A)$.
2. **Approximation:**
   Approximate all the row leverage-scores $\{\mathcal{L}_i : i \in [n]\}$ of A.
3. **Sampling:**
   Sampling according to $\{\mathcal{L}_i/r : i \in [n]\}$, if i is sampled, then set the i-th row of S as $\mathbf{e}_i^T/\sqrt{qp_i}$, where $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ is the standard basis of $\mathbb{R}^n$.

The main cost comes from the second step of computing all leverage scores, which is $\widetilde{O}(\mathrm{nnz}(A) + r^3)$.[10]

[9] Drinea et al: 0710.1435

[10] Clarkson, Woodruff: 1207.6365

Solving linear regressions: quantum

The quantum speedup:

Solving linear regressions: quantum

The quantum speedup:

1. Initialize $S = 0 \in \mathbb{R}^{q \times n}$, where $q = O(r/\varepsilon)$, $r = \mathrm{rank}(A)$.

Solving linear regressions: quantum

The quantum speedup:

1. Initialize $S = 0 \in \mathbb{R}^{q \times n}$, where $q = O(r/\varepsilon)$, $r = \mathrm{rank}(A)$.
2. **Sampling:**
   Generate the state $|\mathcal{L}_A\rangle$ of A. Perform q measurements to $|\mathcal{L}_A\rangle$, denote the results as $J \subseteq [n]$.

Solving linear regressions: quantum

The quantum speedup:

1. Initialize $S = 0 \in \mathbb{R}^{q\times n}$, where $q = O(r/\varepsilon)$, $r = \mathrm{rank}(A)$.
2. **Sampling:**
   Generate the state $|\mathcal{L}_A\rangle$ of A. Perform q measurements to $|\mathcal{L}_A\rangle$, denote the results as $J \subseteq [n]$.
3. **Approximation:**
   Approximate $p_j = \mathcal{L}_j/r$ for all $j \in J$ up to a small constant relative error. Denote the results as $\tilde{p}_j$, set the j-th row of S as $\mathbf{e}_j^T/\sqrt{q\tilde{p}_j}$.

Comparison

	Complexity (of finding S)
Classical	$\widetilde{O}(\mathsf{nnz}(A) + r^3)$
Quantum	$\widetilde{O}\left(\mathcal{K} r \sqrt{n}/\varepsilon^{1.5}\right)$

Comparison

	Complexity (of finding S)
Classical	$\widetilde{O}(\mathsf{nnz}(A) + r^3)$
Quantum	$\widetilde{O}\left(\mathcal{K} r \sqrt{n}/\varepsilon^{1.5}\right)$

When A is dense and low-rank, then in the worst case

$$\mathsf{nnz}(A) = nd, \quad \mathcal{K}\sqrt{n} = n\sqrt{d}\,\|A\|_{\max}/\sigma_{\min}.$$

Comparison

	Complexity (of finding S)
Classical	$\widetilde{O}(\mathsf{nnz}(A) + r^3)$
Quantum	$\widetilde{O}\left(\mathcal{K} r \sqrt{n}/\varepsilon^{1.5}\right)$

When A is dense and low-rank, then in the worst case

$$\mathsf{nnz}(A) = nd, \quad \mathcal{K}\sqrt{n} = n\sqrt{d}\,\|A\|_{\max}/\sigma_{\min}.$$

When A is sparse and low-rank, then in the worst case

$$\mathsf{nnz}(A) = n + d, \quad \mathcal{K}\sqrt{n} = s\sqrt{n}\,\|A\|_{\max}/\sigma_{\min}.$$

Summary

- Classically, to perform leverage score sampling, we first do approximation, then do sampling.

Summary

- Classically, to perform leverage score sampling, we first do approximation, then do sampling.
- Quantumly, to perform leverage score sampling, we first do sampling, then do approximation.

Summary

- Classically, to perform leverage score sampling, we first do approximation, then do sampling.
- Quantumly, to perform leverage score sampling, we first do sampling, then do approximation.
- Comparison of the complexity for leverage score sampling: $A \in \mathbb{R}^{n\times d}, r = \text{rank}(A)$.

	Upper bound	Lower bound
Classical	$O(nd)$	$\Omega(n+d)$
Quantum	$\widetilde{O}(\mathcal{K}\sqrt{\min(n,d)/r})$	$\Omega(\mathcal{K}+\sqrt{\min(n,d)/r})$

Summary

- Classically, to perform leverage score sampling, we first do approximation, then do sampling.
- Quantumly, to perform leverage score sampling, we first do sampling, then do approximation.
- Comparison of the complexity for leverage score sampling: $A \in \mathbb{R}^{n \times d}, r = \text{rank}(A)$.

	Upper bound	Lower bound
Classical	$O(nd)$	$\Omega(n+d)$
Quantum	$\widetilde{O}(\mathcal{K}\sqrt{\min(n,d)/r})$	$\Omega(\mathcal{K} + \sqrt{\min(n,d)/r})$

$\Rightarrow$ Quadratic speedup, may have superpolynomial speedup when $\mathcal{K} \ll n + d$.

Outlook

Explore other applications, such as quantum speedups of CX and CUR decomposition.[11]

[11] Drineas, Mahoney, Muthukrishnan: arXiv:0708.3696

Outlook

Explore other applications, such as quantum speedups of CX and CUR decomposition.[11]

It would be interesting to find applications such that $\mathcal{K}$ is small.

[11] Drineas, Mahoney, Muthukrishnan: arXiv:0708.3696

Outlook

Explore other applications, such as quantum speedups of CX and CUR decomposition.[11]

It would be interesting to find applications such that $\mathcal{K}$ is small.

Thanks very much for your attention!

[11] Drineas, Mahoney, Muthukrishnan: arXiv:0708.3696