Provable Quantum Speed-ups for Optimization and Machine Learning

Xiaodi Wu

Based on (1) arXiv:1809.01731v1 (QIP 2019);
(2) arXiv:1710.02581v2 (QIP 2019);
(3) arXiv:1904.02276v1 (ICML 2019).

UNIVERSITY OF MARYLAND

JOINT CENTER FOR QUANTUM INFORMATION AND COMPUTER SCIENCE

Outline

Interplay: Quantum, Optimization & Machine Learning

- Optimization and Machine Learning – two sides of the same coin!

Interplay: Quantum, Optimization & Machine Learning

- Optimization and Machine Learning – two sides of the same coin!
- Many ways to bring *quantum* into this picture!

Interplay: Quantum, Optimization & Machine Learning

- Optimization and Machine Learning – two sides of the same coin!
- Many ways to bring *quantum* into this picture!

Interplay: Quantum, Optimization & Machine Learning

- Optimization and Machine Learning – two sides of the same coin!
- Many ways to bring *quantum* into this picture!

Quantum Speed-up for Optimization and Machine Learning

- one of important quantum applications.

Interplay: Quantum, Optimization & Machine Learning

- Optimization and Machine Learning – two sides of the same coin!
- Many ways to bring *quantum* into this picture!

Quantum Speed-up for Optimization and Machine Learning

- one of important quantum applications.
- **Heuristic**: variational, annealer, QAOA, ...

Interplay: Quantum, Optimization & Machine Learning

- Optimization and Machine Learning – two sides of the same coin!
- Many ways to bring *quantum* into this picture!

Quantum Speed-up for Optimization and Machine Learning

- one of important quantum applications.
- **Heuristic**: variational, annealer, QAOA, ...
- **Provable**: (1) thorough understanding of heuristics; (2) valuable guideline when empirical results are scarce.

Interplay: Quantum, Optimization & Machine Learning

- Optimization and Machine Learning – two sides of the same coin!
- Many ways to bring *quantum* into this picture!

Quantum Speed-up for Optimization and Machine Learning

- one of important quantum applications.
- **Heuristic**: variational, annealer, QAOA, ...
- **Provable**: (1) thorough understanding of heuristics; (2) valuable guideline when empirical results are scarce.
- This talk focuses on *quantization* of classical algorithms.

Quantization of Classical Algorithms

A typical classical iterative algorithm:

- Assume a feasible set P. Want to optimize $f(x)$ s.t. $x \in P$.
- A generic iterative algorithm with T iterations:
- $x_1 \to x_2 \to \cdots \to x_T$. Cost for each step: (1) store x_i; (2) determine x_i based on $x_{i-1}, \cdots, x_1$, P, $f(x)$.

Quantization of Classical Algorithms

A typical classical iterative algorithm:

- Assume a feasible set P. Want to optimize $f(x)$ s.t. $x \in P$.
- A generic iterative algorithm with T iterations:
- $x_1 \to x_2 \to \cdots \to x_T$. Cost for each step: (1) store x_i; (2) determine x_i based on $x_{i-1}, \cdots, x_1$, P, $f(x)$.

How quantum potentially speeds up this procedure?

- Reduce the cost for each step. Make it quantum and/or store x_is quantumly. However, this could **complicate** the determination of next x_is.
- Not clear how to reduce the number of iterations T.

Cases where we can make it work

- **Convex Optimization**: a quantum algorithm using $\tilde{O}(n)$ queries to the *evaluation* and the *membership* oracles, whereas the best known classical algorithms makes $O(n^2)$ such queries.

Cases where we can make it work

- **Convex Optimization**: a quantum algorithm using $\tilde{O}(n)$ queries to the *evaluation* and the *membership* oracles, whereas the best known classical algorithms makes $O(n^2)$ such queries.
- **Quantum SDP solvers** : a quantum algorithm solves n-dimensional semidefinite programs with m constraints, sparsity s and error ϵ in time $\tilde{O}((\sqrt{m} + \sqrt{n})s^2(Rr/\epsilon)^8)$ where R, r are bounds on the primal/dual solutions.

Cases where we can make it work

- **Convex Optimization**: a quantum algorithm using $\tilde{O}(n)$ queries to the *evaluation* and the *membership* oracles, whereas the best known classical algorithms makes $O(n^2)$ such queries.
- **Quantum SDP solvers** : a quantum algorithm solves n-dimensional semidefinite programs with m constraints, sparsity s and error ϵ in time $\tilde{O}((\sqrt{m}+\sqrt{n})s^2(Rr/\epsilon)^8)$ where R, r are bounds on the primal/dual solutions.
- **Classification**: a sublinear quantum algorithm for training linear and kernel-based classifiers that runs in $O(\sqrt{n}+\sqrt{d})$ given n data points in R^d, whereas the state-of-the-art (and optimal) classical algorithm runs in $O(n+d)$.

Cases where we can make it work

- **Convex Optimization**: a quantum algorithm using $\tilde{O}(n)$ queries to the *evaluation* and the *membership* oracles, whereas the best known classical algorithms makes $O(n^2)$ such queries.
- **Quantum SDP solvers** : a quantum algorithm solves n-dimensional semidefinite programs with m constraints, sparsity s and error ϵ in time $\tilde{O}((\sqrt{m}+\sqrt{n})s^2(Rr/\epsilon)^8)$ where R, r are bounds on the primal/dual solutions.
- **Classification**: a sublinear quantum algorithm for training linear and kernel-based classifiers that runs in $O(\sqrt{n}+\sqrt{d})$ given n data points in R^d, whereas the state-of-the-art (and optimal) classical algorithm runs in $O(n+d)$.

Cases where we can make it work

- **Convex Optimization**: a quantum algorithm using $\tilde{O}(n)$ queries to the *evaluation* and the *membership* oracles, whereas the best known classical algorithms makes $O(n^2)$ such queries.
- **Quantum SDP solvers** : a quantum algorithm solves n-dimensional semidefinite programs with m constraints, sparsity s and error ϵ in time $\tilde{O}((\sqrt{m}+\sqrt{n})s^2(Rr/\epsilon)^8)$ where R, r are bounds on the primal/dual solutions.
- **Classification**: a sublinear quantum algorithm for training linear and kernel-based classifiers that runs in $O(\sqrt{n}+\sqrt{d})$ given n data points in R^d, whereas the state-of-the-art (and optimal) classical algorithm runs in $O(n+d)$.

Yes, we do have accompanying **lower bounds**.

Outline

- Motivation
- Convex Optimization
- Semidefinite programs
- Classification
- Lower bounds
- Future research

Convex optimization

Convex optimization is a central topic in computer science with applications in:

- Machine learning: training a model is equivalent to optimizing a loss function.
- Algorithm design: LP/SDP-relaxation, such as various graph algorithms (vertex cover, max cut,...)
- ......

Classically, it is a major class of optimization problems that has polynomial time algorithms.

Convex optimization

In general, convex optimization has the following form:

$$\min f(x) \quad \text{s.t. } x \in \mathcal{C},$$

where $\mathcal{C} \subseteq \mathbb{R}^n$ is promised to be a convex body and $f\colon \mathbb{R}^n \to \mathbb{R}$ is promised to be a convex function.

Convex optimization

In general, convex optimization has the following form:

$$\min f(x) \quad \text{s.t. } x \in \mathcal{C},$$

where $\mathcal{C} \subseteq \mathbb{R}^n$ is promised to be a convex body and $f\colon \mathbb{R}^n \to \mathbb{R}$ is promised to be a convex function.

It is common to be provided with two oracles:

- *membership oracle*: input an $x \in \mathbb{R}^n$, tell whether $x \in \mathcal{C}$;
- *evaluation oracle*: input an $x \in \mathcal{C}$, output $f(x)$.

Convex optimization

In general, convex optimization has the following form:

$$\min f(x) \quad \text{s.t. } x \in \mathcal{C},$$

where $\mathcal{C} \subseteq \mathbb{R}^n$ is promised to be a convex body and $f\colon \mathbb{R}^n \to \mathbb{R}$ is promised to be a convex function.

It is common to be provided with two oracles:

- *membership oracle*: input an $x \in \mathbb{R}^n$, tell whether $x \in \mathcal{C}$;
- *evaluation oracle*: input an $x \in \mathcal{C}$, output $f(x)$.

Given a parameter $\epsilon > 0$ for accuracy, the goal is to output an $\tilde{x} \in \mathcal{C}$ such that

$$f(\tilde{x}) \leq \min_{x \in \mathcal{C}} f(x) + \epsilon.$$

Convex optimization

Classically, it is well-known that such an $\tilde{x}$ can be found in polynomial time using the ellipsoid method, cutting plane methods or interior point methods.

Convex optimization

Classically, it is well-known that such an $\tilde{x}$ can be found in polynomial time using the ellipsoid method, cutting plane methods or interior point methods.

Currently, the state-of-the-art result by Lee, Sidford, and Vempala uses $\tilde{O}(n^2)$ queries and additional $\tilde{O}(n^3)$ time.

Convex optimization

Classically, it is well-known that such an $\tilde{x}$ can be found in polynomial time using the ellipsoid method, cutting plane methods or interior point methods.

Currently, the state-of-the-art result by Lee, Sidford, and Vempala uses $\tilde{O}(n^2)$ queries and additional $\tilde{O}(n^3)$ time.

Quantumly, we are promised to have unitaries $O_{\mathcal{C}}$ and O_f s.t.

- for any $x \in \mathbb{R}^n$, $O_{\mathcal{C}}|x\rangle|0\rangle = |x\rangle|I_{\mathcal{C}}(x)\rangle$, where $I_{\mathcal{C}}(x) = 1$ if $x \in \mathcal{C}$ and $I_{\mathcal{C}}(x) = 0$ if $x \notin \mathcal{C}$;
- for any $x \in \mathcal{C}$, $O_f|x\rangle|0\rangle = |x\rangle|f(x)\rangle$.

Convex optimization

Main result. Convex optimization takes

- $\tilde{O}(n)$ and $\Omega(\sqrt{n})$ quantum queries to $O_{\mathcal{C}}$;
- $\tilde{O}(n)$ and $\tilde{\Omega}(\sqrt{n})$ quantum queries to O_f.

Furthermore, the quantum algorithm also uses $\tilde{O}(n^3)$ additional time.

Convex optimization

Main result. Convex optimization takes

- $\tilde{O}(n)$ and $\Omega(\sqrt{n})$ quantum queries to $O_{\mathcal{C}}$;
- $\tilde{O}(n)$ and $\tilde{\Omega}(\sqrt{n})$ quantum queries to O_f.

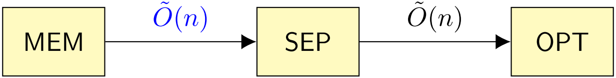
Furthermore, the quantum algorithm also uses $\tilde{O}(n^3)$ additional time.

As a result, we obtain:

- The first nontrivial quantum upper bound on general convex optimization.
- Impossibility of generic exponential quantum speedup of convex optimization! The speedup is at most polynomial.

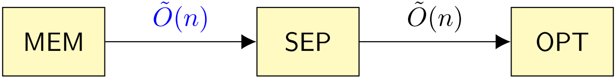
Convex optimization: quantum upper bound

Lee-Sidford-Vempala gives classical oracle reductions:

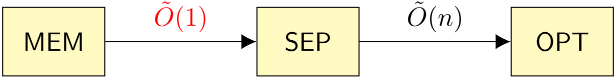


Convex optimization: quantum upper bound

Lee-Sidford-Vempala gives classical oracle reductions:

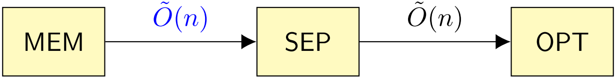


We give corresponding quantum oracle reductions:

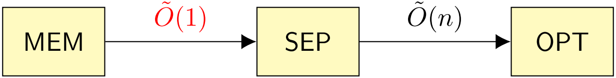


Convex optimization: quantum upper bound

Lee-Sidford-Vempala gives classical oracle reductions:



We give corresponding quantum oracle reductions:



Both papers use the same cutting plane based reduction from OPT to SEP. We show an improved upper bound by reducing the query complexity of the reduction from SEP to MEM.

Construction of SEP from MEM

- A $O(n)$ classical reduction w/ sub-gradient computation.

Construction of SEP from MEM

- A $O(n)$ classical reduction w/ sub-gradient computation.

Jordan's algorithm for gradients!

- Prepare the state $e^{if(x)}|x\rangle$ with $\tilde{O}(1)$ queries.
- Since $f(x) \approx \sum_{k=1}^{n} \frac{\partial f}{\partial x_k} x_k$,

$$\sum_x e^{if(x)}|x\rangle \approx \sum_x \bigotimes_{k=1}^{n} e^{i\frac{\partial f}{\partial x_k} x_k}|x_k\rangle.$$

Apply QFT reveals $\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}$.

Construction of SEP from MEM

- A $O(n)$ classical reduction w/ sub-gradient computation.

Jordan's algorithm for gradients!

- Prepare the state $e^{if(x)}|x\rangle$ with $\tilde{O}(1)$ queries.
- Since $f(x) \approx \sum_{k=1}^{n} \frac{\partial f}{\partial x_k} x_k$,

$$\sum_x e^{if(x)}|x\rangle \approx \sum_x \bigotimes_{k=1}^{n} e^{i\frac{\partial f}{\partial x_k}x_k}|x_k\rangle.$$

  Apply QFT reveals $\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}$.

From gradients to sub-gradients

- Compute the gradient of the *mollification* of the original function!
- Achieve so by carefully sampling from the neighborhood.

Outline

- Motivation
- Convex Optimization
- Semidefinite programs
- Classification
- Lower bounds
- Future research

Semidefinite programming (SDP)

Given m real numbers $a_1, \ldots, a_m \in \mathbb{R}$, s-sparse $n \times n$ Hermitian matrices $A_1, \ldots, A_m, C$, the SDP is defined as

$$\begin{aligned} \max \quad & \mathrm{tr}[CX] \\ \text{s.t.} \quad & \mathrm{tr}[A_i X] \leq a_i \quad \forall\, i \in [m]; \\ & X \succeq 0. \end{aligned}$$

Semidefinite programming (SDP)

Given m real numbers $a_1, \ldots, a_m \in \mathbb{R}$, s-sparse $n \times n$ Hermitian matrices $A_1, \ldots, A_m, C$, the SDP is defined as

$$\begin{aligned} \max \quad & \operatorname{tr}[CX] \\ \text{s.t.} \quad & \operatorname{tr}[A_i X] \leq a_i \quad \forall\, i \in [m]; \\ & X \succeq 0. \end{aligned}$$

SDPs can be solved in polynomial time. Classical *state-of-the-art* algorithms include:

- Cutting-plane method: $\tilde{O}(m(m^2 + n^{2.374} + mns)\,\mathrm{poly}\log(Rr/\epsilon))$.
- Matrix multiplicative weight: $\tilde{O}(mns(Rr/\epsilon)^7)$.

Quantum algorithms for SDPs

Brandão and Svore gave a quantum algorithm with complexity $\tilde{O}(\sqrt{mn}s^2(Rr/\epsilon)^{32})$, a quadratic speed-up in m, n, (later improved to $\tilde{O}(\sqrt{mn}s^2(Rr/\epsilon)^{8})$, based on the **Matrix Multiplicative Weight Update** method.

Quantum algorithms for SDPs

Brandão and Svore gave a quantum algorithm with complexity $\tilde{O}(\sqrt{mn}s^2(Rr/\epsilon)^{32})$, a quadratic speed-up in m, n, (later improved to $\tilde{O}(\sqrt{mn}s^2(Rr/\epsilon)^{8})$, based on the **Matrix Multiplicative Weight Update** method.

No exponential speed-up: also proved $\Omega(\sqrt{m}+\sqrt{n})$ as a lower bound.

Quantum algorithms for SDPs

Brandão and Svore gave a quantum algorithm with complexity $\tilde{O}(\sqrt{mn}s^2(Rr/\epsilon)^{32})$, a quadratic speed-up in m, n, (later improved to $\tilde{O}(\sqrt{mn}s^2(Rr/\epsilon)^{8})$, based on the **Matrix Multiplicative Weight Update** method.

No exponential speed-up: also proved $\Omega(\sqrt{m}+\sqrt{n})$ as a lower bound.

Input model

An oracle that takes input $j \in [m+1]$, $k \in [n]$, $l \in [s]$, and performs the map

$$|j, k, l, 0\rangle \mapsto |j, k, l, (A_j)_{k, s_{jk}(l)}\rangle,$$

where $(A_j)_{k, s_{jk}(l)}$ is the l^{th} nonzero element in the k^{th} row of matrix A_j.

Optimal quantum algorithms for SDPs

Can we close the gap between $\tilde{O}(\sqrt{mn})$ and $\Omega(\sqrt{m} + \sqrt{n})$?

Optimal quantum algorithms for SDPs

Can we close the gap between $\tilde{O}(\sqrt{mn})$ and $\Omega(\sqrt{m}+\sqrt{n})$?Yes!

Theorem

For any $\epsilon > 0$, there is a quantum algorithm that solves the SDP using at most

$$\tilde{O}\big((\sqrt{m}+\sqrt{n})s^2(Rr/\epsilon)^8\big)$$

quantum gates and queries to oracles.

Optimal quantum algorithms for SDPs

Can we close the gap between $\tilde{O}(\sqrt{mn})$ and $\Omega(\sqrt{m}+\sqrt{n})$?Yes!

Theorem

For any $\epsilon > 0$, there is a quantum algorithm that solves the SDP using at most

$$\tilde{O}\big((\sqrt{m}+\sqrt{n})s^2(Rr/\epsilon)^8\big)$$

quantum gates and queries to oracles.

paper	result
BS17	$\tilde{O}(\sqrt{mn}s^2(Rr/\epsilon)^{32})$
vAGGdW17	$\tilde{O}(\sqrt{mn}s^2(Rr/\epsilon)^8)$
this talk	$\tilde{O}((\sqrt{m}+\sqrt{n})s^2(Rr/\epsilon)^8)$

Optimal quantum algorithms for SDPs

The behavior of the algorithm:

- The good: optimal in m, n
- The bad: dependence on R, r, ϵ^{-1} is too high: $(Rr/\epsilon)^8$

Optimal quantum algorithms for SDPs

The behavior of the algorithm:

- The good: optimal in m, n
- The bad: dependence on R, r, ϵ^{-1} is too high: $(Rr/\epsilon)^8$

Applications:

- The good: Some machine learning, especially compressed sensing problems have $Rr/\epsilon = O(1)$ (Ex. quantum compressed sensing by Gross et al. 09).
- The bad: The SDP in the Goeman-Williams algorithm for MAX-CUT has $Rr/\epsilon = \Theta(n)$ (and many other algorithmic SDP applications).

Matrix multiplicative weight method (MMW)

Versatile Framework

- MMW: (matrix) boosting, online learning, ...

Matrix multiplicative weight method (MMW)

Versatile Framework

- MMW: (matrix) boosting, online learning, ...
- A way to get good advices from a few mediocre experts.

Matrix multiplicative weight method (MMW)

Versatile Framework

- MMW: (matrix) boosting, online learning, ...
- A way to get good advices from a few mediocre experts.
- Arora-Kale: use MMW to solve the primal-dual problem of SDPs, which leads to a classical SDP solvers.

Matrix multiplicative weight method (MMW)

Versatile Framework

- MMW: (matrix) boosting, online learning, ...
- A way to get good advices from a few mediocre experts.
- Arora-Kale: use MMW to solve the primal-dual problem of SDPs, which leads to a classical SDP solvers.
- A good candidate to quantize:

Matrix multiplicative weight method (MMW)

Versatile Framework

- MMW: (matrix) boosting, online learning, ...
- A way to get good advices from a few mediocre experts.
- Arora-Kale: use MMW to solve the primal-dual problem of SDPs, which leads to a classical SDP solvers.
- A good candidate to quantize:
  - The number of iterations T is poly-log in terms of n and m.

Matrix multiplicative weight method (MMW)

Versatile Framework

- MMW: (matrix) boosting, online learning, ...
- A way to get good advices from a few mediocre experts.
- Arora-Kale: use MMW to solve the primal-dual problem of SDPs, which leads to a classical SDP solvers.
- A good candidate to quantize:
  - The number of iterations T is poly-log in terms of n and m.
  - Each intermediate solution is

$$\rho^{(t)} = \frac{\exp\left[\frac{\epsilon}{4}\sum_{\tau=1}^{t-1} M^{(\tau)}\right]}{\mathrm{Tr}[\exp\left[\frac{\epsilon}{4}\sum_{\tau=1}^{t-1} M^{(\tau)}\right]]},$$

which is a **Gibbs** state that quantumly can generate efficiently! (e.g, PW09)

Matrix multiplicative weight method (MMW)

Consider the following SDP feasibility problem:

$$\mathrm{tr}[A_i X] \leq a_i + \epsilon \quad \forall\, i \in [m]; \tag{1}$$

$$X \succeq 0; \quad \mathrm{Tr}[X] = 1. \tag{2}$$

Matrix multiplicative weight method (MMW)

Consider the following SDP feasibility problem:

$$\mathrm{tr}[A_i X] \leq a_i + \epsilon \quad \forall\, i \in [m]; \tag{1}$$

$$X \succeq 0; \quad \mathrm{Tr}[X] = 1. \tag{2}$$

Zero-sum Game View

- Player 1: a feasible $X \in \mathcal{S}_\epsilon$.
- Player 2: any violation of any proposed X.

Matrix multiplicative weight method (MMW)

Consider the following SDP feasibility problem:

$$\mathrm{tr}[A_i X] \leq a_i + \epsilon \quad \forall\, i \in [m]; \tag{1}$$

$$X \succeq 0; \quad \mathrm{Tr}[X] = 1. \tag{2}$$

Zero-sum Game View

- Player 1: a feasible $X \in \mathcal{S}_\epsilon$.
- Player 2: any violation of any proposed X.
- Feasibility implies a feasible point X_0 (provided by Player 1) with no violation found by Player 2.

An *equilibrium* point of such can be found by MMW.

Matrix multiplicative weight method (MMW)

Consider the following SDP feasibility problem:

$$\mathrm{tr}[A_i X] \le a_i + \epsilon \quad \forall\, i \in [m]; \tag{1}$$

$$X \succeq 0; \quad \mathrm{Tr}[X] = 1. \tag{2}$$

Zero-sum Game View

- Player 1: a feasible $X \in \mathcal{S}_\epsilon$.
- Player 2: any violation of any proposed X.
- Feasibility implies a feasible point X_0 (provided by Player 1) with no violation found by Player 2.

An *equilibrium* point of such can be found by MMW.

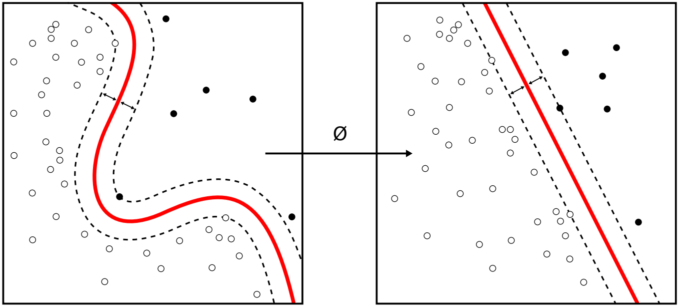
Efficiency of Implementation

- Player 1 is due to quantum Gibbs sampling.
- Player 2 is due to a faster quantum OR lemma.

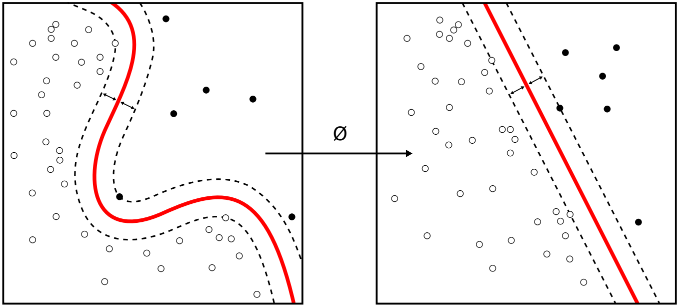
Outline

- Motivation
- Convex Optimization
- Semidefinite programs
- Classification
- Lower bounds
- Future research

A visualization of classification



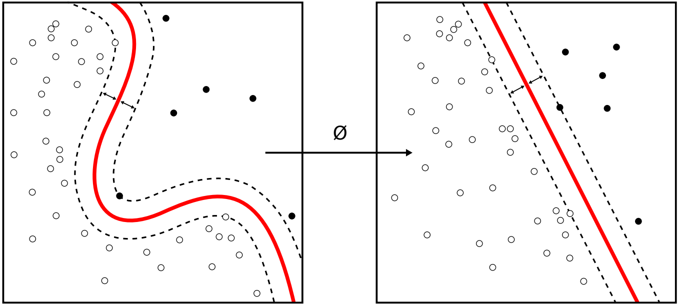
A visualization of classification



- (linear) Given $X_1, \cdots, X_n \in R^d$ and a label vector $y \in \{-1, +1\}^n$, find a hyperplane $w \in R^d$, s.t.

$$y_i \cdot X_i^T w \geq 0, \forall i \in [n].$$

A visualization of classification



- (linear) Given $X_1, \cdots, X_n \in R^d$ and a label vector $y \in \{-1, +1\}^n$, find a hyperplane $w \in R^d$, s.t.

$$y_i \cdot X_i^T w \geq 0, \forall i \in [n].$$

- (kernel-based) $X_i \to \Phi(X_i)$ for some kernel function $\Phi(\cdot)$.

Input/Output Model & Result

Input/Output Model

- (input) Standard coherent access to each entry of X_i.
- (output) Classical efficient representation of w (recover any w_i with $\tilde{O}(1)$ overhead).

Input/Output Model & Result

Input/Output Model

- (input) Standard coherent access to each entry of X_i.
- (output) Classical efficient representation of w (recover any w_i with $\tilde{O}(1)$ overhead).

Result & Comparison

paper	result	technique
CHW12	$\Theta(n + d)$	classical efficient sampling
KWS16	$\tilde{O}(\sqrt{n}d)$	quantized Perceptron
this talk	$\tilde{O}(\sqrt{n} + \sqrt{d})$	quantized fast sampling

Input/Output Model & Result

Input/Output Model

- (input) Standard coherent access to each entry of X_i.
- (output) Classical efficient representation of w (recover any w_i with $\tilde{O}(1)$ overhead).

Result & Comparison

paper	result	technique
CHW12	$\Theta(n+d)$	classical efficient sampling
KWS16	$\tilde{O}(\sqrt{n}d)$	quantized Perceptron
this talk	$\tilde{O}(\sqrt{n}+\sqrt{d})$	quantized fast sampling

Similar results apply to *kernel-based classification, minimum enclosing ball, and ℓ_2-SVM.*

High-level Technique

- This is an *equilibrium value* problem in disguise. Take $X_i \leftarrow (-1)^{y_i} X_i$, it reduces to $\max_w \min_i X_i^\top w \geq 0.$

High-level Technique

- This is an *equilibrium value* problem in disguise. Take $X_i \leftarrow (-1)^{y_i} X_i$, it reduces to $\max_w \min_i X_i^\top w \geq 0$.
- However, this is an example over ℓ_2 unit balls.

High-level Technique

- This is an *equilibrium value* problem in disguise. Take $X_i \leftarrow (-1)^{y_i} X_i$, it reduces to $\max_w \min_i X_i^\top w \geq 0.$
- However, this is an example over ℓ_2 unit balls.
- Fortunately, there exists a classical ℓ_2 *sampling* approach with *O(n + d) cost* for multiplicative weight updates. (*analysis relies on martingale concentration bounds.*)

High-level Technique

- This is an *equilibrium value* problem in disguise. Take $X_i \leftarrow (-1)^{y_i} X_i$, it reduces to $\max_w \min_i X_i^\top w \geq 0$.
- However, this is an example over ℓ_2 unit balls.
- Fortunately, there exists a classical ℓ_2 *sampling* approach with *O(n + d) cost* for multiplicative weight updates. (*analysis relies on martingale concentration bounds.*)
- Extend ℓ_2 sampling to quantum is equivalent to *state preparation* of particular quantum states.

High-level Technique

- This is an *equilibrium value* problem in disguise. Take $X_i \leftarrow (-1)^{y_i} X_i$, it reduces to $\max_w \min_i X_i^\top w \geq 0$.
- However, this is an example over ℓ_2 unit balls.
- Fortunately, there exists a classical ℓ_2 *sampling* approach with *O(n + d) cost* for multiplicative weight updates. (*analysis relies on martingale concentration bounds.*)
- Extend ℓ_2 sampling to quantum is equivalent to *state preparation* of particular quantum states.
- Main contributions:

High-level Technique

- This is an *equilibrium value* problem in disguise. Take $X_i \leftarrow (-1)^{y_i} X_i$, it reduces to $\max_w \min_i X_i^\top w \geq 0.$
- However, this is an example over ℓ_2 unit balls.
- Fortunately, there exists a classical ℓ_2 *sampling* approach with *$O(n + d)$ cost* for multiplicative weight updates. (*analysis relies on martingale concentration bounds.*)
- Extend ℓ_2 sampling to quantum is equivalent to *state preparation* of particular quantum states.
- Main contributions:
  - $O(\sqrt{n} + \sqrt{d})$ quantum sampling of the desired state.

High-level Technique

- This is an *equilibrium value* problem in disguise. Take $X_i \leftarrow (-1)^{y_i} X_i$, it reduces to $\max_w \min_i X_i^\top w \geq 0$.
- However, this is an example over ℓ_2 unit balls.
- Fortunately, there exists a classical ℓ_2 *sampling* approach with *$O(n + d)$ cost* for multiplicative weight updates. (*analysis relies on martingale concentration bounds.*)
- Extend ℓ_2 sampling to quantum is equivalent to *state preparation* of particular quantum states.
- Main contributions:
  - $O(\sqrt{n} + \sqrt{d})$ quantum sampling of the desired state.
  - Extension of the concentration analysis to quantum.

High-level Technique

- This is an *equilibrium value* problem in disguise. Take $X_i \leftarrow (-1)^{y_i} X_i$, it reduces to $\max_w \min_i X_i^\top w \geq 0$.
- However, this is an example over ℓ_2 unit balls.
- Fortunately, there exists a classical ℓ_2 *sampling* approach with *$O(n + d)$ cost* for multiplicative weight updates. (*analysis relies on martingale concentration bounds.*)
- Extend ℓ_2 sampling to quantum is equivalent to *state preparation* of particular quantum states.
- Main contributions:
  - $O(\sqrt{n} + \sqrt{d})$ quantum sampling of the desired state.
  - Extension of the concentration analysis to quantum.

High-level Technique

- This is an *equilibrium value* problem in disguise. Take $X_i \leftarrow (-1)^{y_i} X_i$, it reduces to $\max_w \min_i X_i^\top w \geq 0.$
- However, this is an example over ℓ_2 unit balls.
- Fortunately, there exists a classical ℓ_2 *sampling* approach with *O(n + d) cost* for multiplicative weight updates. (*analysis relies on martingale concentration bounds.*)
- Extend ℓ_2 sampling to quantum is equivalent to *state preparation* of particular quantum states.
- Main contributions:
  - $O(\sqrt{n} + \sqrt{d})$ quantum sampling of the desired state.
  - Extension of the concentration analysis to quantum.

Feature of the quantum algorithm

classical output, highly classical-quantum hybrid, state sampling

Outline

The lower bound

- **Convex Optimization**: Convex optimization takes
  - $\tilde{O}(n)$ and $\Omega(\sqrt{n})$ quantum queries to $O_{\mathcal{C}}$;
  - $\tilde{O}(n)$ and $\tilde{\Omega}(\sqrt{n})$ quantum queries to O_f.
- **Semidefinite Programs**:
  - Upper bound: $\tilde{O}((\sqrt{m}+\sqrt{n})s^2(Rr/\epsilon)^8)$.
  - Lower bound: $\Omega(\sqrt{m}+\sqrt{n})$.
- **Classification**:
  - Upper bound: $\tilde{O}(\sqrt{n}+\sqrt{d})$.
  - Lower bound: $\Omega(\sqrt{n}+\sqrt{d})$.

The lower bound

- **Convex Optimization**: Convex optimization takes
  - $\tilde{O}(n)$ and $\Omega(\sqrt{n})$ quantum queries to $O_{\mathcal{C}}$;
  - $\tilde{O}(n)$ and $\tilde{\Omega}(\sqrt{n})$ quantum queries to O_f.
- **Semidefinite Programs**:
  - Upper bound: $\tilde{O}((\sqrt{m}+\sqrt{n})s^2(Rr/\epsilon)^8)$.
  - Lower bound: $\Omega(\sqrt{m}+\sqrt{n})$.
- **Classification**:
  - Upper bound: $\tilde{O}(\sqrt{n}+\sqrt{d})$.
  - Lower bound: $\Omega(\sqrt{n}+\sqrt{d})$.

High-level difficulty:

- (1) continuous domain (vs Boolean oracle query);
- (2) classical lower bounds are not studied comprehensively;
- (3) how to go beyond $\Omega(\sqrt{n})$?

Outline

- Motivation
- Convex Optimization
- Semidefinite programs
- Classification
- Lower bounds
- Future research

What could we do next?

Continue on this thread:

- More quantization of classical MCMC algorithms (hitting or mixing)?
  upcoming: quantum algorithms for log-concave sampling and volume estimation in high dimension.

What could we do next?

Continue on this thread:

- More quantization of classical MCMC algorithms (hitting or mixing)?
  upcoming: quantum algorithms for log-concave sampling and volume estimation in high dimension.
- Near-term implementation: quantum sampler, variational solver, annealer, or QAOA?

What could we do next?

Continue on this thread:

- More quantization of classical MCMC algorithms (hitting or mixing)?
  upcoming: quantum algorithms for log-concave sampling and volume estimation in high dimension.
- Near-term implementation: quantum sampler, variational solver, annealer, or QAOA?

Find genuine quantum algorithms!

- Go beyond classical framework! Make use of quantum dynamics (e.g., tunneling).

What could we do next?

Continue on this thread:

- More quantization of classical MCMC algorithms (hitting or mixing)?
  upcoming: quantum algorithms for log-concave sampling and volume estimation in high dimension.
- Near-term implementation: quantum sampler, variational solver, annealer, or QAOA?

Find genuine quantum algorithms!

- Go beyond classical framework! Make use of quantum dynamics (e.g., tunneling).
- Non-convex optimization: (1) ubiquitous in ML; (2) numerical evidence of quantum speed-up. *Anything provable?*

Technical Open Questions I:

- Can we close the gap for both membership and evaluation queries? Our upper bounds on both oracles use $\tilde{O}(n)$ queries, whereas the lower bounds are only $\tilde{\Omega}(\sqrt{n})$.

- Can we improve the time complexity of our quantum algorithm? The time complexity $\tilde{O}(n^3)$ of our current quantum algorithm matches that of the classical state-of-the-art algorithm.

- What is the quantum complexity of convex optimization with a first-order oracle (i.e., with direct access to the gradient of the objective function)?

Technical Open Questions II:

- Concrete applications where quantum algorithms (both for convex optimization and SDPs) can have provable speed-ups?
- The use of QRAM (or non-trivial quantum data structure) in the state preparation steps in both quantum algorithms for SDPs and classification? Advantage for amortized complexity?
- Quantum algorithms for equilibrium point problems over other domain (e.g., game theory, learning theory)? The efficiency will depend on specific sampling techniques.

Thank you!

Q & A