CMSC 657 Final Report : Quantum Algorithms for Optimization

Ali Rad, Nhung Nguyen, Sohitri Ghosh
(Group 8)

Universtiy of Maryland, College Park, United States

1. Motivation

Quantum computers help to solve hard problems in computer science by substantially speeding up the algorithms (As we have seen exponential speed up in Shor's algorithm , polynomial speed up in Grover's algorithm). Some of the most interesting classical algorithms that have lots of applications nowadays are the machine learning algorithms. Can quantum computers provide speed-up to these algorithms? To tackle this question we have found it really interesting to study the quantum optimization algorithms which are the core building blocks of the machine learning algorithms.

2. Literature Review

One common class of optimization problems are finding the minimum (convex optimization) of a linear or quadratic function with convex constraints. These problems are found to be efficiently solved by classical semidefinite programming in polynomial time [1][5]. Recently it is also found that quantum computers can provide a speed up over classical computers to this type of problems. Brandao and Svore [2] and Aberdoorn et. al.[3] proved that a quantum SDP solver can have complexity of $\mathcal{O}(\sqrt{mn}s^2/\epsilon^8)$ compared to a classical SDP solver's complexity of $\mathcal{O}(m(m^2 + n\omega + mns)\text{polylog}(m, n, 1/\epsilon))$ [1] or $\mathcal{O}(mns/\epsilon^4 + ns/\epsilon^7)$ [5], with n and s to be the size and sparsity of the input matrix, m is the size of the constraints and ϵ is the additive error of the solution. Brandao et. al. [4] recently show that the complexity can be further improved to $\mathcal{O}(\sqrt{m} + \text{poly}(r))\text{poly}(\log m, \log n, 1/\epsilon))$ by using a fully quantum input model (r is the rank of the input matrix). This new quantum algorithms can be applied to find a description for an unknown l-qubit quantum state ρ subject to a set of measurements m in time $\sqrt{m}\text{poly}(\log m, \log n, r, 1/\epsilon)$. This method requires much less resource than quantum state tomography (using only $\text{poly}(\log m, \log n, r, 1/\epsilon)$ copies of the state instead of n^2) hence are more practical [4].

3. Goal

The quantum devices are able to speed-up the classical optimization algorithm with the advantage of Gibbs sampling. In this project, we would like to study how quantum SDP algorithm can help analyze classical and quantum data. It would be intersting to explore the applications of quantum SDP solvers as described in the references. Also we are interested to study how the quantum algorithm can solve more general convex optimization problems. Going through this topic would enable us to better understand the use of SDP solvers in different problems including those related to machine learning.

4. Introduction to SDP

Let us first look at semidefinite programming. We would like to optimize a linear function with linear constarints over a set of positive semidefinite (PSD) matrices. The primal problem is to maximize the trace of CX where C is $n \times n$ s-sparse matrix, with m constraints as defined below.

$$\begin{gathered} \max \operatorname{tr}(CX) \\ \operatorname{tr}(A_j X) \leq b_j \ \forall j \in [m] \\ X \geq 0 \end{gathered}$$

Our goal is to find what is X which optimizes the above linear function with inputs $C, A_1, ..., A_m$ which are $n \times n$ s-sparse matrices and $b_1, ..., b_m$ which are just numbers.

An equivalent dual problem is to to minimise $b \cdot y$ given m linear constrains.

$$\begin{gathered} \min(b \cdot y) \\ \sum_{j=1}^{m} y_j A_j \geq C \\ y \geq 0 \end{gathered}$$

Here our goal is to find the vector y which optimizes the above scenario. Under mild conditions the optimal solution of the primal problem is similar to the optimal solution of the dual problem. So, we can choose which problem to be solved.
The parameters R and r are defined in terms of the size of the solutions to the optimal or dual problem $[\operatorname{tr}(X_{opt}) \leq R]$ or $[\sum_i y_i \leq r]$.

4.1. SDP Lower Bounds

If we want to write down the solutions X or y it would take $O(n^2)$ or $O(m)$ to write down the solutions for primal and dual problems respectively. So, instead of writing down the whole solutions we would like to output the optimal value which would make the algorithm faster. From an easy reduction to search problem, the lower bound for the classical scenario is $O(n + m)$ whereas the quantum algorithm can have $O(\sqrt{n} + \sqrt{m})$ where r, R, s, δ are constants (s is the sparsity of matrix and δ is the error).

4.2. Input Output Model

We assume that there exists an oracle which can perform the following task.

$$|j,k,l,z\rangle \rightarrow |j,k,l,z \oplus (A_j)_{kf_{jk}l}\rangle$$

So the oracle outputs a chosen nonzero element of $C, A_1, ..., A_m$ at unit cost. Here j is the choice of A_j with k^{th} row with l^{th} nonzero element.

The outputs are samples from the distribution $\frac{y}{||y||_2}$ and the value $||y||_2$ or samples from the distribution $\frac{X}{\text{tr}X}$ and the value $\text{tr}X$. The quantum algorithm given in Brandao et . al. [2] for solving SDP runs in time $\sqrt{n}\sqrt{m}s^2$ poly$(\log(n,m)R,r,\delta)$. Though this algorithm has a very bad scaling with respect to R and r, this gives an unconditional quadratic speed-up over classical algorithm. The details of the quantum algorithm is given below.

5. The Arora-Kale Algorithm

The quantum algorithm is built on a classical algorithm of Arora and Kale for solving SDPs. In this algorithm for given density matrix ρ the algorithm searches for $D_\alpha = \{y \in R^m : y \geq 0, b \cdot y \leq \alpha\}$ such that $\sum_{j=1}^m y_i \text{tr}(A_j\rho) \geq \text{tr}(C\rho)$ or output fails if no such vector exist. The algorithm description is as following:

. Set $\rho^{(1)} = I/n$. Put $\epsilon = \frac{\delta\alpha}{2R^2}$ and $\epsilon' = \ln(1-\epsilon)$. Then for $T \geq \frac{16R^4\ln(n)}{\alpha^2\delta^2}$ run the algorithm for $t = 1, \cdots, T$

- Run Oracle$(\rho(t))$ if it fails, stop and output $\rho(t)$.
- Else, let $y(t)$ be vector generated by oracle.

Then let

$$M^{(t)} = \frac{\sum_{j=1}^m A_j y_j^{(t)}}{2\omega}$$

where ω is defined as $\max||\sum_j y_j A_j - C||$. Then compute

$$W^{(t+1)} = e^{-\epsilon'(\sum_{\tau=1}^t)M^\tau}$$

Then set

$$\rho^{(t+1)} = \frac{W^{(t+1)}}{\text{tr}(W^{(t+1)})}$$

and the output of the algorithm is

$$\bar{y} = \frac{\delta\alpha}{R}e_1 + \frac{1}{T}\sum_{t=1}^T y^t, \qquad e_1 = (1,0,\cdots,0)$$

5.1. Why Arora-Kale works?

Since $y_t \in D_\alpha := \{y : y \geq 0, b \cdot y \leq \alpha\}$, we realize that $\bar{y} \cdot b \leq \frac{\delta\alpha}{R} b_1 + \frac{1}{T}\sum_{t=1}^{T} y^t \cdot b \leq (1+\delta)\alpha$. Then we need to check that $\bar{y}$ is feasible. To see that, at first we need to know that from oracle for all t we have:

$$\mathrm{tr}((\sum_{j=1}^{m} y_j^t A_j - C)\rho^t) \geq 0$$

Then we need to get help from the Matrix Multiplicative Weight(MMW) lemmas (Arora,Kale'07) which states that for given $n \times n$ matrices M^t and $\epsilon < 0.5$ we have:

$$\frac{1}{T}\sum_{t=1}^{T} \mathrm{tr}(M^t \rho^t) \leq (\frac{1+\epsilon}{T})\lambda_n(\sum_{t=1}^{T} M^t) + \frac{\ln(n)}{T\epsilon}$$

with ρ^t proportional to $\exp(-\epsilon'(\sum_{\tau=1}^{t-1} M^\tau))$ with $\epsilon' = -\ln(1-\epsilon)$ and λ_n as min eigenvalue. So by using the this lemma we can easily show that

$$\lambda_{\min}((\sum_{j=1}^{m}(\frac{1}{T}\sum_{t=1}^{T} y_j^t)A_j - C)) \geq 0$$

5.2. Quantizing Arora-Kale Algorithm

We can enhance the classical SDP solver with the following three aspects via quantum mechanics. First, we can improve the multiplication weight method simply by preparing the quantum thermal Gibbs states of Hamiltonian given by linear combination of the input matrices $A_1, \cdots, A_m, C$. Therefore with quantum computer we can efficiently speed up in time polynomial in $\log(n)$. The second and third enhancement comes from the amplitude amplification and speed up in number of constrain m by using the Jayenes' principle of maximum entropy. So in the first part, we implement the oracle by Gibbs sampling to produce y^t and apply amplitude amplification to solve it in time $O(s^2 n^{1/2} m^{1/2})$. Then we sparsify M^t to be a sum of $O(\log(m))$ terms:

$$\bar{M}^t = (||y^t||_1 Q^{-1}\sum_{j=1}^{Q} A_{ij} - c + RI)\frac{1}{2R}, \qquad Q = O(\log(m))$$

Then we use from quantum Gibbs sampling and amplitude amplification to prepare

$$\bar{\rho}^t \sim e^{-\epsilon' \sum_{\tau=1}^{t} \bar{M}^\tau})$$

in time $O(s^2 n^{1/2})$.

Algorithm 1: Quantum Arora-Kale [2]

Let $\rho^1 = I/n$, $\epsilon = \frac{\delta\alpha}{2\omega R}$, $\epsilon' = -\ln(1-\epsilon)$, $\frac{8\omega^2 R^2 \ln(n)}{\delta^2\alpha^2}$;

for $t = 1, 2, \cdots, T$ **do**

1. Prepare y^t by Gibbs sampling: $y^t \leftarrow \text{oracle}(\rho^t)$
2. $M^t = \sum_{j=1}^{m}(y_j^t A_j - C + \omega I)\frac{1}{2\omega}$
3. Sparsify M^t to $(M')^t$
4. $\rho^{t+1} = \exp(-\epsilon(\sum(\sum_{\tau=1}^{t}(M')^\tau))/\text{tr}(\cdots)$

end

;

Output: $\bar{y} = \frac{\delta\alpha}{R}e_1 + \frac{1}{T}\sum_{t=1}^{T} y^t$;

6. Quantum SDPs solver with better dependence on n and m

6.1. SDP feasibility problem

The SDP optimization problem can be reduced to a feasibility by using a binary search: one can guess a candidate X_0 and convert it to a constraint.

$$\max \text{tr}(CX) \Rightarrow \text{tr}(-CX) \leq \text{tr}(-CX_0)$$

Hence the problem becomes finding a convex region S_ϵ of all X such that

$$\text{tr}(A_j X) \leq b_j \forall j \in [m] \tag{1}$$

$$X \geq 0 \tag{2}$$

$$\text{tr}(X) = 1 \tag{3}$$

If S_ϵ is not empty, we can do a binary search in $S\epsilon$ to find X.

6.2. Zero sum game approach and fast quantum OR lemma

Instead of follow the primal-dual approach, we can use zero sum game approach to solve the feasibility problem. The zero sum game approach is basically a competition to find a counter example to the feasibility problem. There is some benefits of using this simpler and more intuitive approach. First, the dependence on the dual solution is eliminated. Secondly, this intuitively leads to an improved version of quantum OR lemma, which speeds up the process of generating a new copy of X.

Oracle 6.1. Input a density matrix X, output an $i \in [m]$ so that *Eq.* (1) is violated. If no such i exists, output "FEASIBLE".

In the following algorithm, Brandao et. al. use the Gibbs sampler to generate potential candidates for the zero sum game and use Oracle 6.1 to determine whether it is a feasible solution [4].

Algorithm 2: Matrix multiplicative weights for testing feasibility [4]

Initialize the weight matrix $W^{(1)} = I_n$, and $T = \frac{16\ln n}{\delta^2}$;
for $t = 1, 2, \cdots, T$ **do**
  1. Prepare the Gibbs state $\rho^{(t)} = \frac{W^{(t)}}{\mathrm{Tr}[W^{(t)}]}$
  2. Find a $j^{(t)} \in \{1, 2, \cdots, m\}$ such that $\mathrm{Tr}(A_{j^{(t)}}\rho^{(t)}) > aj^{(t)} + \delta$.
  if $j_{(t)}$ *is found* **then**
    Take $M^{(t)} = \frac{1}{2}(I_n - A_{j^{(t)}})$
  else
    Claim $S_\delta \neq \varnothing$ and output $\rho^{(t)}$;
  end
  3. Define a new weight matrix: $W^{(t+1)} = \exp[-\frac{\delta}{2}\sum_{\tau=1}^{t} M^{(\tau)}]$
end
;
Claim that $S_0 = \varnothing$ and terminate;

The improvement from previous quantum SDP solvers comes from the fast amplification algorithm used to implement the Oracle 6.1 in step 2. Using the fast amplification algorithm , Brandao et. al. prove a fast quantum OR lemma, which allows the reduction of gate complexity from $\sqrt{m}\sqrt{n}$ to $\sqrt{m} + \sqrt{n}$.

6.3. Quantum SDP solver with quantum inputs

If we further restrict the form of A_j to be $A_j = A_j^+ - A_j^-$ (where A_j^+, A_j^- are positive semi-definite), we can implement the following oracles to find $\mathrm{Tr}[A_j]$, prepare A_j and the constraints a_j efficiently. Theses oracles allow the Gibbs state of the form $\frac{\exp(-K)}{\mathrm{Tr}(\exp(-K))}$ (where $K = K^+ - K^-, K^\pm = \sum_{j\in S} c_j A_j^\pm, S$ is a set of violated constraints) to be prepared efficiently. The gate complexity for preparing this Gibbs state is $poly(\log m, \log n)$ compared to $\sqrt{n}$ as in the plain input model[4].

Oracle 6.2. $O_{\mathrm{Tr}} |j\rangle |0\rangle |0\rangle = |j\rangle \mathrm{Tr}[A_j^+]\mathrm{Tr}[A_j^-]$

Oracle 6.3. $O |j\rangle \langle j| \otimes |0\rangle \langle 0| \otimes |0\rangle \langle 0| O^\dagger = |j\rangle \langle j| \otimes |\psi_j^+\rangle \langle \psi_j^+| \otimes |\psi_j^-\rangle \langle \psi_j^-|$,
where $|\psi_j^+\rangle, |\psi_j^-\rangle$ are some purifications of $\frac{A_j^+}{\mathrm{Tr}[A_j^+]}, \frac{A_j^-}{\mathrm{Tr}[A_j^-]}$.

Oracle 6.4. $O_a |j\rangle \langle j| \otimes |0\rangle \langle 0| O_a^\dagger = |j\rangle \langle j| \otimes |a_j\rangle \langle a_j|$

7. Application : Learnability of Quantum States

An important task in quantum mechanics is to learn the state of the system i.e. to compute the density matrix ρ given many realizations of an experiment. We can achieve that by state tomography. For such full quantum state tomography we need n^2 copies for a $n = 2^k$ dimensional qubit state, which is a lot even for a

lower dimensional qubit state. Instead, we can try to learn the statistics about the quantum states like their expectation values with respect to a set of measurements. Basically, we are looking for $\mathrm{Tr}(\rho E_i)$ upto an error ϵ for a set of POVM $E_1, ...E_m$ where $I \geq E_i \geq 0 \in C^{n \times n}\ \forall i \in m$.

Question 7.1. *Given a set of measurements E_i and having access to copies of unknown state ρ, we want to find a description of the state σ such that the following constraint is satisfied .*

$$|\,Tr(\sigma E_i) - Tr(\rho E_i)| \leq \epsilon, \forall i \in m$$

7.1. Reduction to Feasibility Problem

Our job is now reduced to the SDP feasibility problem as described in the previous section in the following way.

$$\begin{aligned} |\mathrm{Tr}(\sigma E_i) &\leq \mathrm{Tr}(\rho E_i)| + \epsilon \\ |\mathrm{Tr}(\sigma E_i) &\geq \mathrm{Tr}(\rho E_i)| - \epsilon \\ \mathrm{Tr}(\sigma) &= 1 \\ \sigma &\geq 0 \end{aligned}$$

Here our task is to search for the state σ which follows the same statistics as of ρ upto an error ϵ. Comparing with the primal SDP problem, we find that the constraints here can be expressed as following,

$$b_i = \pm \mathrm{Tr}(\rho E_i) + \epsilon$$

We can learn the constraints b_i by implementing an oracle which measures E_i on ρ and we assume that there is an efficient quantum circuit for performing these measurements in poly(log(n)) time (as shown in next section).

Jaynes' principle states that there is always a Gibbs state of the following form with real numbers λ_i , which has the same expectation values on the E_is as the original state ρ where λ_i s are real numbers.

$$\frac{\exp(\sum_i \lambda_i E_i)}{\mathrm{Tr}(\exp(\sum_i \lambda_i E_i))}$$

We have to find out these λ_i s and our solution would then take a form of

$$\sigma = \exp(\sum_i \lambda_i E_i)$$

.

For low rank matrices, there is a huge speed up.

7.2. Quantum Input Model

The quantum input model is relevant for low rank measurements.

Oracle 7.1. ***for traces of Ei*** *: A unitary O_{Tr} such that for any $i \in [m], O_{Tr} |i\rangle |0\rangle = |i\rangle |Tr[E_i]\rangle$*

Oracle 7.2. ***for preparing Ei*** *: A unitary O such that for any $i \in [m]$, $O |i\rangle \langle i| \otimes |0\rangle \langle 0| O^\dagger = |i\rangle \langle i| \otimes |\psi_i\rangle \langle \psi_i|$ where $|\psi_i\rangle \langle \psi_i|$ is any purification of $E_i/Tr(E_i)$*

If the measurement operators are of the form, $E_i = V_i P_i V_i^\dagger$ and $P_i = \sum_{i=1}^{r_i} |i\rangle \langle i|$, then the following purification of $E_i/\mathrm{Tr}(E_i)$ can be done by preparing the state $\frac{1}{\sqrt{r_i}} \sum_{i=1}^{r_i} |i\rangle |i\rangle$ in r_i time and then applying $V_i \otimes I$ in time poly(log(n)).

$$|\psi_i\rangle = \frac{1}{\sqrt{r_i}} \sum_{i=1}^{r_i} (V_i \otimes I) |i\rangle |i\rangle$$

7.3. Algorithm

The MMW algorithm to address this specific question can be briefly outlined in the following way for a particular case of $\mathrm{Tr}(\rho E_i) \leq b_i$.

Algorithm 3: Algorithm for Learnability of quantum States [4] [7]

Start with $\rho_0 = I_n$, and iterate the following loop for $T = \frac{16\log(n)}{\epsilon^2}$ times ;
for $t = 1, 2, \cdots, T$ **do**
  1. Gibbs Sampling : Prepare $O(m^{\frac{1}{2}})$ copies of ρ_t
  2. Search for a violated constraint i such that $\mathrm{Tr}(\rho_t E_i) > b_i + \epsilon/2$.
  if $i_{(t)}$ *is found* **then**
    Follow step 3
  else
    Claim no violation and output ρ_t
  end
  3. Define $\rho_{(t+1)} = \exp[\log(\rho_t) - \epsilon^2 E_{i(t)}]/\mathrm{Tr}(...)$
end
;
Claim that $S_0 = \varnothing$ and terminate;

So, starting with a maximally mixed state, we are basically searching for a violation of those constraints mentioned earlier. If no such violation is found at t^{th} iteration then ρ_t is our answer, the state with thhe same statistics as ρ (upto an error). If not, then in the $(t+1)^{th}$ step we add that violated constraint in the definition of the Hamiltonian as a penalty. This algorithm converges within the iterations T for the following reason which we describe in the next section.

7.4. Why it works ?

There are many ways to look at this question of why this algorithm converges within the mentioned steps. We will approach the answer in terms of the relative entropy. From, Peierls-Bogoliubov inequality, we obtain the following in each step of the iteration of the algorithm,

$$S(\rho||\rho_t) - S(\rho||\rho_{t-1}) \leq -\frac{\epsilon}{8}\mathrm{Tr}((\rho - \rho_t)E_{i(t)})$$

If the Grover search in our algorithm never fails i.e. if at every step a violated constraint is found, then at each iteration we have,

$$\mathrm{Tr}((\rho - \rho_t)E_{i(t)}) \leq -\frac{\epsilon}{2}$$

But as we started with a maximally mixed state, the maximum relative entropy in our case can be

$$S(\rho||\rho_0) \leq \log(n)$$

If the algorithm does not converge within $T = \frac{16\log(n)}{\epsilon^2}$, then after T^{th} iteration, the relative entropy becomes negative which is not physically possible. So, the Grover search must fail at some $t < T$ implying ρ_t is our feasible solution.

7.5. Complexity

The Gibbs sampling in this case is done in $O(n^{\frac{1}{2}})$ time with phase estimation and amplitude amplification [8] (and $O(m^{\frac{1}{2}})$ copies of ρ), giving the worst case complexity as $O(m^{\frac{1}{2}}n^{\frac{1}{2}})$. This can be further improved upon use of faster quantum OR bound to $O((m+n)^{\frac{1}{2}})$.

In conclusion to the algorithm considered here, we can find λ_is using at most poly(log m, log n,r, ϵ^{-1}) copies of ρ and at most $\sqrt{m}$ poly(log m, log n,r, ϵ^{-1}) quantum gates and queries to Oracle 7.1 and Oracle 7.2.

8. Quantum SDP in the small scale Quantum computer

In the previous sections, we realized the essential part of the quantum SDP algorithms are the for loop that should calculate the expectation value of A_i operators in the specific way: $c_{i,t} = \mathrm{tr}(\frac{A_i e^{\sum_i \lambda_{i,t} A_i}}{Z})$. This loop should be executed with quantum computer and then the Lagrange multiplier needs to be calculated and fed to the next iteration of the loop. But this calculation can be done by classical computer. so we can divide the task to one quantum computer and one classical computer. Unfortunately we don't have access to full quantum computers as of today, but what if we can test the quantum algorithm on the Noisy Intermediate-Scale Quantum (NISQ)

chips? For example, we consider the Google processor with 72 qubits and a model Hamiltonian like the following:

$$A_i = \delta_i \sum_{j=1}^{72} n_j + U_i \sum_{j=1}^{72} n_j(n_j - 1) + g_i \sum_{l \sim k} (a_l^\dagger a_k + a_k^\dagger a_l) + f_i \sum_{j=1}^{72} (a_j e^{i\phi_i} + a_j^\dagger e^{-i\phi_i}) \quad (4)$$

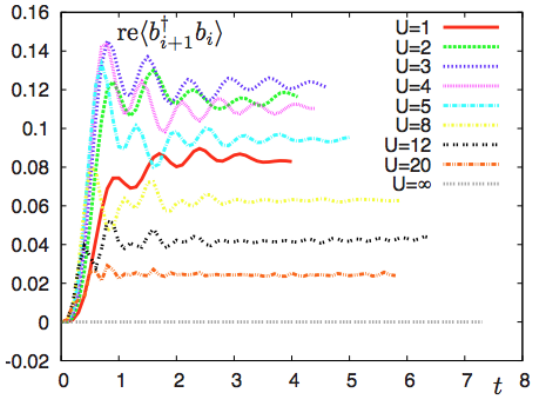
where δ and f are parameters which can be varied in the range of $[-30, 30]$MHZ and U is a constant in the range of 200MHZ [9]. The idea is that if we consider the superposition of the operators as a new Hamiltonian:

$$H = \sum_i \lambda_i A_i \quad (5)$$

then the ergodic evolution provides the following approximation:

$$\langle \psi_t | A_i | \psi_t \rangle \sim \mathrm{tr}(\frac{A_i e^{-\beta H}}{Z}) \quad (6)$$

which is exactly what we need to calculate in the SDP operations. The simulations show that the results of quantum algorithm can be ready after $0.5\mu s$ which can beat the time complexity of classical algorithm. The following graph shows the real part of correlations to neighbors $\langle a_{i+1}^\dagger a_i \rangle$ as a function of time for different values of U. As we see, the result becomes stable after the certain time scale [10]



9. Beyond SDP and Convex optimization

SDP problems are a subclass of a more general problem which is called convex optimization. The problem can be defined as follows:

$$\min_{x \in K} f(x), \text{ s.t.} K \in R^n, \text{be a convex set and } f : K \to R \text{ is a convex function} \quad (7)$$

and the convex function based on the definition follows from the following rule: $f(\alpha x + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha)f(x_2)$. It is easy to show that the above problem is equivalent to the following problem:

$$\min x' \in R, x \in K x' \text{s.t.} f(x) \leq x' \leq M \tag{8}$$

Therefore the task of optimizing the linear function $\min_{x \in K} \vec{c} \cdot \vec{x}$ is equivalent to finding the answer of the general convex optimization problem. As we can see that the target function of convex optimization and SDP are the same. The only difference lies on the extra bound for domain of feasibility. A natural question that can arise is that how quantum computer can speed up the general convex optimizations problems? The recent paper [5] explored this question and has shown that there exist at least one quantum algorithm that speeds up the oracle performance. In general, for convex optimization problem we can define two oracles: A membership oracle, O_K which checks if a selected point x is inside the feasible set and also the evaluation oracle, O_f which calculates the value of $f(x)$ for a given point x. Simple calculation shows that the computational complexity of these oracles is order of $O(n)$. Similarly we can define these two oracles in the quantum version of the algorithms by: $O_K|x, 0\rangle = [x, \delta[x \in K]\rangle$ where δ is the Dirac function and $O_f|x, 0\rangle = |x, f(x)\rangle$.

The theorem in [5] proves that: There exists a convex body $K \in R^n$, a convex function f on K, and a precision $\epsilon > 0$, such that a quantum algorithm needs at least $\Omega(\sqrt{n})$ queries to a membership oracle for K and $\Omega(\frac{\sqrt{n}}{\log n})$ queries to an evaluation oracle for f to output a point x satisfying

$$f(x) \leq \min_{x \in K} f(x) + \epsilon \tag{9}$$

with high success probability. This research direction is new and we hope we can see faster quantum algorithm proposals in near future that enhance the performance of convex optimization problems.

10. Conclusion

In this project-work, we have studied the SDP solver algorithms in deltail and understood the technicalities and importance of moving to the quantum regime in order to have a speed up over their classical counterparts. We have explored in detail one application of this optimization algorithm in efiiciently learning the state of a given quantum system. It was interesting to see the recent efforts towards physically implementing these models. We started looking beyond the SDP algorithms to a more general class of convex optimization. But due to shortage of time, we could not delve into detailed discussion of this topic. Meanwhile, we found the following open problems in SDP solvers to be quite interesting and would be happy to explore those in future.

11. Open Problems

Following are the open problems from the papers which we found to be really important and interesting.

- The initial quantum SDP algorithm [2] had a strong functionality with respect to error parameters. The main concern can be whether can we improve the parameters (in terms of R, r, δ).

  Bad: Many interesting SDP problems(e.g Goesman-Williamson) $Rr/\delta = O(n)$
  Good: Machine Learning and Compressed Sensing problems:$Rr/\delta = O(1)$

- Is it possible to find more efficient algorithm and move towards the theoretical lower limit on the complexity of algorithms in terms of n, m and s?

- Is it possible to find an algorithm with superpoly speed-ups?

- How robust are the algorithms to error and noises?

- In these algorithms quantum computers are only used for the preparation of Gibbs states. Is it possible to implement the algorithm by some small sized quantum computer with no error correction ?

12. References

[1] Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In Proceedings of the Fifty-sixth Annual IEEE Symposium on Foundations of Computer Science, pages 10491065. IEEE, 2015

[2] Fernando G. S. L. Brandao and Krysta Svore. Quantum speed-ups for semidefinite program- ming. In 58th Annual IEEE Symposium on Foundations of Computer Science. IEEE, 2017.

[3] Joran van Apeldoorn, Andras Gilyen, Sander Gribling, and Ronald de Wolf. Quantum SDP- solvers: Better upper and lower bounds. In 58th Annual IEEE Symposium on Foundations of Computer Science. IEEE, 2017.

[4] Fernando G. S. L. Brandao, Amir Kalev, Tongyang Li, Cedric Yen-Yu Lin, Krysta M. Svore, and Xiaodi Wu. Quantum SDP Solvers: Large Speed-ups, Optimality, and Applications to Quantum Learning. arXiv:1710.02581v2, 2018.

[5] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite pro- grams. In Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing, pages 227236. ACM, 2007.

[6] Shouvanik Chakrabarti, Andrew M. Childs, Tongyang Li, Xiaodi Wu. Quantum algorithms and lower bounds for convex optimization. arXiv:1809.01731v2

[7] Quantum speed-ups for semidefinite programming by Fernando Brando (https://www.youtube.com/watch?v=ZgxkmBvwfLA&t=1882s)

[8] David Poulin, Pawel Wocjan. Preparing ground states of quantum many-body systems on a quantum computer. Phys.Rev.Lett.102:130503,2009

[9] The complexity of antiferromagnetic interactions and 2D lattices, Stephen Piddock, and Ashley Montanaro. arXiv:1506.04014v2

[10] Relaxation and thermalization in the one-dimensional Bose-Hubbard model: A case study for the interaction quantum quench from the atomic limit, S. Sorg, L. Vidmar, L. Pollet, and F. Heidrich-Meisner. arXiv:1405.5404v3