CMSC 657, Introduction to Quantum Information Processing
Fall 2018, University of Maryland

Lecture on November 13th, 2018

Prepared by: *Shouvanik Chakrabarti, Xiaodi Wu*

Quantum algorithms for solving linear systems of equations

1 Introduction

In this course we have seen a variety of quantum algorithms that are more efficient than their classical counterparts. Some of these have been solutions to toy problems, such as the Deutsch-Josza and Bernstein-Vazirani problems. We have also seen some quantum speedups in problems of independent interest, namely the integer factorization problem (via Shor's algorithm) and the unstructured search problem (via Grover's algorithm). This lecture will focus on another such problem, that of solving linear systems of equations.

A linear system of equations with N variables, and M constraints is defined as follows: Given an M-dimensional vector $b \in \mathbb{R}^M$ and an $M \times N$ matrix A, the task is to find a vector $x \in \mathbb{R}^n$ such that $Ax = b$. We shall focus on the case where $M = N$ and there are as many constraints or variables. In this case, if A is invertible (full rank), the system of equations has a unique solution.

Linear systems of equations are ubiquitious throughout science and engineering. The number of variables (N) in many important applications can be very large, requiring massive amounts of storage and processing power to solve. Classical algorithms for such systems include Gaussian Elimination (which requires $O(N^3)$ operations), and the Coppersmith-Winograd algorithm ($O(N^{2.376})$). Furthermore, in many classical applications the linear systems are *sparse*, ie. any row of A has at most s non-zero elements, for some constant s. Approximating the solution to a sparse system of linear equations requires $\tilde{O}(N)$ operations (where $\tilde{O}(f(N))$ is used as shorthand for $O(f(N)\,\mathrm{poly}(\log(f(N)))))$, via conjugate gradient descent.

Consider now the case where the vector $b \in \mathbb{R}^n$ is encoded as a quantum state $|b\rangle \in \mathbb{C}^n$, such that $|b\rangle = \sum b_i|i\rangle$. Then the solution to the system of equations would be encoded in a state $|x\rangle$ such that $A|x\rangle = |b\rangle$. Harrow, Hassidim and Lloyd [2], demonstrated an algorithm that given $|b\rangle$, and oracle access to a sparse A, finds $|x\rangle$ using only $\tilde{O}(\log(N))$ queries.

Note that this is not a speedup over the classical algorithms in the strictest sense, since the classical algorithm finds all the coefficients x_i for x while the quantum state $|x\rangle = \sum_i x_i|i\rangle$ does not directly make this information available. Obtaining the coefficients would require a measurement of $|x\rangle$, and since any measurement destroys the state, the algorithm must be repeated $\Omega(N)$ times to obtain multiple copies of the final state in order to find all the coefficients. However, in many applications we are more interested in some global property of the vector x rather than the coefficients x_i themselves. Such a property can often be easily extracted by a measurement of the state $|x\rangle$.

Example: Consider a stochastic process defined by $x^t = Ax^{t-1} + b$. The steady state of such a process is given by $|x\rangle = (I - A)^{-1}b$. Thus in order to check if two stochastic processes (A_1, b_1) and (A_2, b_2) have the same stable state we can construct the states $(I - A_1)^{-1}|b_1\rangle$ and $(I - A_2)^{-1}|b_2\rangle$ and perform the $SWAP$ test to test their similarity.

The remainder of the note is devoted to a sketch of the HHL algorithm.

2 Preliminaries

In order to apply the HHL algorithm we must have access to a procedure that encodes the coefficients b_i in a quantum state $|b\rangle$. $|b\rangle$ may be a product of some other quantum computation, or be a standard state preparation operation given the coefficients b_i. For the rest of this note we assume the existence of an efficient unitary B that when applied to an initial state $|initial\rangle$ produces the state $|b\rangle$. Denote the gate complexity of B by t_B.

We assume that A is an s-sparse matrix whose coefficients are efficiently computable, ie. given any row i and a column j, we can compute A_{ij} in time $O(s)$. We also assume oracular access to the coefficients via an oracle $U_A|i\rangle|j\rangle|k\rangle = |i\rangle|j\rangle|k \oplus A_{ij}\rangle$.

We can also assume without loss of generality that A is Hermitian. If A is not Hermitian, define

$$A' = \begin{bmatrix} 0 & A \\ A' & 0 \end{bmatrix}, b' = \begin{bmatrix} 0 \\ b \end{bmatrix}, x' = \begin{bmatrix} x \\ 0 \end{bmatrix} \tag{1}$$

Then A' is Hermitian, and $A'|x'\rangle = |b'\rangle$. Furthermore, any property of x can be computed given x'. Thus in the rest of the note we focus on Hermitian A.

Hamiltonian Simulation: A very common task in quantum computing is to simulate the unitary evolution $U = e^{iHt}$ generated by a Hermitian matrix H(which we call the Hamiltonian). We shall use the following fact about Hamiltonian simulation in our analysis. (Refer to [1] for proof)

Theorem 2.1. *Given unitary oracle access to an s-sparse Hermitian matrix A, we can simulate the unitary e^{iHt} with gate complexity $\tilde{O}(\log(N)s^2t)$.*

3 Sketch of algorithm

Let $A = \lambda_i|u_i\rangle\langle u_i|$ be the spectral decomposition of A. The task of the algorithm is to compute $|x\rangle = A^{-1}|b\rangle$ given $|b\rangle$, thus we wish to efficiently simulate the operator A^{-1} using a quantum circuit. Notice that in the basis $\{|u_i\rangle\}$, the inverse matrix has the simple representation $A^{-1} = \frac{1}{\lambda_i}|u_i\rangle\langle u_i|$. Thus if we can implement an operation that maps $|u_i\rangle \to \frac{1}{\lambda_i}|u_i\rangle$ for every eigenvector $|u_i\rangle$, we shall have implemented A^{-1} by linearity.

Broadly speaking, our quantum circuit will proceed as follows:

$$\begin{aligned}
|u_i\rangle &\to |\lambda_i\rangle|u_i\rangle && (2)\\
&\to |u_i\rangle\left(\frac{C}{\lambda_i}|success\rangle + \sqrt{1 - \frac{C^2}{\lambda_i^2}}|failure\rangle\right) && (3)\\
&\to \frac{|u_i\rangle|success\rangle}{\lambda_i} && (4)\\
&\to \frac{|u_i\rangle}{\lambda_i} && (5)
\end{aligned}$$

Thus the first step involves computing the eigenvalues of a Hermitian operator given an eigenvector. The second step involves adjoining an ancilla state and applying a 2-block unitary, which can be efficiently implemented. Finally we measure the ancilla, where obtaining the result *success* would perform the necessary operation. In order to boost the probability of measuring well, we perform *amplitude amplification* on the state before measurement.

The above steps are outlined in more detail in the following sections. In the rest of the note we make the simplifying assumption that all the singular values of A lie between $\frac{1}{\kappa}$ and 1. This corresponds to the condition number of A being $\leq \kappa$ and thus $\frac{1}{\kappa} \leq \|A\|_\infty \leq 1$.

3.1 Eigenvalue estimation

We have already seen an algorithm that estimates an eigenvalue of a unitary operator given the corresponding eigenvector. In particular, given a unitary U and eigenvector u_j such that $U|u_j\rangle = e^{\frac{2\pi i\lambda_j}{T}}|u_j\rangle$, we perform the operations

$$(\sum_{t=0}^{T-1}|t\rangle\langle t|\otimes U^t)\sum_{t=0}^{N}\frac{1}{\sqrt{T}}|t\rangle|u_j\rangle=\frac{1}{\sqrt{T}}\sum_{t=0}^{T-1}e^{\frac{2\pi i\lambda_j}{T}}|t\rangle|u_j\rangle \tag{6}$$

$$QFT_T^{-1}\frac{1}{\sqrt{T}}\sum_{t=0}^{T-1}e^{\frac{2\pi i\lambda_j}{T}}|t\rangle|u_j\rangle=\frac{1}{T}\sum_{k=0}^{T-1}\left(\sum_{t=0}^{T-1}e^{\frac{2\pi i(\lambda_j-k)}{T}}\right)|k\rangle|u_j\rangle \tag{7}$$

After this procedure, we are left with the state $\alpha_{k|j}|k\rangle|u_j\rangle$ where

$$\alpha_{k|j}=\frac{1-e^{\frac{2\pi i(\lambda_j-k)T}{T}}}{T\left(1-e^{\frac{2\pi i(\lambda_j-k)}{T}}\right)}$$

Define, $\delta=\frac{\lambda_j-k}{T}$. Thus,

$$|\alpha_{k|j}|^2=\frac{\sin^2(\pi\delta T)}{T^2 sin^2(\pi\delta)}=O(\frac{1}{T^2\delta^2}) \tag{8}$$

In eigenvalue estimation, the first register is measured at this point and the eigenvalue is reported as $\tilde{\lambda}=\tilde{k}/T$ where $\tilde{k}$ is the result of the measurement.

We extend this procedure to find the eigenvalues of a Hermitian A, by using Hamiltonian simulation to implement the unitaries $U^t=e^{\frac{2\pi iAt}{T}}$. Clearly, if $A|u_j\rangle=\lambda_j|u_j\rangle$, then $U|u_j\rangle=e^{\frac{2\pi i\lambda_j}{T}}|u_j\rangle$. Thus eigenvalue estimation can be now be used with U in order to find the eigenvalues of A. Since we do not want to report the eigenvalues we do not measure the first register and relabel the ket $|k\rangle$ with $|\tilde{\lambda}_{k|j}\rangle$ to obtain

$$\sum_{k=0}^{T-1}\alpha_{k|j}|\tilde{\lambda}_{k|j}\rangle|u_j\rangle \tag{9}$$

Applying this procedure to $|b\rangle$ we obtain by linearity, the state $\sum_j\sum_{k=0}^{T-1}\beta_j\alpha_{k|j}|\tilde{\lambda}_{k|j}\rangle|u_j\rangle$ is obtained. For each j it is evident that the concentration $\tilde{\lambda}_{k|j}$ are concentrated around the true eigenvalue λ_j. In latter sections, we assume for simplicity that we obtain the correct eigenvalue and the state in question is simply

$$\sum_j\beta_j|\lambda_j\rangle|u_j\rangle \tag{10}$$

The complexity of performing this eigenvalue estimation operation is given by the complexity of simulating U upto T times. From the known result Theorem 2.1 on Hamiltonian simulation, this complexity is given by $O(\log(n)s^2T)$. Later, we will choose T appropriately to bound the error in the matrix inversion.

3.2 Rotation conditioned on the eigenvalue

Given the state $\sum_j\beta_j|\lambda_j\rangle|u_j\rangle$ we perform a conditional rotation based on the eigenvalue in order to introduce a phase of $\frac{1}{\lambda_j}$ associated with each $|u_j\rangle$. Explicitly, we adjoin an ancilla in the state $|failure\rangle$ and perform a conditional rotation,

$$\sum_j\beta_j|\lambda_j\rangle|u_j\rangle\to\sum_j\beta_j|\lambda_j\rangle|u_j\rangle\left(\frac{C}{\lambda_j}|success\rangle+\sqrt{1-\frac{C^2}{\lambda_j^2}}|failure\rangle\right) \tag{11}$$

For this rotation to be a unitary we must choose C such that $C/\lambda_j\le 1$ for all j, thus we choose $C=\frac{1}{\kappa}$. We also uncompute the phase estimation procedure along with any garbage produced therein, to obtain simply

$$|\psi\rangle=\sum_j\beta_j|u_j\rangle\left(\frac{C}{\lambda_j}|success\rangle+\sqrt{1-\frac{C^2}{\lambda_j^2}}|failure\rangle\right) \tag{12}$$

3.3 Amplitude Amplification

Measuring the state ψ and obtaining the result $|success\rangle$ would mean that the matrix has been successfully inverted upto a normalization. Suppose measuring $|success\rangle$ has probability p. Then classically we could repeat the experiment $O(p)$ times to have a high probability of measuring $|well\rangle$ at least once. However, quantumly we can do better by applying amplitude amplification, which is also the idea behind Grover's algorithm.

Let the algorithm begin with an initial state $|initial\rangle$. Let U_{inv} be the composite operation of computing the eigenvalue, performing the rotation and finally uncomputing the eigenvalue. Thus $|\psi\rangle = U_{inv}B|initial\rangle$. Our goal is to boost the amplitude of the $|success\rangle$ states within $|\psi\rangle$. Consider the 2-dimensional subspace spanned by $|G\rangle = \sum_j \beta_j |u_j > |success\rangle$, and $|B\rangle = \sum_j \beta_j |u_j\rangle |failure\rangle$. Clearly, $|\psi\rangle$ lies in this subspace and the initial angle between $|\psi\rangle$ and B is $\theta = \sin^{-1}(\sqrt{p})$. With the same geometric idea as Grover's algorithm, it is clear to see that a rotation around $|B\rangle$ followed by a rotation around $|\psi\rangle$ shifts the state by 2θ towards ϕ. The composition of these two rotations forms a Grover iterate, which needs to be applied only $O(\frac{1}{\sqrt{p}})$ times in order to obtain a high probability of measuring $|success\rangle$ on the third register.

The rotations that make up the Grover iterate can be written as $R_B = I - |B\rangle\langle B|$ and $R_\psi = I - |\psi\rangle\langle\psi|$. In general, these cannot be implemented without knowledge of $|B\rangle$ and $|\psi\rangle$. However, for vectors in the space spanned by $|B\rangle$ and $|G\rangle$, R_B can be implemented with the single bit qubit operation

$$R_{succ} = I - |success\rangle\langle success|$$

. It is also easy to verify that $R_\psi = B^\dagger U_{inv}^\dagger R_{init} U_{inv} B$, where

$$R_{init} = I_|initial\rangle\langle initial|$$

Thus we apply the unitary $(R_\psi = B^\dagger U_{inv}^\dagger R_{init} U_{inv} B)^k (U_{inv}B)$ where $k = O(1/\sqrt{p})$.

3.4 Final Runtime

From the above analysis, the total complexity of the HHL algorithm is $O(\log(N) s^2 T \sqrt{1/p})$. To get a final runtime we must bound T and $1/p$. First we notice,

$$p = \sum_j \frac{\beta_j^2 C^2}{\lambda_j^2} \geq \frac{C^2}{\lambda_j^2} \geq C^2 \geq \frac{1}{\kappa^2} \tag{13}$$

Notice that eigenvalue estimation with parameter T gives us error $\leq \frac{1}{T}$ with high probability. Suppose that the operation we simulate is $\tilde{A}$. If we require $\left\|\tilde{A} - A^{-1}\right\|_\infty \leq \epsilon \left\|A^{-1}\right\|_\infty$. Thus we require a relative error $\leq \epsilon$ on each $1/\lambda_j$ which translates into a relative error of ϵ on each λ_j. Thus we must choose T such that $\frac{1/T}{\lambda_j} \leq \epsilon$. Thus $T = \frac{\kappa}{\epsilon}$ suffices, and we have a total runtime of $O(\frac{\log(N)s^2\kappa^2}{\epsilon})$. Thus the HHL algorithm has a worse runtime than conjugate gradient descent in terms of the error and the condition number, but is exponentially better in terms of the dimension of the system.

References

[1] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders, *Efficient quantum algorithms for simulating sparse hamiltonians*, Communications in Mathematical Physics **270** (2006), no. 2, 359371.

[2] A. W. Harrow, A. Hassidim, and S. Lloyd, *Quantum Algorithm for Linear Systems of Equations*, Physical Review Letters **103** (2009), no. 15, 150502, 0811.3171.