CS 880: Quantum Information Processing 9/30/2010

Lecture 12: Order Finding

Instructor: Dieter van Melkebeek Scribe: Kenneth Rudinger

We continue our discussion of quantum algorithms based on eigenvalue estimation. We finish up the topic of solving sparse well-conditioned systems of linear equations over the reals. Additionally, we examine the topic of order finding and begin the development of an efficient order finding algorithm.

1 "Solving" Linear Equations

To briefly recap from last lecture, we wish to solve a system of linear equations $A\mathbf{x} = \mathbf{b}$, where A is an $N \times N$ invertible hermitian matrix. We are given A and $|\mathbf{b}\rangle$ (normalized); it is our goal to find $|\mathbf{x}\rangle$ (renormalized, as we don't require A to be unitary). Note that this is *not* quite the same as solving for $\mathbf{x}$, because we are not actually solving for all components of $\mathbf{x}$; rather we hope to get as an output the normalized state $|\mathbf{x}\rangle$. The quantum algorithm we will develop will run in $\widetilde{O}((\log N)s^2\kappa^2\frac{1}{\epsilon})$. s is the sparseness of the matrix (the maximum number of non-zero elements in any row of A), ϵ is the allowed error ($|| \, |\widetilde{\mathbf{x}}\rangle - |\mathbf{x}\rangle \, || < \epsilon$, where $|\widetilde{\mathbf{x}}\rangle$ is the actual output state) and $||A||^{-1} \leq \kappa$. We also have the requirement that $\Omega(1) \leq ||A|| \leq 1$.

Because A is hermitian, we know we can decompose $|\mathbf{b}\rangle$ into a linear combination of the eigenvectors of A:

$$|\mathbf{b}\rangle = \sum_j \beta_j \, |\varphi_j\rangle \tag{1}$$

where $|\varphi_j\rangle$ are the eigenvectors of A, with eigenvalues λ_j. It is this state we wish to transform into $|\mathbf{x}\rangle$, as we know $|\mathbf{x}\rangle$ can be decomposed in the following manner (up to normalization):

$$|\mathbf{x}\rangle = \sum_j \beta_j \lambda_j^{-1} \, |\varphi_j\rangle \tag{2}$$

The proof of (2) is straightforward: $A \, |\mathbf{x}\rangle = |\mathbf{b}\rangle = \sum_j \beta_j \, |\varphi_j\rangle$; applying A^{-1} yields $A^{-1}A \, |\mathbf{x}\rangle = |\mathbf{x}\rangle = \sum_j \beta_j A^{-1} \, |\varphi_j\rangle = \sum_j \beta_j \lambda_j^{-1} \, |\varphi_j\rangle$. Step 1 in creating $|\mathbf{x}\rangle$ is performing the transformation $|\mathbf{b}\rangle \, |\mathbf{0}\rangle \rightarrow \sum_j \beta_j \, |\varphi_j\rangle \left|\widetilde{\lambda_j}\right\rangle$ using eigenvalue estimation. Step 2 will be transforming this new state into $|\mathbf{x}\rangle$.

1.1 Step 1

To perform the first step of this algorithm, we need a unitary operator U with the same eigenvectors as A, and with eigenvalues in close relation to the eigenvalues of A. If we choose $U = e^{iA}$, these conditions are satisfied, as $U \, |\varphi_j\rangle = e^{i\lambda_j} \, |\varphi_j\rangle$. (The process of matrix exponentiation is discussed in Lecture 11.)

Because $||A|| \leq 1$, we know that each eigenvalue λ_j satisfies $|\lambda_j| \leq 1$. Therefore, we know there is a one-to-one correspondence between λ_j and $e^{i\lambda_j} = e^{2\pi i\omega_j}$ where $|\omega_j| \leq \frac{1}{2}$. (In fact, this even

holds if we relaxe the condition $||A|| \leq 1$ to $||A|| < \pi$.) We can, for a given eigenvector of U $|\varphi_j\rangle$, obtain $|\widetilde{\omega_j}\rangle$ through eigenvalue estimation on U. We are then able to obtain the state $\left|\widetilde{\lambda_j}\right\rangle$, because of the one-to-one correspondence between λ_j and ω_j.

If A is sparse, we can efficiently compute U with fixed error in a time of approximately $\widetilde{O}(\log(N)s^2)$. (This computation will be discussed at greater length in the upcoming lecture on random walks.) Once we have our hands on U, we can run eigenvalue approximation on $\sum_j \beta_j |\varphi_j\rangle |\mathbf{0}\rangle$ to get $\sum_j \beta_j |\varphi_j\rangle \left|\widetilde{\lambda_j}\right\rangle$ (which, we recall, we will not actually observe). $\left|\widetilde{\lambda_j}\right\rangle$ will be concentrated near $|\lambda_j\rangle$, but they will not be exactly equal. In order to realize our overall error of ϵ, we must ensure that the relative error for each $\left|\widetilde{\lambda_j}\right\rangle$ is at most ϵ. If we apply U k times in the eigenvalue estimation step, then absolute error is at most k^{-1} with high probability. To ensure that the relative error for all j is at most ϵ with high probability, we want $k^{-1}/|\lambda_j| \leq \epsilon$, i.e., $k \geq 1/(\epsilon|\lambda_j|)$ for all j. Because of our condition on κ, we see that this inequality is satisfied if $k \geq \frac{\kappa}{\epsilon}$. (Recall that $||A^{-1}|| \leq \kappa$ means that $|\lambda_j^{-1}| \leq \kappa$.)

This is the first step in our algorithm. The complexity cost is the cost of eigenvalue estimation, so we see that we have a complexity of $O(\frac{\kappa}{\epsilon}(\log(N)s^2)$. The cost of running U once is $O(\log(N)s^2)$, and we do it $\frac{\kappa}{\epsilon}$ times. As we are now assured that each $\left|\widetilde{\lambda_j}\right\rangle$ is sufficiently concentrated near each respective $|\lambda_j\rangle$ for our total error to be less that ϵ, we will simplify the rest of our analysis by assuming $\left|\widetilde{\lambda_j}\right\rangle = |\lambda_j\rangle$.

1.2 Step 2

Next we discuss the second step of our algorithm. We now have the state $\sum_j \beta_j |\varphi_j\rangle |\lambda_j\rangle$. We cannot simply extract $|\mathbf{x}\rangle$ by multiplying each term in the sum by λ_j^{-1}, as this would not be a unitary transformation. However, if we attach a single $|0\rangle$ qubit to our state, we can perform a pseudo-non-unitary action on the $|\lambda_j\rangle$ part of the state, absorbing the "non-unitary" part of the transformation into the added $|0\rangle$ state, preserving unitarity. Such a transformation can be achieved with elementary operations, and will have the following action for some C to be determined:

$$|\lambda_j\rangle |0\rangle \to |\lambda_j\rangle \left(\sqrt{1 - (\frac{C}{\lambda_j})^2} |0\rangle + \frac{C}{\lambda_j} |1\rangle\right)$$

For this operation to be unitary, we need that $\frac{C}{|\lambda_j|} \leq 1$ for all j, so we choose the value of C to be $C = \frac{1}{\kappa}$. We now make a *partial observation*, only observing the last qubit, measuring either $|0\rangle$ or $|1\rangle$, and thus collapsing the rest of the (now renormalized) state into the subspace consistent with our observation. If we observe $|1\rangle$, the new state of the system is (modulo normalization) $\sum_j \beta_j \lambda_j^{-1} |\varphi_j\rangle |\lambda_j\rangle$, which is the state $|\mathbf{x}\rangle$ we had wanted to extract from the beginning, with the addition of the $|\lambda_j\rangle$ qubits. To reset these $|\lambda_j\rangle$ states to their initialized state of $|0\rangle$ (and thus independent of j), we simply run eigenvalue estimation in reverse. As $|1\rangle$ is the "good" state, we want to be guaranteed to have a sufficiently high probability of observing it. We see that the probability of this event is

$$\Pr[\text{observing } |1\rangle] = \sum_j |\frac{\beta_j C}{\lambda_j}|^2 \geq \min_j |\frac{C}{\lambda_j}|^2 \geq C^2 = \frac{1}{\kappa^2}$$

Therefore, if we run this algorithm κ^2 times, we will have a good probability of observing $|1\rangle$. This would make our overall runtime $O(\kappa^3(\log(N)s^2\frac{1}{\epsilon})$. However, we can use amplitude amplification (as is done in Grover's algorithm) to decrease runtime, as it is desirable to increase the probability of observing a "good" state (and decrease the probability of observing a "bad" state). We recall that if, after one run, the probability of observing a good state is p, amplitude amplification reduces the number of iterations to have a good probability of success from $O(\frac{1}{p})$ to $O(\frac{1}{\sqrt{p}})$. Therefore, as probability of success (observing $|1\rangle$) after one run is on the order of $\frac{1}{\kappa^2}$, we can save a factor of κ in runtime, yielding a final runtime of $O(\kappa^2(\log(N)s^2\frac{1}{\epsilon})$.

2 Order Finding

We now turn our attention to the problem of order finding. We are given integers M and a such that $M > 0$ and $0 \leq a < M$. Our goal is to find the order of a mod M, that is, find the smallest $r > 0$ such that $a^r = 1 \bmod M$.

Exercise 1. *Prove that r exists if and only if GCD(a,M)=1. (M and a are relatively prime.)*

From now on, we shall assume that $GCD(a, M) = 1$, and will not concern ourselves with the need to check this condition, which can be done in $O(\text{poly-log}(M))$ time.

Classically, the best known algorithms solve this problem in time exponential in the bit length of M. However, we will demonstrate the existence of a quantum algorithm that solves this problem in time polynomial in the bit length of M. Our algorithm will be an application of eigenvalue estimation. However, this problem of order finding can also be solved as a special case of period finding, as was done by Shor (which will be discussed in a future lecture, when we discuss Shor's algorithm for factoring).

In order to find r using eigenvalue estimation, we need a unitary operator with its eigenvalues connected to a in some manner. The following is such an operator:

$$U_a |x\rangle = |ax \mod M\rangle \text{ (if } 0 \leq x \leq M)$$
$$U_a |x\rangle = |x\rangle \text{ (otherwise)}$$

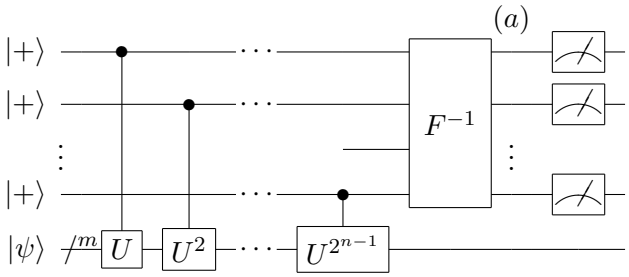
For future reference, we note that we can efficiently compute high powers of U. If we simply applied U as a black box, this would take time linear in the exponent, and recall that we need an exponent of $N = 2^n$ if we want n bits of accuracy in the eigenvalue estimation procedure for U. However, we can compute the powers of U_a more efficiently because $U_a^N |x\rangle = \left|a^N x \bmod M\right\rangle$. Since $a^N \bmod M$ is modular exponentiation, we can compute it with a number of steps polynomial in $\log(M + N)$. Therefore, computing high powers of U_a is relatively easy.

Now, let us note that $U_a^r = I$, because we know that $a^r \bmod M$=1. Therefore, for any eigenvalue λ of U, $\lambda^r = 1$ and thus $\lambda_j = e^{\frac{2\pi ij}{r}}$ for $0 \leq j < r$.

Exercise 2. *Show that the state* $|\varphi_j\rangle = \frac{1}{\sqrt{r}} \sum_{l=0}^{r-1} e^{\frac{-2\pi ijl}{r}} \left|a^l \bmod M\right\rangle$ *is an eigenvector of* U_a *with eigenvalue* $\lambda_j = e^{\frac{-2\pi ijl}{r}}$ *for* $0 \leq j < r$. *Also, find the remaining eigenvectors of* U_a.

Note that $\sum_{j=0}^{r-1} |\varphi_j\rangle = \sqrt{r}\,|1\rangle$. So, if we run eigenvalue estimation on U starting from the state $|1\rangle$, we obtain the state $\frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} |\varphi_j\rangle\,|\widetilde{\omega_j}\rangle$, where $\widetilde{\omega_j} = \frac{\widetilde{j}}{r}$. If we increase our number of bits of

accuracy n, the more concentrated around $\left|\frac{\widetilde{j}}{r}\right\rangle$ our state will be. When we observe this state, we will get a good approximation of ω_j for some j, where j is uniformly distributed in $\{0, 1, ..., r-1\}$. The following is a circuit diagram of this procedure, with $|\psi\rangle = |1\rangle$ (the n bit string equal to 1).



The state at point (a) is the desired state $\frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} |\varphi_j\rangle |\widetilde{\omega_j}\rangle$. Our goal is to now recover r. We will do this using the following two facts.

Fact 1. If $|\widetilde{\omega_j} - \frac{j}{r}| \leq \frac{1}{2M^2}$, then we can efficiently recover $\frac{j}{r}$ in reduced form, i.e., we can obtain j' and r' where $j' = \frac{j}{GCD(j,r)}$ and $r' = \frac{j}{GCD(j,r)}$.

Fact 2. If you pick j_1 and j_2 uniformly from $\{0, 1, ..., r-1\}$, then:

$$Pr[GCD(j_1, j_2) = 1] \geq 1 - \sum_{p \ prime} \frac{1}{p^2} \geq 0.54.$$

As we will explain in the next lecture, if $GCD(j_1, j_2) = 1$, then $r = LCM(r'_1, r'_2)$, where we have run the procedure from Fact 1 both j_1 and j_2 to yield r'_1 and r'_2, respectively.

The proofs of the two facts, along with the rest of the algorithm, will also be provided in the next lecture.