Fast quantum algorithms for Least Squares Regression and Statistic Leverage Scores

Yang Liu* Shengyu Zhang*

Abstract

Least squares regression is the simplest and most widely used technique for solving overdetermined systems of linear equations $Ax = b$, where $A \in \mathbb{R}^{n\times p}$ has full column rank and $b \in \mathbb{R}^n$. Though there is a well known unique solution $x^* \in \mathbb{R}^p$ to minimize the squared error $\|Ax - b\|_2^2$, the best known classical algorithm to find x^* takes time $\Omega(n)$, even for sparse and well-conditioned matrices A, a fairly large class of input instances commonly seen in practice. In this paper, we design an efficient quantum algorithm to generate a quantum state proportional to $|x^*\rangle$. The algorithm takes only $O(\log n)$ time for sparse and well-conditioned A. When the condition number of A is large, a canonical solution is to use regularization. We give efficient quantum algorithms for two regularized regression problems, including ridge regression and δ-truncated SVD, with similar costs and solution approximation.

Given a matrix $A \in \mathbb{R}^{n\times p}$ of rank r with SVD $A = U\Sigma V^T$ where $U \in \mathbb{R}^{n\times r}$, $\Sigma \in \mathbb{R}^{r\times r}$ and $V \in \mathbb{R}^{p\times r}$, the statistical leverage scores of A are the squared row norms of U, defined as $s_i = \|U_i\|_2^2$, for $i = 1, ..., n$. The matrix coherence is the largest statistic leverage score. These quantities play an important role in many machine learning algorithms. The best known classical algorithm to approximate these values runs in time $\Omega(np)$. In this work, we introduce an efficient quantum algorithm to approximate s_i in time $O(\log n)$ when A is sparse and the ratio between A's largest singular value and smallest non-zero singular value is constant. This gives an exponential speedup over the best known classical algorithms. Different than previous examples which are mainly modern algebraic or number theoretic ones, this problem is linear algebraic. It is also different than previous quantum algorithms for solving linear equations and least squares regression, whose outputs compress the p-dimensional solution to a $\log(p)$-qubit quantum state.

Keywords least square regression, statistical leverage score, quantum algorithms

1 Introduction

Quantum algorithms for solving linear systems, and the controversy The past two decades witnessed the development of quantum algorithms [Mos09], and one recent discovery is quantum speedup for solving linear systems $Ax = b$ for sparse and well-conditioned matrices $A \in \mathbb{R}^{n\times p}$. Solving linear systems is a ubiquitous computational task, and sparse and well-conditioned matrices form a fairly large class of inputs frequently arising in many practical applications, especially in recommendation systems where the data set can be very sparse [ZWSP08].

*Department of Computer Science and Engineering and The Institute of Theoretical Computer Science and Communications, The Chinese University of Hong Kong. Email: {yliu, syzhang}@cse.cuhk.edu.hk

The best known classical algorithm for solving linear systems for this class of matrices runs in time $O(\sqrt{\kappa}sn)$ [She94], where κ is the condition number of A (i.e. the ratio between A's largest and smallest singular values), and the sparseness parameter s is the maximum number of non-zero entries in each row of A. Harrow, Hassidim and Lloyd [HHL09] introduced an efficient quantum algorithm, thereafter referred to as HHL algorithm, for the linear system problem, and the algorithm runs in time $O(s^2\kappa^2 \log n)$. The dependence on κ is later improved by Ambainis [Amb12] and the algorithm was used for solving least squares regression (defined next) by Wiebe, Braun and Lloyd [WBL12]. HHL algorithm was also extended in [CJS13] to more general problem specifications.

Though the costs of these quantum algorithms are exponentially smaller than those of the best known classical algorithms, there is a catch that these quantum algorithms do not output the entire solution x^*, but compress $x^* \in \mathbb{R}^n$ (assuming $n = p$) into a $\log n$-qubit quantum state. More precisely, the output is a quantum state $|\overline{x^*}\rangle$ proportional to $\sum_{i=1}^n x_i^*|i\rangle$. This important distinction between outputs of classical and quantum algorithms caused some controversy for these quantum algorithms. After all, one cannot read out the values x_i^* from $|\overline{x^*}\rangle$. Indeed, if outputting all x_i^* is required as classical algorithms, then any quantum algorithm needs $\Omega(n)$ time even for just writing down the answer, thus no exponential speedup is possible.

Despite this drawback, the quantum output $|\overline{x^*}\rangle$ can be potentially useful in certain context where only global information of x^* is needed. For instance, sometimes only the expectation value of some operator associated with x^*, namely $x^{*T}Mx^*$ for some matrix M is needed [HHL09]. Another example is when one desires to compute only the weighted sum $\sum c_i x_i^*$, then SWAP test can be used on $|\overline{c}\rangle = \sum_i \frac{c_i}{\|c\|_2}|i\rangle$ and $|\overline{x^*}\rangle$ to get a good estimate of $\sum c_i x_i^*$ in time $O(\log n)$. As argued in [Amb12], this is impossible for classical algorithms unless P = BQP.

In this paper, we give new quantum algorithms, which also address the controversial issue on two levels. First, we design an efficient quantum algorithm for least squares regression, which runs in time $O(\log n)$ for sparse and well-conditioned A. Same as the one in [WBL12], our quantum algorithm outputs a quantum sketch $|\overline{x^*}\rangle$ only, but our algorithm is simpler, and more efficient with a better dependence on s and κ.

In addition, we consider the case that A is ill-conditioned, or even not full-rank. Classical resolutions for such cases use regularization. We give efficient quantum algorithms for two popular regularized regression problems, including ridge regression and δ-truncated SVD, based on our algorithm for least squares regression.

Second, we also design new efficient quantum algorithms for calculating statistic leverage scores (SLS) and matrix coherence (MC), two quantities playing important roles in many machine learning algorithms [Sar06][DMM08][MD09][BMD09][DMMS11]. Our algorithm has cost $O(\log n)$ for approximately calculating the k-th statistic leverage score s_k for any index $k \in [n]$, exponentially faster than the best known classical algorithms. Repeatedly applying this allows us to approximately calculate all the statistic leverage scores in time $O(n \log n)$ and to calculate matrix coherence in time $O(\sqrt{n} \log n)$, which has a polynomial speedup to their classical counterparts of cost $O(n^2)$ [DMIMW12]. Note that different than all aforementioned quantum algorithm that outputs a quantum sketch only, our algorithms for calculating SLS and MC indeed produce the requested values, same as their classical counterpart algorithms' output. Our algorithms are based on the phase estimation idea as in the HHL algorithm, and the results showcase the usefulness of the HHL algorithm even in the standard computational context without controversial issue any more.

Next we explain our results in more details.

Least Squares Regression Least squares regression (LSR) is the simplest and most widely used technique for solving overdetermined systems. In its most important application – data fitting, it finds a hyperplane through a set of data points while minimizing the sum of squared errors. The formal definition of LSR is as follows. Given an $n \times p$ matrix A ($n \geq p$) together with an n-dimensional vector b, the goal of LSR is to compute a p-dimensional vector

$$x^* = \arg\min_{x \in \mathbb{R}^p} \|Ax - b\|_2^2 . \tag{1}$$

For well-conditioned problems, i.e. those with the condition number of A being small (which in particular implies that A has full column rank), it is well known that Eq.(1) has a unique and closed-form solution

$$x^* = A^+ b, \tag{2}$$

where A^+ is the Moore-Penrose pseudoinverse of A. If one computes x^* naively by first computing A^+ and then the product A^+b, then the cost is $O(\min\{p^2n, n^2p\})$, which is prohibitively slow in the big data era[1]. Therefore, finding fast approximation algorithms which output a vector $\tilde{x} \approx x^*$ is of great interest. Classically, there are known algorithms that output an $\tilde{x}$ with a relative error bound $\|\tilde{x} - x^*\|_2 \leq \epsilon \|x^*\|_2$ for any constant error $0 < \epsilon < 1$, and run in time $\tilde{O}(\text{nnz}(A) + nr)$ [CW13][NN13], where $\text{nnz}(A)$ is the number of non-zero entries in A, r is the rank of A and the $\tilde{O}$ notation hides a logarithmic factor. These algorithms are much faster than the naive ones for the special case of sparse or low rank matrices, but remain linear in size of A for general cases. Given that it is impossible to have classical approximation algorithms to run in time $o(np)$ for general cases, it would be great if there exist much faster quantum algorithms for LSR. Similar to [HHL09], one can only hope to produce a quantum state close to $|\overline{x^*}\rangle$ fast. [WBL12] gives a quantum algorithm which outputs a quantum state close to $|\overline{x^*}\rangle$ in time $O(\log(n+p)s^3\kappa^6)$. Their algorithm is based on the observation that $x^* = A^+b = (A^TA)^{-1}A^Tb$ when A has full column rank, and their main idea is to construct the quantum state $|\overline{x^*}\rangle$ by applying the operator $(A^TA)^{-1}$ to the state $|A^Tb\rangle$. In this paper, we propose another quantum algorithm that outputs a quantum state close to $|\overline{x^*}\rangle$ in time $\tilde{O}(\log(n+p)s\kappa^3)$, where the $\tilde{O}$ notation hides the slower growing functions.

We highlight four advantages of our algorithm compared to [WBL12]. First, our algorithm is much simpler since we directly apply the operator A^+ to the state $|b\rangle$, while they first applied A^T to $|b\rangle$ to get $|A^Tb\rangle$, then prepared $(A^TA)^{-1}$ and applied it to $|A^Tb\rangle$. (Thus, despite solving the same LSR problem as in [WBL12], our algorithm is more similar to the one in [HHL09] than to [WBL12].) Second, the simplicity also leads to a better dependence on s and κ in our algorithm. Third, [WBL12] assumes that A is Hermitian, which is usually not the case for typical machine learning applications[2]. Our algorithm works for non-Hermitian matrices as well, for which we need to work on singular Hermitian matrices A. Fourth, note that $|\overline{x^*}\rangle$ misses one important information of x^*, namely its ℓ_2 norm, which is actually crucially needed when we want to compute $\sum_i c_i x_i$ by SWAP test. Our algorithm also gives a good estimate to $\|x^*\|_2^2$ without introducing much extra running time.

[1] Though theoretically more efficient algorithms for matrix multiplication exist [Sto10], in practice they are seldom used due to the complication in implementing, parallelization and non-robustness. Thus in machine learning algorithms matrix multiplication $A_{m\times n}B_{n\times k}$ is assumed to take time $O(mnk)$. In any case it is just a polynomial saving, in contrast to the exponential gap to the quantum algorithm cost.

[2] Although they mentioned a standard pre-processing technique to deal with the non-Hermitian case, but they seem to have overlooked the fact that after the pre-processing, the new input matrix is not full (column) rank (unless $n = p$, which is hardly the case in machine learning settings).

The more precise description of the performance of our algorithm is stated in the next theorem. As in [HHL09], we assume that the vector b is in a nice form in the sense that each b_i and $\sum_{i_1}^{i_2} |b_i|^2$ can be efficiently computed, which enables us to prepare $|b\rangle$ efficiently [GR02].

Theorem 1. *Let $A \in \mathbb{R}^{n\times p}$ and $b \in \mathbb{R}^n$ be the input of the least squares regression problem and suppose that x^* is its optimal solution. Assume that each row and column of A has at most s non-zero entries, and all the non-zero singular values of A are in the range $[\frac{1}{\kappa}, 1]$. Then there exists a quantum algorithm that, with probability 0.99, outputs a quantum state proportional to $\tilde{x}$ satisfying*

$$\|\tilde{x} - x^*\|_2 \leq \epsilon \cdot \max\{\|x^*\|_2, \|b\|_2\},$$

and outputs a value ℓ satisfying

$$|\ell - \|x^*\|_2^2| \leq \epsilon(\|x^*\|_2^2 + \|b\|_2^2),$$

in time $O(s\kappa^3\epsilon^{-2}(\log(n+p) + poly\log(s,\kappa)))$.

Ridge regression and truncated SVD For ill-conditioned problems, i.e. when the condition number of A is large, the solution given by Eq.(2) becomes very sensitive to errors in A and b. A prevailing solution in practice is to use regularization. Two of the most commonly used regularization methods are ridge regression [GHO99] (a.k.a Tikhonov regularization) and truncated singular value decomposition [Han87].

For ridge regression (RR) problem, we are given an $n \times p$ matrix A, an n-dimensional vector b together with a parameter $\lambda > 0$, and we want to compute

$$x^* = \arg\min_{x\in\mathbb{R}^p} \|Ax - b\|_2^2 + \lambda \|x\|_2^2 . \tag{3}$$

The unique minimizer of Eq.(3) is $x^* = (A^TA+\lambda I_p)^{-1}A^Tb$ [Tik63], which takes $O(np^2+p^3)$ time to compute in the naive way. An alternative solution uses the dual space approach by computing an equivalent expression $x^* = A^T(AA^T+\lambda I_n)^{-1}b$ [SGV98], which takes $O(n^2p+n^3)$ time to compute in the naive way, and is faster than the original one when $p \gg n$. When approximation is allowed, the best known classical algorithm for ridge regression outputs an approximation solution $\tilde{x}$ satisfying $\|\tilde{x} - x^*\|_2 \leq \epsilon \|x^*\|_2$ in time $\tilde{O}(\text{nnz}(A) + n^2r)$ [CLL+15]. This algorithm has an significant speedup over the previous algorithms when A is sparse or of low rank, but still slow for general cases. Based on the algorithm in Theorem 1, we design a quantum algorithm to solve ridge regression problem efficiently (in the sense of generating quantum sketch of the solution).

Theorem 2. *Let $A \in \mathbb{R}^{n\times p}$, $b \in \mathbb{R}^n$ and λ be the input of the ridge regression problem and suppose that x^* is its optimal solution. Assume that each row and column of A has at most s non-zero entries, and all the non-zero singular values of A are in the range $[\frac{1}{\kappa}, 1]$. Then there exists a quantum algorithm that, with probability 0.99, generates a quantum state proportional to $|\tilde{x}\rangle$ satisfying*

$$\|\tilde{x} - x^*\|_2 \leq \epsilon \cdot \max\{\|x^*\|_2, \|b\|_2\},$$

and outputs a value ℓ satisfying

$$|\ell - \|x^*\|_2^2| \leq \epsilon(\|x^*\|_2^2 + \|b\|_2^2),$$

in time $O(s\kappa'^3\epsilon^{-2}(\log(n+p) + poly\log(s,\kappa')))$, for $\kappa' = \frac{\max\{1,\sqrt{\lambda}\}}{\min\{\frac{1}{\kappa},\sqrt{\lambda}\}}$.

Next we discuss truncated singular value decomposition (truncated SVD). In this problem we are given an $n \times p$ matrix A, an n-dimensional vector b together with a parameter $k < \texttt{rank}(A)$, and we want to compute

$$x^* = \arg\min_{x \in \mathbb{R}^p} \|A_k x - b\|_2^2, \tag{4}$$

where A_k is the best rank-k approximation of A obtained through SVD. More specifically, let $A = \sum_i^r \lambda_i u_i v_i^T$ be the SVD of A, where r is the rank of A, λ_i is the i-th largest singular value of A, and $u_i \in \mathbb{R}^n, v_i \in \mathbb{R}^p$ are the corresponding left and right singular vectors for $i = 1, ..., r$. Then it is well known ([EY36]) that A_k is equal to $A_k = \sum_{i=1}^k \lambda_i u_i v_i^T$.

The basic idea of truncated SVD is to impose an additional requirement that the ℓ_2 norm of the solution x^* should be small by removing the large influence from the small non-zero singular values of A. If the number k is chosen properly, the ratio between λ_1 and λ_k is small and then the solution $x^* = A_k^+ b$ to Eq.(4) becomes not sensitive to errors in A and b. A more direct way to remove the influence by the small non-zero singular values of A is to set a gap δ on the singular values and neglect all those singular values smaller than δ. Define the δ-truncated singular value decomposition (δ-TSVD) problem as follows.

Given an $n \times p$ matrix A, an n-dimensional vector b together with a parameter $\delta > 0$ and we want to find

$$x^* = \arg\min_{x \in \mathbb{R}^p} \|A_\delta x - b\|_2^2, \tag{5}$$

where $A_\delta = \sum_{i:\lambda_i \geq \delta} \lambda_i u_i v_i^T$, assuming $A = \sum_i^r \lambda_i u_i v_i^T$ is the SVD of A.

A naive algorithm to solve δ-TSVD needs to first compute the matrix A_δ and then solve the least squares regression problem with the new input A_δ and b in $O(\min\{n^2 p, np^2\})$ time. Our algorithm in Theorem 1 can be also adapted to solve δ-TSVD efficiently (again in the sense of generating quantum sketch of the optimal solution).

Theorem 3. *Let $A \in \mathbb{R}^{n \times p}$, $b \in \mathbb{R}^n$ and δ be the input of the δ-truncated singular value decomposition problem and let x^* be the optimal solution of this problem. Assume that each row and column of A has at most s non-zero entries, and that the largest singular value of A is at most 1. Let $\Lambda_1 = \max\{\lambda_i : \lambda_i < \delta, i \in [n]\}$, $\Lambda_2 = \min\{\lambda_i : \lambda_i \geq \delta, i \in [n]\}$ where λ_i is the i-th largest singular value of A. Let $\Lambda = \Lambda_2 - \Lambda_1$. Then there exists a quantum algorithm that, with probability 0.99, generates a quantum state proportional to $|\tilde{x}\rangle$ satisfying*

$$\|\tilde{x} - x^*\|_2 \leq \epsilon \cdot \max\{\|x^*\|_2, \|b\|_2\},$$

and outputs a value ℓ satisfying

$$|\ell - \|x^*\|_2^2| \leq \epsilon(\|x^*\|_2^2 + \|b\|_2^2),$$

in time $O(s(\log(n+p) + poly\log(s, \kappa))/(\min\{\Lambda, \delta\epsilon\}\delta^2\epsilon))$.

Calculating statistic leverage scores and matrix coherence The definition of statistic leverage scores (SLS) and matrix coherence (MC) are as follows. Though the definition uses A's SVD, which is not necessarily unique, it is not hard to see that each s_i depends on A only, not on any specific SVD decomposition of A.

Definition 1. *Given an $n \times p$ matrix A of rank r with SVD $A = U\Sigma V^T$ where $U \in \mathbb{R}^{n\times r}$, $\Sigma \in \mathbb{R}^{r\times r}$ and $V \in \mathbb{R}^{p\times r}$, the statistic leverage scores of A are defined as $s_i = \|U_i\|_2^2$, $i \in \{1, ..., n\}$, where U_i is the i-th row of U. The matrix coherence of A is defined as $c = \max_{i\in\{1,...,n\}} s_i$, the largest statistic leverage score of A.*

Statistic leverage scores measure the correlation between the singular vectors of a matrix and the standard basis and they are very useful in large-scale data analysis and randomized matrix algorithms [MD09][DMM08]. These quantities have been used in statistical data analysis since a long time ago. Actually they are equal to the diagonal entries of the "hat-matrix" which interprets the influence associated with the data points and so they are widely used to indicate possible outliers in regression diagnostics [HW78][CH$^+$86]. They have also been found useful in many theoretical computer science and machine learning problems. Many random sampling algorithms for matrix problems like least-squares regression [Sar06][DMMS11] and low-rank matrix approximation [Sar06][DMM08][MD09][BMD09] use them as an important indicator to design the sampling distribution which are used to sample the input data matrix.

The related parameter, matrix coherence, has also been of interest recently in problems like Nystrom-based low-rank matrix approximation [TR10] and matrix completion [CR09].

A naive algorithm to compute the statistic leverage scores and matrix coherence first performs SVD or QR decomposition to get an orthogonal basis of A, and then calculates the squared ℓ_2 norm of the rows of the basis matrix to get the statistic leverage scores. This takes $O(np^2)$ time (assuming $n \geq p$), which is extremely slow when n and p are large. The best known classical approximation algorithm for the problem of calculating statistic leverage scores runs in time $O((np + p^3)\log n)$ [DMIMW12], which is much faster than the naive algorithm for the case when $n >> p >> 1$. This algorithm is also the best for calculating matrix coherence for now.

An important difference of calculating SLS/MC than previously mentioned LSR related problems is that each SLS s_i (and the MC c) is a scalar instead of a vector. Thus the previous barrier of outputting a vector does not exist, which makes it possible to design efficient quantum algorithm for complete solution to the problem rather than generating a quantum sketch as before. In this work, we design a fast quantum algorithm to approximate the statistic leverage score s_i for any index $i \in \{1, ..., n\}$ in time $O(\log n)$ when A is sparse and the ratio between A's largest singular value and smallest non-zero singular value is small. And thus we can approximate all the statistic leverage scores in time $O(n \log n)$ by running the algorithm for index $i = 1, ..., n$. And we can approximate the matrix coherence in time $O(\sqrt{n}\log n)$ by using amplitude amplification. More specifically, we have the following theorem and corollary.

Theorem 4. *Let $A \in \mathbb{R}^{n\times p}$, let s_i be the i-th statistic leverage score of A for $i = 1, ..., n$. Assume that each row and column of A has at most s non-zero entries, and all the non-zero singular values of A are in the range $[\frac{1}{\kappa}, 1]$. Then there exists a quantum algorithm that, on any requested $i \in [n]$, returns $\tilde{s}_i$ satisfying $|\tilde{s}_i - s_i| \leq \epsilon$ in time $O(s\kappa\epsilon^{-1}(\log(n + p) + poly\log(s, \kappa)))$.*

Corollary 5. *Let $A \in \mathbb{R}^{n\times p}$, let c be the coherence of A. Assume that each row and column of A has at most s non-zero entries, and all the non-zero singular values of A are in the range $[\frac{1}{\kappa}, 1]$. Then there exists a quantum algorithm that returns $\tilde{c}$ satisfying $|\tilde{c} - c| \leq \epsilon$ in time $O(\sqrt{n} \cdot s\kappa\epsilon^{-1}(\log(n + p) + poly\log(s, \kappa)))$.*

2 Preliminaries

Given a matrix $A \in \mathbb{R}^{n\times p}$ with rank $r \le \min\{n,p\}$, let A_i denote the transpose of the i-th row of A, namely take the i-th row and view it as a column vector $\in \mathbb{R}^p$. Let $\mathrm{nnz}(A)$ denote the number of non-zero entries of A. Let λ_i denote the i-th largest singular value of A, and let $\lambda_{\max}$ denote the largest singular value of A, unless specified otherwise. Let I_r denote the identity matrix of dimension $r \times r$, e_i the unit vector with the i-th coordinate being 1 and all the rest being 0, and 0_n the zero vector of dimension n.

For a rank-r matrix $A \in \mathbb{R}^{n\times p}$, its thin SVD is $A = U\Sigma V^T$ where $U \in \mathbb{R}^{n\times r}$, $V \in \mathbb{R}^{p\times r}$ satisfy $U^TU = V^TV = I_r$, and $\Sigma = \mathrm{diag}(\lambda_1, ..., \lambda_r)$ with the λ_i's being the singular values of A. The Moore-Penrose pseudoinverse of A is defined to be $A^+ = V\Sigma^{-1}U^T$. The full SVD of A is $A = U_F\Sigma_F V_F^T$ where $U_F \in \mathbb{R}^{n\times n}, \Sigma_F \in \mathbb{R}^{n\times p}, V_F \in \mathbb{R}^{p\times p}$ and $U_F^TU_F = U_FU_F^T = I_n, V_F^TV_F = V_FV^T = I_p$. When a matrix $A \in \mathbb{R}^{n\times n}$ is full rank, the thin SVD and full SVD are the same.

Quantum phase estimation [Kit95][CEMM97][BDM99] or quantum eigenvalue estimation allows one to estimate the eigenphase of an eigenvector of a unitary operator. It has been widely used as subroutine in other algorithms. In the Phase Estimation problem, we are given a unitary matrix U by black-boxes of controlled-U, controlled-U^2, controlled-U^{2^2}, $\cdots$, controlled-$U^{2^{t-1}}$ operations, and an eigenvector $|u\rangle$ of U with eigenvalue $e^{2\pi i\varphi}$ with the value of $\varphi \in [0,1)$ unknown. The task is to output an n-bit estimation of φ.

Theorem 6. *There is a quantum algorithm that, on input $|0^t\rangle|u\rangle$ where $t = \log\frac{1}{\epsilon\delta} + O(1)$, outputs $|\tilde{\varphi}\rangle|u\rangle$ in time $O(t)$ using each controlled-U^{2^i} once, and $|\varphi - \varphi'| \le \delta$ with probability at least $1-\epsilon$.*

Two comments are in order. First, if we do not have controlled-U^{2^i}, and need to implement them, then the total time becomes $O(\frac{1}{\delta}\log\frac{1}{\epsilon})$ assuming that implementing controlled-U takes unit time. Second, when the input is $|0^t\rangle|b\rangle$ where $b = \sum_i \beta_i|u_i\rangle$, then the output is $b = \sum_i \beta_i|\tilde{\varphi}_i\rangle|u_i\rangle$ where each $\tilde{\varphi}_i$ approximates φ_i.

The next Amplitude Estimation theorem estimates the success probability of an algorithm.

Theorem 7 ([BHMT00])**.** *Suppose that an algorithm A has success probability $p < 1 - \Omega(1)$, then there exists an algorithm B running A exactly M times to output a number p' satisfying that*

$$|p' - p| \le O\Big(\frac{\sqrt{p}}{M} + \frac{1}{M^2}\Big).$$

Hamiltonian simulation is one of the central tasks in quantum information processing. The task is to implement an operation close to e^{iHt}, where the error is measured by diamond norm. The simulation can be made efficient if the Hamiltonian H is sparse. We say H is s sparse if it contains at most s non-zero entries in each row, and in this case, H can be specified by an oracle that gives the index j of the ℓ-th non-zero entry of any row i, and can also answer ℓ when queried (i,j). The best known result is as follows [BCK15].

Theorem 8. *If a Hamiltonian H acting on m qubits has at most s non-zero entries in each row, then one can simulate e^{iHt} with $O(\tau\log(\tau/\epsilon)/\log\log(\tau/\epsilon))$ queries and $O(\tau(m + \mathit{poly}\log(\tau/\epsilon)))$ additional 2-qubit gates, where $\tau = s\cdot t\cdot\max_{ij}|H_{ij}|$, within error ϵ.*

When we later apply this theorem, $m = \log n$, $t = 1$ and $\max_{ij}|H_{ij}| \le 1$, thus the time complexity is $O(s(\log n + \mathit{polylog}(s,\kappa)))$.

3 Quantum algorithm for LSR

In this section, we present our quantum approximation algorithm QLSR for Least Squares Regression, then analyze its error rate and running time.

Algorithm QLSR
Input: $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$. A is Hermitian with spectral decomposition $A = \sum_{i=1}^{n} \lambda_i |v_i\rangle\langle v_i|$, where all the eigenvalues $\lambda_1, ..., \lambda_n$ satisfy $\frac{1}{\kappa} \leq |\lambda_i| \leq 1$ for $i = 1, ..., r$ for some known value κ and $\lambda_i = 0$ for $i = r+1, ..., n$. Suppose that $b = \sum_{i=1}^{n} \beta_i |v_i\rangle$.
Output: A quantum state proportional to $|\tilde{x}\rangle$ where $\tilde{x} \approx x^* \stackrel{\text{def}}{=} A^+ b$, and a value $\ell \approx \|x^*\|_2^2$.
Algorithm:

1. Prepare the quantum state $|b\rangle = \frac{1}{\|b\|_2} \sum_{i=1}^{n} \beta_i |v_i\rangle$.
2. Perform phase estimation to create the state $\frac{1}{\|b\|_2} \sum_{i=1}^{n} \beta_i |v_i\rangle|\tilde{\lambda_i}\rangle$, where $\tilde{\lambda_i}$ is the estimated value of λ_i satisfying $|\tilde{\lambda_i} - \lambda_i| \leq \delta_{\text{PE}} \stackrel{\text{def}}{=} \frac{\epsilon}{2\kappa}$ for $i = 1, \ldots, n$.
3. Add a qubit $|0\rangle$ to the state and perform a controlled rotation as follows. If $\tilde{\lambda}_i \geq \frac{1}{2\kappa}$, rotate the qubit to $(\frac{1}{2\kappa\tilde{\lambda_i}}|1\rangle + \sqrt{1 - \frac{1}{4\kappa^2\tilde{\lambda_i}^2}}|0\rangle)$; otherwise do nothing. The resulting state is
$$\frac{1}{\|b\|_2} \sum_{i=1}^{r} \beta_i |v_i\rangle|\tilde{\lambda}_i\rangle \left(\frac{1}{2\kappa\tilde{\lambda_i}}|1\rangle + \sqrt{1 - \frac{1}{4\kappa^2\tilde{\lambda_i}^2}}|0\rangle \right) + \frac{1}{\|b\|_2} \sum_{i=r+1}^{n} \beta_i |v_i\rangle|\tilde{\lambda}_i\rangle|0\rangle. \quad (6)$$
4. Use amplitude amplification by repeating the previous steps $O(\kappa^2/\epsilon)$ times.
5. Measure the last qubit.
6. **if** we observe $|1\rangle$,
   (a) The remaining state is proportional to $\sum_{i=1}^{r} \frac{\beta_i}{\tilde{\lambda}_i} |v_i\rangle|\tilde{\lambda}_i\rangle$.
   (b) Reverse the phase estimation process and get the state proportional to $\sum_{i=1}^{r} \frac{\beta_i}{\tilde{\lambda}_i} |v_i\rangle = |\tilde{x}\rangle$ as our output.

   else output 0 as an estimate to $|x^*\rangle$.
7. Use amplitude estimation to get an estimate p' to the probability p of observing $|1\rangle$ when measuring the state in Eq.(6), to precision $\delta = \epsilon/(4\kappa^2)$ and with success probability 0.99. Output $\ell = p' \cdot 4\|b\|_2^2 \kappa^2$.

Without loss of generality, we can assume that $A \in \mathbb{R}^{n\times n}$ with rank r is Hermitian and $b \in \mathbb{R}^n$. See appendix B for discussions on how to deal with the non-Hermitian case. Recall that our goal is to compute $x^* = A^+ b$. Now we will analyze the precision, error probability and the cost. For convenience, we summarize the parameters here: the phase estimation error $\delta_{\text{PE}} = \frac{\epsilon}{2\kappa}$, the Hamiltonian simulation error $\delta_{HS} = O(\delta_{PE}^2) \cdot O(\epsilon/\kappa^2)$ and last-step measurement precision $\delta = \frac{\epsilon}{4\kappa^2}$. Note that in one phase estimation procedure, we need to simulate H for $O(1/\delta_{PE})$ time, thus the $O(\kappa^2/\epsilon)$ repetitions in Step 4 gives the total error $\delta_{HS} O(1/\delta_{PE}) O(\kappa^2/\epsilon)$, which can be made $\delta_{PE}/100$ by choosing the constant small enough in the big-O in definition of δ_{HS}. Thus the imperfection of the Hamiltonian simulation is negligible compared to the phase estimation error δ_{PE}.

We first analyze the quality of the solution $|\tilde{x}\rangle$.

Lemma 9. *With probability at least 0.99, the outputted vector $\tilde{x}$ satisfies*

$$\|\tilde{x} - x^*\|_2 \le \epsilon \cdot \max\{\|x^*\|_2, \|b\|_2\}.$$

Proof. First we will show that if we observe $|1\rangle$ in Step 6, then the outputted state (normalized) $|\tilde{x}\rangle$ satisfies $\|\tilde{x} - x^*\|_2 \le \epsilon \|x^*\|_2$. Indeed, if we observe $|1\rangle$, then the remaining state is proportional to $|\tilde{x}\rangle = \sum_{i=1}^r \frac{\beta_i}{\tilde{\lambda}_i}|v_i\rangle$. Recall that $|x^*\rangle = \sum_{i=1}^r \frac{\beta_i}{\lambda_i}|v_i\rangle$. Thus

$$\begin{aligned} \|\tilde{x} - x^*\|_2^2 &= \sum_{i=1}^r \left(\frac{\beta_i}{\tilde{\lambda}_i} - \frac{\beta_i}{\lambda_i}\right)^2 = \sum_{i=1}^r \frac{\beta_i^2}{\lambda_i^2}\left(1 - \frac{\lambda_i}{\tilde{\lambda}_i}\right)^2 = \sum_{i=1}^r \frac{\beta_i^2}{\lambda_i^2}\frac{(\tilde{\lambda}_i - \lambda_i)^2}{\tilde{\lambda}_i^2} \\ &\le \left(\frac{\delta_{\mathrm{PE}}}{\frac{1}{\kappa} - \delta_{\mathrm{PE}}}\right)^2 \|x^*\|_2^2 \le \epsilon^2 \|x^*\|_2^2. \end{aligned} \tag{7}$$

Next we show that if we do not observe $|1\rangle$, then the outputted 0 vector is still a good estimation to $|x^*\rangle$, because $|x^*\rangle$ itself is too short. More precisely, define $\rho = \frac{1}{\|b\|_2^2}\sum_{i=1}^r \beta_i^2$, the fraction of $|b\rangle$ falling into the non-zero eigenspace of A. Note that the probability of observing $|1\rangle$ in Step 6 is

$$p = \frac{1}{\|b\|_2^2}\sum_{i=1}^r \frac{\beta_i^2}{4\kappa^2\tilde{\lambda}_i^2} = \frac{1}{4\kappa^2\|b\|_2^2}\sum_{i=1}^r \frac{\beta_i^2}{\tilde{\lambda}_i^2} \ge \frac{1}{4\kappa^2\|b\|_2^2}\sum_{i=1}^r \beta_i^2 = \frac{1}{4\kappa^2}\rho.$$

If $\rho \ge \epsilon^2/\kappa^2$, then $p \ge \epsilon^2/(4\kappa^4)$, thus the amplitude amplification already boosts the probability to 0.99 with $O(\kappa^2/\epsilon)$ repetitions, enabling us to observe $|1\rangle$ almost for sure. When $\rho < \epsilon^2/\kappa^2$, if we observe $|1\rangle$, then Eq.(7) still holds. If we observe $|0\rangle$ and output 0 as an estimate to x^*, then the error is

$$\|0 - x^*\|_2^2 = \|x^*\|_2^2 = \sum_{i=1}^r \frac{\beta_i^2}{\lambda_i^2} \le \kappa^2\rho\|b\|_2^2 < \epsilon^2\|b\|_2^2.$$

□

Next we analyze the estimated norm.

Lemma 10. *With probability at least 0.99, the outputted value ℓ satisfies*

$$\left|\ell - \|x^*\|_2^2\right| \le \epsilon\big(\|x^*\|_2^2 + \|b\|_2^2\big).$$

Proof. Recall that $\ell = p' \cdot 4\|b\|_2^2\kappa^2$, and $|p - p'| \le \delta$.

$$\begin{aligned} \left|\ell - \|x^*\|_2^2\right| &\le \left|p \cdot 4\|b\|_2^2\kappa^2 - \|x^*\|_2^2\right| + \delta \cdot 4\|b\|_2^2\kappa^2 \\ &= \Big|\sum_{i=1}^r \Big(\frac{\beta_i^2}{\tilde{\lambda}_i^2} - \frac{\beta_i^2}{\lambda_i^2}\Big)\Big| + \delta \cdot 4\|b\|_2^2\kappa^2 \end{aligned}$$

Using the fact that $\lambda_i \ge 1/\kappa$ and that $|\lambda_i - \tilde{\lambda}_i| \le \delta_{\mathrm{PE}}$, it is not hard to see that

$$\Big|\sum_{i=1}^r \Big(\frac{\beta_i^2}{\tilde{\lambda}_i^2} - \frac{\beta_i^2}{\lambda_i^2}\Big)\Big| \le 2\kappa\delta_{\mathrm{PE}}\sum_{i=1}^r \frac{\beta_i^2}{\lambda_i^2} = 2\kappa\delta_{\mathrm{PE}}\|x^*\|_2^2 = \epsilon\|x^*\|_2^2.$$

Since $\delta = \epsilon/4\kappa^2$, we have $\delta \cdot 4\|b\|_2^2\kappa^2 = \epsilon\|b\|_2^2$. Thus $\left|\ell - \|x^*\|_2^2\right| \le \epsilon\big(\|x^*\|_2^2 + \|b\|_2^2\big)$. □

The error probability is a small constant as guaranteed by the error rate of phase estimation, amplitude amplification, and amplitude estimation. Finally let us analyze the cost. For Step 1, we can efficiently prepare $|b\rangle$ in time $O(\log n)$ provided that b_i $(i = 1, ..., n)$ and $\sum_{i_1}^{i_2} |b_i|^2 (1 \leq i_1 < i_2 \leq n)$ are efficiently computable by using the procedure of [GR02].

For Step 2, we perform quantum phase estimation by simulating e^{iA}, which takes time $O(s(\log n + poly \log(s, \kappa)))$ by Theorem 8. In order that the eigenvalue estimation has error at most $\delta_{\mathrm{PE}} = \frac{\epsilon}{2\kappa}$, the phase estimation algorithm needs $O(\kappa/\epsilon)$ calls of e^{iA} simulation. Thus the total time for one phase estimation is $O(s(\log n + poly \log(s, \kappa))\kappa/\epsilon)$. Repeating this for $O(\kappa^2/\epsilon)$ time in Step 4 needs time $O(s(\log n + poly \log(s, \kappa))\kappa^3/\epsilon^2)$.

Therefore, if we do not need to estimate the norm $\|x^*\|_2$, then the algorithm can just stop before Step 7. The total time cost is $O((\log n) \cdot s^2 \cdot \frac{\kappa}{\epsilon} \cdot \frac{\kappa^2}{\epsilon}) = O((\log n) s^2 \kappa^3/\epsilon^2)$.

To estimate the norm $\|x^*\|_2$, the Amplitude Estimation needs to repeat Step 1 to 3 at most $O(1/\delta) = O(\kappa^2/\epsilon)$ times. So the total cost is $O((\log n) \cdot s^2 \cdot \frac{\kappa}{\epsilon} \cdot \frac{\kappa^2}{\epsilon}) = O((\log n) s^2 \kappa^3/\epsilon^2)$.

This completes the proof of Theorem 1.

Our quantum algorithms for the two extensions, Ridge Regression and Truncated SVD problem are deferred to Appendix (Section A).

4 Quantum algorithm for calculating statistic leverage scores and matrix coherence

In this section, we present quantum algorithms for calculating statistic leverage scores and matrix coherence, and analyze their performance. Given an $n \times p$ matrix A of rank r with SVD $A = U\Sigma V^T$ where $U \in \mathbb{R}^{n \times r}$, $\Sigma \in \mathbb{R}^{r \times r}$ and $V \in \mathbb{R}^{p \times r}$, the statistic leverage scores of A are defined as $s_i = \|U_i\|_2^2$, $i \in \{1, ..., n\}$, and the matrix coherence c is the largest statistic leverage score of A.

Without loss of generality, we assume that A is Hermitian. (See Appendix C for the detailed technique to deal with the non-Hermitian case.) We have the following quantum algorithm for calculating the k-th statistic leverage score of A, s_k for any index $k \in [n]$. Denote the k-th computational basis by $|e_k\rangle$, which has the form $e_k = \sum_{i=1}^{n} \beta_i |v_i\rangle$ as a decomposition into A's eigenvectors $|v_i\rangle$.

Algorithm QSLS
Input: $A \in \mathbb{R}^{n\times n}$, $k \in [n]$. A is Hermitian with rank r and spectral decomposition $A = \sum_{i=1}^{n} \lambda_i |v_i\rangle\langle v_i|$. The eigenvalues $\lambda_1, ..., \lambda_n$ satisfy $\frac{1}{\kappa} \leq |\lambda_i| \leq 1$ for $i \leq r$ and $\lambda_i = 0$ for $i > r$. Suppose that $e_k = \sum_{i=1}^{n} \beta_i |v_i\rangle$.
Output: A value $\tilde{s_k} \approx s_k$.
Algorithm:

1. Prepare the quantum state $|e_k\rangle = \sum_{i=1}^{n} \beta_i |v_i\rangle$.
2. Perform phase estimation to create the state proportional to $\sum_{i=1}^{n} \beta_i |v_i\rangle|\tilde{\lambda_i}\rangle$, where $\tilde{\lambda_i}$ is the estimated value of λ_i satisfying $|\tilde{\lambda_i} - \lambda_i| \leq \delta_{\mathrm{PE}} \stackrel{\mathrm{def}}{=} \frac{1}{3\kappa}$ for $i = 1, \ldots, n$.
3. Add one qubit $|0\rangle$ to the state and perform a controlled rotation as follows. If $\tilde{\lambda_i} \geq \frac{1}{2\kappa}$, rotate the qubit to $|1\rangle$; otherwise do nothing. The resulting state is proportional to
$$\sum_{i=1}^{r} \beta_i |v_i\rangle|\tilde{\lambda_i}\rangle|1\rangle + \sum_{i=r+1}^{n} \beta_i |v_i\rangle|\tilde{\lambda_i}\rangle|0\rangle. \tag{8}$$
4. Measure the last qubit. Denote by p the probability of observing $|1\rangle$.
5. Use Amplitude Estimation to get an estimate p' of p to precision ϵ. Output $\tilde{s_k} = p'$.

Now we analyze the running time and performance. The analysis of Hamiltonian simulation error and running time is similar to that for Algorithm QLSR and is omitted here, and we will focus on the difference. In order that the eigenvalue estimation has error $\leq \delta_{\mathrm{PE}} = \frac{1}{3\kappa}$, i.e. $|\tilde{\lambda_i} - \lambda_i| \leq \delta_{\mathrm{PE}}, \forall i = 1, ..., n$, we need to run the procedure $O(\kappa)$ times.

To estimate the probability p (of observing $|1\rangle$ in Step 4) to precision ϵ, the Amplitude Estimation needs to repeat Step 1 to 4 at most $O(1/\epsilon)$ times. So the total cost is $O(s(\log n + polylog(s,\kappa)) \times O(\kappa) \times O(\frac{1}{\epsilon}) = O(s\kappa\epsilon^{-1}(\log n + polylog(s,\kappa)))$.

The probability of seeing $|1\rangle$ in Step 4 of Algorithm QSLS is $p = \sum_{i=1}^{r} \beta_i^2$. On the other hand, note that
$$AA^+ = U\Sigma V^T (U\Sigma^{-1}V^T)^T = U\Sigma V^T V \Sigma^{-1} U^T = U\Sigma\Sigma^{-1}U^T = UU^T,$$
where we used the fact that $V^T V = I_r$ and $\Sigma\Sigma^{-1} = I_r$. Therefore, we can relate s_k to the probability p of observing $|1\rangle$ in the following way.
$$s_k = \|U_k\|_2^2 = \left\|e_k^T U\right\|_2^2 = e_k^T UU^T e_k = e_k^T AA^+ e_k.$$
Plugging $e_k = \sum_{i=1}^{n} \beta_i |v_i\rangle$, $A = \sum_{i=1}^{r} \lambda_i |v_i\rangle\langle v_i|$ and $A = \sum_{i=1}^{r} \lambda_i^{-1} |v_i\rangle\langle v_i|$ in the above expression, we have
$$e_k^T AA^+ e_k = \sum_{i,\ell\in[r],j,k\in[n]} \beta_i \lambda_j \lambda_k^{-1} \beta_\ell \langle v_i|v_j\rangle\langle v_j|v_k\rangle\langle v_k|v_\ell\rangle = \sum_{i=1}^{r} \beta_i^2.$$
Putting the above two inequalities together yields
$$s_k = \sum_{i=1}^{r} \beta_i^2 = p.$$
Since the outputted estimate $\tilde{s_k} = p'$, and $|p' - p| \leq \epsilon$, we have that $|\tilde{s_k} - s_k| \leq \epsilon$. This proves Theorem 4. Using a simple amplitude amplification, we easily get Corollary 5.

5 Concluding remarks

In this paper we give efficient algorithms for least squares regression, ridge regression, δ-truncated singular value decomposition and calculating statistical leverage scores and matrix coherence. The latter is particularly interesting because it gives an example of exponential speedup for a natural linear algebraic problem, instead of number theoretic or modern algebraic ones (such as problems on groups, fields, polynomials, etc). We hope that phase estimation can find more applications in quantum algorithm designing for linear algebraic problems.

Acknowledgments

The work was supported by Research Grants Council of the Hong Kong S.A.R. (Project no. CUHK419413). Part of the work was done when the S.Z. visited Centre of Quantum Technologies partially under its support.

References

[Amb12] Andris Ambainis. Variable time amplitude amplification and quantum algorithms for linear algebra problems. In *Proceedings of the 29th International Symposium on Theoretical Aspects of Computer Science*, pages 636–647, 2012.

[BACS07] Dominic W Berry, Graeme Ahokas, Richard Cleve, and Barry C Sanders. Efficient quantum algorithms for simulating sparse hamiltonians. *Communications in Mathematical Physics*, 270(2):359–371, 2007.

[BCK15] Dominic W. Berry, Andrew M. Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *Proceedings of the 56th Annual Symposium on Foundations of Computer Science*, pages 792–809, 2015.

[BDM99] Vladimír Bužek, Radoslav Derka, and Serge Massar. Optimal quantum clocks. *Physical review letters*, 82(10):2207, 1999.

[BHMT00] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *arXiv preprint quant-ph/0005055*, 2000.

[BMD09] Christos Boutsidis, Michael W Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 968–977. Society for Industrial and Applied Mathematics, 2009.

[CEMM97] Richard Cleve, Artur Ekert, Chiara Macchiavello, and Michele Mosca. Quantum algorithms revisited. *arXiv preprint quant-ph//9708016*, 1997.

[CH$^+$86] Samprit Chatterjee, Ali S Hadi, et al. Influential observations, high leverage points, and outliers in linear regression. *Statistical Science*, 1(3):379–393, 1986.

[CJS13] B. D. Clader, B. C. Jacobs, and C. R. Sprouse. Preconditioned quantum linear system algorithm. *Physical Review Letters*, page 250504, 2013.

[CLL+15] Shouyuan Chen, Yang Liu, Michael R. Lyu, Irwin King, and Shengyu Zhang. Fast relative-error approximation algorithm for ridge regression. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 201–210, 2015.

[CR09] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[CW13] Kenneth L Clarkson and David P Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 81–90. ACM, 2013.

[DMIMW12] Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. *The Journal of Machine Learning Research*, 13(1):3475–3506, 2012.

[DMM08] Petros Drineas, Michael W Mahoney, and S Muthukrishnan. Relative-error cur matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008.

[DMMS11] Petros Drineas, Michael W Mahoney, S Muthukrishnan, and Tamás Sarlós. Faster least squares approximation. *Numerische Mathematik*, 117(2):219–249, 2011.

[EY36] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[GHO99] Gene H Golub, Per Christian Hansen, and Dianne P O'Leary. Tikhonov regularization and total least squares. *SIAM Journal on Matrix Analysis and Applications*, 21(1):185–194, 1999.

[GR02] Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint quant-ph/0208112*, 2002.

[Han87] Per Christian Hansen. The truncatedsvd as a method for regularization. *BIT Numerical Mathematics*, 27(4):534–553, 1987.

[HHL09] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[HW78] David C Hoaglin and Roy E Welsch. The hat matrix in regression and anova. *The American Statistician*, 32(1):17–22, 1978.

[Kit95] Alexei Yu Kitaev. Quantum measurements and the abelian stabilizer problem. *arXiv preprint quant-ph/9511026*, 1995.

[MD09] Michael W Mahoney and Petros Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

[Mos09] Michele Mosca. *Quantum Algorithms,* chapter in *Encyclopedia of Complexity and Systems Science*. Springer, 2009.

[NN13] Jelani Nelson and Huy L Nguyên. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 117–126. IEEE, 2013.

[Sar06] Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 143–152. IEEE, 2006.

[SGV98] Craig Saunders, Alexander Gammerman, and Volodya Vovk. Ridge regression learning algorithm in dual variables. In *(ICML-1998) Proceedings of the 15th International Conference on Machine Learning*, pages 515–521. Morgan Kaufmann, 1998.

[She94] Jonathan Richard Shewchuk. An introduction to the conjugate gradient method without the agonizing pain, 1994.

[Sto10] Andrew James Stothers. On the complexity of matrix multiplication. 2010.

[Tik63] Andrey Tikhonov. Solution of incorrectly formulated problems and the regularization method. In *Soviet Math. Dokl.*, volume 5, pages 1035–1038, 1963.

[TR10] Ameet Talwalkar and Afshin Rostamizadeh. Matrix coherence and the nystrom method. *arXiv preprint arXiv:1004.2008*, 2010.

[WBL12] Nathan Wiebe, Daniel Braun, and Seth Lloyd. Quantum algorithm for data fitting. *Physical Review Letters*, 109:050505, 2012.

[ZWSP08] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. In *Algorithmic Aspects in Information and Management*, pages 337–348. Springer, 2008.

A Applications of our quantum algorithm for LSR

In this section, we present two applications of our quantum algorithm for LSR. First, we directly apply our techniques to RR. Second, we design a quantum algorithm for δ-TSVD.

A.1 Quantum algorithm for ridge regression

Since we already handled the case for A being singular in previous section, we can use the algorithm to the ridge regression problem, by a reduction based on the following key observation. Ridge regression Eq.(3) can be reformulated into the following problem:

$$x^* = \arg\min_{x \in \mathbb{R}^p} \left\| \begin{bmatrix} A \\ \sqrt{\lambda} I_p \end{bmatrix} x - \begin{bmatrix} b \\ 0_p \end{bmatrix} \right\|_2^2, \tag{9}$$

which becomes a least squares regression problem. Thus we can use the quantum algorithm for least squares regression with input $\begin{bmatrix} A \\ \sqrt{\lambda} I_p \end{bmatrix}$ and $\begin{bmatrix} b \\ 0_p \end{bmatrix}$ to solve this problem and prove Theorem 2. We give detailed explanation on how the κ' comes.

Let the SVD of A be $A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$ where $u_i \in \mathbb{R}^n$ and $v_i \in \mathbb{R}^p$. Let $u_{r+1}, ..., u_n$ be the rest vectors in the orthonormal basis containing $u_1, ..., u_r$ and let $v_{r+1}, ..., v_p$ be the rest vectors in the orthonormal basis containing $v_1, ..., v_r$. Then the SVD of $\begin{bmatrix} A \\ \sqrt{\lambda} I_p \end{bmatrix}$ is $\begin{bmatrix} A \\ \sqrt{\lambda} I_p \end{bmatrix} = \sum_{i=1}^{r} \sigma_i u_i' v_i^T + \sum_{i=r+1}^{p} \sqrt{\lambda} u_i' v_i^T$ where $u_i' = \begin{bmatrix} u_i \\ 0_p \end{bmatrix}$ for $i = 1, ..., r$ and $u_i' = \begin{bmatrix} 0_n \\ v_i \end{bmatrix}$ for $i = r+1, ..., p$. So the singular values of $\begin{bmatrix} A \\ \sqrt{\lambda} I_p \end{bmatrix}$ are $\sigma_1, ..., \sigma_r$ and $\sqrt{\lambda}, ..., \sqrt{\lambda}$ ($p - r$ times). If $\sigma_1, ..., \sigma_r$ are in the range $[\frac{1}{\kappa}, 1]$, the all the singular values of $\begin{bmatrix} A \\ \sqrt{\lambda} I_p \end{bmatrix}$ will be in the range $[\min\{\frac{1}{\kappa}, \sqrt{\lambda}\}, \max\{1, \sqrt{\lambda}\}]$, thus we set $\kappa' = \frac{\max\{1, \sqrt{\lambda}\}}{\min\{\frac{1}{\kappa}, \sqrt{\lambda}\}}$.

A.2 Quantum algorithm for δ-Truncated SVD

In this part, we generalize our techniques to solve the δ-TSVD problem. Let λ_i be the i-th largest singular value of A for $i = 1, ..., n$, we assume that $\lambda_i \leq 1$. Let r be the rank of A. Specially, let $\Lambda_1 = \max\{\lambda_i : \lambda_i < \delta, i \in [r]\}$, $\Lambda_2 = \min_i\{\lambda_i : \lambda_i \geq \delta, i \in [r]\}$, and let $\Lambda = \Lambda_2 - \Lambda_1$.

We have the following algorithm QTSVD for δ-TSVD. The main difference between QTSVD and QLSR is that in QLSR we need to distinguish the non-zero singular values and zero singular values, while in QTSVD we need to distinguish the large singular values that are $\geq \delta$ and the small singular values that are $< \delta$. More specifically, in Algoritm QTSVD we have a different requirement on the phase estimation error δ_{PE} in Step 2 and a different rule to rotate the last register in Step 3; the rest are similar.

Algorithm QTSVD
Input: $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$, δ, where A is Hermitian with rank r and spectral decomposition $A = \sum_{i=1}^n \lambda_i |v_i\rangle\langle v_i|$. $|\lambda_i| \leq 1$ for $i \in [n]$. $\Lambda_1 = \max\{|\lambda_i| : |\lambda_i| < \delta, i \in [r]\}$, $\Lambda_2 = \min_i\{|\lambda_i| : |\lambda_i| \geq \delta, i \in [r]\}$, $\Lambda = \Lambda_2 - \Lambda_1$, $b = \sum_{i=1}^n \beta_i |v_i\rangle$.
Output: A quantum state proportional to $|\tilde{x}\rangle$ where $\tilde{x}$ is the approximation to the optimal solution $x^* = A^+ b$ and a value ℓ as the approximation to $\|\tilde{x}\|_2^2$.
Algorithm:

1. Prepare the quantum state proportional to $|b\rangle = \sum_{i=1}^n \beta_i |v_i\rangle$.
2. Perform phase estimation to precision $\delta_{\mathrm{PE}} \stackrel{\mathrm{def}}{=} \min\{\frac{\epsilon\delta}{2}, \frac{\Lambda}{2}\}$ to create the state proportional to $\sum_{i=1}^n \beta_i |v_i\rangle|\tilde{\lambda_i}\rangle$, where $|\tilde{\lambda_i} - \lambda_i| \leq \delta_{\mathrm{PE}}$ for $i = 1, ..., n$.
3. Add a qubit $|0\rangle$ to the state and make a controlled rotation as follows. If $\tilde{\lambda}_i \geq \frac{\Lambda_1+\Lambda_2}{2} - \frac{\Lambda}{2}$, rotate it to $(\frac{\delta}{2\tilde{\lambda}_i}|1\rangle + \sqrt{1 - \frac{\delta^2}{4\tilde{\lambda}_i^2}}|0\rangle)$; otherwise do nothing. We get the state proportional to
$$\sum_{i:\lambda_i \geq \delta} \beta_i |v_i\rangle|\tilde{\lambda}_i\rangle \left(\frac{\delta}{2\tilde{\lambda}_i}|1\rangle + \sqrt{1 - \frac{\delta^2}{4\tilde{\lambda}_i^2}}|0\rangle\right) + \sum_{i:\lambda_i < \delta} \beta_i |v_i\rangle|\tilde{\lambda}_i\rangle|0\rangle. \tag{10}$$
4. Use amplitude amplification, which repeats the previous steps $O(\frac{1}{\delta^2\epsilon})$ times.
5. Measure the last qubit.
6. **if** we observe $|1\rangle$,
   (a) The remaining state is proportional to $\sum_{i:\lambda_i \geq \delta} \frac{\beta_i}{\tilde{\lambda}_i} |v_i\rangle|\tilde{\lambda}_i\rangle$.
   (b) Reverse the phase estimation process and get the state proportional to $\sum_{i:\lambda_i \geq \delta} \frac{\beta_i}{\tilde{\lambda}_i} |v_i\rangle = |\tilde{x}\rangle$ as our output.

   else output 0 as an estimate to $|x^*\rangle$.
7. Use Amplitude Estimation to get an estimate p' to the probability p of observing $|1\rangle$ when measuring the state in Eq.(6) to precision $\epsilon\delta^2/4$. Output $\ell = p' \cdot 4\|b\|_2^2/\delta^2$.

The proof of Theorem 3 is very similar to that of Theorem 1, and we omit the details here.

B Non-Hermitian case for LSR

Assume $A \in \mathbb{R}^{n\times p}$ with rank $r \leq \min\{n, p\}$, $b \in \mathbb{R}^n$, and we want to calculate $x^* = A^+ b \in \mathbb{R}^p$.

Define
$$A' = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}, \tag{11}$$
and
$$b' = \begin{bmatrix} b \\ 0_p \end{bmatrix}. \tag{12}$$
Now $A' \in \mathbb{R}^{(n+p)\times(n+p)}$ is Hermitian with rank $2r$. It is easily seen that if A has at most s nonzero entries in each row and column, so is A'. Furthermore, the optimal solution $(x^*)'$ to the new LSR problem (A', b') is related to that x^* to the original one (A, b) as follows:
$$(x^*)' = A'^+ b' = \begin{bmatrix} 0_n \\ x^* \end{bmatrix}. \tag{13}$$

Thus, if A is not Hermitian, we can work on the instance (A', b') of LSR and read out the optimal solution x^* from Eq.(13).

C Non-Hermitian case for calculating statistic leverage scores

Given a matrix $A \in \mathbb{R}^{n\times p}$, suppose the thin SVD of A is $A = U\Sigma V^T$, we want to calculate the values $\|U_i\|_2^2, i = i, \cdots, n$.

Let

$$A' = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}, \tag{14}$$

which is a Hermition. Note that

$$A' = \frac{1}{\sqrt{2}} \begin{bmatrix} U & U \\ V & -V \end{bmatrix} \cdot \begin{bmatrix} \Sigma & \\ & -\Sigma \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} U & U \\ V & -V \end{bmatrix}^T. \tag{15}$$

We can see that the statistic leverage scores of A are exactly the first n statistic leverage scores of A', which is a Hermition. Computing an individual leverage score of A reduces to computing an individual leverage score of A', and computing the maximum leverage score of A reduces to computing the maximum, over the first n rows, leverage score of A'.