Fast quantum algorithms for Least Squares Regression and Statistic Leverage Scores

Yang Liu Shengyu Zhang

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]. The best 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].

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. 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 touch upon the controversial issue from two perspectives. 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 design 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 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 algorithms that output a quantum sketch only, our algorithm for calculating SLS and MC indeed produces 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. 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 $x^* = \arg\min_{x \in \mathbb{R}^p} \|Ax - b\|_2^2$. 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 the above has a unique and closed-form solution $x^* = A^\dagger b$, where $A^\dagger$ is the Moore-Penrose pseudoinverse of A. If one computes x^* naively by first computing $A^\dagger$ and then the product $A^\dagger b$, then the cost is $O(p^2 n + n^2 p)$, which is prohibitively slow in the big data era. 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 $O(\log(n+p)s^3\kappa^6)$. In this paper, we propose a better quantum algorithm that outputs a quantum state close to $|\overline{x^*}\rangle$ in time $O(\log(n+p)s^2\kappa^3)$. Compared to [WBL12], our algorithm is much simpler since we directly apply the operator $A^\dagger$ to the state $|b\rangle$, while they first applied A^T to $|b\rangle$ to get $|A^T b\rangle$, then prepared $(A^T A)^{-1}$ and applied it to $|A^T b\rangle$. This simplicity also leads to a better dependence on s and κ in our algorithm. Another important difference is that [WBL12] assumes that A is Hermitian, which is usually not the case for typical machine learning applications[1]. Our algorithm works for non-Hermitian matrices as well, for which we need to work on singular Hermitian matrices A. Finally, note that $|\overline{x^*}\rangle$ misses one important information of x^*, namely its ℓ_2 norm, which is actually crucial if we want to compute $\sum_i c_i x_i$ by SWAP test in aforementioned applications. Our algorithm gives a good estimate to $\|x^*\|_2^2$ without within the same running time bound.

[1] Although the authors 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).

Ridge regression and truncated SVD For ill-conditioned problems, i.e. when the condition number of A is large, the solution given by $x^* = A^\dagger b$ 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 $x^* = \arg\min_{x\in\mathbb{R}^p} \|Ax - b\|_2^2 + \lambda \|x\|_2^2$. The unique minimizer of this 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 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+14]. 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 our algorithm for LSR, we design a quantum algorithm which generates a quantum state close to $|\bar{x^*}\rangle$ and estimates the norm $\|\tilde{x}\|_2^2$, in time $O(\log(n+p)s^2\kappa'^3/\epsilon^2)$, for $\kappa' = \max\{1, \sqrt{\lambda}\}/\min\{\frac{1}{\kappa}, \sqrt{\lambda}\}$.

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 $\delta > 0$ and we want to find $x^* = \arg\min_{x\in\mathbb{R}^p} \|A_\delta x - b\|_2^2$, 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 $O(n^2p + np^2)$ time. Our algorithm for LSR can be also adapted to solve δ-TSVD efficiently.

Calculating statistic leverage scores and matrix coherence 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 (SLS) of A are defined as $s_i = \|U_i\|_2^2$, $i \in \{1, ..., n\}$. The matrix coherence (MC) of A is defined as $c = \max_{i\in\{1,...,n\}} s_i$, the largest statistic leverage score of A. 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 any specific SVD decomposition 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]. Equal to the diagonal entries of the "hat-matrix", they are widely used to indicate possible outliers in regression diagnostics [HW78][CH+86]. In addition, 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].

The best classical algorithm for both calculating the statistic leverage scores and calculating matrix coherence takes time $O((np+p^3)\log n)$ [DMIMW12]. 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.