Theoretical Computer Science 657 (2017) 38–47

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Fast quantum algorithms for least squares regression and statistic leverage scores

Yang Liu, Shengyu Zhang *

Department of Computer Science and Engineering and The Institute of Theoretical Computer Science and Communications, The Chinese University of Hong Kong, Hong Kong

ARTICLE INFO

Article history:
Received 2 November 2015
Received in revised form 28 April 2016
Accepted 2 May 2016
Available online 8 June 2016

Keywords:
Least square regression
Statistical leverage score
Quantum algorithms

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 $i = 1, \ldots, n$. 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.

1. Introduction

Quantum algorithms for solving linear systems, and the controversy The past two decades witnessed the development of quantum algorithms [22], 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 [31]. The best known classical algorithm for solving linear

* Corresponding author.
E-mail addresses: yliu@cse.cuhk.edu.hk (Y. Liu), syzhang@cse.cuhk.edu.hk (S. Zhang).

http://dx.doi.org/10.1016/j.tcs.2016.05.044
0304-3975/

systems for this class of matrices runs in time $O(\sqrt{\kappa}sn)$ [26], 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 [18] 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 [1] and the algorithm was used for solving least squares regression (defined next) by Wiebe, Braun and Lloyd [30]. HHL algorithm was also extended in [8] 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 [18]. 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 [1], 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 [30], 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 [24,13,21,5,14]. 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 $O(n^2)$ [12]. 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 $O(p^2n + n^2p)$, 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 $\left\|\tilde{x} - x^*\right\|_2 \leq \epsilon \|x^*\|_2$ for any constant error $0 < \epsilon < 1$, and run in time $\tilde{O}(\text{nnz}(A) + nr)$ [11,23], 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 [18], one can only hope to

[1] Though theoretically more efficient algorithms for matrix multiplication exist [27], 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.