Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension

András Gilyén[1,*]   Seth Lloyd[2]   Ewin Tang[3]

November 13, 2018

Abstract

We construct an efficient classical analogue of the quantum matrix inversion algorithm [HHL09] for low-rank matrices. Inspired by recent work of Tang [Tan18a], assuming length-square sampling access to input data, we implement the pseudoinverse of a low-rank matrix and sample from the solution to the problem $Ax = b$ using fast sampling techniques. We implement the pseudo-inverse by finding an approximate singular value decomposition of A via subsampling, then inverting the singular values. In principle, the approach can also be used to apply any desired "smooth" function to the singular values. Since many quantum algorithms can be expressed as a singular value transformation problem [GSLW18], our result suggests that more low-rank quantum algorithms can be effectively "dequantised" into classical length-square sampling algorithms.

1   Introduction

Quantum computing provides potential exponential speed-ups over classical computers for a variety of linear algebraic tasks, including an operational version of matrix inversion [HHL09]. Recently, inspired by the quantum algorithm for recommendation systems [KP17], Tang showed how to generalize the well-known FKV algorithm [FKV04] to sample from the singular vectors of low-rank matrices [Tan18a] and to implement principal component component analysis [Tan18b]. Intriguingly, Tang's work suggests that many of the quantum algorithms for low-rank matrix manipulation [RSML18] can be extended to provide fast *classical* algorithms under suitable sampling assumptions, achieving logarithmic dependence on the dimension. In this work, we show that such exponential speed-ups are indeed possible in the case of low-rank matrix inversion. Our treatment is self-contained and improves some aspects of previous approaches [Tan18a], leading to smaller exponents in our runtime bounds.

Suppose we want to solve $Ax = b$, where we are given $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, and wish to recover $x \in \mathbb{R}^n$. The equation might not have a solution, but we can always find an x minimizing $\|Ax - b\|$. Namely, $x = A^+ b$ works, where A^+ is the pseudoinverse of A. If

$$A = \sum_{\ell=1}^{k} \sigma_\ell \, u^{(\ell)} v^{(\ell)T}$$

is a singular value decomposition of A, such that $\sigma_\ell > 0$, then the pseudoinverse is simply $A^+ = \sum_{\ell=1}^{k} v^{(\ell)} u^{(\ell)T}/\sigma_\ell$, and $x = \sum_{\ell=1}^{k} v^{(\ell)} \langle u^{(\ell)}, b \rangle / \sigma_\ell$. The problem of finding the pseudoinverse of

---

[1]QuSoft, CWI and University of Amsterdam, the Netherlands. Supported by ERC Consolidator Grant QPROGRESS and partially supported by QuantERA project QuantAlgo 680-91-034.

[2]Massachusetts Institute of Technology, Departments of Mechanical Engineering and Physics, Xanadu

[3]University of Washington

[*]Corresponding author: `gilyen@cwi.nl`

1

a low-rank matrix occurs widely, such as in problems of data fitting, stochastic regression, and quadratic optimization with linear constraints.

Applications of the classical stochastic regression algorithm presented here include a wide variety of data analysis problems. Consider, for example, the problem of finding the optimal investment portfolio amongst n stocks. Let r_i be the m vectors of historical returns on the stocks (e.g., the returns on the i'th day or the i'th tick of the stock market), and let $\vec{r} = (1/m) \sum_i r_i$, so that A is the matrix with rows r_i^T. The covariance matrix is $C = \sum_i r_i r_i^T / m = A^T A$. Typically, the covariance matrix C is (approximately) low-rank, with each singular value corresponding to an underlying trend of correlated motion of investment returns. Classical portfolio management operates by finding the vector of investments w that maximizes the expected return $w^T r$ subject to a constraint on the variance $w^T C w$. As shown in [RL18], finding and sampling from the optimal portfolio and mapping out the optimal risk-return curve is a low-rank matrix inversion problem which can be solved on a quantum computer in time $\mathcal{O}(\mathrm{polylog}(mn))$, exponentially faster than conventional classical portfolio optimization methods. The results presented here show that our quantum-inspired classical algorithm can similarly allow one to map out the risk-return curve and sample from the optimal portfolio using $\mathcal{O}(\mathrm{polylog}(mn))$ *classical* time.

2 The algorithm

We use the following notation. For $v \in \mathbb{C}^d$ we denote the Euclidean norm by $\|b\|$. For a matrix $A \in \mathbb{C}^{m \times n}$ we denote by $\|A\|$ the operator norm, and by $\|A\|_F$ the Frobenius norm. We use notation A_i for the i-th row, $A_{\cdot j}$ for the j-th column, and $A^\dagger$ for the adjoint of A. We use "bra-ket" notation: for $v \in \mathbb{C}^d$ we denote the corresponding column vector by $|v\rangle \in \mathbb{C}^{d \times 1}$, and we denote by $\langle v| \in \mathbb{C}^{1 \times d}$ its adjoint. Accordingly we denote the inner product $\langle v, w \rangle$ by $\langle v|w \rangle$. We call the probability distribution $|v_i|^2/\|v\|^2$ over $i \in [d]$ the *length-square distribution* of v.

In this paper for simplicity we treat the case when the matrix A has rank $k \ll m, n$, and does not have too small singular values.[1] For normalization purposes we assume that $\|A\| \leq 1$ and $\|A^+\| \leq \kappa$. Our program is the following: we first show how to describe an approximate singular value decomposition $\sum_{\ell=1}^k \tilde{\sigma}_\ell |\tilde{u}^{(\ell)}\rangle\langle\tilde{v}^{(\ell)}|$ using a succinct representation of the vectors. Then we show how to estimate the values $\langle \tilde{u}^{(\ell)}|b\rangle/\tilde{\sigma}_\ell$ via sampling, and how to sample from the corresponding linear combination of the vectors $\tilde{v}^{(\ell)}$. The overall algorithm allows us to sample or query elements of an approximate solution $\tilde{x} \approx A^+ b$ to the equation $Ax = b$ in time poly-logarithmic in the size of the matrix.

The idea of the approximate singular value decomposition of A using length-square sampling comes from [FKV04]. Consider the following: first use length-square sampling to sample some rows R. If we sample enough rows, then $\|A^T A - R^T R\|$ is small as shown by Theorem 3. Then the singular values and right singular vectors of R are very close to the singular values and right singular vectors of A. Using the approximate right singular vectors we can recover approximate left singular vectors as well, e.g., by applying the matrix A to the right singular vector, as shown by Lemma 4. This is promising, but since R still can have very high-dimensional rows, computing the singular value decomposition of R can still be prohibitive. However, we can apply the trick once more! We can sample columns of R resulting in the matrix C. Again the singular values and left singular vectors of C are very close to the singular values and left singular vectors of R provided that $\|RR^T - CC^T\|$ is small.

Theorem 1 (Correctness)**.** *If A has rank at most[2] k, $\|A\| \leq 1$, $\|A^+\| \leq \kappa$, and the projection of b to the column space of A has norm $\Omega(\|b\|)$, then Algorithm 1 solves $Ax = b$ up to ε-multiplicative accuracy, such that $\|\tilde{x} - A^+ b\| \leq \varepsilon \|A^+ b\|$ with probability at least $1 - \eta$.*

---

[1]This assumption can be relaxed by inverting A only on the "well-conditioned" subspace, and dealing with small singular values similarly to the earlier works [FKV04, Tan18a], but we leave such details for future work.

[2]While running the algorithm we can actually detect if A has lower rank and adapt the algorithm accordingly.

In order to execute Algorithm 1 we use length-square sampling techniques, which have found great applications in randomized linear algebra [KV17], as well as recent quantum-inspired classical algorithms for recommendation systems [Tan18a]. For simplicity we assign unit cost to arithmetic operations such as addition or multiplication of real or complex numbers, assuming that all numbers are represented with a small number of bits. If $A \in \mathbb{C}^{m \times n}$ is stored in an efficient tree-like data-structure as in [FKV04, KP17, Tan18a], then we can implement the sampling and query operations of A required by Algorithm 1 in complexity $\mathcal{O}(\log(mn))$ assuming that data-structure queries have unit cost. Under these assumption we get the following bound:

Theorem 2 (Complexity). *If we have $\widetilde{\mathcal{O}}(1)$-time[3] query access to b and $\widetilde{\mathcal{O}}(1)$-time length-square access to A, then under the conditions of Theorem 1 we can execute Algorithm 1 in complexity $\widetilde{\mathcal{O}}\left(\kappa^{16}k^6\|A\|_F^6/\varepsilon^6\right)$, outputting an implicit description of $\tilde{x}$ suitable for query and sample access.*

Note that our complexity bound has smaller exponents than e.g. [Tan18a]. This partly comes from the fact that we only consider low-rank matrices, but we also get improvements by adapting and reanalysing the FKV algorithm [FKV04]. We only work out the constant factors for the number of rows and columns to be sampled, because these parameters dominate the complexity.

For comparison with the quantum analogue, note that under the assumption that the data structure for A is stored in quantum memory, an ε-approximate quantum state $|\tilde{x}\rangle/\|\tilde{x}\|$ can be prepared in complexity $\widetilde{\mathcal{O}}(\kappa\|A\|_F \mathrm{polylog}(1/\varepsilon))$, as shown in [CGJ18, GSLW18]. This directly enables length-square sampling, and its entries can be estimated with $\mathrm{poly}(\kappa/\varepsilon)$ overheads.

Algorithm 1 Low-rank stochastic regression via length-square sampling

---

Input: A vector $b \in \mathbb{C}^m$ and a matrix $A \in \mathbb{C}^{m \times n}$ s.t. $\|A\| \leq 1$, $\mathrm{rank}(A) = k$ and $\|A^+\| \leq \kappa$.

Goal 1: Query elements of a vector $\tilde{x}$ such that $\|\tilde{x} - x\| \leq \varepsilon\|x\|$ for $x = A^+b$.

Goal 2: Sample from a distribution 2ε-close in total-variation distance to $\frac{|x_j|^2}{\|x\|^2}$.

1: **Init:** Set $r = 2^{10}\ln\left(\frac{8n}{\eta}\right)\frac{\kappa^4k^2\|A\|_F^2}{\varepsilon^2}$ and $c = 2^6 \cdot 3^4 \ln\left(\frac{8r}{\eta}\right)\frac{\kappa^8k^2\|A\|_F^2}{\varepsilon^2}$.

2: **Sample rows:** Sample r row indices $i_1, i_2, \ldots, i_r$ according to the row norm squares $\frac{\|A_{i.}\|^2}{\|A\|_F^2}$. Define R to be the matrix whose s-th row is $\frac{\|A\|_F}{\sqrt{r}}\frac{A_{i_s.}}{\|A_{i_s.}\|}$.

3: **Sample columns:** Sample $s \in [r]$ uniformly, then sample a column index j distributed as $\frac{|R_{sj}|^2}{\|R_{s.}\|^2} = \frac{|A_{i_sj}|^2}{\|A_{i_s.}\|^2}$. Sample a total number of c column indices $j_1, j_2, \ldots, j_c$ this way. Define the matrix C whose t-th column is $\frac{\|R\|_F}{\sqrt{c}}\frac{R_{.j_t}}{\|R_{.j_t}\|} = \frac{\|A\|_F}{\sqrt{c}}\frac{R_{.j_t}}{\|R_{.j_t}\|}$.

4: **SVD:** Query all elements of A corresponding to elements of C. Compute the left singular vectors and singular values of C and denote the left singular vectors by $w^{(1)}, \ldots, w^{(k)}$ and the corresponding singular values by $\tilde{\sigma}_1, \ldots, \tilde{\sigma}_k$.

5: **Approximate right singular vectors of A:** Implicitly define $\tilde{v}^{(\ell)} := \left(\sum_{s=1}^r R_{i_s.}^\dagger \frac{w_s^{(\ell)}}{\tilde{\sigma}_\ell}\right)$.

6: **Matrix elements:** For each $\ell \in [k]$ compute $\tilde{\lambda}_\ell$ such that $\left|\tilde{\lambda}_\ell - \langle\tilde{v}^{(\ell)}|A^\dagger|b\rangle\right| = \mathcal{O}\left(\frac{\varepsilon\tilde{\sigma}_\ell^2\|b\|}{\sqrt{k}}\right)$.

7: **Output:** Row indices $i_1, i_2, \ldots, i_r$ and $w := \frac{\tilde{\lambda}_\ell}{\tilde{\sigma}_\ell^3}w^{(\ell)} \in \mathbb{C}^r$ such that $\|w\| = \mathcal{O}\left(\kappa^2\sqrt{k}\|b\|\right)$.

Queries to $\tilde{x}$: Define $\tilde{x} := R^\dagger w$, where R is implicitly defined by the row indices. A query to $\tilde{x}_j$ can be computed by querying $R_{s,j}$ for all $s \in [r]$ and taking their linear combination.

Sampling from $|\tilde{x}_j|^2/\|\tilde{x}\|^2$: Rejection sample (Lemma 12), using T samples from the distribution $\frac{|R_{s,j}|^2}{\|R_{s,.}\|^2}$ for some $s \in [r]$, and querying rT entries of R, s.t. $\mathbb{E}[T] \leq \|w\|^2\|A\|_F^2/\|\tilde{x}\|^2$.

In the above algorithm, we first convert left singular vectors of C ($w^{(\ell)}$) to approximate right singular vectors of R ($\tilde{v}^{(\ell)}$), which also approximate right singular vectors of A. Then we

---

[3]In this paper by $\widetilde{\mathcal{O}}(T)$ we hide poly-logarithmic factors in T, the dimensions m, n and the failure probability η.

"convert" these to left singular vectors of A in the form $(\langle\tilde{v}^{(\ell)}|A^\dagger/\tilde{\sigma}_\ell)$. To clarify the formula for $\tilde{x}$, notice the following sequence of approximations:

$$\tilde{x} \approx \sum_{\ell=1}^{k} \frac{1}{\tilde{\sigma}_\ell^4} |R^\dagger w^{(\ell)}\rangle\langle R^\dagger w^{(\ell)}|A^\dagger b \approx \sum_{\ell=1}^{k} \frac{1}{\tilde{\sigma}_\ell^2} |\tilde{v}^{(\ell)}\rangle\langle\tilde{v}^{(\ell)}|A^\dagger b \approx (R^\dagger R)^+ A^\dagger b \approx (A^\dagger A)^+ A^\dagger b = A^+ b$$

The conversion step from right to left singular vectors of A magnifies previous inaccuracies. For this reason, it is beneficial to sample a higher number of columns than rows, unlike in earlier works [FKV04, Tan18a].

3 Correctness of Algorithm 1

The goal is to output a description of an approximate solution $\tilde{x}$. If $\|\tilde{x} - x\| \le \varepsilon\|x\|$, then the length-square distributions of x and $\tilde{x}$ are 2ε-close as shown by Lemma 10. Thus for our purposes it suffices to find approximate right singular vectors $\tilde{v}^{(\ell)}$ and approximate singular values $\tilde{\sigma}_\ell$ such that

$$\left\|\sum_{\ell=1}^{k} \frac{|\tilde{v}^{(\ell)}\rangle\langle\tilde{v}^{(\ell)}|}{\tilde{\sigma}_\ell^2} A^\dagger A - \Pi_{\mathrm{rows}(A)}\right\| \le \frac{\varepsilon}{2}. \tag{1}$$

Let us define

$$|x'\rangle = \sum_{\ell=1}^{k} \frac{|\tilde{v}^{(\ell)}\rangle\langle\tilde{v}^{(\ell)}|}{\tilde{\sigma}_\ell^2} A^\dagger|b\rangle = \sum_{\ell=1}^{k} \frac{\langle\tilde{v}^{(\ell)}|A^\dagger|b\rangle}{\tilde{\sigma}_\ell^2}|\tilde{v}^{(\ell)}\rangle = \sum_{\ell=1}^{k} \frac{\lambda_\ell}{\tilde{\sigma}_\ell^2}|\tilde{v}^{(\ell)}\rangle, \tag{2}$$

then due to Equation (1) we have that

$$\|x' - x\| = \left\|\sum_{\ell=1}^{k} \frac{|\tilde{v}^{(\ell)}\rangle\langle\tilde{v}^{(\ell)}|}{\tilde{\sigma}_\ell^2} A^\dagger|b\rangle - |x\rangle\right\| = \left\|\left(\sum_{\ell=1}^{k} \frac{|\tilde{v}^{(\ell)}\rangle\langle\tilde{v}^{(\ell)}|}{\tilde{\sigma}_\ell^2} A^\dagger A - \Pi_{\mathrm{rows}(A)}\right)|x\rangle\right\| \le \frac{\varepsilon}{2}\|x\|.$$

Remember that we assumed that the projection of b to the column space of A has norm $\Omega(\|b\|)$. Since $\|A\| \le 1$ we also have that $\|x\| = \Omega(\|b\|)$. Therefore it suffices to find $\tilde{x}$ such that $\|\tilde{x} - x'\| = \mathcal{O}(\varepsilon\|b\|)$ in order to ensure $\|\tilde{x} - x'\| \le \frac{\varepsilon}{2}\|x\|$. If we compute approximate values $\tilde{\lambda}_\ell$ such that $\left|\lambda_\ell - \tilde{\lambda}_\ell\right| = \mathcal{O}\left(\frac{\varepsilon\tilde{\sigma}_\ell^2\|b\|}{\sqrt{k}}\right)$ then the magnitude of perturbation $\|\tilde{x} - x'\|$ can be bounded by $\mathcal{O}(\varepsilon\|b\|)$, and $\|w\| = \mathcal{O}(\kappa^2 k\|b\|)$ as we show in Section 3.2.

3.1 Finding approximate singular values and right singular vectors

First we invoke some improved bounds on length-square sampling from [KV17, Theorem 4.4].[4]

Length-square row sampling of a matrix $A \in \mathbb{C}^{m\times n}$ is as follows: pick a row index $i \in [m]$ with probability $p_i = \frac{\|A_{i.}\|^2}{\|A\|_F^2}$, and upon picking index i set the random output $Y = \frac{A_{i.}}{\sqrt{p_i}}$. Notice that in Algorithm 1 both R and C can be characterized as length-square (row) sampled matrices (the latter holding because every row of R has the same norm).

Theorem 3. *Let $A \in \mathbb{C}^{m\times n}$ be a matrix and let $R \in \mathbb{C}^{s\times n}$ be the sample matrix obtained by length-squared sampling and scaling to have $\mathbb{E}[R^\dagger R] = A^\dagger A$. ($R$ consists of rows $Y_1, Y_2, \ldots, Y_s$, which are i.i.d. copies of $Y/\sqrt{s}$, as defined above.) Then, for all $\varepsilon \in [0, \|A\|/\|A\|_F]$,[5] we have*

$$\mathbb{P}\left[\left\|R^\dagger R - A^\dagger A\right\| \ge \varepsilon\|A\|\|A\|_F\right] \le 2n e^{-\frac{\varepsilon^2 s}{4}}.$$

Hence, for $s \ge \frac{4\ln(2n/\eta)}{\varepsilon^2}$, with probability at least $(1-\eta)$ we have

$$\left\|R^\dagger R - A^\dagger A\right\| \le \varepsilon\|A\|\|A\|_F.$$

---

[4]In [KV17] the theorem is stated for real matrices, but the proof works for complex matrices as well.
[5]If $\varepsilon \ge \|A\|/\|A\|_F$, then the zero matrix is a good enough approximation to $AA^\dagger$.

In the following lemma $\|M\|$ denotes the operator norm, but the proof would also work for the Frobenius norm. Note that the following lemmas are independent of the dimensions of the matrices, which is the reason why we do not specify the dimensions. We use δ_{ij} to denote the Kronecker delta function, which is defined to be 1 if $i = j$ and 0 otherwise.

Lemma 4 (Converting approximate left and right singular vectors). *Suppose that $w^{(\ell)}$ is a system of orthonormal vectors spanning the column space of C such that*

$$\sum_{\ell=1}^{k} |w^{(\ell)}\rangle\langle w^{(\ell)}| = \Pi_{\mathrm{cols}(C)}$$

and

$$\langle w^{(i)}|CC^\dagger|w^{(j)}\rangle = \delta_{ij}\tilde{\sigma}_i^2.$$

Suppose that $\mathrm{rank}(R) = \mathrm{rank}(C) = k$ and $\left\|RR^\dagger - CC^\dagger\right\| \leq \gamma$. Let $\tilde{v}^{(\ell)} := \frac{R^\dagger w^{(\ell)}}{\tilde{\sigma}_\ell}$, then

$$\left|\langle \tilde{v}^{(i)}|\tilde{v}^{(j)}\rangle - \delta_{ij}\right| \leq \frac{\gamma}{\tilde{\sigma}_i\tilde{\sigma}_j},$$

and

$$\left|\langle \tilde{v}^{(i)}|R^\dagger R|\tilde{v}^{(j)}\rangle - \delta_{ij}\tilde{\sigma}_i^2\right| \leq \frac{\gamma\left(2\|R\|^2 + \gamma\right)}{\tilde{\sigma}_i\tilde{\sigma}_j}.$$

Proof. Let V be the matrix whose ℓ-th column is the vector $\tilde{v}^{(\ell)}$ and let us define the Gram matrix $G = V^\dagger V$. We have that

$$\begin{aligned}
G_{ij} = |\langle \tilde{v}^{(i)}, \tilde{v}^{(j)}\rangle - \delta_{ij}| &= \left|\frac{\langle w^{(i)}|RR^\dagger|w^{(j)}\rangle}{\tilde{\sigma}_i\tilde{\sigma}_j} - \delta_{ij}\right| \\
&\leq \left|\frac{\langle w^{(i)}|CC^\dagger|w^{(j)}\rangle}{\tilde{\sigma}_i\tilde{\sigma}_j} - \delta_{ij}\right| + \frac{\gamma}{\tilde{\sigma}_i\tilde{\sigma}_j} \\
&= \frac{\gamma}{\tilde{\sigma}_i\tilde{\sigma}_j}.
\end{aligned}$$

Now observe that

$$\left\|RR^\dagger RR^\dagger - CC^\dagger CC^\dagger\right\| \leq \left\|RR^\dagger(RR^\dagger - CC^\dagger)\right\| + \left\|(RR^\dagger - CC^\dagger)CC^\dagger\right\| \leq \gamma\left(2\|R\|^2 + \gamma\right).$$

Let $i, j \in [k]$, then

$$\langle w^{(i)}|CC^\dagger CC^\dagger|w^{(j)}\rangle = \langle w^{(i)}|CC^\dagger\left(\sum_{\ell=1}^{k} |w^{(\ell)}\rangle\langle w^{(\ell)}|\right)CC^\dagger|w^{(j)}\rangle = \delta_{ij}\sigma_i^4.$$

Finally we get that

$$\begin{aligned}
\left|\langle \tilde{v}^{(i)}|R^\dagger R|\tilde{v}^{(j)}\rangle - \delta_{ij}\sigma_i^2\right| &= \left|\frac{\langle w^{(i)}|RR^\dagger RR^\dagger|w^{(j)}\rangle}{\tilde{\sigma}_i\tilde{\sigma}_j} - \delta_{ij}\sigma_i^2\right| \\
&\leq \left|\frac{\langle w^{(i)}|CC^\dagger CC^\dagger|w^{(j)}\rangle}{\tilde{\sigma}_i\tilde{\sigma}_j} - \delta_{ij}\sigma_i^2\right| + \frac{\gamma\left(2\|R\|^2 + \gamma\right)}{\tilde{\sigma}_i\tilde{\sigma}_j} \\
&= \frac{\gamma\left(2\|R\|^2 + \gamma\right)}{\tilde{\sigma}_i\tilde{\sigma}_j}. \qquad \square
\end{aligned}$$

Lemma 5. *Let B be a matrix of rank at most k, and suppose that V has k columns that span the row and column spaces of B. Then*

$$\|B\| \leq \left\|(V^\dagger V)^{-1}\right\|\left\|V^\dagger B V\right\|.$$

Proof. Let $G := V^\dagger V$ be the Gram matrix of V and let $\tilde{V} := VG^{-\frac{1}{2}}$. It is easy to see that $\tilde{V}$ is an isometry and its columns still span the the row and column spaces of B. Since $\tilde{V}$ is an isometry we get that

$$\|B\| = \left\|\tilde{V}^\dagger B \tilde{V}\right\| = \left\|G^{-\frac{1}{2}} V^\dagger B V G^{-\frac{1}{2}}\right\| \leq \left\|G^{-1}\right\|\left\|V^\dagger B V\right\| = \left\|(V^\dagger V)^{-1}\right\|\left\|V^\dagger B V\right\|. \qquad \square$$

Lemma 6 (Approximate left and right singular vectors). *Suppose that $\tilde{v}^{(i)}$ is a system of approximately orthonormal vectors spanning the row space of A such that*

$$|\langle \tilde{v}^{(i)}|\tilde{v}^{(j)}\rangle - \delta_{ij}| \leq \alpha \leq \frac{1}{4k}, \tag{3}$$

and

$$\left|\langle \tilde{v}^{(i)}|R^\dagger R|\tilde{v}^{(j)}\rangle - \delta_{ij}\tilde{\sigma}_i^2\right| \leq \beta,$$

where $\tilde{\sigma}_i^2 \geq \frac{4}{5\kappa^2}$. Suppose that $\mathrm{rank}(A) = \mathrm{rank}(R) = k$ and $\left\|A^\dagger A - R^\dagger R\right\| \leq \theta$, then

$$\left\|\Pi_{\mathrm{rows}(A)} - \sum_{\ell=1}^{k} \frac{|\tilde{v}^{(\ell)}\rangle\langle\tilde{v}^{(\ell)}|}{\tilde{\sigma}_\ell^2} A^\dagger A\right\| \leq \frac{8k}{3}(\beta\kappa^2 + \theta\kappa^2 + \alpha). \tag{4}$$

Proof. Let $B := \sum_{\ell=1}^{k} \frac{|\tilde{v}^{(\ell)}\rangle\langle\tilde{v}^{(\ell)}|}{\tilde{\sigma}_\ell^2} A^\dagger A - \Pi_{\mathrm{rows}(A)}$, we will apply Lemma 5. For this observe

$$\left|\langle\tilde{v}^{(i)}|B|\tilde{v}^{(j)}\rangle\right| = \left|\sum_{\ell=1}^{k} \frac{\langle\tilde{v}^{(i)}|\tilde{v}^{(\ell)}\rangle\langle\tilde{v}^{(\ell)}|A^\dagger A|\tilde{v}^{(j)}\rangle}{\tilde{\sigma}_\ell^2} - \langle\tilde{v}^{(i)}|\tilde{v}^{(j)}\rangle\right|$$

$$\leq \left|\sum_{\ell=1}^{k} \frac{\langle\tilde{v}^{(i)}|\tilde{v}^{(\ell)}\rangle\langle\tilde{v}^{(\ell)}|R^\dagger R|\tilde{v}^{(j)}\rangle}{\tilde{\sigma}_\ell^2} - \delta_{ij}\right| + \sum_{\ell=1}^{k} \frac{|\langle\tilde{v}^{(i)}|\tilde{v}^{(\ell)}\rangle|\theta\|\tilde{v}^{(\ell)}\|\|\tilde{v}^{(j)}\|}{\tilde{\sigma}_\ell^2} + \alpha$$

$$\leq \left|\sum_{\ell=1}^{k} \frac{\langle\tilde{v}^{(i)}|\tilde{v}^{(\ell)}\rangle\langle\tilde{v}^{(\ell)}|R^\dagger R|\tilde{v}^{(j)}\rangle}{\tilde{\sigma}_\ell^2} - \delta_{ij}\right| + 2\theta\kappa^2 + \alpha$$

$$\leq \left|\sum_{\ell\neq j}^{k} \frac{\langle\tilde{v}^{(i)}|\tilde{v}^{(\ell)}\rangle\langle\tilde{v}^{(\ell)}|R^\dagger R|\tilde{v}^{(j)}\rangle}{\tilde{\sigma}_\ell^2}\right| + \left|\langle\tilde{v}^{(i)}|\tilde{v}^{(j)}\rangle\frac{\langle\tilde{v}^{(j)}|R^\dagger R|\tilde{v}^{(j)}\rangle}{\tilde{\sigma}_j^2} - \delta_{ij}\right| + 2\theta\kappa^2 + \alpha$$

$$\leq \left|\sum_{\ell\neq j}^{k} \frac{\langle\tilde{v}^{(i)}|\tilde{v}^{(\ell)}\rangle\langle\tilde{v}^{(\ell)}|R^\dagger R|\tilde{v}^{(j)}\rangle}{\tilde{\sigma}_\ell^2}\right| + \alpha(1 + \beta/\sigma_j^2) + \delta_{ij}\beta/\sigma_j^2 + 2\theta\kappa^2 + \alpha$$

$$\leq (1 + k\alpha)\beta\frac{5}{4}\kappa^2 + 2\theta\kappa^2 + 2\alpha$$

$$\leq 2(\beta\kappa^2 + \theta\kappa^2 + \alpha).$$

Let $e_\ell \in \mathbb{C}^k$ denote the ℓ-th standard basis vector and let us define $V := \sum_{\ell=1}^{k} |\tilde{v}^{(\ell)}\rangle\langle e_\ell|$. It follows that $\left\|V^\dagger B V\right\| \leq 2k(\beta\kappa^2 + \theta\kappa^2 + \alpha)$. By (3) we have that $\left\|V^\dagger V - I\right\| \leq k\alpha \leq 1/4$, and thus $\left\|(V^\dagger V)^{-1}\right\| \leq 4/3$. By Lemma 5 we get that $\|B\| \leq 8k(\beta\kappa^2 + \theta\kappa^2 + \alpha)/3$. $\qquad \square$

If $\gamma \leq \frac{1}{10\kappa^2}$ and $\theta \leq \frac{1}{10\kappa^2}$, we get $\tilde{\sigma}_{\min}^2 \geq \frac{4}{5\kappa^2}$. Then by Lemma 4 we get that $\alpha \leq \frac{5}{4}\kappa^2\gamma$ and $\beta \leq 3\kappa^2\gamma$. Substituting this into Equation (4) we get the upper bound

$$\gamma\left(8k\kappa^4 + 10k\kappa^2/3\right) + \theta\frac{8k}{3}\kappa^2 \leq \gamma 12k\kappa^4 + \theta\frac{8k}{3}\kappa^2. \tag{5}$$

Choosing $\theta = \frac{1}{16}\frac{\varepsilon}{k\kappa^2}$ and $\gamma = \frac{1}{36}\frac{\varepsilon}{k\kappa^4}$, the above bound (5) becomes $\varepsilon/2$. Therefore to succeed with probability at least $1 - \eta/2$ it suffices to sample $r = 2^{10}\ln(8n/\eta)\kappa^4 k^2\|A\|_F^2/\varepsilon^2$ row indices, and then subsequently $c = 2^6 \cdot 3^4 \ln(8r/\eta)\kappa^8 k^2\|A\|_F^2/\varepsilon^2$ column indices as shown by Theorem 3.

3.2 The required precision for matrix element estimation

Recall from Equation (2) that

$$x' = \sum_{\ell=1}^{k} \frac{\lambda_\ell}{\tilde{\sigma}_\ell^2}\tilde{v}^{(\ell)},$$

and $\tilde{x}$ is as above except we replace λ_ℓ with $\tilde{\lambda}_\ell$. As we argued in the beginning of the section, for the correctness of Algorithm 1 it suffices to ensure $\|\tilde{x} - x'\| = \mathcal{O}(\varepsilon)$, assuming that $\|b\| = 1$. Now we show that if we have $\left|\lambda_\ell - \tilde{\lambda}_\ell\right| = \mathcal{O}\left(\frac{\varepsilon\tilde{\sigma}_\ell^2\|b\|}{\sqrt{k}}\right)$, then the magnitude of perturbation can be bounded by $\mathcal{O}(\varepsilon)$, and we also get that $\|w\| = \mathcal{O}\left(\kappa^2\sqrt{k}\right)$. Let $e_\ell \in \mathbb{C}^k$ denote the ℓ-th standard basis vector; we rewrite $\|\tilde{x} - x'\|$ as

$$\left\|\sum_{\ell=1}^{k}\frac{\lambda_\ell - \tilde{\lambda}_\ell}{\tilde{\sigma}_\ell^2}|\tilde{v}^{(\ell)}\rangle\right\| = \sqrt{\left\|\sum_{\ell=1}^{k}\frac{\lambda_\ell - \tilde{\lambda}_\ell}{\tilde{\sigma}_\ell^2}|\tilde{v}^{(\ell)}\rangle\langle e_\ell|e_\ell\rangle\right\|^2} = \sqrt{\left\|\left(\sum_{\ell=1}^{k}|\tilde{v}^{(\ell)}\rangle\langle e_\ell|\right)\left(\sum_{\ell=1}^{k}\frac{\lambda_\ell - \tilde{\lambda}_\ell}{\tilde{\sigma}_\ell^2}|e_\ell\rangle\right)\right\|^2}.$$

Let us define $V := \sum_{\ell=1}^{k}|\tilde{v}^{(\ell)}\rangle\langle e_\ell|$, and $|z\rangle := \sum_{\ell=1}^{k}\frac{\lambda_\ell - \tilde{\lambda}_\ell}{\tilde{\sigma}_\ell^2}|e_\ell\rangle$, then we have that

$$\left\|\sum_{\ell=1}^{k}\frac{\lambda_\ell - \tilde{\lambda}_\ell}{\tilde{\sigma}_\ell^2}|\tilde{v}^{(\ell)}\rangle\right\| = \sqrt{\langle z|V^\dagger V|z\rangle} \le \sqrt{\|V^\dagger V\|}\|z\| = \mathcal{O}(\varepsilon),$$

where we used that $\left\|V^\dagger V\right\| \le 1 + k\alpha \le \frac{4}{3}$ as we have shown in the proof of Lemma 6.

Now we show that $\|w\| = \mathcal{O}\left(\kappa^2\sqrt{k}\right)$. Remember that $\tilde{v}^{(\ell)} = \frac{R^\dagger w^{(\ell)}}{\tilde{\sigma}_\ell}$, thus $\tilde{x} = R_{.\ell}^\dagger\sum_{\ell=1}^{k}\frac{\tilde{\lambda}_\ell}{\tilde{\sigma}_\ell^3}w^{(\ell)}$. Let $w := \sum_{\ell=1}^{k}\frac{\tilde{\lambda}_\ell}{\tilde{\sigma}_\ell^3}w^{(\ell)}$, then we get

$$\|w\| = \sqrt{\sum_{\ell=1}^{k}\frac{|\tilde{\lambda}_\ell|^2}{\tilde{\sigma}_\ell^2}} \le \sqrt{\sum_{\ell=1}^{k}\frac{|\lambda_\ell|^2}{\tilde{\sigma}_\ell^6}} + \sqrt{\sum_{\ell=1}^{k}\frac{|\tilde{\lambda}_\ell - \lambda_\ell|^2}{\tilde{\sigma}_\ell^6}} \le \mathcal{O}(\kappa^2)\sqrt{\sum_{\ell=1}^{k}\frac{|\lambda_\ell|^2}{\tilde{\sigma}_\ell^2}} + \mathcal{O}(\kappa^2\varepsilon).$$

Finally observe that

$$\sum_{\ell=1}^{k}\frac{|\lambda_\ell|^2}{\tilde{\sigma}_\ell^2} = \sum_{\ell=1}^{k}\frac{\langle b|A|\tilde{v}^{(\ell)}\rangle\langle\tilde{v}^{(\ell)}|A^\dagger|b\rangle}{\tilde{\sigma}_\ell^2} \le \mathrm{Tr}\left[\sum_{\ell=1}^{k}\frac{A|\tilde{v}^{(\ell)}\rangle\langle\tilde{v}^{(\ell)}|A^\dagger}{\tilde{\sigma}_\ell^2}\right] = \mathrm{Tr}\left[\sum_{\ell=1}^{k}\frac{\langle\tilde{v}^{(\ell)}|A^\dagger A|\tilde{v}^{(\ell)}\rangle}{\tilde{\sigma}_\ell^2}\right]$$

$$\le \sum_{\ell=1}^{k}\frac{\langle\tilde{v}^{(\ell)}|R^\dagger R|\tilde{v}^{(\ell)}\rangle}{\tilde{\sigma}_\ell^2} + \mathcal{O}(k\kappa^2)\left\|A^\dagger A - R^\dagger R\right\| \le \mathcal{O}(k + k\beta\kappa^2 + k\theta\kappa^2) \le \mathcal{O}(k + \varepsilon),$$

where the last two inequalities follow from Lemma 6 and its follow-up discussion.

4 Complexity of Algorithm 1

The complexity is dominated by two parts of the algorithm: finding the left singular vectors of an r by c matrix, and estimating some matrix elements of A. If we use naive matrix multiplication, then computing the singular value decomposition of $CC^\dagger$ costs

$$\mathcal{O}\left(r^2 c\right) = \tilde{\mathcal{O}}\left(\kappa^{16}k^6\frac{\|A\|_F^6}{\varepsilon^6}\right).$$

In this section, we prove that this dominates the runtime of the algorithm. First, we use length-square sampling techniques similarly to Tang [Tan18a] to approximate the matrix elements $\lambda_\ell := \langle \tilde{v}^{(\ell)} | A^\dagger | b \rangle$, which has complexity $\widetilde{\mathcal{O}}\left(\kappa^8 k^4 \frac{\|A\|_F^4}{\varepsilon^4} \right)$ as we show in Section 4.2. Second, we show how to efficiently length-square sample from $\tilde{x} := \sum_{\ell=1}^{k} \frac{\tilde{\lambda}_\ell}{\tilde{\sigma}_\ell^2} \tilde{v}^{(\ell)}$ using rejection sampling.

4.1 Length-square sampling techniques

Definition 7 (Length-square distribution). *For a non-zero vector $v \in \mathbb{C}^n$ we define the probability distribution $q^{(v)}$ on $[n]$ such that $q_i^{(v)} = \frac{|v_i|^2}{\|v\|^2}$.*

Note that if v describes a normalized pure quantum state, the above distribution is exactly the distribution we get through measurement in the computational basis by the Born rule.

Definition 8 (Length-square access to a vector). *We say that we have length-square access to the vector $v \in \mathbb{C}^n$ if we can request a sample from the distribution $q^{(v)}$ that takes cost $S(v)$. We also assume that we can query the elements of v with cost $Q(v)$, and that we can query the value of $\|v\|$ with cost $N(v)$.[6] We denote by $L(v) := S(v) + Q(v) + N(v)$ the overall access cost.*

If the matrix A is stored in an tree-like dynamic data structure [FKV04, KP17, Tan18a], then the complexity of length-square accessing A is $\mathcal{O}(\log(mn))$. During the complexity analysis we will assume such efficient access.

Definition 9 (Length-square access to a matrix). *We say that we have (row) length-square access to the matrix $A \in \mathbb{C}^{m \times n}$ if we have length-square access to the rows $A_{i\cdot}$ of A for all $i \in [m]$ and length-square access to the vector of row norms $a \in \mathbb{R}^m$, where $a_i := \|A_{i\cdot}\|$. We denote by $L(A)$ the complexity of the length-square access to A.*

Note that length-square access to A implies the ability to determine $\|A\|_F$ in $N(A)$ time. We will use that the closeness of two vectors in Euclidean distance implies closeness of their corresponding distributions.

Lemma 10 (Bounding Total Variation distance by Euclidean distance [Tan18a, Lemma 6.1]). *For $v, w \in \mathbb{C}^n$, $\left\| q^{(v)}, q^{(w)} \right\|_{TV} \leq \frac{2\|v-w\|}{\max(\|v\|, \|w\|)}$.*

4.2 Estimating the matrix element $\langle \tilde{v}^{(\ell)} | A^\dagger | b \rangle$

We use the inner product estimation method of Tang [Tan18a] for matrix element estimation.

Lemma 11 (Trace inner product estimation). *Suppose that we have length-square access to $A \in \mathbb{C}^{m \times n}$ and query access to the matrix $B \in \mathbb{C}^{m \times n}$ in complexity $Q(B)$. Then we can estimate $\mathrm{Tr}\left[A^\dagger B \right]$ to precision $\xi \|A\|_F \|B\|_F$ with probability at least $1 - \eta$ in time*

$$\mathcal{O}\left(\frac{\log(1/\eta)}{\xi^2} (L(A) + Q(B)) \right).$$

Proof. This exactly follows from [Tan18a, Proposition 6.2], since $\mathrm{Tr}\left[A^\dagger B \right]$ is the inner product of order-two tensors. Let X be the random variable given by length-square sampling i from a, the vector of row norms of A, sampling j from $A_{i\cdot}$, and setting the random output to $X = \frac{\|A\|_F^2}{A_{ij}} B_{ij}$. Then

$$\mathbb{E}[X] = \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{|A_{ij}|^2}{\|A\|_F^2} \frac{\|A\|_F^2}{A_{ij}} B_{ij} = \sum_{j=1}^{n} \sum_{i=1}^{m} (A^\dagger)_{ji} B_{ij} = \mathrm{Tr}\left[A^\dagger B \right]$$

$$\mathbb{E}[|X|^2] = \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{|A_{ij}|^2}{\|A\|_F^2} \frac{\|A\|_F^4}{|A_{ij}|^2} |B_{ij}|^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} \|A\|_F^2 |B_{ij}|^2 = \|A\|_F^2 \|B\|_F^2.$$

---

[6] We assume for simplicity that $S(v), Q(v)$, and $N(v) \geq 1$.

So X is an unbiased estimator of $\mathrm{Tr}\big[A^\dagger B\big]$. To compute the expectation, we use standard techniques: it suffices to estimate the the real and imaginary parts separately to additive precision $\xi\|A\|_F\|B\|_F/\sqrt{2}$ with success probability at least $1-\eta/2$. For each, we compute the mean of $\frac{9}{\xi^2}$ copies of X, and take the median of $6\log(2/\eta)$ such empirical mean estimators, giving the desired result. $\square$

We can estimate $\lambda_\ell = \langle\tilde{v}^{(\ell)}|A^\dagger|b\rangle = \mathrm{Tr}\big[\langle\tilde{v}^{(\ell)}|A^\dagger|b\rangle\big] = \mathrm{Tr}\big[A^\dagger|b\rangle\langle\tilde{v}^{(\ell)}|\big]$ using this lemma. Observe that $\big\||b\rangle\langle\tilde{v}^{(\ell)}|\big\|_F = \big\|\tilde{v}^{(\ell)}\big\|\|b\| \le (1+\varepsilon)\|b\|$, and we can query the (i,j) matrix element of $|b\rangle\langle\tilde{v}^{(\ell)}|$ by querying b_i and $\tilde{v}_j^{(\ell)}$, which has $\widetilde{\mathcal{O}}(1)$ and $\widetilde{\mathcal{O}}(r)$ cost respectively. We desire to estimate λ_ℓ to additive precision $\mathcal{O}\Big(\frac{\varepsilon\sigma_\ell^2\|b\|}{\sqrt{k}}\Big)$ with success probability $\frac{\eta}{2k}$. By applying Lemma 11, we can compute such an estimate $\tilde{\lambda}_\ell$ with complexity

$$\widetilde{\mathcal{O}}\left(\log(2k/\eta)\frac{k\|A\|_F^2}{\varepsilon^2\sigma_\ell^4}r\right) = \widetilde{\mathcal{O}}\left(\kappa^4 k\frac{\|A\|_F^2}{\varepsilon^2}r\right) = \widetilde{\mathcal{O}}\left(\kappa^8 k^3\frac{\|A\|_F^4}{\varepsilon^4}\right).$$

4.3 Sampling from the approximate solution

Our goal is to sample from the length-square distribution of $\tilde{x} = R^\dagger w$. In order to tackle this problem we invoke a result from [Tan18a] about length-square sampling a vector that is a linear-combination of length-square accessible vectors. For completeness we present its proof too, following the approach of Tang [Tan18a].

Lemma 12 (Length-square sample a linear combination of vectors [Tan18a, Proposition 6.4]). *Suppose that we have length-square access to $R \in \mathbb{C}^{r\times n}$ having normalized rows, and we are given $w \in \mathbb{C}^r$ (as a list of numbers in memory). Then we can implement queries to the vector $y := R^\dagger w \in \mathbb{C}^n$ with complexity $Q(y) = \mathcal{O}(rQ(R))$ and we can length-square sample from $q^{(y)}$ with complexity $S(y)$ such that $\mathbb{E}[S(y)] = \mathcal{O}\Big(\frac{r\|w\|^2}{\|y\|^2}(S(R)+rQ(R))\Big)$.*

Proof. The algorithm is simple, it proceeds by rejection sampling: one should first sample a row index $i \in [r]$ uniformly, then draw a column index j distributed as $|R_{ij}|^2$, then compute $|y_j|^2 = |\langle w|R_{\cdot j}\rangle|^2$, $\|R_{\cdot j}\|^2$ and either output j with probability $\frac{|\langle w|R_{\cdot j}\rangle|^2}{\|w\|^2\|R_{\cdot j}\|^2}$ or sample (i,j) again. In each round the probability that we pick the column index j is $\sum_{j=1}^r |R_{ij}|^2/r = \|R_{\cdot j}\|^2/r$, and the probability that we output j is $|y_j|^2/(r\|w\|^2)$. The success probability in each round is $\|y\|^2/(r\|w\|^2)$, therefore the expected number of rounds is $r\|w\|^2/\|y\|^2$. $\square$

Since all rows of R have norm $\|A\|_F/\sqrt{r}$, and $\|\tilde{x}\| = \Omega(1)$ by Lemma 12 we can length-square sample from $\tilde{x}$ in expected complexity

$$\mathcal{O}\left(\frac{r\|w\|^2\|A\|_F^2/r}{\|\tilde{x}\|^2}r\right) = \mathcal{O}\left(\frac{\|w\|^2\|A\|_F^2}{\|\tilde{x}\|^2}r\right) = \mathcal{O}\left(\kappa^4 k\|A\|_F^2 r\right) = \widetilde{\mathcal{O}}\left(\frac{\kappa^8 k^3\|A\|_F^4}{\varepsilon^2}\right).$$

5 Discussion

We presented a proof-of-principle algorithm for approximately inverting low-rank matrices in runtime that is logarithmic in the dimensions. For simplicity we analysed the case when the matrix has low rank, however it should be possible to devise similar results when the matrix does not have low rank, but one only intends to invert the matrix on a "well-conditioned" subspace. We expect that the complexity can be further improved by using optimized algorithms for finding an approximate singular vale decomposition of the subsampled matrix C. Also, one might use another variant of the algorithm where the left singular vectors of A are approximated using some variant of the FKV algorithm [FKV04], instead of the right singular vectors. Another approach

could be to use a different low-rank approximation method, which reconstructs an approximation of A by using a linear combination of rows and columns such as described in [KV17].

Although in this paper we focus on implementing the pseudo-inverse of a matrix by inverting the singular values, one could in principle apply any desired function to the singular values. It has recently been shown that many quantum algorithms can be expressed as a singular value transformation problem [GSLW18]. This supports Tang's suggestion [Tan18b] that many quantum algorithms can be effectively turned to randomized classical algorithms via length-square sampling techniques incurring only polynomial overheads. Our work gives evidence that this conversion can be done in general for low-rank problems, suggesting that exponential quantum speed-ups are tightly related to problems where high-rank matrices play a crucial role, like in Hamiltonian simulation or the Fourier transform. However, more work remains to be done on understanding the class of problems for which exponential quantum speed-up can be achieved.

Acknowledgments

References

[CGJ18]   Shantanav Chakraborty, András Gilyén, and Stacey Jeffery. The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation. arXiv: 1804.01973, 2018.

[FKV04]   Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, 2004.

[GSLW18] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. arXiv: 1806.01838, 2018.

[HHL09]   Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009. arXiv: 0811.3171

[KP17]   Iordanis Kerenidis and Anupam Prakash. Quantum recommendation systems. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 49:1–49:21, 2017. arXiv: 1603.08675

[KV17]   Ravindran Kannan and Santosh Vempala. Randomized algorithms in numerical linear algebra. *Acta Numerica*, 26:95–135, 2017.

[RL18]   Patrick Rebentrost and Seth Lloyd. Quantum computational finance: quantum algorithm for portfolio optimization. arXiv: 1811.03975, 2018.

[RSML18] Patrick Rebentrost, Adrian Steffens, Iman Marvian, and Seth Lloyd. Quantum singular-value decomposition of nonsparse low-rank matrices. *Physical Review A*, 97:012327, 2018. arXiv: 1607.05404

[Tan18a] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. *Electronic Colloquium on Computational Complexity*, page 128, 2018. arXiv: 1807.04271

[Tan18b] Ewin Tang. Quantum-inspired classical algorithms for principal component analysis and supervised clustering. arXiv: 1811.00414, 2018.