J. Appl. Math. & Informatics Vol. **4x***(2023), No. x, pp. xx - xx*
https://doi.org/10.14317/jami.201x.xxx

A SURVEY ON ALTERNATIVE QUANTUM INPUT MODELS[†]

HYEONMIN ROH*, TAEWON KIM**

Abstract. This paper provides an expository expansion to an open problem in quantum machine learning; whether there are alternative data structures that prevent efficient classical counterparts for quantum machine learning algorithms.

AMS Mathematics Subject Classification : 65H05, 65F10.
Key words and phrases :

1. Introcution

Quantum machine learning (QML) was activated by the HHL algorithm [3] that approximately solves a system of linear equations in a logarithmic time. However, as Aaronson [4] critiqued, HHL and other QML algorithms involved quantum advantageous *input assumptions* or *data structures.* Critically, Tang [9] introduced *dequantization*, a method of providing efficient classical counterparts for large amount of QML algorithms by randomized-linear algebraic exploitations of quantum-advantageous assumptions, hence demystifying claimed exponential speedups of QML algorithms. This paper focus on the problem of constructing data structrues that prevent dequantization, that is, an open problem stated in Tang's thesis [12] with implicit examples by Chakraborty, Gilyén and Jeffry [8] and Kerenidis and Prakash [10].

2. Preliminaries

By practical or industrial reasons, the majority of data given for QML are classical.

Received , . Revised , . Accepted , . *Corresponding author.
[†]Please add: "This research received no external funding" or "This research was funded by NAME OF FUNDER grant number XXX." and and "The APC was funded by XXX". Check carefully that the details given are accurate and use the standard spelling of funding agency names, any errors may affect your future funding. Example:This work was supported by the research grant of the University

Definition 2.1. Classical data is an element of n-fold Cartesian product $\{0,1\}^n$.

To utillize classical data within a quantum computer, there must be an efficient way or a structure to prepare those data inputs into quantum states, where quantum states are defined as follows.

Definition 2.2. For $a_x \in \mathbb{C}$, if $\sum_x |a_x|^2 = 1$, sum of ℓ_2-normalized vectors in a complex vector space

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} a_x \, |x\rangle$$

is the state of n qubits.

We simply assumme that such efficient preperation by supposing Quantum Random Access Memory (QRAM), a classical memory that can be accessed by quantum algorithms. Incidentally, QRAM is quite a broad term as noted in survey of Jaques and Rattew [11], but following abstraction might suffice for our discussion. Beforehand, it is useful to introduce following abbreviation

$$|\psi\rangle \otimes |\phi\rangle = |\psi\rangle \, |\phi\rangle = |\psi, \phi\rangle$$

since $|\cdot\rangle$ denotes a column vector.

Definition 2.3. For classical data $t_i \in \{0,1\}^n$, existence of QRAM is an assumption that there exist unitaries of the form

$$U \, |i, 0\rangle = |i, t_i\rangle$$

and such action of U requires $\mathcal{O}(\text{polylog}(n))$ time.

Classical data t_i constitues a classical memory T. Register of $|i\rangle$ is called the address register. Time complexity of U expressed in big-oh notation means that there exists constants c and n_0 both equal or greater than 0 such that $T(n)$, a time function of U for n, is equal or less than the polylogarithmic function of n multiplied by c for all $n \geq n_0$. Another useful notation in algorithmic analysis is $\widetilde{\mathcal{O}}(f(n))$, shorthand for $\mathcal{O}(g(n) \log^k g(n))$.

3. Quantum accesibble classical data structure

QRAM enables input models or *data structures* for processing matrix input in a quantum adventageous way. We restate such data structure utilized in the quantum recommendation algorithm by Kerenidis [5], first QML algorithm to be dequantized by Tang [9]. In advance, we check state preparation method by Grover and Rudolph [2].

Corollary 3.1. *Let a probability distribution $\{p_i\}$ such that $p_i^{(m)}$ is the probability for the random variable x to lie in the ith region, and $p_s^{(m)}$ is the probability for x to lie in sth region. For an orthonormal state $|i\rangle$, let a m-qubit state as follows*

$$|\psi_m\rangle = \sum_{i=0}^{2^m-1} \sqrt{p_i^{(m)}} \, |i\rangle \, .$$

Then, we can get the mapping of

$$|\psi_{m+1}\rangle = \sum_{i=0}^{2^{m+1}-1} \sqrt{p_i^{(m+1)}}\, |i\rangle\, .$$

Proof. Let $n = \log N$ for the total number of points N to discretize the distribution $p(x)$. That is, we divide the distribution with 2^m of regions for some m. We subdivide these 2^m regions to yield a 2^{m+1} region discretization of $p(x)$.

First, we define the left and right boundaries of region i as x_L^i and x_R^i with the function

$$f(i) = \frac{\int_{x_L^i}^{(x_R^i - x_L^i)/2} p(x)dx}{\int_{x_L^i}^{x_R^i} p(x)dx}.$$

That is, for x in the region i, the probability of x also lying in the left half of this region. Since an integral on $p(x)$ can be classicaly computed efficiently, we take an ancillia register initially in the state $|0\ldots0\rangle$ and construct a circuit which efficiently performs the computation

$$\sqrt{p_i^{(m)}}\, |i, 0\cdots0\rangle \mapsto \sqrt{p_i^{(m)}}\, |i, \theta_i)\rangle$$

where $\theta_i \equiv \arccos\sqrt{f(i)}$. Then, by a controlled rotation of angle θ_i on the $m+1$'th qubit, we have

$$\sqrt{p_i^{(m)}}\, |i, \theta_i, 0\rangle \mapsto \sqrt{p_i^{(m)}}\, |i, \theta_i\rangle\, (\cos\theta_i\, |0\rangle + \sin\theta_i\, |1\rangle)$$

where we can uncompute the register containing $|\theta_i\rangle$ with

$$\sqrt{p_i^{(m)}}\, |i\rangle \mapsto \sqrt{\alpha_i}\, |i, 0\rangle + \sqrt{\beta_i}\, |i, 1\rangle$$

where α and β are the probability for x to lie in the left or right half of the region i, respectivly. We apply the mapping above for m-qubit states given as

$$|\psi_m\rangle = \sum_{i=0}^{2^m-1} \sqrt{p_i^{(m)}}\, |i\rangle\, .$$

Then, we get $\sqrt{\psi_{m_1}}$ as desired. □

Lemma 3.2. *Given a certain probability distribution $\{p_i\}$, we can efficiently construct a superposition of*

$$|\psi(\{p_i\})\rangle = \sum_i \sqrt{p_i}\, |i\rangle\, .$$

Proof. By repeating the cororally above until $m = n$, we can efficiently construct a superposition as desired. □

Example 3.3. Let a 4-dimensional state of $|\phi\rangle = 0.4\,|00\rangle + 0.4\,|01\rangle + 0.8\,|10\rangle + 0.2\,|11\rangle$. Then, by Lemma 3.2, we apply rotation on the first qubit, that yields:

$$|0, 0\rangle \mapsto (\sqrt{0.32}\, |0\rangle + |0.68\rangle\, |1\rangle)\, |0\rangle\, .$$

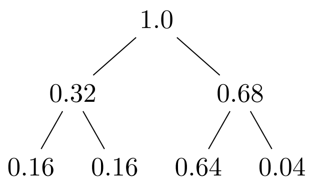

FIGURE 1. Vector state preparation for 4-dimensional state $|\phi\rangle$

Untill $m = n$, we apply rotation on the next qubit, conditioned on the last one:

$$(\sqrt{0.32}\,|0\rangle + \sqrt{0.68})\,|0\rangle \mapsto \begin{array}{l} \sqrt{0.32}\,|0\rangle \dfrac{1}{\sqrt{0.32}}(0.4\,|0\rangle + 0.4\,|1\rangle) \\ \quad + \\ \sqrt{0.68}\,|1\rangle \dfrac{1}{\sqrt{0.68}}(0.8\,|0\rangle + 0.2\,|1\rangle). \end{array}$$

This result, which may have alternatives by construction details, is used in the data structure below to to efficiently perform $\widetilde{U}$ and $\widetilde{V}$.

Theorem 3.4 ([5] Theorem 15)**.** *For a real matrix $A \in \mathbb{R}^{m\times n}$, let (i, j, A_{ij}) be given data. If we assume QRAM, then, there exists a data structure representable by binary search trees to store the matrix A with following properties:*

(1) *The size of the data structure is $\mathcal{O}(w \log^2(mn))$ where w is the number of data entries already in the tree.*
(2) *The time to store a new data (i, j, A_{ij}) is $\mathcal{O}(\log^2(mn))$.*
(3) *There exists a quantum algorithm that takes* polylog(mn) *time to perform the mapping*

$$\widetilde{U} : |i, 0\rangle \to |i, A_i\rangle$$

and for $\widetilde{A} \in \mathbb{R}^m$ with entries $\widetilde{A_i} = \|A_i\|$,

$$\widetilde{V} : |0, j\rangle \to |\widetilde{A}, j\rangle\,.$$

Proof. The data structure consits m binary trees B_i, where $i \in [m]$. Leaf node is created or updated by the arrival of new entry (i, j, A_{ij}), storing the value A_{ij}^2, where internal node v stores the sum of the values of all leaves in the subtree rooted at v. Hence, the value stored at the root is $\|A_i\|^2$.

The depth of the tree is at most $\lceil \log n \rceil$ for most n leaves, which is the number of updates required for a new entry. If we store each tree sorted as ordered list, update node address retrival time and bits required for memory would be $\mathcal{O}(\log mn)$. Thus, the size of the data structure is $\mathcal{O}(w \log^2(mn))$ and the time to store a new entry is $\mathcal{O}(\log^2(mn))$.

Now let a quantum algorithm has an access to these m-binary trees. We apply Lemma 3.2, that is, we apply a sequence of conditional rotations to the initial state $|0\rangle^{\lceil \log n \rceil}$ to obtain $|A_i\rangle$. For B_{i,t_i}, conditioned on the first register

$$\begin{tikzcd}
 & & & \|A_i\|^2 \arrow[dll] \arrow[drr] & & & \\
 & \|A_{i1}\|^2 + \|A_{i2}\|^2 \arrow[dl] \arrow[dr] & & & & \|A_{i3}\|^2 + \|A_{i4}\|^2 \arrow[dl] \arrow[dr] & \\
\|A_{i1}\|^2 \arrow[d] & & \|A_{i2}\|^2 \arrow[d] & & \|A_{i3}\|^2 \arrow[d] & & \|A_{i4}\|^2 \arrow[d] \\
A_{i1}^2 & & A_{i2}^2 & & A_{i3}^2 & & A_{i4}^2
\end{tikzcd}$$

FIGURE 2. example for $A^{i\times 4}$

being $|i\rangle$ and the first d qubits begin in state $|t_i\rangle$ for the depth d, the rotation is applied to the $d+1$ qubit as follows

$$|i, t_i, 0\rangle \mapsto |i, t_i\rangle \frac{1}{\sqrt{B_{i,t_{i0}}}} \left(|0\rangle + \sqrt{B_i, t_{i1}}\, |1\rangle\right).$$

So there are $\lceil \log n \rceil$ rotations applied and for each rotations we need two quantum queries from each node in the superposition. This process also holds for $\widetilde{V}$. Hence, polylog(mn) time is required. $\square$

This data structure constitues an array of m binary trees. The value stored at the root is $\|A_i\|^2$ for $i \in [m]$, and depth of each tree is at most $\lceil \log n \rceil$. For a fair evaluation of quantum speedup provided by such QRAM-dependant data stucture, dequantization starts from constructing equivalent binary search tree with only a polynomial slowdown.

Definition 3.5 ([12] Definition 4.1)**.** For all $i \in [n]$, if we can query for $v(i)$, we have *query access* to a vector $v \in \mathbb{C}^n$, denoted by $Q(v)$. For all $(i, j) \in [m] \times [n]$, if we can query for A_{ij}, we have $Q(A)$ to a matrix $A \in \mathbb{C}^{m\times n}$. Time cost of such query is denoted by $q(v)$ and $q(A)$, respectively.

Definition 3.6 ([12] Definition 4.2)**.** For a vector $v \in \mathbb{C}^n$, we have *sampling and query access* to v, denoted by $SQ(v)$, if we can:

(1) have query access to v;
(2) obtain independent samples $i \in [n]$ following the distribution $\mathcal{D}_v \in \mathbb{R}^n$ with $\mathcal{D}_v(i) := |v(i)|^2/\|v\|^2$;
(3) have query access to $\|v\|$.

Cost of entry querying, index sampling, norm querying, are denoted as $q(v)$, $s(v)$, and $n(v)$, respectively. Also, let $sq(v) := \max(q(v), s(v), n(v))$.

Samples obtained from sampling and query access are analogue to the quantum state $|v\rangle := 1/\|v\| \sum v_i\, |i\rangle$ in the computational basis. Such sampling and query access may be generalized by some oversampling rate.

Definition 3.7. For $v \in \mathbb{C}^n$ and $\phi \geq 1$, we have $\widetilde{v} \in \mathbb{C}^n$ if $\|\widetilde{v}\|^2 = \phi\|v\|^2$ and $|\widetilde{v}_i|^2 \geq |v_i|^2$ for all $i \in [n]$.

Definition 3.8 ([12] Definition 4.3)**.** For $v \in \mathbb{C}^n$ and $\phi \geq 1$, if $Q(v)$ and $SQ(\widetilde{v})$ for $\widetilde{v} \in \mathbb{C}^n$, we have *$\phi$-oversampling and query access to v* or $SQ_\phi(v)$. Also,

$$s_\phi(v) := s(\widetilde{v}),\ q_\phi(v) := q(\widetilde{v}),\ n_\phi(v) := n(\widetilde{v}),\ sq_\phi(v) := \max(s_\phi(v), q_\phi(v), n_\phi(v)).$$

$$\begin{tikzcd}
 & & & \|\widetilde{A}\|^2 \arrow[dll] \arrow[drr] & & & & \\
 & \|\widetilde{A_1}\|^2 \arrow[dl] \arrow[dr] & & & & \|\widetilde{A_2}\|^2 \arrow[dl] \arrow[dr] & & \\
\|\widetilde{A_{11}}\|^2 + \|\widetilde{A_{12}}\|^2 \arrow[d] \arrow[dr] & & \|\widetilde{A_{13}}\|^2 + \|\widetilde{A_{14}}\|^2 \arrow[d] \arrow[dr] & & \|\widetilde{A_{21}}\|^2 + \|\widetilde{A_{22}}\|^2 \arrow[d] \arrow[dr] & & \|\widetilde{A_{23}}\|^2 + \|\widetilde{A_{24}}\|^2 \arrow[d] \arrow[dr] & \\
\|\widetilde{A_{11}}\|^2 & \|\widetilde{A_{12}}\|^2 & \|\widetilde{A_{13}}\|^2 & \|\widetilde{A_{14}}\|^2 & \|\widetilde{A_{21}}\|^2 & \|\widetilde{A_{22}}\|^2 & \|\widetilde{A_{23}}\|^2 & \|\widetilde{A_{24}}\|^2
\end{tikzcd}$$

FIGURE 3. example of a ϕ-sampling and qeury access data structure

Note that $SQ_\phi(v)$ constructs an identical tree structure to the one from QRAM. So, we can dequantize whenever QML algorithm relies on QRAM based data structure.

4. Alternative Data Structures

However, there have been variations for such definitions of QRAM, which might prevent dequantization. We first present such variant with general approach, then with more concrete form of presentation by *block-encoding.* Theorem below constructs a more efficient variant of QRAM, where the memory requirement [10] is $\widetilde{\mathcal{O}}(mn)$ and time cost of update, insertion, or deletion for a single entry is $\mathcal{O}(\text{polylog}(n))$. of art.

Theorem 4.1 ([10] Theorem IV.2)**.** *Let $M = \max_{i\in[m]} \|a_i\|^2$ and $A \in \mathbb{R}^{m\times n}$. There eixsts an efficient QRAM data structure for storing matrix entries (i, j, a_{ij}) such that access to this data structure allows a quantum algorithm to implement following unitary in time $\widetilde{\mathcal{O}}(\log(mn))$.*

$$U\,|i, 0^{\lceil\log(n+1)\rceil}\rangle = |i\rangle \frac{1}{\sqrt{M}} \left(\sum_{j\in[n]} a_{ij}\,|j\rangle + (M - \|a_i\|^2)^{1/2}\,|n+1\rangle \right) .k$$

Before heading to the proof, it is helpful to understand utility of M.

Definition 4.2. The normalized vector state corresponding to vector $x \in \mathbb{R}^n$ and $M \in \mathbb{R}$ such that $\|x\|^2 \leq M$ is the quantum state

$$|\bar{x}\rangle = \frac{1}{\sqrt{M}} \left(\sum_{i \in [n]} x_i |i\rangle + (M - \|x\|^2)^{1/2} |n+1\rangle \right).$$

So the key variation is the norm in the input state.

Proof. m binary trees $B_i \longrightarrow$ use **rotation**

•

•

•

□

Now we define the notion of block-encoding to clarify the prevention of result above. Block-encdoing was devised during a optimal solution of Hamiltonian simulation probem, [7] originally termed as 'qubitization'. Simply put, block-encoding is a technique of representing Hermitian or subnormalized matrix as the top-left block of a unitary matrix, that is;

$$U = \begin{bmatrix} A/\alpha & \cdot \\ \cdot & \cdot \end{bmatrix}$$

where $\cdot$ denotes arbitrary elements of U.

Definition 4.3 (Gilyén 2019)**.** For $A \in \mathbb{C}^{n \times m}$, $\alpha, \epsilon \in \mathbb{R}_+$ and $a \in \mathbb{N}$, $(s+a)$-qubit unitary U is an (α, a, ϵ)-block-encoding of A if

$$\|A - \alpha(\langle 0|^{\otimes a} \otimes I)U(|0\rangle^{\otimes a} \otimes I)\| \leq \epsilon.$$

For $n, m \leq 2^s$ we may define an embedding matrix $A_e \in \mathbb{C}^{2^s \times 2^s}$ such that the top-left block of A_e is A and all other entries are 0.

Theorem 4.4 (Chakraborty, 2019, Lemma 25)**.** *Let* $A \in \mathbb{C}^{m \times n}$.

(1) *Fix* $p \in [0,1]$. *If* $A^{(p)}$ *and* $(A^{(1-p)})^\dagger$ *are both stored in quantum-accessible data structures, then there exist unitaries* U_R *and* U_L *that can be implemented in time* $\mathcal{O}(\text{polylog}(mn/\epsilon))$ *such that* $U_R^\dagger U_L$ *is a* $(\mu_p(A), \lceil \log(n+m+1) \rceil, \epsilon)$*-block-encoding of* $\bar{A}$.

(2) *On th other hand, if* A *is stored in a quantum-accessible data structure, then there exist unitaries* U_R *and* U_L *that can be implemented in time* $\mathcal{O}(\text{polylog}(mn/\epsilon))$ *such that* $U_R^\dagger U_L$ *is a* $(\|A\|_F, \lceil \log(m+n) \rceil, \epsilon)$*-block-encoding of* $\bar{A}$.

Theorem 4.5. *Chakraborty2019*

Theorem 4.6. *main result*

5. Examples and Applications

Conflicts of interest : Declare conflicts of interest or state "The authors declare no conflict of interest." Authors must identify and declare any personal circumstances or interest that may be perceived as inappropriately influencing the representation or interpretation of reported research results.

Data availability : In this section, please provide details regarding where data supporting reported results can be found, including links to publicly archived datasets analyzed or generated during the study. If the study did not report any data, you might add "Not applicable" here.

Acknowledgments : In this section you can acknowledge any support given which is not covered by the author contribution or funding sections.

References

1. C. Baiocchi and A. Capelo, *Variational and Quasi Variational Inequalities*, J. Wiley and Sons, New York, 1984.
2. L. Grover and T. Rudolph, *Creating superpositions that correspond to efficiently integrable probability distributions,"*, arXiv preprint quant-ph/0208112, (2002).
3. A. W. Harrow, A. Hassidim and S. Lloyd, *Quantum algorithm for linear Systems of equations*, Phys. Rev. Lett **103** (2009), 150502.
4. S. Aaronson, *Read the fine print*, Nature Phys **11** (2015), 291-293.
5. I. Kerenidis and A. Prakash, *Quantum Recommendation Systems*, ITCS **67** (2017), 49:1–49:21.
6. R. Babbush, C. Gidney, B. Dominic, N. Weibe, J. McClean, A. Paler, A. Fowler and H. Neven, *Encoding Electronic Spectra in Quantum Circuits with Linear T Complexity* Phys. Rv. **8** (2018), 041015.
7. H. Low and I. Chuang, *Hamiltonian Simulation by Qubitization*, Quantum **3** (2019), 163.
8. Chakraborty, Gilyén and Jeffery. *The Power of Block-Encoded Matrix Powers*, ICALP **132** (2019), 33:1-33:14.
9. E. Tang. *A Quantum-Inspired Classical Algorithm for Recommendation Systems*, Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (2019), 217–228.
10. Kerenidis, Iordanis and Prakash, Anupam, *Quantum gradient descent for linear systems and leaste squares*, Phys.Rev.A **2** (2020), 022316.
11. S. Jaques and A. Rattew, *QRAM: A Survey and Critique*, arXiv: 2305.10310 [quant-ph] (2023).
12. E. Tang, *Quantum Machine Learning Without Any Quantum*, Ph.D. diss, University of Washington (2023).
13. D. Chan and J.S. Pang, *The generalized quasi variational inequality problems*, Math. Oper. Research **7** (1982), 211-222.
14. C. Belly, *Variational and Quasi Variational Inequalities*, J. Appl. Math. and Computing **6** (1999), 234-266.
15. D. Pang, *The generalized quasi variational inequality problems*, J. Appl. Math. and Computing **8** (2002), 123-245.

1st Author name received M.Sc. from Seoul National University and Ph.D. at University of Minnesota. Since 1992 he has been at Chungnam National University. His research interests include numerical optimization and biological computation.

Department of Mathematics, Chungnam National University, Daejeon 305-764, Korea.
`e-mail: soh@cnu.ac.kr`

2nd Author name received M.Sc. from Kyungpook National University, and Ph.D. from Iowa State University. He is currently a professor at Chungbuk National University since 1991. His research interests are computational mathematics, iterative method and parallel computation.

Department of Mathematics, College of Natural Sciences, Chungbuk National University, Cheongju 361-763, Korea.
`e-mail: gmjae@chungbuk.ac.kr`