QUANTUM MACHINE LEARNING WITH SUBSPACE STATES

Anupam Prakash

QCWare.

ArXiv: 2202.00054 with Iordanis Kerenidis

QUANTUM SPEEDUPS CRITERIA

- Three criteria for quantum speedups whose intersection has proved to be remarkably hard to meet:

1. Quantum algorithm is directly ***comparable*** to classical and performs the *same* task.

2. ***Exponential speedups*** over best known classical algorithm.

3. Problems useful in practice and widely ***applicable.***

EXPONENTIAL QML SPEEDUPS?

- The amplitude encoding of N dimensional vector x requires log(N) qubits compared to an O(N) sized classical array. For N=16,

x_1	x_2	x_3	$\cdots$	x_{15}	x_{16}

$$|x\rangle = \frac{1}{||x||}(x_1|0000\rangle + x_2|0001\rangle + x_3|0010\rangle + \cdots + x_{15}|1110\rangle + x_{16}|1111\rangle)\,.$$

- Such an exponential compression raises the possibility of an exponential speedup for quantum machine learning (QML).

- Bottlenecks: State preparation requires a circuit of depth N and measurements perform ℓ_2−sampling from the output.

- QML Algorithms: Inner product estimation/kernels, Hamiltonian simulation (HHL), quantum singular value transformation (QSVE/QSVT).

READ THE FINE PRINT!

- *Input Issues:* The initial state for a linear algebra procedure may need exponential resources to prepare.

- *Output Issues:* Quantum algorithms sample from the output, a classical algorithm reconstructs the output.

- *Running time parameters:* Condition number is difficult to bound making it hard to establish speedups.

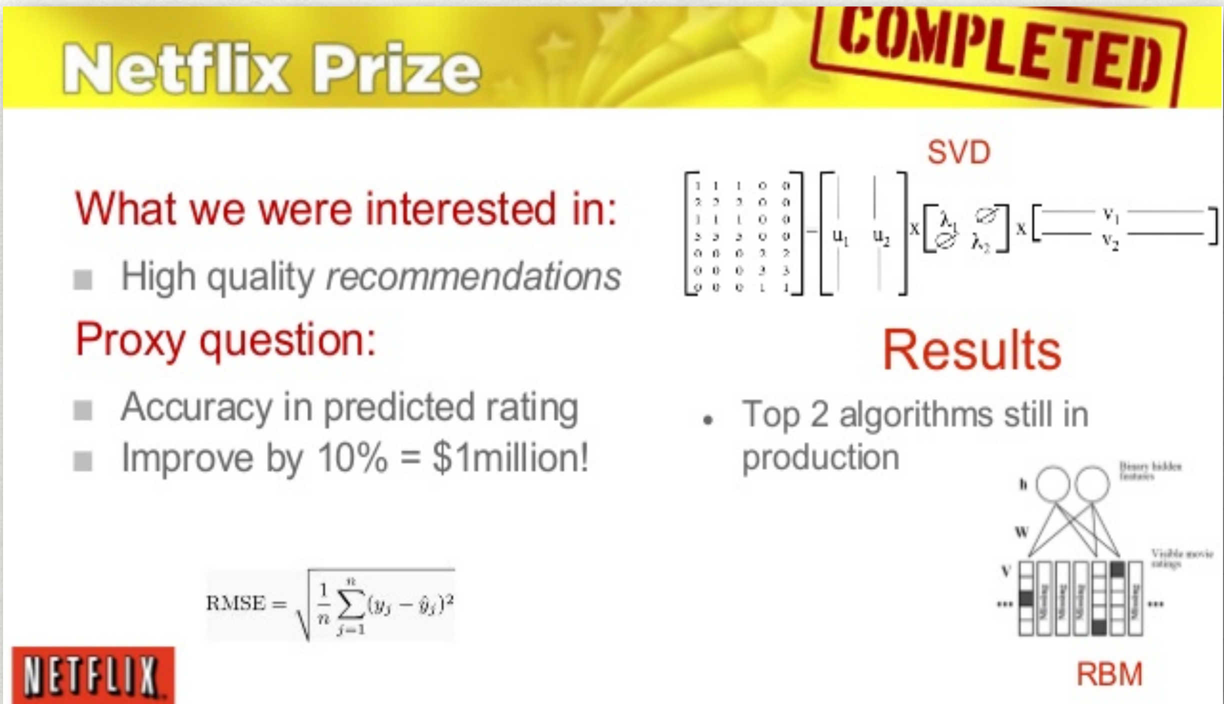
- *Dense matrices:* Matrices arising in machine learning are dense, but may often have good low rank approximations.

- *[Aaronson 14]:* Caveats make it difficult to establish end-to-end speedups for QML algorithms.

TOWARDS END-TO-END QML

- *Input Issues: QRAM data structures can be replaced with parametrized circuits, logarithmic depth circuits for unary encoding.*
- *Output Issues: End-to-end quantum speedups for sampling tasks: recommendation systems [KP17], determinant sampling [KP22], fermion and boson samplers.*
- *Running time parameters: Low rank quantum linear algebra depends on numerical rank. Improved understanding of parameters for quantum linear systems.*
- *Dense matrices: We can operate with dense matrices that are sub-matrices of efficient unitaries using block-encodings.*
- *[KP22]: A new approach to QML with subspace states and some candidate exponential speedups.*

NETFLIX PROBLEM



- The preference matrix P.

	P_1	P_2	P_3	P_4	$\cdots$	$\cdots$	P_{n-1}	P_n
U_1	.1	.4	?	?	$\cdots$	$\cdots$	?	.9
U_2	.2	?	.6	?	$\cdots$	$\cdots$	.85	?
U_3	?	?	.8	.9	$\cdots$	$\cdots$	?	.2
$\vdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
U_m	?	.75	?	?	$\cdots$	$\cdots$	?	.2

Solutions based on reconstruction of the incomplete preference matrix under low rank assumptions.

Quantum algorithm samples from high value entries without reconstructing matrix.

QRS/DEQUANTIZATION

- *Low rank assumption:* The 'completed' preference matrix has a good low rank approximation as users fall into k types for $k \ll n$.
- *Theorem:* The quantum algorithm outputs good recommendations for most users in time $O(k\text{polylog}(mn))$. **[Kerenidis, P. 2017].**
- **[Tang 2019]:** There is a classical recommendation systems algorithm with running time poly(k, polylog(mn)).
- Exponential speedups for several other QML algorithms including PCA, SVM, k-means, semidefinite programming have since been refuted *[CGLLTW19].*

QML POST DEQUANTIZATION

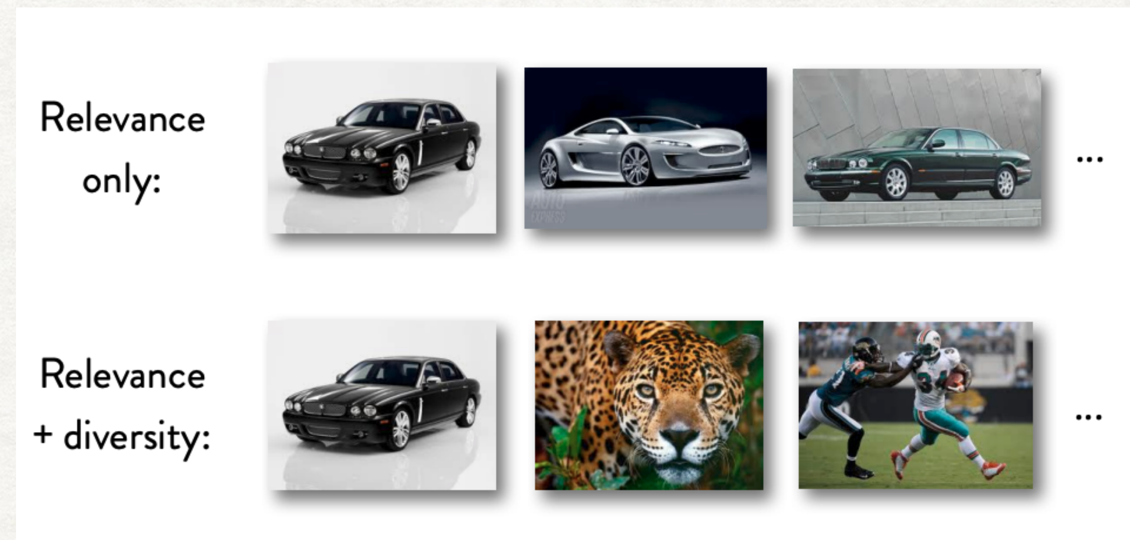
ALGORITHM	RUNNING TIME	PARAMETERS USERS, PRODUCTS:10^8. TYPES: 10^3.
QUANTUM [KP17]	$O(kpolylog(mn))$	10^3
CLASSICAL-CUR DECOMPOSITION [2002]	$O(k^2n)$	10^{12}
QUANTUM INSPIRED [T19, CGLLTW19]	$O(k^8polylog(mn))$	10^{17}

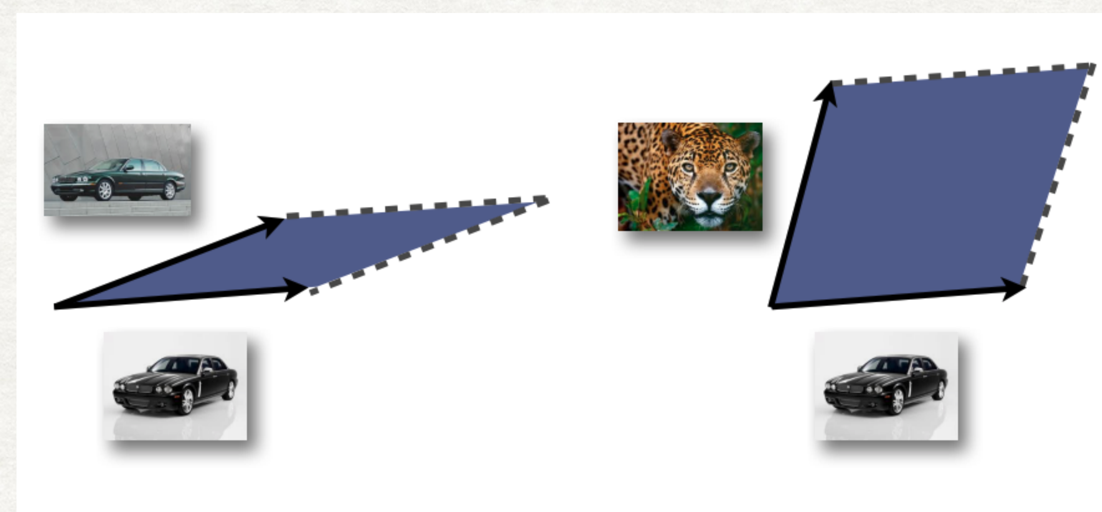
- Sparse HHL based approach and the low rank approach both have their limitations, new techniques needed.

NEW RESULTS IN QUANTUM MACHINE LEARNING

1. DETERMINANT SAMPLING

- Relevance vs diversity in machine learning. (query: Jaguar).
- Diverse results are obtained using volume sampling.
- Problem: Sample a set of rows S of matrix $A \in \mathbb{R}^{N \times D}$ with probability proportional to squared volume.
- APPLICATIONS: LOW RANK APPROXIMATION, LEAST SQUARES, REPRESENTATIVE SUMMARY, MONTE CARLO METHODS, SPARSE VECTOR IN SUBSPACE AND MORE!





1. DETERMINANT SAMPLING

- Determinant sampling: Given full rank matrix $A \in \mathbb{R}^{N \times D}$, sample from probability distribution on row subsets S of size d,

$$\Pr[S] = \frac{\det(A_S)^2}{\det(A^T A)}.$$

- Distribution is invariant under column operations on A, matrix is pre-processed to have orthonormal columns.

- Classical algorithm: Subsequent samples using $O(D^3)$ arithmetic operations *[Derezinski, Clarkson, Mahoney, Warmuth 19].*

- THEOREM: THERE IS A QUANTUM DETERMINANT SAMPLING ALGORITHM USING O(ND) GATES AND WITH CIRCUIT DEPTH O(DLOG(N)).

2. COMPOUND MATRICES

- The k-th order compound matrix $\mathscr{A}^{[k]}$ corresponding to $A \in \mathbb{R}^{n \times n}$ is the matrix indexed by size-k subsets I, J of the rows and columns of A and with entries,

$$\mathscr{A}^{[k]}_{I,J} = det(A_{I,J})\,.$$

- The compound matrix is an exponentially large matrix with dimension $\binom{n}{k}$ and the minors of the matrix as entries.

THEOREM: THERE IS A QUANTUM ALGORITHM FOR SINGULAR VALUE ESTIMATION/ TRANSFORMATION FOR COMPOUND MATRICES OF ALL ORDERS WITH COMPLEXITY O(POLY(N)).

- Kernel property: $\mathscr{A}^{[k]}\mathscr{B}^{[k]} = (\mathscr{A}\mathscr{B})^{[k]}$. Compound matrix/Cauchy Binet kernels are widely applicable in ML. *[Vishwanathan, Smola 08].*

2. COMPOUND MATRICES

- Naively, classical algorithms would have complexity $O(n^k)$ for compound matrix SVD, this is a potentially exponential speedup.
- Quantum inspired algorithms have running time polynomial in the matrix rank.
- If matrix A has a rank r approximation, the rank of the compound matrix $\mathscr{A}^{[k]}$ is $O(r^k)$, which is exponential in r.
-

OPEN QUESTION: FIND END-TO-END QML APPLICATION WITH EXPONENTIAL QUANTUM SPEEDUPS USING COMPOUND MATRIX SVD.

3. TOPOLOGICAL DATA ANALYSIS

- Topological data analysis: Captures topological features of the data-set, complementary to classical ML methods.
- Simplicial complexes generalize graphs, besides vertices and edges they have higher order faces.
- The Vietoris-Rips complex VR(X, n, d) includes all point sets with diameter at most d.
- A Dirac operator D and the Laplacian L= DD*+ D*D can be defined for every simplicial complex.
- The Betti numbers are the dimensions of the kernels of the 'graded' Laplacian L.
- Topological data analysis: The persistent Betti numbers for the VR complexes capture topological/'shape' features of the data.

3. TOPOLOGICAL DATA ANALYSIS

- The Dirac Operator is an exponentially large sparse operator, hence Hamiltonian simulation can be used for quantum TDA *[Lloyd, Garnerone, Zanardi 16].*
- Large polynomial overheads $O(n^5)$ for Dirac operator simulation.
- The depth overhead has been reduced to $O(n)$ in a series of recent works with more efficient Dirac operator constructions.
- THEOREM: THERE IS A O(LOG N) DEPTH EMBEDDING FOR THE DIRAC OPERATOR, THAT YIELDS A QUANTUM TDA ALGORITHM WITH POLY-LOGARITHMIC OVERHEAD.

SUBSPACE STATES

- The orthogonal group acts not only on vectors, but on subspaces of arbitrary dimension.
- A d-dimensional subspace $\mathcal{X} \subset \mathbb{R}^n$ is represented by a matrix $X \in \mathbb{R}^{n \times d}$ with orthonormal columns such that $\mathcal{X} = Col(X)$.
- Quantum subspace state:

$$|\mathcal{X}\rangle = |Col(X)\rangle = \sum_{|S|=d} det(X_S)|S\rangle .$$

- A one-to-one encoding for subspaces, depends on Col(X) and not the representing matrix X, that is |Col(X)> = |Col(XV)> for orthogonal V.
- Subspace states are a small fraction of the Hamming weight d quantum states, standard basis states are subspace states.

SUBSPACE STATES FOR QML

- Near term QML algorithms use unary encodings of vectors and inner product estimation.

- Inner product between subspace states is the product of the principal angles between the corresponding subspaces,

$$< Col(X) | Col(Y) > = det(X^T Y) = \prod_i \cos(\theta_i) .$$

- Subspace based classification and clustering methods have been useful in classical machine learning ***[Rene Vidal].***

- Particularly relevant to settings where data can be represented by a k-dimensional PCA, like images and videos.

QUANTUM OPERATIONS ON SUBSPACE STATES

- ***Measurement:*** Measuring a subspace state in the standard basis is equivalent to the determinant sampling problem.

- ***Addition/deletion of vector:*** Efficient quantum circuits C(y) for adding or deleting the vector y from the subspace |Col(X)>.

- ***Rotations: G***iven |Col(X)> it is possible to create the rotated subspace state |Col(UX)> for an orthogonal matrix U.

- ***Givens complexity:*** The gate complexity for the rotation |Col(UX)> is the number of elementary Givens rotations in a decomposition of U.

- Compound matrices and TDA algorithms follow from considering matrices embedded in the addition and rotation circuits.

CLIFFORD ALGEBRAS

- *Clifford algebras:* Operator algebras generated by mutually anti-commuting operators.
- *Theorem:* If A(1), A(2), ..., A(2N+1) are mutually anti-commuting operators acting on a Hilbert space H, then dim(H) > 2^N.
- *Quantum Proof:* A pair of anti-commuting operators induces a tensor product factorization $H = H_1 \otimes H'$ where H_1 is a one qubit system. *[Reichardt, Unger, Vazirani 13].*
- There is a canonical set of (N+1) mutually anti-commuting operators on the n-qubit real Hilbert space:

$$A(i) = Z^{\otimes(i-1)} \otimes X \otimes I^{\otimes(n-i)}.$$

THE GAMMA MAP

- Introduced by Dirac, the Gamma map lifts vectors, matrices and higher order tensors to the Clifford algebra.

- Definition: The Gamma map for a vector x is defined as:

$$\Gamma(x) = \sum_{i \in [n]} x_i Z^{\otimes i-1} \otimes X \otimes I^{\otimes n-i}.$$

- It follows from anti-commutativity that if ||x||=1, then the Gamma map squares to I and is thus a unitary operator.

- ***Definition:*** A Clifford loader circuit C(x) is an implementation of the unitary operator $\Gamma(x)$.

CLIFFORD LOADERS

- ***Theorem:*** The action of the Clifford loader C(x) on |Col(Y)>:

1. *Addition:* If x is orthogonal to Col(Y), then
   C(x) |Col(Y)> = |Col(Y, x)>.

2. *Deletion:* If x belongs to Col(Y) and Col(Y)=Col(Y', x), then
   C(x)|Col(Y)>=|Col(Y')>.

- In general, x will have components perpendicular and parallel to Y, and the result will be a superposition of case A and B.

- Given matrix A with orthonormal columns the subspace state Col(A) can be prepared using Clifford loaders,

$$|Col(A)\rangle = C(a_d)C(a_{d-1})\cdots C(a_1)\,|0^n\rangle\,.$$

UNARY DATA LOADERS

- Logarithmic depth state preparation circuits.
- Circuit structured as binary tree.
- Uses two qubit RBS gates: rotations on subspace spanned by |01> and |10>, identity otherwise.
- N qubit and O(log N) depth: Optimal trade-off for circuit depth vs number of qubits.
- Figure: 8 dimensional data loader with depth 4.

```
T  : |0|1|2|3|

q0 : -X-B-B-B-
        | | |
q1 : ---|-|-S-
        | |
q2 : ---|-S-B-
        |   |
q3 : ---|---S-
        |
q4 : ---S-B-B-
          | |
q5 : -----|-S-
          |
q6 : -----S-B-
            |
q7 : -------S-

T  : |0|1|2|3|
```

FERMIONIC BEAM SPLITTER (FBS) GATES

- The two qubit gates used in the data loader are the RBS (Reconfigurable beam splitter) gates:

$$RBS(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

- The Clifford loader uses the Fermionic analog of the RBS gates that we call the FBS (Fermionic beam splitter) gate:

$$FBS(i,j,\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & (-1)^{\oplus_{i<k<j} x_k} \sin(\theta) & 0 \\ 0 & (-1)^{1+\oplus_{i<k<j} x_k} \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

CLIFFORD LOADER CONSTRUCTION

- A 'fermionic' data loader D'(x) is obtained by replacing all the RBS gate in a data loader D(x) by the corresponding FBS gates.

- A Clifford loader C(x) is defined to be a circuit such that,

$$C(x) = D'(x)(X \otimes I^{n-1})D'(x)^{\dagger} .$$

- Theorem: The C(x) defined above implements the Gamma map unitary:

$$\Gamma(x) = \sum_{i \in [n]} x_i Z^{\otimes i-1} \otimes X \otimes I^{\otimes n-i} .$$

- The construction works for all circuits D(x) composed of RBS gates such that $D(x) | 10^{N-1} > = | x > .$

```
T  : |0|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|16|

q0 : -B-Z-B-Z---Z-B-Z-X-Z-B--Z-----Z--B--Z--B--
      | | | |   | | |   | |  |     |  |  |  |
q1 : -S-@-|-@-X-@-|-@---@-|--@--X--@--|--@--S--
          |   |   |       |     |     |
q2 : -B---S-X-@---|-------|-----@--X--S-----B--
      |     |     |       |        |        |
q3 : -S-----@-----|-------|--------@--------S--
                  |       |
q4 : -B-Z-B-Z-----S-------S--------Z--B--Z--B--
      | | | |                      |  |  |  |
q5 : -S-@-|-@----------------------@--|--@--S--
          |                           |
q6 : -B---S---------------------------S-----B--
      |                                     |
q7 : -S-------------------------------------S--

T  : |0|1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|16|
```

LOGARITHMIC DEPTH CLIFFORD LOADER ON 8 QUBITS.

ROTATIONS ON SUBSPACE STATES

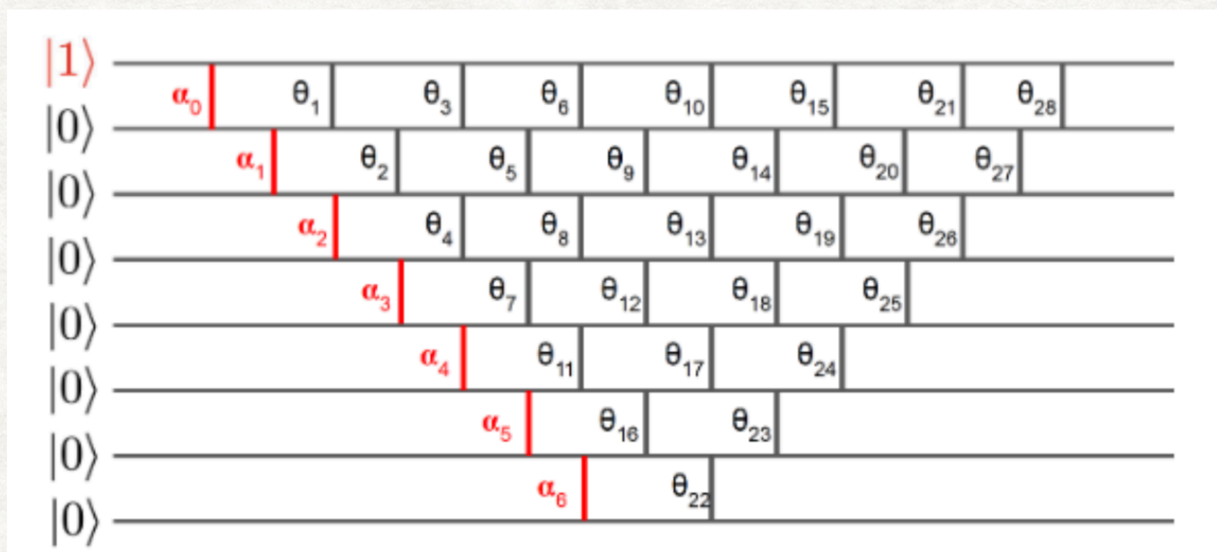
- The Givens rotations $G(i, j, \theta)$ is rotation by angle theta on coordinates (i,j) and identity on other coordinates.
- Givens rotations generate the special orthogonal group, there are different methods for decomposing an orthogonal matrix into Givens rotations. [Cosine Sine, pyramid decompositions].
- *Givens complexity: Minimum number of Givens rotations in a decomposition of U.*
- *Givens complexity is O(N log N) for Fourier and related transforms.*

QUANTUM GIVENS ROTATIONS

- Lemma: A quantum Givens rotation is implemented by a single quantum gate, $|Col(G(i,j,\theta)X) > = FBS(i,j,\theta)|Col(X) >$.

- Starting with $|Col(I_K)\rangle = |1^k 0^{n-k}\rangle$, a sequence of Givens rotations can be applied to obtain the desired subspace state $|Col(X)\rangle$.

$$I_K = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ . & \cdots & \cdots & . \\ 0 & 0 & \cdots & 1 \\ 0 & 0 & \cdots & 0 \\ . & \cdots & \cdots & . \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$



$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \cdots & \cdots & \cdots & \cdots \\ x_{d1} & x_{d2} & \cdots & x_{dd} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix}$$

- A classical Givens rotation on rows (i, j) costs O(k) arithmetic operations and maps $(x_i, x_j) \rightarrow (cx_i + sx_j, cx_j - sx_i)$.

THE ACTION OF A GIVENS CIRCUIT

- Decompose U into a sequence of elementary Givens rotations.

- *The Givens circuit G(U)* is a quantum circuit obtained by replacing every Givens rotation by the corresponding RBS or FBS gate.

- *Claim:* On the standard basis G(U)|S> = |Col(U_S)>, it selects the columns and prepares corresponding subset state.

- *Proof: Follows from the claim G(U)|Col(X)>= |Col(UX)> applied to the standard basis state |S>.*

- Computationally hard case: G(U) applied to an entangled initial state, for example $(|00> + |11>)^{\otimes n/2}$. ***[Ivanov]***

COMPOUND MATRIX SVD

- The k-th order compound matrix $\mathscr{A}^{[k]}$ corresponding to $A \in \mathbb{R}^{n \times n}$ is the matrix indexed by size-k subsets I, J of the rows and columns of A and with entries $\mathscr{A}^{[k]}_{I,J} = \det(A_{IJ})$.
- ***Observation 1:*** The Givens circuit G(U) acts as the compound matrix $\mathscr{U}^{[k]}$ on bit strings of Hamming weight k.
- ***Observation 2:*** If U is a block encoding for A, then compound matrices $\mathscr{U}^{[k]}$ are block encodings for $\mathscr{A}^{[k]}$.
- Thus, we can perform SVE and SVT and quantum linear algebra for compound matrices in a black box manner using G(U).

DIRAC OPERATOR EMBEDDING

- ***Observation 3:*** The Dirac operator for the complete simplicial complex is implemented by the C(z) circuit for $z = \frac{1}{\sqrt{N}}(1,1,\cdots,1)$.
- ***Observation 4:*** The Dirac operator for an arbitrary simplicial complex is a sub matrix of C(z), giving an efficient block encoding for D.
- Logarithmic depth Clifford loader constructions reduce the depth for quantum TDA from O(n) to O(log n).
- Potential exponential speedups were found by looking at matrices embedded in exponentially large unitaries associated with subspace states: *Givens circuits and Clifford Loaders.*

NEW QUANTUM SPEEDUPS

PROBLEM	QUANTUM	CLASSICAL	SPEEDUP
DETERMINANT SAMPLING	CLIFFORD LOADERS	IMPORTANCE SAMPLING	$O(n^2)$ $O(n^3)$
COMPOUND MATRIX SVD	GIVENS CIRCUITS	SINGULAR VALUE DECOMPOSITION	$O(poly(n))$ $O(poly(n^k))$
TOPOLOGICAL DATA ANALYSIS	DIRAC OPERATOR EMBEDDING	----	$O(\log(N))$ CIRCUIT DEPTH $O(N)$

RESEARCH QUESTIONS

- Find end-to-end QML application using compound matrix SVD with potentially exponential speedup.
- Find a QML application with exponential speedup for quantum TDA taking into account the implicit assumptions.
- Bosonic compound matrices are indexed by multi-sets with scaled permanents as entries, develop NISQ methods for quantum SVD for bosonic compound matrices and find applications to ML.

SOME TAKEAWAYS

- A promising avenue for exponential quantum linear algebra speedups is to find matrices that embed in unitary operators associated with non interacting fermions.
- Quantum Machine learning algorithms can be formulated to work not only with vectors (1-dimensional subspaces) but for sub-spaces of arbitrary dimension.
- The way to obtain quantum speedups in an applied domain is not to fit an applied problem to a known quantum technique, but rather to look at problems closely related to quantum mechanics.