Quantum-Inspired Algorithms from Randomized Numerical Linear Algebra

Nadiia Chepurko
MIT

Kenneth L. Clarkson
IBM Research

Lior Horesh
IBM Research

Honghao Lin
CMU

David P. Woodruff
CMU

Outline

➢ Background

➢ Our Results

➢ High Level Intuitions

➢ Experiments

Quantum Machine Learning

➢ In recent years, quantum algorithms for various ML problems have been proposed.
  - [HHL09] sparse matrix inversion
  - [LGZ16] topological data analysis
  - [WBL12] data fitting
  - [LMB14] principal compenent analysis
  - [RML14] support vector machine
  - [KP17] recommendation system
  - [BKL+19] semidefinite programming
  - ...

Quantum Machine Learning

- Some of these algorithms have the striking property that their running times do not depend on the input size
- That is, for a given matrix A, the running times for these proposed quantum algorithms are at most polylogarithmic in n and d, and polynomial in other parameters of A, such as $rank(A), \kappa(A), ||A||_F$.
- Question: Is actual speed up

  exponential/high polynomial/quadratic?

Quantum-Inspired Model

- Obeservation in [Tan19]: these quantum algorithms depend on a particular input representation of A, which is a strong assumption.
- Classical algorithms without corresponding assumptions are underpowered.
- A number of works study the de-quantizing QML problems in the quamtum-inspired model, as a classical analogue to state preparation.

Quantum-Inspired Model

- Obeservation in [Tan19]: these quantum algorithms depend on a particular input representation of A, which is a strong assumption.
- Classical algorithms without corresponding assumptions are underpowered.
- A number of works study the de-quantizing QML problems in the quamtum-inspired model, as a classical analogue to state preparation.

Quantum-Inspired Model

- Matrix-based data-structure, SQ(A)
  - SAMPLE1(): samples a row based on the ℓ_2 norm of the rows.
  - SAMPLE 2(i): takes a row index and sample the column entry based on their squared values
  - Query(i, j): outputs A_{ij}
  - Norm(i): outputs the norm of the i-th row
  - Norm(): outputs the norm of the Frobenius norm of A
- Note: such dynamic data structure can be implemented in $O(\log(nd))$ time in classical setting, which only increases runtime by a logarithmic factor. See [Tan19, GST20]

Quantum-Inspired Model

- Matrix-based data-structure, SQ(A).
- [GST20]: $\tilde{O}\left(\frac{\|A\|_F^8 \kappa(A)^2}{(\sigma_{\min}^6 \varepsilon^4)}\right)$ query time algorithms for linear regression
- [Tan19]: $\Omega(\text{poly}(\kappa k \varepsilon^{-1} \eta))$ query time algorithms for low-rank approximation.
- Both algorithms runs in time independent of dimensions.
- Actually shows that previous quantum algorithms do not give an exponential speedup

Quantum-Inspired Model

- Matrix-based data-structure, SQ(A).
- [GST20]: $\tilde{O}\left(\frac{\|A\|_F^8 \kappa(A)^2}{(\sigma_{\min}^6 \varepsilon^4)}\right)$ query time algorithms for linear regression
- [Tan19]: $\Omega(\mathrm{poly}(\kappa k \varepsilon^{-1} \eta))$ query time algorithms.
- Question:

  Can the sublinear terms in the running time be reduced significantly?

Our Results

Problem	Time		Prior Work	
	Update	Query	Update	Query
Ridge Regression	$O(\log(n))$	$\tilde{O}\left(\frac{d'\kappa^3\|A\|_F^2\log(d)}{\varepsilon^4\|A\|_2^2}\right)$ Thm. 4	$O(\log(n))$	$\tilde{O}\left(\frac{k^6\|A\|_F^6\kappa^{16}}{\varepsilon^6}\right)$ [GLT18] $\tilde{O}\left(\frac{\|A\|_F^8\kappa(A)^2}{(\sigma_{\min}^6\varepsilon^4)}\right)$ [GST20]
Low Rank Sampling	$O(\log(n))$	$\tilde{O}\left(\frac{\|A\|_F^2\left(\frac{\|A\|_F^2}{\|A\|_2^2}+k^2+k\psi_k\right)}{\varepsilon^4\|A\|_2^2}+\frac{k^3}{\varepsilon^6}\right)$ Thm. 5	$O(\log(n))$	$\Omega(\text{poly}(\kappa k\varepsilon^{-1}\eta))$ [Tan19]

High Level Intuition

- Observation: the reason the quantum-inspired literature obtains all of these matrix-dependent parameters is because such parameters relate squared-length sampling to leverage score sampling. But the previous works do not utilize this well.
- The algorithms build data structures for sampling according to the squared row and column lengths of a matrix.
- Is well-known that leverage score sampling often gives stronger guarantees. Writing $A = U\Sigma V^T$ in its singular value decomposition (SVD). The i-th leverage score of A is the $|U_i|_2^2$.

High Level Intuition

- Writing $A = U\Sigma V^T$ in its singular value decomposition (SVD). The i-th leverage score of A is the $|U_i|_2^2$.
- $|A_i|_2^2 \geq \sigma_{min}^2 |U_i|_2^2$ and $|A_i|_2^2 \leq \sigma_{max}^2 |U_i|_2^2$. The two ways are with ratio distance at most $\kappa(A)^2 = \frac{\sigma_{max}^2}{\sigma_{min}^2}$.

High Level Intuition

- Writing $A = U\Sigma V^T$ in its singular value decomposition (SVD). The i-th leverage score of A is the $|U_i|_2^2$.
- $|A_i|_2^2 \geq \sigma_{min}^2 |U_i|_2^2$ and $|A_i|_2^2 \leq \sigma_{max}^2 |U_i|_2^2$. The two ways are with ratio distance at most $\kappa(A)^2 = \frac{\sigma_{max}^2}{\sigma_{min}^2}$.
- If we oversample by a factor of $\kappa(A)^2$, we can get the same guarantee.

High Level Intuition

- ➢ [GST20]: $\tilde{O}\left(\frac{\|A\|_F^8 \kappa(A)^2}{(\sigma_{\min}^6 \varepsilon^4)}\right)$
- ➢ [GLT18]: $\tilde{O}\left(\frac{k^6 \|A\|_F^6 \kappa^{16}}{\varepsilon^6}\right)$
- ➢ The previous analysis of algorithms are actually implicitly doing leverage score sampling, or in the case of ridge regression, ridge leverage score sampling.
- ➢ We show how to obtain simpler algorithms and analysis by using existing techniques in the literature of randomized numerical linear algebra.

Experiments

- ➢ Low-Rank Approximation on two real-world dataset: KOS and MovieLens 100K
- ➢ Compare to [ADBL20], a implementation of the previous quantum-inspired algorithms.

Experiments

	$k = 10$	$k = 15$	$k = 20$
ε(Ours)	0.0416	0.0557	0.0653
ε(ADBL)	0.0262	0.0424	0.0538
Runtime (Ours, Query)	0.125s	0.131s	0.135s
Runtime (Ours, Total)	0.181s	0.183s	0.184s
Runtime (ADBL, Query)	0.867s	0.913s	1.024s
Runtime (ADBL, Total)	0.968s	1.003s	1.099s
Runtime of SVD	2.500s		

	$k = 10$	$k = 15$	$k = 20$
ε(Ours)	0.0397	0.0478	0.0581
ε(ADBL)	0.0186	0.0295	0.0350
Runtime (Ours, Query)	0.292s	0.296s	0.295s
Runtime (Ours, Total)	0.452s	0.455s	0.452s
Runtime (ADBL, Query)	1.501s	1.643s	1.580s
Runtime (ADBL, Total)	1.814s	1.958s	1.897s
Runtime of SVD	36.738s		

➢ See our papers for the results for the ridge regression.