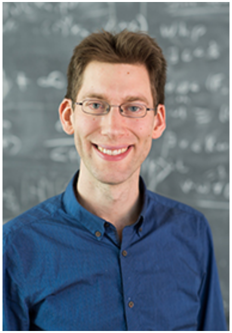
Small Quantum Computers and Large Classical Data Sets [3]

Miguel Ángel Ortiz Marín

December 1st, 2020

Section 1

Introduction



Aram W. Harrow (MIT, Boston)

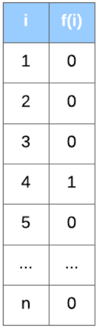
According to [1], the algorithm at the center of the "quantum machine learning" mini-revolution is called HHL [4].

But, existing quantum algorithms for optimization and machine learning are often less complete than their classical counterparts because they do not use realistic models of their input data.

Quantum Algorithm Models

Model	Definition	Example
Standard	Input $x \in 0, 1^n$ Output: $y \in 0, 1$	factoring / number theory combinatorial optimization
Oracle	given access to O $O\|i, a\rangle = \|i, a + x_i\rangle$	Grover, NAND tree, collision, ... Hidden subgroup problem, welded trees
Quantum Data	given state $\|\psi\rangle$	QFT SWAP test, Schur transform tomography linear systems
Quantum Oracle	Given U	phase estimation quantum sensing, process tomography qubitization, singular value transform

Quantum Search

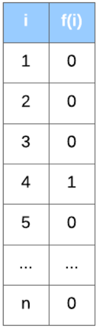


i	f(i)
1	0
2	0
3	0
4	1
5	0
...	...
n	0

Given the ability to compute
$f(i) := x_i$
Find i^* such that $f(i*) = 1$

Classical: $\mathcal{O}(n)$ time needed

Quantum Search

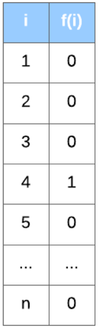


i	f(i)
1	0
2	0
3	0
4	1
5	0
...	...
n	0

Given the ability to compute
$f(i) := x_i$
Find i^* such that $f(i*) = 1$

Classical: $\mathcal{O}(n)$ time needed
Grover's algorithm [2]: (1996)
$\mathcal{O}(\sqrt{n})$ on quantum computer

Quantum Search



i	f(i)
1	0
2	0
3	0
4	1
5	0
...	...
n	0

Given the ability to compute
$f(i) := x_i$
Find i^* such that $f(i*) = 1$

Classical: $\mathcal{O}(n)$ time needed
Grover's algorithm [2]: (1996)
$\mathcal{O}(\sqrt{n})$ on quantum computer

similar speedups for maximizing, approximate counting, collisions, triangle finding, game-tree evaluation, sampling, backtracking algorithms, rapidly mixing Markov chains

Oracle search?

usual
model

$$\boxed{\begin{array}{c}\text{"oracle"} = \text{subroutine} \\ O|i\rangle|0\rangle = |i\rangle|x_i\rangle\end{array}}$$

Oracle search?

usual model $\boxed{\begin{array}{c}\text{"oracle"} = \text{subroutine} \\ O|i\rangle|0\rangle = |i\rangle|x_i\rangle\end{array}}$ $\Rightarrow$ provable speedups, sometimes exponential

Oracle search?

usual model

$$\boxed{\begin{array}{c}\text{"oracle"} = \text{subroutine} \\ O|i\rangle|0\rangle = |i\rangle|x_i\rangle\end{array}}$$

$\Rightarrow$

provable speedups, sometimes exponential

However

- We cannot query a data center in superposition
- Classical memory is $\approx$ parallel.
  For large enough n, n bits of memory $\leftrightarrow \mathcal{O}(n)$ CPUs.
- Proposed “quantum RAM” could be queried in superposition, but not easy to build.

Oracles from classical memory

$x_1, \ldots, x_n$

Oracles from classical memory

$x_1, \ldots, x_n$

$\Rightarrow$

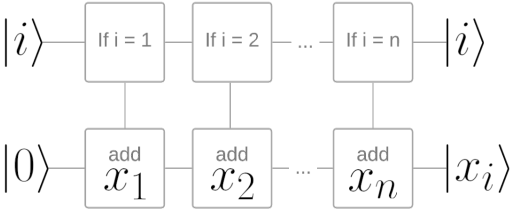
$O|i\rangle|0\rangle = |i\rangle|x_i\rangle$

Oracles from classical memory

$x_1, \ldots, x_n$

$\Rightarrow$

$O|i\rangle|0\rangle = |i\rangle|x_i\rangle$



Big data quantum speedups

Queries with overhead $\mathcal{O}(n)$ mean no oracle speedup.

Big data quantum speedups

Queries with overhead $\mathcal{O}(n)$ mean no oracle speedup.

Usual solutions

1. QRAM: quantumly-accesible database
2. $x_i = f(i)$ for efficient f, e.g. $f(i) = a^i modN$.

Big data quantum speedups

Queries with overhead $\mathcal{O}(n)$ mean no oracle speedup.

Usual solutions

1. QRAM: quantumly-accesible database
2. $x_i = f(i)$ for efficient f, e.g. $f(i) = a^i modN$.

New approach

Find problems where we can reduce the size of the data set

- Clustering
- Bayesian Inference
- Saddle-Point optimization

Section 2

Statistics using data

Maximum-likelihood estimation:

Given a set of models Y and a data set X
Compute $\max_y [r(y) + \sum_{x \in X} f(x, y)]$
Where $f(x, y) = \log p(x|y)$.

Maximum-likelihood estimation:

Given a set of models Y and a data set X
Compute $\max\limits_{y}[r(y) + \sum_{x \in X} f(x, y)]$
Where $f(x, y) = \log p(x|y)$.

$\mathbf{X}, \mathbf{Y}$ appear almost symmetrically. ($\max$ *vs* $\sum$ don't matter for Grover) but
X lives on a **hard drive** $\rightarrow$ **no oracle access or superposition**
Y is **synthetic** (e.g weights and centers of clusters) $\rightarrow$ **can superpose**

Maximum-likelihood estimation:

Given a set of models Y and a data set X
Compute $\max_y [r(y) + \sum_{x\in X} f(x, y)]$
Where $f(x, y) = \log p(x|y)$.

$\mathbf{X}, \mathbf{Y}$ appear almost symmetrically. (max *vs* $\sum$ don't matter for Grover) but
X lives on a **hard drive** $\rightarrow$ **no oracle access or superposition**
Y is **synthetic** (e.g weights and centers of clusters) $\rightarrow$ **can superpose**

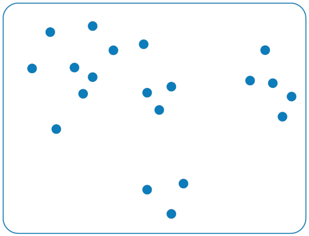
Classical computer controlling a quantum computer: Can run Grover with $\mathcal{O}(|X| \cdot |Y|^{1/2})$

Key question: Can we reduce the dependence on $|X|$?

Section 3

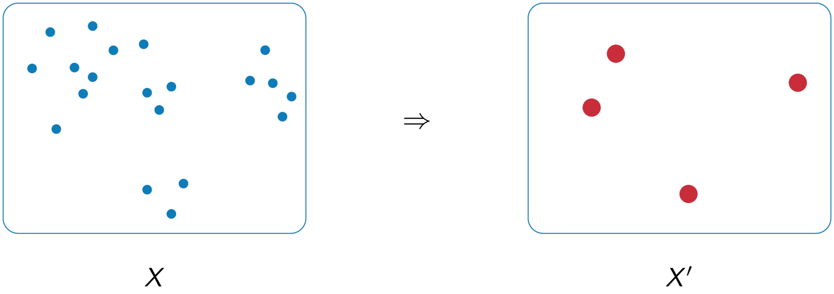
Data Reduction

Coresets

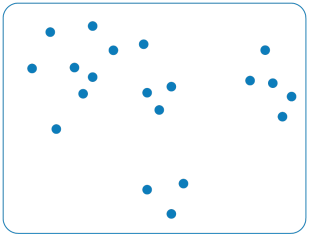


X

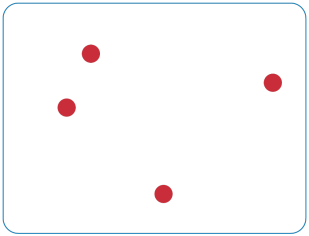
Coresets



Coresets



$\Rightarrow$



X X'

X' is a **coreset** if for all y,
$\sum_{x \in X} f(x, y) \approx \sum_{x \in X'} w(x) f(x, y)$

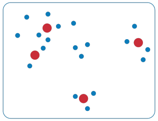
Hybrid algorithms for machine learning

$$X \rightarrow \boxed{\text{classical computer}} \xrightarrow{X'} \boxed{\text{quantum computer}} \rightarrow y$$

- Finds quick, crude guess of y'.
- Use y' to construct importance sampling weights
- Draw X' using importance sampling.

- Use Grover / q walks / adiabatic / etc to find y.
- Inner loop cost is $\mathcal{O}(|X'|)$ Total cost for Grover $\mathcal{O}(|X'| \cdot |Y|^{\frac{1}{2}})$
- Provable speedups include clustering.

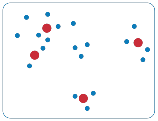
Importance sampling



X'

Goal: for all y, $\sum_{x \in X} f(x,y) \approx \sum_{x \in X'} w(x) f(x,y)$

Importance sampling



X'

Goal: for all y, $\sum_{x \in X} f(x,y) \approx \sum_{x \in X'} w(x) f(x,y)$

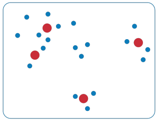
Challenges

1. X' estimator should be unbiased
2. For any fixed y, need to control the variance

$$\sigma^2 = E_x[f(x,y)^2] - E_x[f(x,y)]^2$$

3. X' needs to work simultaneously for all y.

Importance sampling



X'

Goal: for all y, $\sum_{x \in X} f(x,y) \approx \sum_{x \in X'} w(x) f(x,y)$
Challenges

1. X' estimator should be unbiased
2. For any fixed y, need to control the variance

$$\sigma^2 = E_x[f(x,y)^2] - E_x[f(x,y)]^2$$

3. X' needs to work simultaneously for all y.

Solutions

1. Sample x with probability $\propto s(x)$. Assign weight $\propto \frac{1}{s(x)}$
2. Importance Sampling: Choose $s(x) \geq max_y[\frac{f(x,y)}{\sum_{x'} f(x',y)}]$
3. Bound pseudo-dimension D of family $f(.,y)$.

Non-adaptive coresets

Thm: Importance sampling $x_1, ..., x_m$ yields a $\epsilon - \mathit{coreset}$ with prob. $\geq 1 - \delta$ if

$$m \gtrsim \frac{S^2}{\epsilon^2}\left(D + \log\frac{1}{\delta}\right)$$

Given:
$s(x) \geq max_y[\frac{f(x,y)}{\sum_{x'} f(x',y)}]$.
total sensitivity $S = \sum_x s(x)$.
Pseudo dimesion D.

Non-adaptive coresets

Thm: Importance sampling $x_1, ..., x_m$ yields a $\epsilon - \textit{coreset}$ with prob. $\geq 1 - \delta$ if

$$m \gtrsim \frac{S^2}{\epsilon^2}\left(D + \log\frac{1}{\delta}\right)$$

Given:
$s(x) \geq max_y[\frac{f(x,y)}{\sum_{x'} f(x',y)}]$.
total sensitivity $S = \sum_x s(x)$.
Pseudo dimesion D.

Example: k-means in R^d
$y = (y^{(1)}, ..., y^{(k)})$ = positions of k centers.
$f(x, y) = -\min_j dist(x, y^{(j)})^2$
pseudo-dim: $\mathrm{D} = \mathcal{O}(dk \log(k))$
sensitivity: $S \leq \mathcal{O}(k)$

Non-adaptive coresets

Thm: Importance sampling $x_1, ..., x_m$ yields a $\epsilon - \textit{coreset}$ with prob. $\geq 1 - \delta$ if

$$m \gtrsim \frac{S^2}{\epsilon^2}\left(D + \log\frac{1}{\delta}\right)$$

Given:
$s(x) \geq max_y[\frac{f(x,y)}{\sum_{x'} f(x',y)}]$.
total sensitivity $S = \sum_x s(x)$.
Pseudo dimesion D.

Example: k-means in R^d
$y = (y^{(1)}, ..., y^{(k)})$ = positions of k centers.
$f(x,y) = -\min_j dist(x, y^{(j)})^2$
pseudo-dim: D $= \mathcal{O}(dk\log(k))$
sensitivity: $S \leq \mathcal{O}(k)$

$$\Rightarrow m \gtrsim \frac{dk^3 \log k + k^2 \log\frac{1}{\delta}}{\epsilon^2}$$

Non-adaptive coresets

Thm: Importance sampling $x_1, ..., x_m$ yields a $\epsilon - \textit{coreset}$ with prob. $\geq 1 - \delta$ if

$$m \gtrsim \frac{S^2}{\epsilon^2}\left(D + \log\frac{1}{\delta}\right)$$

Given:
$s(x) \geq max_y[\frac{f(x,y)}{\sum_{x'} f(x',y)}]$.
total sensitivity $S = \sum_x s(x)$.
Pseudo dimesion D.

Example: k-means in R^d
$y = (y^{(1)}, ..., y^{(k)}) =$ positions of k centers.
$f(x, y) = -\min_j dist(x, y^{(j)})^2$
pseudo-dim: D $= \mathcal{O}(dk\log(k))$
sensitivity: $S \leq \mathcal{O}(k)$

$$\Rightarrow m \gtrsim \frac{dk^3\log k + k^2\log\frac{1}{\delta}}{\epsilon^2}$$

Algorithm

1. Classical computer finds $s(x)$ for each $x \in X$ and samples $x_1, ..., x_m$
2. Brute force search in time k^m classical or $k^{\frac{m}{2}}$ quantumly

(Or use other algorithms, like adiabatic, quantum walks, etc..)

Adaptive data-reduction

$$X \rightarrow \boxed{\text{classical computer}} \xrightarrow{X'} \boxed{\text{quantum computer}} \rightarrow y$$

Data set is small by the time we turn the quantum computer.

Adaptive data-reduction

$$X \rightarrow \boxed{\text{classical computer}} \overset{X'}{\rightarrow} \boxed{\text{quantum computer}} \rightarrow y$$

Data set is small by the time we turn the quantum computer.

$$X \quad \rightarrow \quad \boxed{\begin{array}{c}\text{classical}\\ \text{computer}\end{array}} \begin{array}{c}\rightleftarrows \\ \cdots \\ \rightarrow \\ \leftarrow\end{array} \boxed{\begin{array}{c}\text{quantum}\\ \text{computer}\end{array}} \quad \rightarrow \quad y$$

Are there benefits to using the quantum computer interactively?

Adapative coresets

[Campbell, Broderick 2017] [Bugalo, Elvira, Martino, Luengo, Míguez, Djuric, 2017]

Bayesian Inference: sample y from:

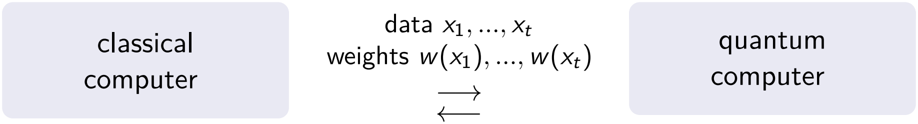
$$\pi(y) = \frac{\pi_0(y) \exp \sum_{x \in X} f(x,y)}{Z}$$

Adapative coresets

[Campbell, Broderick 2017] [Bugalo, Elvira, Martino, Luengo, Míguez, Djuric, 2017]

Bayesian Inference: sample y from:

$$\pi(y) = \frac{\pi_0(y) \exp \sum_{x \in X} f(x,y)}{Z}$$



Adapative coresets

[Campbell, Broderick 2017] [Bugalo, Elvira, Martino, Luengo, Míguez, Djuric, 2017]

Bayesian Inference: sample y from:

$$\pi(y) = \frac{\pi_0(y) \exp \sum_{x \in X} f(x,y)}{Z}$$

$$\begin{tikzcd}
\boxed{\text{classical computer}} & \substack{\text{data } x_1, \ldots, x_t \\ \text{weights } w(x_1), \ldots, w(x_t)} & \boxed{\text{quantum computer}}
\end{tikzcd}$$

$$\overset{\longrightarrow}{\longleftarrow}$$

Sample y_t from $\pi_t(y_t) = \frac{\pi_0(y) \exp \sum_{i=1}^{t} w(x_i) f(x_i,y)}{Z}$

Adapative coresets

[Campbell, Broderick 2017] [Bugalo, Elvira, Martino, Luengo, Míguez, Djuric, 2017]

Bayesian Inference: sample y from:

$$\pi(y) = \frac{\pi_0(y) \exp \sum_{x \in X} f(x,y)}{Z}$$

$$\boxed{\text{classical computer}} \quad \begin{array}{c} \text{data } x_1, \ldots, x_t \\ \text{weights } w(x_1), \ldots, w(x_t) \\ \longrightarrow \\ \longleftarrow \end{array} \quad \boxed{\text{quantum computer}}$$

Sample y_t from $\pi_t(y_t) = \frac{\pi_0(y) \exp \sum_{i=1}^{t} w(x_i) f(x_i,y)}{Z}$

Choose $x_{t+1}, w(x_{t+1})$ using $y_1, \ldots, y_t, x_1, \ldots, x_t, w(x_1), \ldots, w(x_t)$

Comparison with variational algorithms

1. Variational algorithms use a classical outer loop to perform gradient descent on the circuit parameters of a quantum inner loop. These can be extremely generaland in some cases amount to performing a local search over the set of all short circuits implementable in a particular hardware model. As a result, they often lack the provable guarantees of algorithms in this paper. On the other hand, they can be run on even very simple quantum computers and running them can teach us about what we might expect from future quantum hardware.

2. There are two key features of our Algorithm 3 that are not suggested by the usual variational formulation: 1) the ansatz and the classical outer loopare structured carefully to present the quantum computer with a very limited subset of the overalldata set, and; 2) the output of the quantum computer is usable for a form of stochastic mirror descent without any of the dimension dependence that is seen in general (e.g [36]).

Comparison with stochastic gradient descent

1. The essential difference between SGD and coresets is that coresets are sampled once and then used throughout the optimization, while SGD draws fresh samples for each gradient step. For classical gradient descent, this difference may not be important, or it may favor SGD. However, when used as a subroutine inside a Grover or Durr-Høyer search, the stochastic noise introduced by SGD can be harmful. Indeed, the Grover speedup is known to vanish when the oracle is stochastic and has a non-negligible chance of being replaced by the identity operator [49].

Comparison with stochastic gradient descent

2. This is an example of a more general problem. A subset of size k, whether used as a coreset ora SGD minibatch, may be “bad” with probability δ. For coreset algorithms this means the overall algorithm fails with probability δ. For a Grover search using SGD, this means the overall algorithm fails with probability $\sqrt{|Y|}\delta$. (Other algorithms for searching over Y, such as the adiabatic algorithm, may have a more complicated dependence on stochastic noise, and studying this difference is an important open question.)

3. Recently there has been enormous progress in our theoretical and practical understanding of more sophisticated variants of gradient descent [52], and it is an important open question to un-derstand what potential these have for benefiting quantum algorithms.

Perspective

1. Grover is used for concreteness but same ideas apply to adiabatic algorithm, QAOA, quantum walks, etc.
2. QCs will augment not replace classical computers. Let's design algorithms as though we believe this!

Section 4

Recent Work

Coreset Clustering on Small Quantum Computers [6]

Many quantum algorithms for machine learning require access to classical data in superposition. However, for many natural data sets and algorithms, the overhead required to load the data set in superposition can erase any potential quantum speedup over classical algorithms. Recent work by Harrow introduces a new paradigm in hybrid quantum-classical computing to address this issue, relying on coresets to minimize the data loading overhead of quantum algorithms. We investigate using this paradigm to perform k-means clustering on near-term quantum computers, by casting it as a QAOA optimization instance over a small coreset. We compare the performance of this approach to classical k-means clustering both numerically and experimentally on IBM Q hardware. We are able to find data sets where coresets work well relative to random sampling and where QAOA could potentially outperform standard k-means on a coreset. However, finding data sets where both coresets and QAOA work well–which is necessary for a quantum advantage over k-means on the entire data set–appears to be challenging.

A hybrid classical-quantum workflow for natural language processing [5]

Natural language processing (NLP) problems are ubiquitous in classical computing, where they often require significant computational resources to infer sentence meanings. With the appearance of quantum computing hardware and simulators, it is worth developing methods to examine such problems on these platforms. In this manuscript we demonstrate the use of quantum computing models to perform NLP tasks, where we represent corpus meanings, and perform comparisons between sentences of a given structure. We develop a hybrid workflow for representing small and large scale corpus data sets to be encoded, processed, and decoded using a quantum circuit model. In addition, we provide our results showing the efficacy of the method, and release our developed toolkit as an open software suite.

Thanks!

- Scott Aaronson. “Read the fine print”. In: *Nature Physics* 11.4 (Apr. 2015), pp. 291–293. DOI: 10.1038/nphys3272. URL: https://doi.org/10.1038/nphys3272.
- Lov K. Grover. “A Fast Quantum Mechanical Algorithm for Database Search”. In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC ’96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1996, pp. 212–219. ISBN: 0897917855. DOI: 10.1145/237814.237866. URL: https://doi.org/10.1145/237814.237866.
- Aram W. Harrow. *Small quantum computers and large classical data sets*. 2020. arXiv: 2004.00026 [quant-ph].

Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. "Quantum Algorithm for Linear Systems of Equations". In: *Physical Review Letters* 103.15 (Oct. 2009). ISSN: 1079-7114. DOI: 10.1103/physrevlett.103.150502. URL: http://dx.doi.org/10.1103/PhysRevLett.103.150502.

L. J. O'Riordan et al. "A hybrid classical-quantum workflow for natural language processing". In: *ArXiv* abs/2004.06800 (2020).

Teague Tomesh et al. *Coreset Clustering on Small Quantum Computers*. 2020. arXiv: 2004.14970 [quant-ph].