Quantum algorithms for classification

Alessandro Luongo

Twitter: @scinawa
`https://luongo.pro/qml`

15-11-2018, Montpellier, IQFA'9 Colloquium

1. What are we doing here?

2. Toolbox

3. Quantum algorithm for classification..
   - QSFA
   - QFDC
   - q-means

4. ..on real data


0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9
0 1 2 3 4 5 6 7 8 9

Unsupervised methods

$$X \in \mathbb{R}^{n \times d}$$

- Anomaly detection
- Clustering
- Blind signal separation
- Text mining

Supervised methods

$$X \in \mathbb{R}^{n \times d}, Y \in \mathbb{R}^{n \times m}$$

- Regression
- Pattern recognition
- Time series forecasting
- Speech recognition

Unsupervised methods

$$X \in \mathbb{R}^{n \times d}$$

- Anomaly detection
- Clustering
- Blind signal separation
- Text mining

- Classification

Supervised methods

$$X \in \mathbb{R}^{n \times d}, Y \in \mathbb{R}^{n \times m}$$

- Regression
- Pattern recognition
- Time series forecasting
- Speech recognition

- Classification

ML's algos:
$runtime = O(poly(size)) = O(poly(n, d))$...

ML's algos:
$runtime = O(poly(size)) = O(poly(n, d))$...

but $size = O(2^{time})$...

ML's algos:
$runtime = O(poly(size)) = O(poly(n,d))$...

but $size = O(2^{time})$...$\Rightarrow$ problem!.

We need Quantum Machine Learning!
$runtime = O(polylog(size))$

ML's algos:
$runtime = O(poly(size)) = O(poly(n, d))$...

but $size = O(2^{time})$...$\Rightarrow$ problem!.

We need Quantum Machine Learning!
$runtime = O(polylog(size))$

[HHL09] ...

1. Ewin Tang. **A quantum-inspired classical algorithm for recommendation systems.** arXiv:1807.04271, 2018. (undergraduate thesis, advised by Scott Aaronson)
2. Ewin Tang. **Quantum-inspired classical algorithms for principal component analysis and supervised clustering.** arXiv:1811.00414, 2018.
3. Ewin Tang et al. **Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension** arXiv: 1811.04909, 2018

QML team @ IRIF

- Iordanis Kerenidis
- Jonas Landman
- Anupam Prakash

1 - QRAM

Let $X \in \mathbb{R}^{n \times d}$. There is a quantum algorithm that

$$|i\rangle |0\rangle \rightarrow |i\rangle |x_i\rangle \quad |x_i\rangle = \|x_i\|^{-1} |x_i\rangle$$

1 - QRAM

Let $X \in \mathbb{R}^{n\times d}$. There is a quantum algorithm that

$$|i\rangle |0\rangle \rightarrow |i\rangle |x_i\rangle \quad |x_i\rangle = \|x_i\|^{-1} |x_i\rangle$$

$$\frac{1}{\sqrt{\sum_{i=0}^{n} \|x_i\|^2}} \sum_{i=0}^{n} \|x_i\| |i\rangle |x_i\rangle$$

1 - QRAM

Let $X \in \mathbb{R}^{n \times d}$. There is a quantum algorithm that

$$|i\rangle |0\rangle \to |i\rangle |x_i\rangle \quad |x_i\rangle = \|x_i\|^{-1} |x_i\rangle$$

$$\frac{1}{\sqrt{\sum_{i=0}^{n} \|x_i\|^2}} \sum_{i=0}^{n} \|x_i\| \, |i\rangle \, |x_i\rangle$$

- Execution time: $O(\log nd)$

1 - QRAM

Let $X \in \mathbb{R}^{n \times d}$. There is a quantum algorithm that

$$|i\rangle |0\rangle \to |i\rangle |x_i\rangle \qquad |x_i\rangle = \|x_i\|^{-1} |x_i\rangle$$

$$\frac{1}{\sqrt{\sum_{i=0}^{n} \|x_i\|^2}} \sum_{i=0}^{n} \|x_i\| \, |i\rangle \, |x_i\rangle$$

- Execution time: $O(\log nd)$
- Preparation time: $O(nd \log nd)$

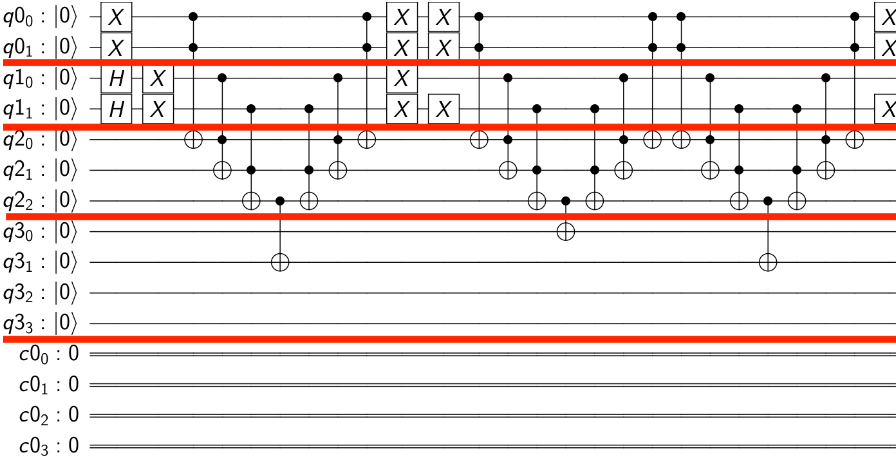
1 - QRAM

Let $X \in \mathbb{R}^{n \times d}$. There is a quantum algorithm that

$$|i\rangle |0\rangle \rightarrow |i\rangle |x_i\rangle \quad |x_i\rangle = \|x_i\|^{-1} |x_i\rangle$$

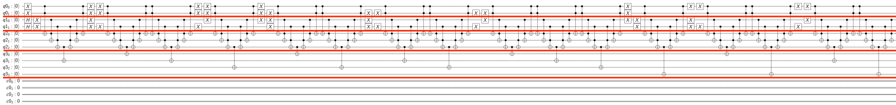
$$\frac{1}{\sqrt{\sum_{i=0}^{n} \|x_i\|^2}} \sum_{i=0}^{n} \|x_i\| |i\rangle |x_i\rangle$$

- Execution time: $O(\log nd)$
- Preparation time: $O(nd \log nd)$
- Size: $O(nd \log nd)$



q0_0 : |0⟩
q0_1 : |0⟩
q1_0 : |0⟩
q1_1 : |0⟩
q2_0 : |0⟩
q2_1 : |0⟩
q2_2 : |0⟩
q3_0 : |0⟩
q3_1 : |0⟩
q3_2 : |0⟩
q3_3 : |0⟩
c0_0 : 0
c0_1 : 0
c0_2 : 0
c0_3 : 0
X
H

QRAM [[2,3,4],[5,6,7],[8,9,10]]



2 - Q-BLAS

- $M := \sum_i \sigma_i u_i v_i^T \in \mathbb{R}^{d \times d}$, $\|M\|_2 = 1$, in QRAM
- $x \in \mathbb{R}^d$ in QRAM.

There is a quantum algorithm that w.h.p. returns :

(i) $|z\rangle$ such that $\left\| |z\rangle - |M^{-1}x\rangle \right\| \leq \epsilon$
in time $\widetilde{O}(\kappa(M)\mu(M)\log(1/\epsilon))$

2 - Q-BLAS

- $M := \sum_i \sigma_i u_i v_i^T \in \mathbb{R}^{d \times d}$, $\|M\|_2 = 1$, in QRAM
- $x \in \mathbb{R}^d$ in QRAM.

There is a quantum algorithm that w.h.p. returns :

(i) $|z\rangle$ such that $\left\| |z\rangle - |M^{-1}x\rangle \right\| \leq \epsilon$
in time $\widetilde{O}(\kappa(M)\mu(M)\log(1/\epsilon))$

(ii) $|z\rangle$ such that $\| |z\rangle - |Mx\rangle \| \leq \epsilon$
in time $\widetilde{O}(\kappa(M)\mu(M)\log(1/\epsilon))$

2 - Q-BLAS

- $M := \sum_i \sigma_i u_i v_i^T \in \mathbb{R}^{d\times d}$, $\|M\|_2 = 1$, in QRAM
- $x \in \mathbb{R}^d$ in QRAM.

There is a quantum algorithm that w.h.p. returns :

(i) $|z\rangle$ such that $\left\| |z\rangle - |M^{-1}x\rangle \right\| \leq \epsilon$
in time $\widetilde{O}(\kappa(M)\mu(M)\log(1/\epsilon))$

(ii) $|z\rangle$ such that $\| |z\rangle - |Mx\rangle \| \leq \epsilon$
in time $\widetilde{O}(\kappa(M)\mu(M)\log(1/\epsilon))$

(iii) a state $|M^+_{\leq\theta,\delta} M_{\leq\theta,\delta} x\rangle$
in time $\tilde{O}(\frac{\mu(M)\|x\|}{\delta\theta \left\| M^+_{\leq\theta,\delta} M_{\leq\theta,\delta} x \right\|})$

2 - Q-BLAS

- $M := \sum_i \sigma_i u_i v_i^T \in \mathbb{R}^{d\times d}$, $\|M\|_2 = 1$, in QRAM
- $x \in \mathbb{R}^d$ in QRAM.

There is a quantum algorithm that w.h.p. returns :

1. $|z\rangle$ such that $\left\| |z\rangle - |M^{-1}x\rangle \right\| \leq \epsilon$
   in time $\widetilde{O}(\kappa(M)\mu(M)\log(1/\epsilon))$
2. $|z\rangle$ such that $\| |z\rangle - |Mx\rangle \| \leq \epsilon$
   in time $\widetilde{O}(\kappa(M)\mu(M)\log(1/\epsilon))$
3. a state $|M^+_{\leq\theta,\delta} M_{\leq\theta,\delta} x\rangle$
   in time $\tilde{O}\left(\frac{\mu(M)\|x\|}{\delta\theta \left\| M^+_{\leq\theta,\delta} M_{\leq\theta,\delta} x \right\|}\right)$

Get estimates of $\|z\| = f(M)x$ (with mult. error ϵ_2, time $O() \cdot \epsilon_2^{-1}$)

Gilyén, András, et al. "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics." arXiv preprint arXiv:1806.01838 (2018).

2.5 - Q-BLAS

- $A := \sum_i \sigma_i u_i v_i^T, \quad B := \sum_i \lambda_i w_i l_i^T \in \mathbb{R}^{d \times d}$ in QRAM
$\|A\|_2 = \|B\|_2 = 1$, in QRAM
- $x \in \mathbb{R}^d$ in QRAM.

There is a quantum algorithm that w.h.p. returns :

- (i) $|z\rangle$ such that $\big\| |z\rangle - |(AB)^{-1}x\rangle \big\| \leq \epsilon$
- (ii) $|z\rangle$ such that $\| |z\rangle - |(AB)x\rangle \| \leq \epsilon$
- (iii) a state $|(AB)^{+}_{\leq\theta,\delta}(AB)_{\leq\theta,\delta}x\rangle$

Get estimates of $\|z\| = f(AB)x$ (with mult. error ϵ_2, time $O() \cdot \epsilon_2^{-1}$)

Gilyén, András, et al. "Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics." arXiv preprint arXiv:1806.01838 (2018).

- Before: Quantum Singular Value Estimation

$$\sum_i \alpha_i |v_i\rangle \mapsto \sum_i \alpha_i |v_i\rangle |\overline{\sigma}_i\rangle$$

- Now: Qubitization:

$$W = e^{i\phi_0\sigma_z} e^{i\theta\sigma_x} e^{i\phi_1\sigma_z} e^{i\theta\sigma_x} \cdots e^{i\phi_k\sigma_z} e^{i\theta\sigma_x}$$

3 - Compute distances

$V \in \mathbb{R}^{n\times d}, C \in \mathbb{R}^{k\times d}$ in the QRAM, $\Delta > 0$ and $\epsilon > 0$
There is a quantum algorithm that w.h.p. and in time
$\widetilde{O}\left(\frac{(T(V)+T(C))Z\log(1/\Delta)}{\epsilon}\right)$

$$|i\rangle\,|j\rangle\,|0\rangle \mapsto \;|i\rangle\,|j\rangle\,|\overline{d(v_i,c_j)}\rangle$$

where $|\overline{d(v_i,c_j)} - d(v_i,c_j)| \leqslant \epsilon$, where
$Z = \max_{i,j}(\|v_i\|^2 + \|c_j\|^2)$.

Based on: Wiebe, N., Kapoor, A., & Svore, K. (2014). Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. arXiv preprint arXiv:1401.2142.

3 - sketch proof

- Use QFD to build:

$$\frac{\|v_i\|}{\sqrt{Z_{ij}}} |i\rangle |j\rangle |0\rangle |v_i\rangle + \frac{\|c_j\|}{\sqrt{Z_{ij}}} |i\rangle |j\rangle |1\rangle |c_j\rangle$$

- Hadamard on 3rd qubit. Note that

$$p(1)_{ij} = \frac{1}{2Z_{ij}}(\|v_i\|^2 + \|c_j\|^2 - 2\|v_i\| \|c_j\| \langle v_i\rangle c_j) = \frac{d(v_i, c_j)^2}{2Z_{ij}}$$

- Perform amplitude estimation on L copies.
- Use Median Lemma (Wiebe et. al.)
- Invert circuit to remove garbage (and multiply by $2Z_{ij}$).

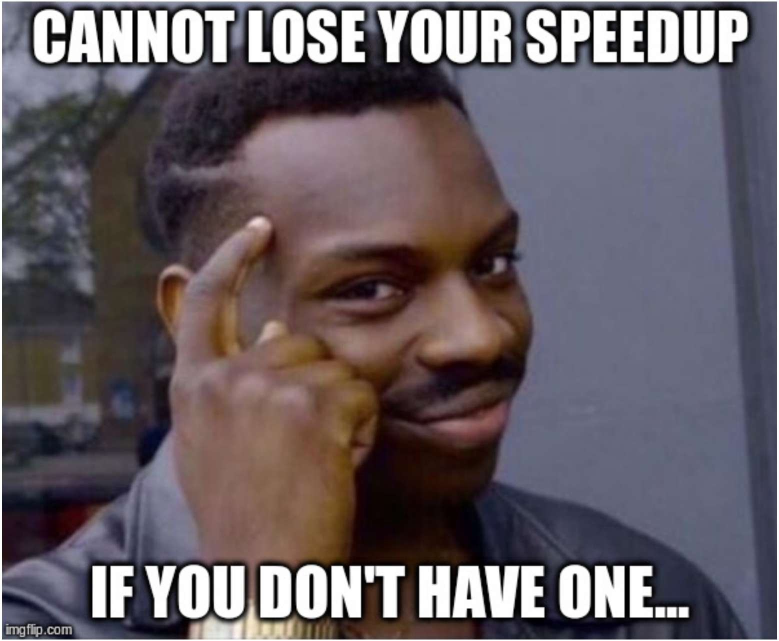
4 - Tomography

For a pure quantum state $|x\rangle$, there is a tomography algorithm with sample and time complexity $O(d \log d/\epsilon^2)$ that produces an estimate $\widetilde{x} \in \mathbb{R}^d$ with $\|\widetilde{x}\|_2 = 1$ such that $\|\widetilde{x} - x\|_2 \leq \epsilon$ with probability at least $(1 - 1/d^{0.83})$.

Kerenidis, Iordanis, and Anupam Prakash. ”A quantum interior point method for LPs and SDPs.” arXiv preprint arXiv:1808.09266 (2018).

Other..

- Amplitude estimation
- Amplitude amplification
- Hamiltonian simulation
- Phase estimation
- Quantum Random Walks
- Swaps


CANNOT LOSE YOUR SPEEDUP
IF YOU DON'T HAVE ONE...
imgflip.com

Slow Feature Analysis (Supervised)

Input signal: $x(i) \in \mathbb{R}^d$. Task: Learn K functions:

$$y(i) = [g_1(x(i)), \cdots, g_K(x(i))]$$

Such that $\forall j \in [K]$. Minimize:

$$\Delta(y_j) = \frac{1}{a} \sum_{k=1}^{K} \sum_{\substack{s,t \in T_k \\ s<t}} \big(g_j(x(s)) - g_j(x(t))\big)^2$$

Constraints on output signal: average of components is 0, variance of components is 1, signals are decorrelated.

Def Cov. matrix $B := X^T X$, Derivative cov. matrix $A := \dot{X}^T \dot{X}$

Slow Feature Analysis (Supervised)

Input signal: $x(i) \in \mathbb{R}^d$. Task: Learn K functions:

$$y(i) = [g_1(x(i)), \cdots, g_K(x(i))]$$

Such that $\forall j \in [K]$. Minimize:

$$\Delta(y_j) = \frac{1}{a} \sum_{k=1}^{K} \sum_{\substack{s,t \in T_k \\ s<t}} \left(g_j(x(s)) - g_j(x(t))\right)^2$$

Constraints on output signal: average of components is 0, variance of components is 1, signals are decorrelated.

Def Cov. matrix $B := X^T X$, Derivative cov. matrix $A := \dot{X}^T \dot{X}$

$$AW = BW\Lambda$$

Step 1: Whitening

Data is whitened (sphered) if $B = X^T X = I$.

Step 1: Whitening

Data is whitened (sphered) if $B = X^T X = I$.

...Whitening its just matrix inversion
(Moore-Penrose inverse)..
$Z := X^{+} X$ ..

Step 1: Whitening

Data is whitened (sphered) if $B = X^T X = I$.

...Whitening its just matrix inversion (Moore-Penrose inverse)..
$Z := X^+ X$ ..now $Z^T Z = I$

Step 1: Whitening

Data is whitened (sphered) if $B = X^T X = I$.

...Whitening its just matrix inversion (Moore-Penrose inverse)..
$Z := X^+ X$ ..now $Z^T Z = I$
Freebie Theorem!
There exists an efficient quantum algorithm for whitening that return $|Z\rangle := X^+ |X\rangle$

Step 2: projection

$$\begin{array}{ccc} X & \xrightarrow{Der.} & \dot{X} \\ \Big\downarrow{\scriptstyle Whit.} & & \Big\downarrow{\scriptstyle Whit.} \\ Z & \xrightarrow{Der.} & \dot{Z} \end{array}$$

- Whiten data $|X\rangle \mapsto |Z\rangle$
- Project data in slow feature space $|Z\rangle \mapsto |Y\rangle$

New algo! QSFA

- Let $X = \sum_i \sigma_i u_i v_i^T \in \mathbb{R}^{n \times d}$, $\dot{X} \in \mathbb{R}^{n \log n \times d}$ QRAM.
- Let $\epsilon, \theta, \delta, \eta > 0$.

There exists a quantum algorithm that produces:

- $|\overline{Y}\rangle$ with $|\,|\overline{Y}\rangle - |A^{+}_{\leq\theta,\delta} A_{\leq\theta,\delta} Z\rangle\,| \leq \epsilon$ in time

$$\tilde{O}\Big(\Big(\kappa(X)\mu(X)\log(1/\varepsilon) + \frac{(\kappa(X) + \kappa(\dot{X}))(\mu(X) + \mu(\dot{X}))}{\delta\theta}\Big)$$

$$\ldots \times \frac{||Z||}{||A^{+}_{\leq\theta,\delta} A_{\leq\theta,\delta} Z||}\Big)$$

- $\overline{\|Y\|}$ with $|\overline{\|Y\|} - \|Y\|\,| \leq \eta \,\|Y\|$ with an additional $1/\eta$ factor.

New algo! QFDC (Supervised)

$X_k \in \mathbb{R}^{|T_k| \times d}$ matrix of elements labeled k
$X_0 \in \mathbb{R}^{|T_k| \times d}$ repeats the row x_0 for $|T_k|$ times.

$$F_k(x_0) = \frac{\|X_k - X_0\|_F^2}{2(\|X_k\|_F^2 + \|X_0\|_F^2)},$$

$$\frac{1}{\sqrt{N_k}}\Big(|0\rangle \sum_{i \in T_k} \|x(0)\| \, |i\rangle \, |x(0)\rangle + |1\rangle \sum_{i \in T_k} \|x(i)\| \, |i\rangle \, |x(i)\rangle \Big)$$

$$h(x_0) = min_k\{F_k(y_0) = p(|1\rangle)\}$$

Combining QSFA and QFDC

Require:

X and $\dot{X}$ in QRAM, test vector $x(0)$, and $\varepsilon, \eta > 0$ error param.

Ensure:

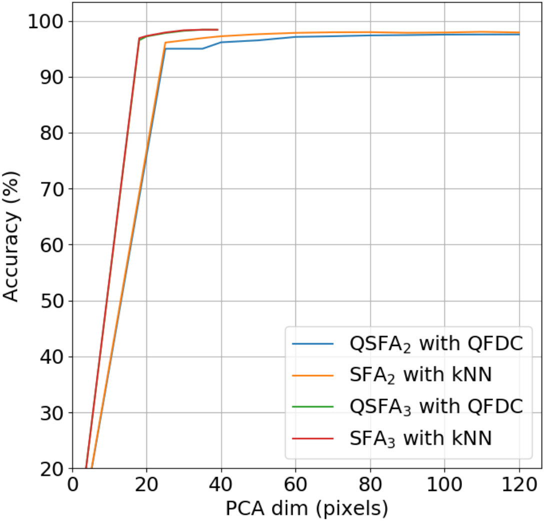
A label for $x(0)$

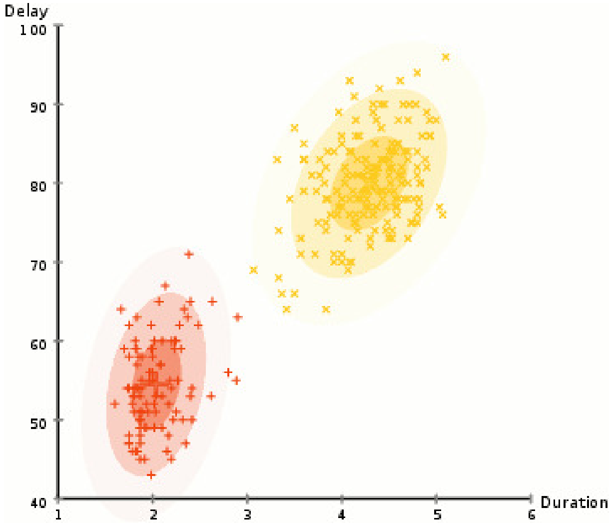
```
for k = 1 → K do
    s_k := 0
    Use QSFA to estimate ‖Y_k‖_F and the norm ‖y(0)‖ (to error η).
    for r = O(1/η^2) do
```

$$\frac{1}{\sqrt{N_k}}\Big(|0\rangle \sum_{i \in T_k} \|y(0)\| \, |i\rangle \, |y(0)\rangle + |1\rangle \sum_{i \in T_k} \|y(i)\| \, |i\rangle \, |y(i)\rangle \Big)$$

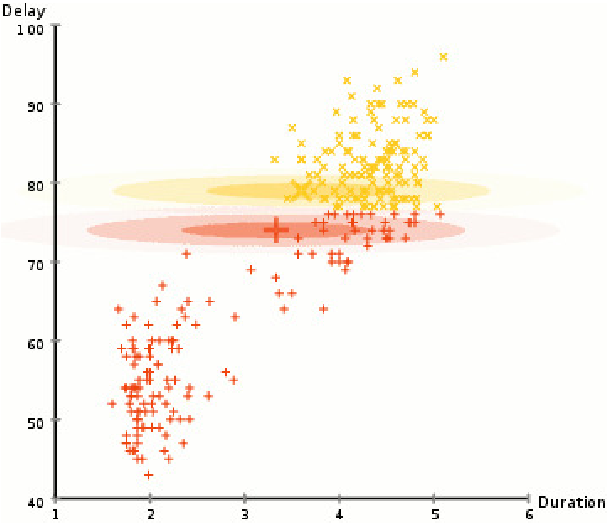
```
        Apply a Hadamard to the first register
        Measure first reg. If |1⟩ then s_k := s_k + 1
    end for
    Estimate F_k(y(0)) := s_k / r (to error O(ε + η)).
end for
```

Accuracy QSFA+QFDC

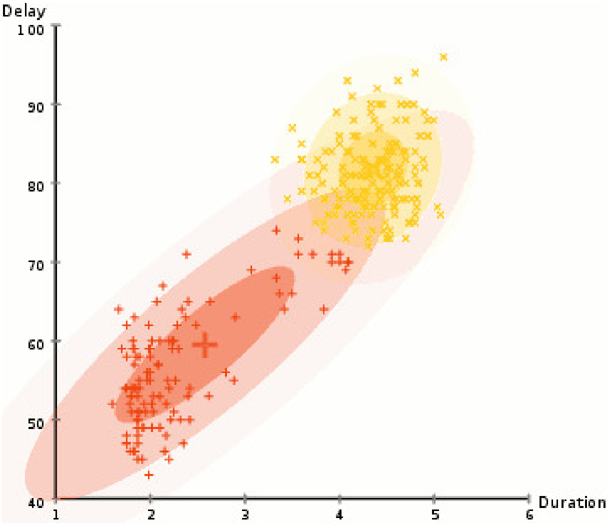




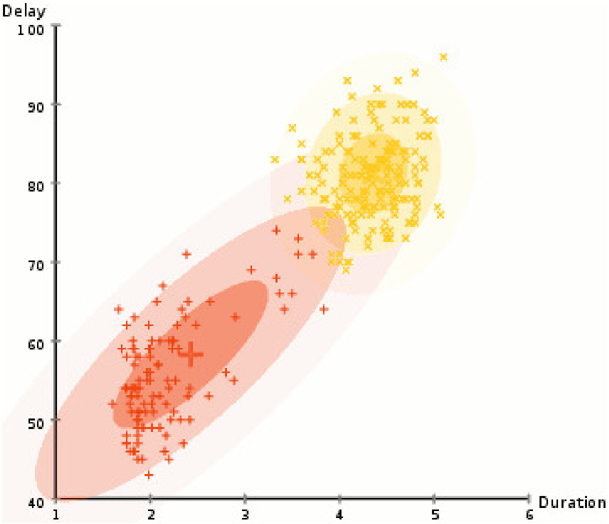
From Wikipedia



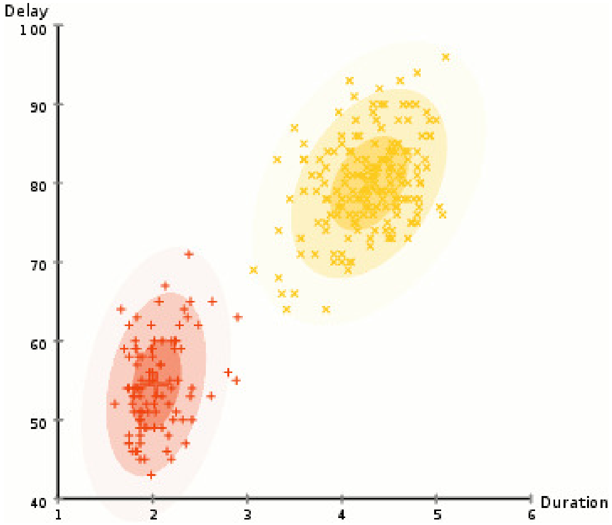
From Wikipedia



From Wikipedia



From Wikipedia



From Wikipedia

Well-clustered data

The data is $(\xi, \beta, \lambda, \eta)$-well clustered if there are $\xi > 0, \quad \beta > 0, \quad 0 \ll \lambda < 1, \quad \eta > 1:$

1. clusters' separation: $d(c_i, c_j) \geq \xi \quad \forall i, j \in [k]$
2. proximity to centroid: A fraction λn of points v_i in the dataset verify: $d(v_i, c_{l(v_i)}) \leq \beta$.
3. dataset's width: All the norms are between 1 and $\eta = \max_i (\|v_i\|)$

k-means (Unsupervised)

Find initial centroids c_j

Repeat until centroids are steady: $|c_j^t - c_j^{t+1}| \leq \tau$

- Calculate distances between all points and all clusters

$$\forall i \in [n], c \in [k] \quad d(v_i, c_i)$$

- Assign points to closer cluster

$$l(v_i) = \arg\min_{c \in [k]} d(v_i, c_i)$$

- Calculate centroids again

$$c_j = \frac{1}{|C_j|} \sum_{i \in C_j} v_i$$

k-means (Unsupervised)

Find initial centroids c_j

Repeat until centroids are steady: $|c_j^t - c_j^{t+1}| \leq \tau$

- Calculate distances between all points and all clusters

$$\forall i \in [n], c \in [k] \quad d(v_i, c_i)$$

- Assign points to closer cluster

$$l(v_i) = \arg\min_{c \in [k]} d(v_i, c_i)$$

- Calculate centroids again

$$c_j = \frac{1}{|C_j|} \sum_{i \in C_j} v_i$$

... is $O(tndk)$:(

δ-k-means (Unsupervised)

Find initial centroids c_j

Repeat until centroids are steady: $|c_j^t - c_j^{t+1}| \leq \tau$

- Calculate distances between all points and all clusters

$$\forall i \in [n], c \in [k] \quad d(v_i, c_i)$$

- Assign points to closer cluster

$$L_\delta(v_i) = \{c_p \ |d^2(c_i^*, v_i) - d^2(c_p, v_i)| \leq \delta \}$$

$$l(v_i) = rand(L_\delta(v_i))$$

- Calculate centroids again

$$c_j = \frac{1}{|C_j|} \sum_{i \in C_j} v_i$$

q-means (Unsupervised)

Find initial centroids c_j

Repeat until centroids are steady: $|c_j^t - c_j^{t+1}| \leq \tau$

- Calculate distances between all points and all clusters

$$\bigotimes_{j=0}^{K} \sum_{i=0}^{n} |i\rangle |j\rangle |d(v_i, c_i)\rangle$$

- Assign points to closer cluster

$$\sum_{i=0}^{n} |i\rangle |l(i)\rangle$$

- Calculate centroids again

$$\frac{1}{\sqrt{Z}} \sum_{j\in[k]} \left\| c_j^{t+1} \right\| |c_j^{t+1}\rangle |j\rangle$$

Recovering centroids

$$(V^T \otimes I) \cdot U_{norms} \cdot \sum_{i=0}^{n} |i\rangle \, |l(i)\rangle$$

$$\left(V^T \otimes I\right) \left(\frac{1}{\sqrt{\|V\|_F}} \sum_{i=0}^{n} \|v_i\| \, |i\rangle \, |\ell(v_i)^t\rangle \right)$$

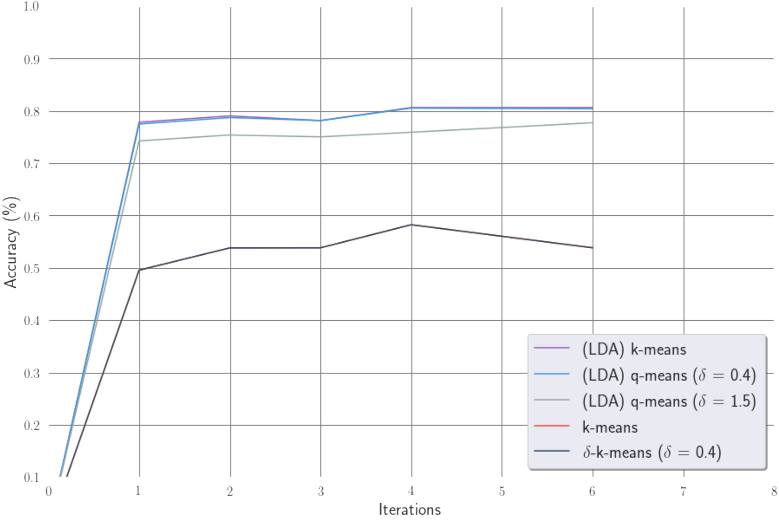
$$= \sum_{j \in [k]} \frac{1}{\sqrt{\sum_{\mathcal{C}_j^{t+1}} \|v_i\|}} \sum_{i \in \mathcal{C}_j^{t+1}} \|v_i\| \, |v_i\rangle \, |j\rangle$$

$$\frac{1}{Z} \sum_{j \in [k]} \left\| c_j^{t+1} \right\| \, |c_j^{t+1}\rangle \, |j\rangle$$

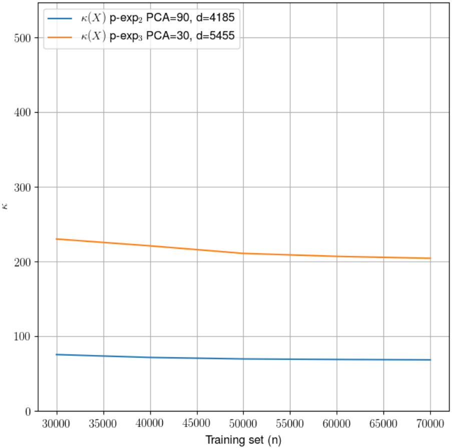
New algo! q-means

For a $(\xi, \beta, \lambda, \eta)$-well clustered dataset $V \in \mathbb{R}^{n \times d}$ in QRAM, there is a quantum algorithm that returns in t steps the k centroids that cluster the dataset consistently with the classical δ-k-means algorithm in time $\widetilde{O}\left(t \cdot \frac{k^2 d Z^{5/2} \kappa(V)}{\delta^3}\right)$.

Accuracy q-means

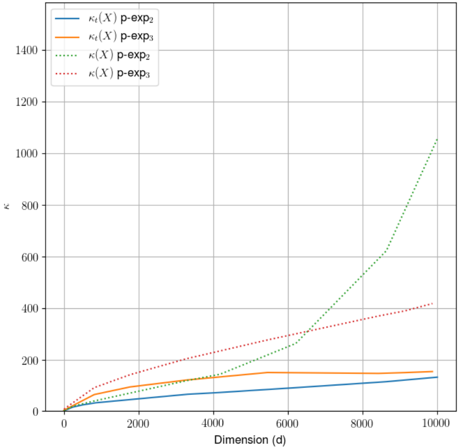


$\lambda_{max}/\lambda_{min}$: more data



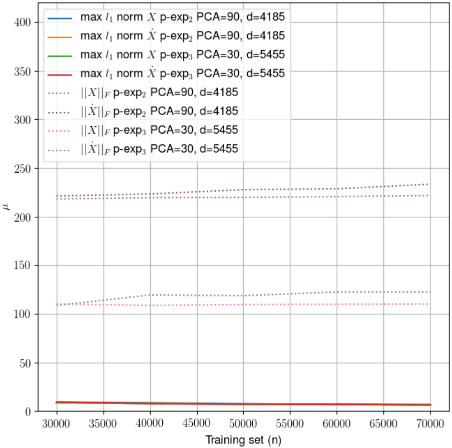
Condition number by increasing the number of elements in training set

$\lambda_{max}/\lambda_{min}$: more feature



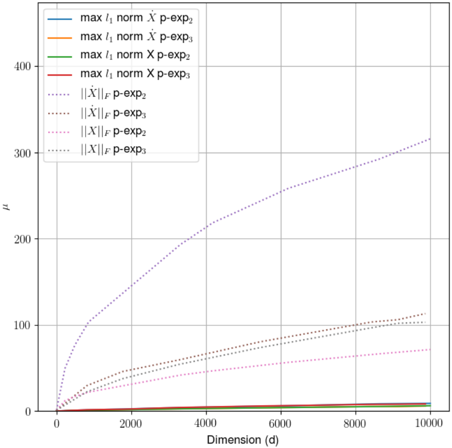
Condition number by increasing the features (pixels)

$\mu(X)$: more data



$\mu(X)$ and $\mu(\dot{X})$ by increasing the number of elements in training set.

$\mu(X)$: more features



$\mu(X)$ and $\mu(\dot{X})$ by increasing the number of features.

#TODOs

- Generalizations...
- Experiments...
- Code...
- New algos...
- Compositions...
- Adversarial QML...
- Privacy preserving QML...

Thanks for your time

there is never enough.
(cit. Dan Geer)

@scinawa

- Quantum Machine Learning ⇒ https://luongo.pro/qml
- QSFA + QFDC ⇒ https://arxiv.org/abs/1805.08837
- q-means ⇒ stay tuned...