Quantum Machine Learning:

Prospects and challenges

Iordanis Kerenidis

iordanis.kerenidis@qcware.com

Why Quantum Machine Learning?

Why Quantum?

Why Quantum?

- Quantum is NOT a faster processor

Why Quantum?

- Quantum is NOT a faster processor
- Quantum is a fundamentally DIFFERENT way of performing computation that can be MUCH FASTER for CERTAIN tasks

Why Quantum?

- Quantum is NOT a faster processor
- Quantum is a fundamentally DIFFERENT way of performing computation that can be MUCH FASTER for CERTAIN tasks
- Quantum computing will NOT replace classical computing, but we expect it to remove certain bottlenecks and open the way to new applications

Why Quantum?

- Quantum is NOT a faster processor
- Quantum is a fundamentally DIFFERENT way of performing computation that can be MUCH FASTER for CERTAIN tasks
- Quantum computing will NOT replace classical computing, but we expect it to remove certain bottlenecks and open the way to new applications
- We need to rethink and invent new algorithmic solutions

Why Quantum Machine Learning?

Why Quantum Machine Learning?

- We KNOW large quantum computers can offer provable theoretical advantages
    - Classification, Rec. Systems, q-means, Boosting, Exp-Max, etc. [ICML, NeurIPS, ICLR]

Why Quantum Machine Learning?

- We KNOW large quantum computers can offer provable theoretical advantages
  - Classification, Rec. Systems, q-means, Boosting, Exp-Max, etc. [ICML, NeurIPS, ICLR]

- We HAVE concrete avenues for bringing QML closer to reality
  - Reducing resource requirements of impactful QML algorithms
  - Proposing new QML specific hardware architectures for overcoming bottlenecks
  - “Noisy” calculations can be handled by ML algorithms

Why Quantum Machine Learning?

- We KNOW large quantum computers can offer provable theoretical advantages
  - Classification, Rec. Systems, q-means, Boosting, Exp-Max, etc. [ICML, NeurIPS, ICLR]

- We HAVE concrete avenues for bringing QML closer to reality
  - Reducing resource requirements of impactful QML algorithms
  - Proposing new QML specific hardware architectures for overcoming bottlenecks
  - “Noisy” calculations can be handled by ML algorithms

- Powerful and very subtle quantum tools for ML

Why Quantum Machine Learning?

- We KNOW large quantum computers can offer provable theoretical advantages
  - Classification, Rec. Systems, q-means, Boosting, Exp-Max, etc. [ICML, NeurIPS, ICLR]

- We HAVE concrete avenues for bringing QML closer to reality
  - Reducing resource requirements of impactful QML algorithms
  - Proposing new QML specific hardware architectures for overcoming bottlenecks
  - “Noisy” calculations can be handled by ML algorithms

- Powerful and very subtle quantum tools for ML

- QML may offer: Efficiency, Accuracy, Interpretability, Trust, Energy savings

Quantum Tools for ML

Quantum Toolbox

1. Quantum Distance Estimators [arXiv:1401.2142, :1805.08837, :1812.03584]
   We can efficiently estimate distances between quantum states/data points.

Quantum Toolbox

1. Quantum Distance Estimators [arXiv:1401.2142, :1805.08837, :1812.03584]
   We can efficiently estimate distances between quantum states/data points.

2. Quantum Dimensionality Reduction techniques
   - Principal Component Analysis [Loyd, Mohseni, Rebentrost 13]
   - Linear Discriminant Analysis [Cong, Duan 15]
   - Slow Feature Analysis [Kerenidis, Luongo 18]

Quantum Toolbox

1. **Quantum Distance Estimators** [arXiv:1401.2142, :1805.08837, :1812.03584]
   We can efficiently estimate distances between quantum states/data points.

2. **Quantum Dimensionality Reduction techniques**
   - Principal Component Analysis [Loyd, Mohseni, Rebentrost 13]
   - Linear Discriminant Analysis [Cong, Duan 15]
   - Slow Feature Analysis [Kerenidis, Luongo 18]

3. **Quantum Linear Algebra**
   - Gradients (applications in NN training, linear regression, etc.)
   - Linear Systems [Harrow, Hassidim, Lloyd 09, GSLW 19]

Supervised Learning

Classification

- Distance-based classification

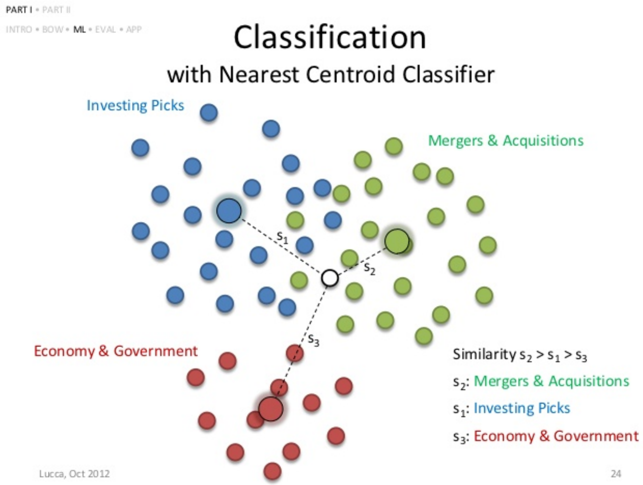
  k-Nearest Neighbors, Nearest Centroids

- Support Vector Machines

  [RML 2013, Kerenidis,Prakash,Szilagyi 2019]

- Quantum Neural Networks

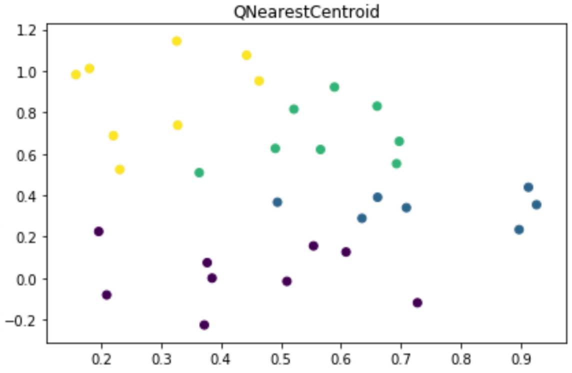
  FF NN, Convolutional NN, Variational circuits, ...

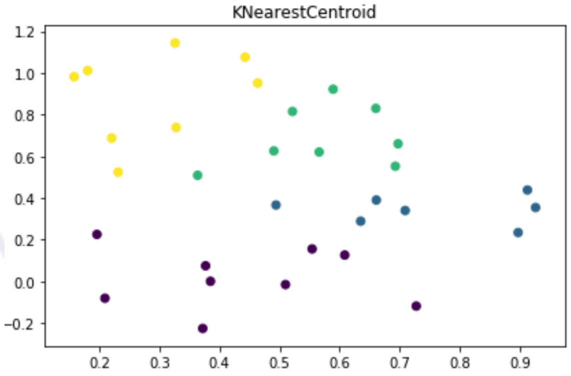


Classification - QNearestCentroid

Nearest Centroid Classifier

- Fit your model
  - Find centroids of each class of training data
- Predict labels of new data
  - Compute distance classically of new data to centroids
  - Assign the label of the nearest centroid

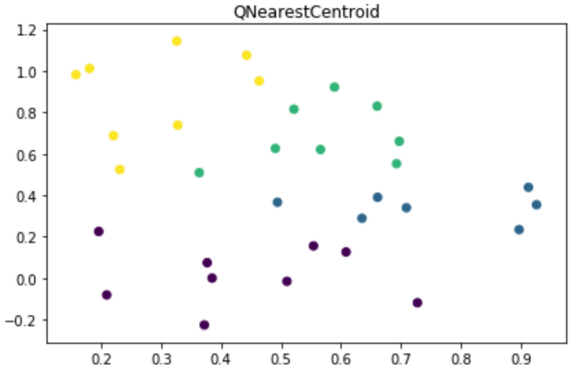


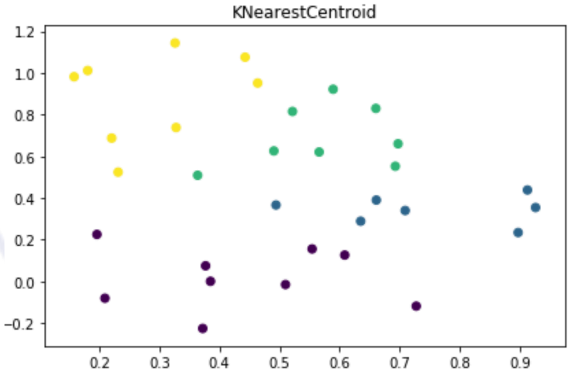


Classification - QNearestCentroid

Nearest Centroid Classifier

- Fit your model
  - Find centroids of each class of training data
- Predict labels of new data
  - Compute distance quantumly of new data to centroids
  - Assign the label of the nearest centroid

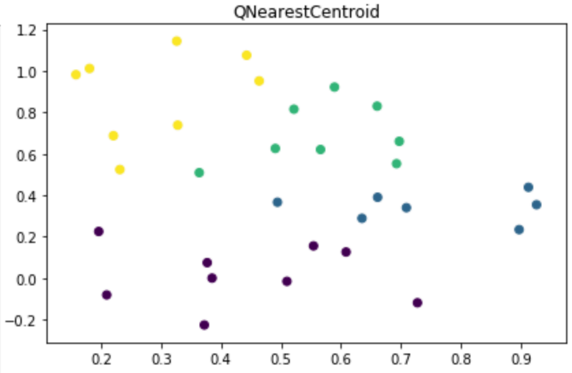


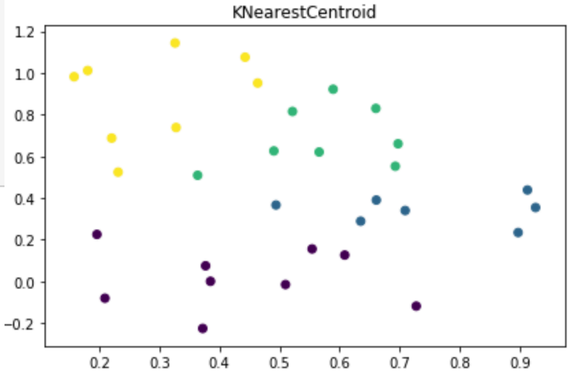


Classification - QNearestCentroid

```
# let's create some synthetic data
X, y = generate_data_clusters()

# let's run the quantum classifier
qlabels = fit_and_predict(X,y=y,model='QNearestCentroid')

#import NearestCentroid from scikit-learn for benchmarking
clabels = sklearn.neighbors.NearestCentroid().fit(X,y).predict(X)

print('Quantum labels\n',qlabels)
print('Classical labels\n',clabels)

# let's plot the data (only for dimension=2)
plot(X,qlabels,'QNearestCentroid')
plot(X,clabels,'KNearestCentroid')
```

```
Quantum labels
 [2 0 0 1 0 0 0 0 1 1 2 0 1 1 0 1 2 3 2 1 2 2 2 3 3 3 3 3 0 3 3 2]
Classical labels
 [2 0 0 1 0 0 0 0 1 1 2 0 1 1 0 1 2 3 2 1 2 2 2 3 3 3 3 3 0 3 3 2]
```





Recommendation Systems [Kerenidis, Prakash, ITCS 17]

amazon

Step 1

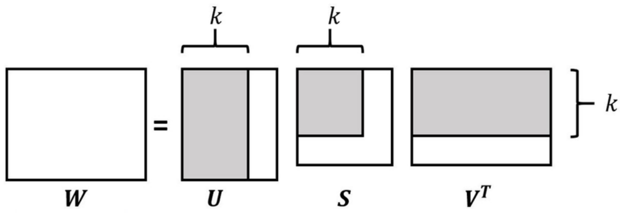
Products

Users

?	**1**	**…**	**?**
?	0	…	?
…	…	…	…
1	?	…	0

Step 2

Singular Value Estimation

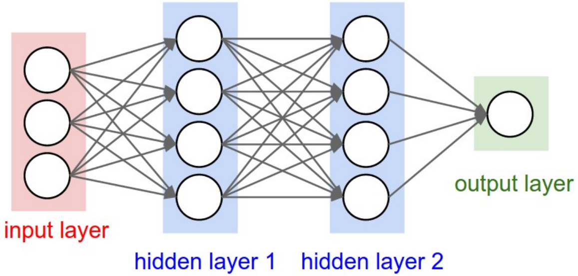


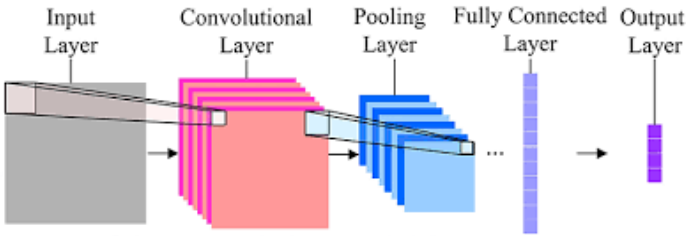
Step 3

Efficient quantum algorithm for Singular Value Estimation

Quantum computers provide competitive recommendations fast!

Quantum Neural Networks





Accuracy/Convergence : at least as good as classical Neural Nets

Running time: gains from IP estimation for training and evaluation

[Allcock, Hsieh, Kerenidis, Zhang 19], [Kerenidis,Landman,Prakash ICLR 20]

Main challenge: Define QNNs with provable guarantees

Unsupervised Learning

Clustering

k-means++

Input: N points in d-dimensions
Output: K clusters/centroids

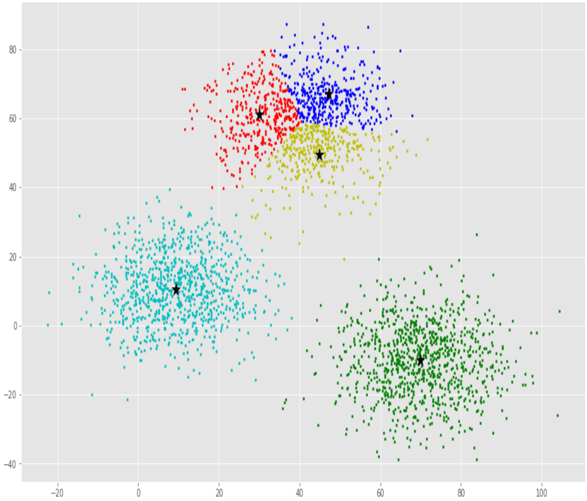
1. Start with some initial centroids (e.g. ++-method)

Repeat until convergence

2. For each point

   estimate distances to centroids
   and assign to nearest centroid

3. Update the centroids



Clustering

q-means++ [Kerenidis,Landman,Luongo, Prakash NeurIPS 2019]

Input: N points in d-dimensions (quantum access)
Output: K clusters/centroids

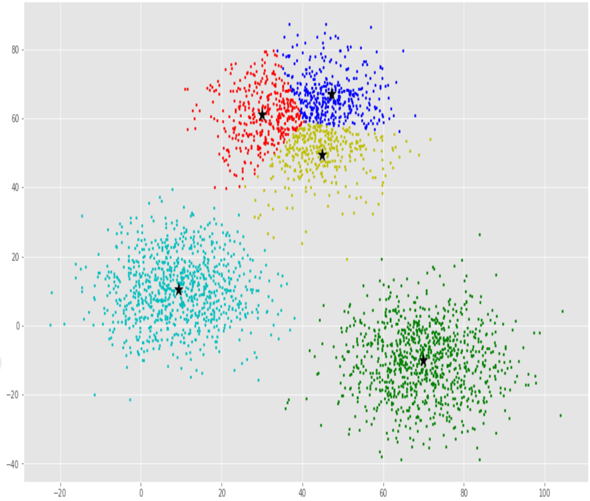
1. Start with some initial centroids (e.g. ++-method)

Repeat until convergence

2. For all points in superposition

   estimate distances to centroids quantumly
   and assign to nearest centroid

3. Update the centroids
   i. Quantum linear algebra to find new centroid
   ii. Tomography to recover classical description

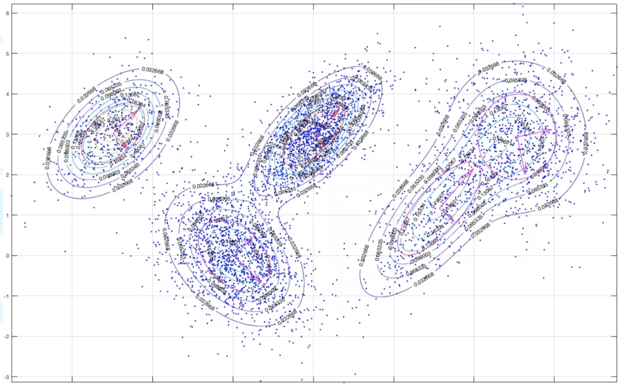


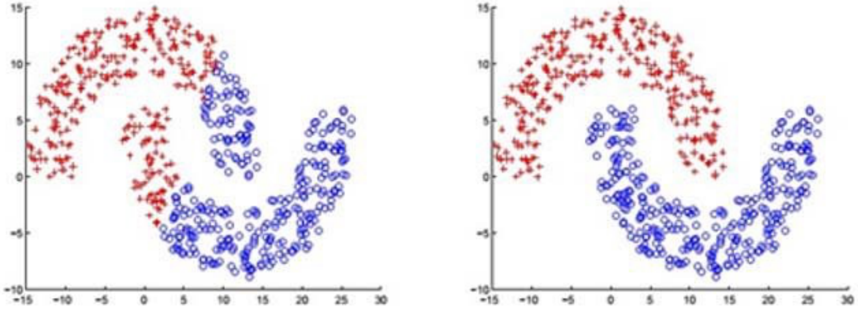
Clustering

Extensions

- Expectation Maximization for Gaussian Mixture Models [Kerenidis,Luongo,Prakash ICML 2020]

- Spectral Clustering [Kerenidis,Landman 2020]





(a) K-means

(b) Spectral Clustering

Reinforcement Learning

Reinforcement Learning

Quantum Policy Iteration [Cherrat,Kerenidis,Prakash 2020]

Input: states S, actions A, transitions P, Rewards R
Output: policy π

1. Start with some initial π_0

Repeat until convergence

2. solve $(I-\gamma P^{\pi})Q=R$ quantum linear systems
3. update π from Q by measurements

Remarks

"No input" / Well-conditioned / ℓ^{∞} guarantees

Conclusions

Prospects - Challenges

- Powerful yet subtle quantum tools

  Power: Linear Algebra, Distance Estimations, Expectations, etc.

  Subtleties: Input, Output, running time parameters

Prospects - Challenges

- Powerful yet subtle quantum tools

  Power: Linear Algebra, Distance Estimations, Expectations, etc.

  Subtleties: Input, Output, running time parameters

- Promising directions

  - Heavy Linear Algebra (Dim. Reduction, SVM, Spectral Clustering)
  - Reinforcement Learning (efficient data, well-conditioned systems, ℓ^∞ approx)
  - Quantum Neural Networks

Prospects - Challenges

- Powerful yet subtle quantum tools

  Power: Linear Algebra, Distance Estimations, Expectations, etc.

  Subtleties: Input, Output, running time parameters

- Promising directions
  - Heavy Linear Algebra (Dim. Reduction, SVM, Spectral Clustering)
  - Reinforcement Learning (efficient data, well-conditioned systems, ℓ^∞ approx)
  - Quantum Neural Networks

- Final remarks
  - ML is about practical solutions to real-world problems.
  - QML applications is a formidable challenge, but certainly worth pursuing

Thank you!

Iordanis Kerenidis

jkeren@irif.fr, iordanis.kerenidis@qcware.com