Programming Quantum Computers (Apps V: Machine Learning)

(**Subtrack of** Quantum Computing: An App-Oriented Approach)

Moez A. AbdelGawad

moez@{cs.rice.edu, alexu.edu.eg, srtacity.sci.eg}

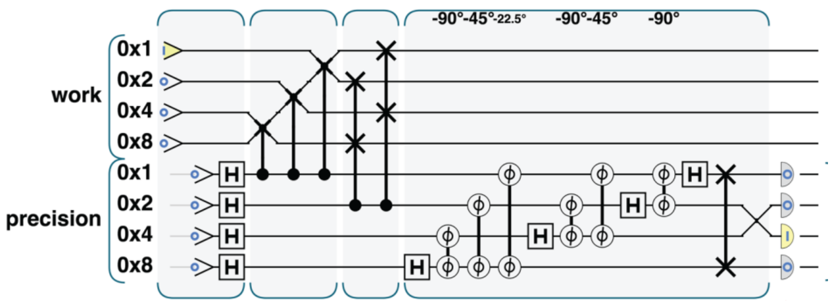
Sat., Jan. 11th, 2020

Quantum Computers are Real

- What are they <u>useful</u> for?
  - Let's discover, by programming them!

- A hands-on approach to programming QCs/QPUs.
  - By doing; i.e., by writing code & building programs.
  - Using simulators, since real QCs are harder-to-access (so far).

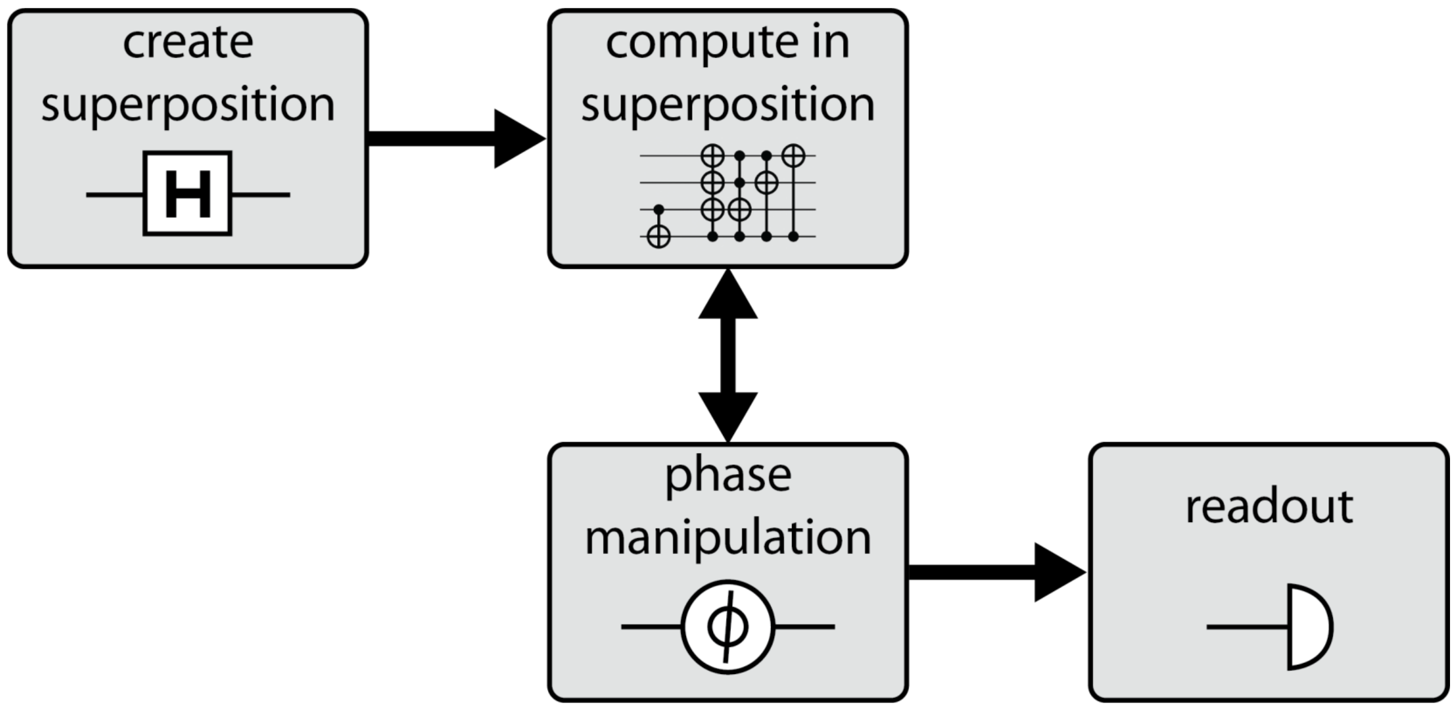
- Goals: Read, understand, write, and *debug* quantum applications.
  - Ones like this program.



Topics Covered

- Qubit, Superposition, Entanglement.
- Single-Qubit Ops: H, NOT and Phase.
- Multi-Qubit Ops: Conditional Ops.
- Teleportation.
- Quantum Arithmetic and Logic.
- (Quantum) Amplitude Amplification.
  - Converting phase info into magnitude info.
- Quantum Fourier Transform.
  - Revealing patterns (frequencies).
- (Quantum) Phase Estimation.
  - Characterization of quantum operations.
- QRAM, Vector & Matrix Quantum Encodings, and Quantum Simulation.

Structure of Quantum Apps



QUANTUM APPLICATIONS

QUANTUM MACHINE LEARNING

Lecture Outline

- What is QML?
- Solving Systems of Linear Equations.
- Quantum Principle Component Analysis.
- Quantum Support Vector Machines.
- Other Machine Learning Applications.

QML

Systems of Linear Equations

- Elementary school puzzles.
  - Fruit salad puzzles.
    - TODO: Fix. 2 bananas and 3 apples and 5 oranges = 17
    - 3 oranges and 7 apples = 15
    - 2 bananas and 6 oranges =??
    - A + A + A = 30. B + B - A = 2. O + O + B = 18. A + B x O = ?? (46).
  - “Cat, tortoise and table” puzzle <<Put pic>>
    - T + C - O = 170. T + O - C = 130. T = ?? (150).
- A System of Linear Equations = A Matrix.
  - Math of quantum *applications* not of quantum programming.

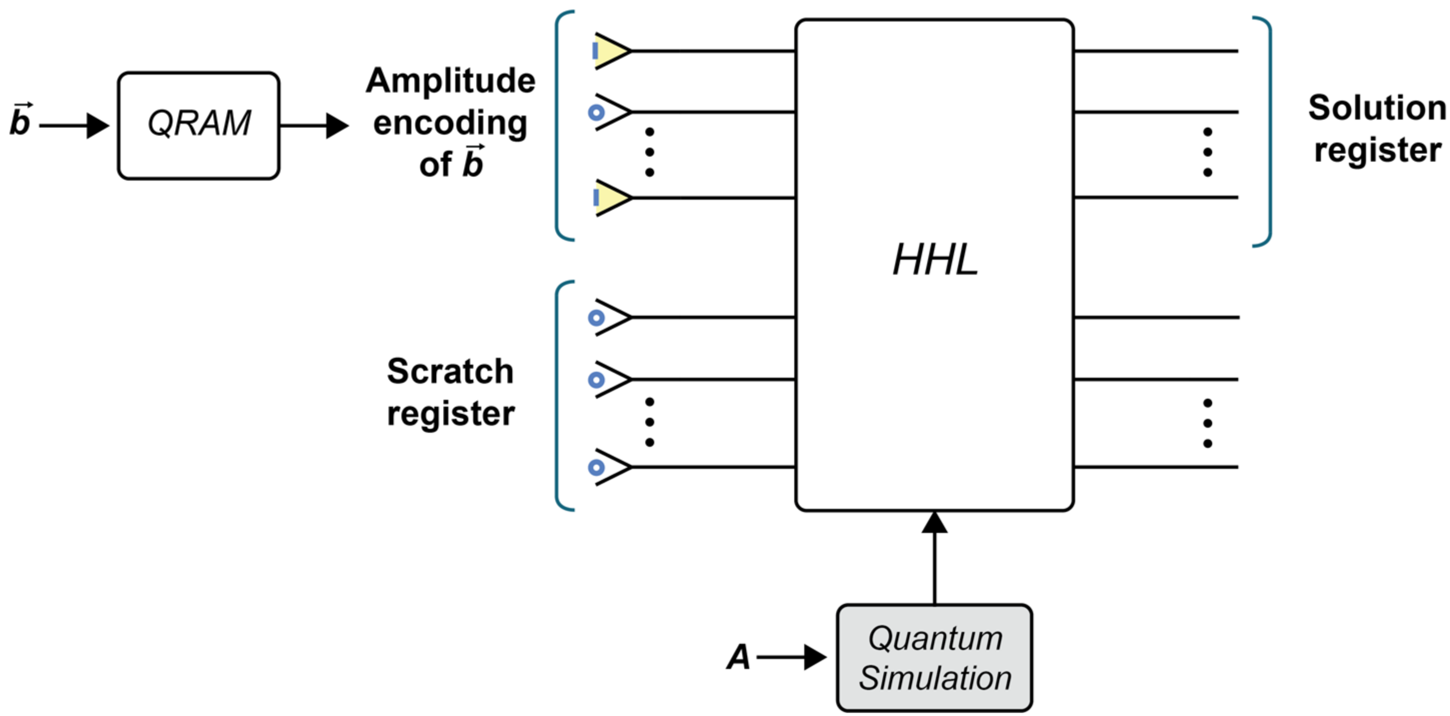
Systems of Linear Equations

- $3x_1 + 4x_2 = 3$ and $2x_1 + x_2 = 3$.
  - $\begin{pmatrix} 3 & 4 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 3 \end{pmatrix}$.
  - General form: $\boldsymbol{Ax} = \boldsymbol{b}$ (matrix $\boldsymbol{A}$, and vectors $\boldsymbol{x}$ and $\boldsymbol{b}$).

- Matrix Inversion.
  - $\boldsymbol{Ax} = \boldsymbol{b} \Rightarrow \boldsymbol{x} = \boldsymbol{A}^{-\mathbf{1}}\boldsymbol{b}$.
  - Has many, *many* applications.
  - *Conjugate gradient descent* (cgd) is probably the most efficient conventional matrix inversion algorithm.
    - Depends on matrix possessing certain helpful properties.
  - The HHL algorithm can efficiently compute (faster than *cgd*) an amplitude-encoding of $\boldsymbol{A}^{-\mathbf{1}}\boldsymbol{b}$.
    - Quantum output. Still very useful; a critical building block in other QML apps.
  - HHL: Provides *quantum* answers to puzzles (oranges=??, apples=??, bananas=??).

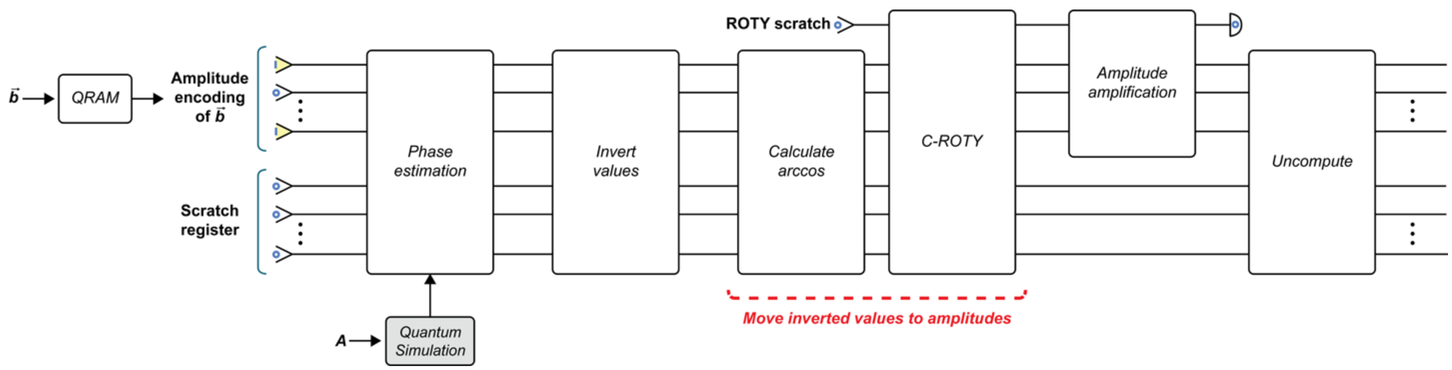
HHL

HHL

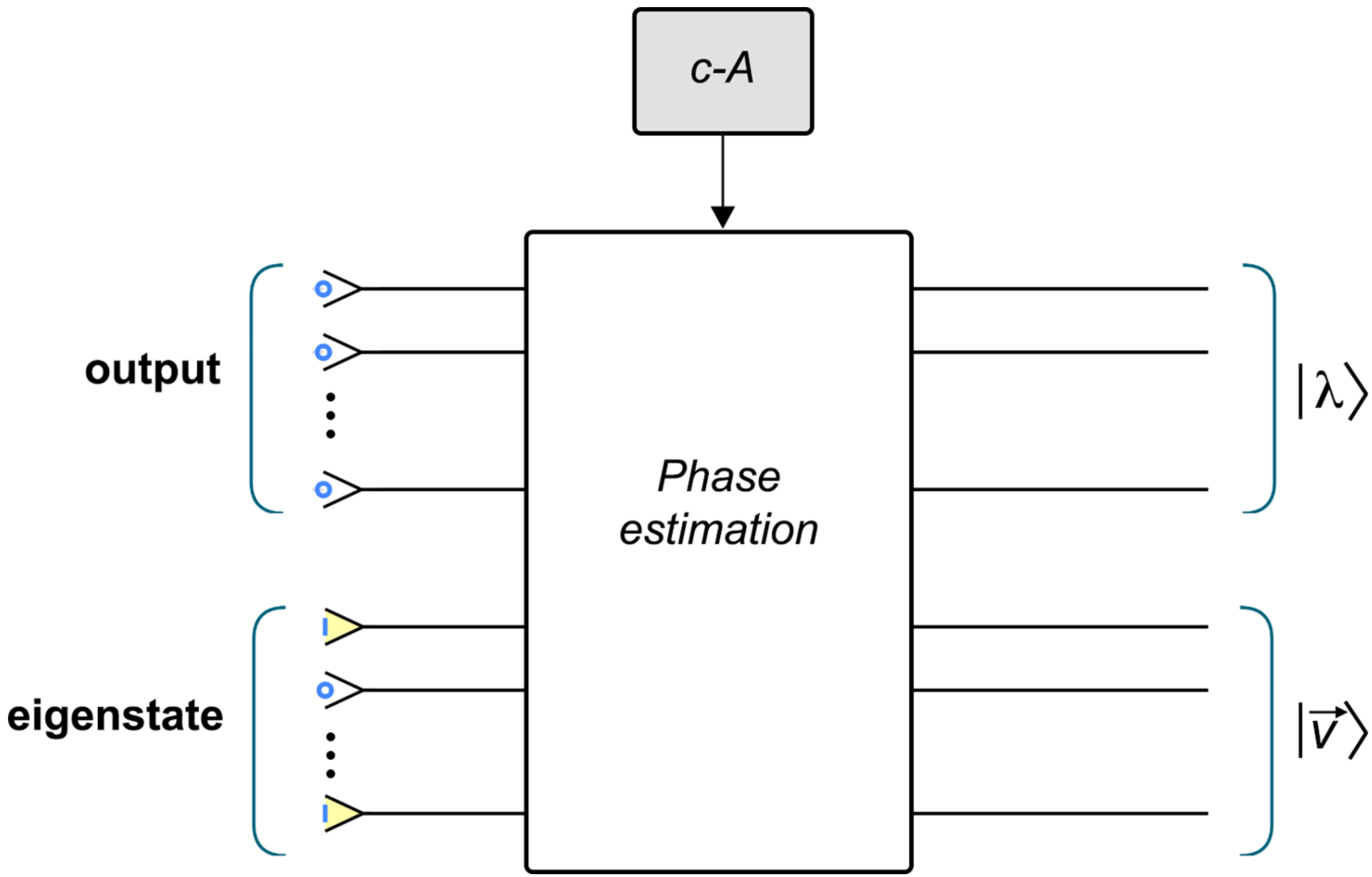


$\vec{b}$
QRAM
Amplitude encoding of $\vec{b}$
Scratch register
HHL
Solution register
A
Quantum Simulation

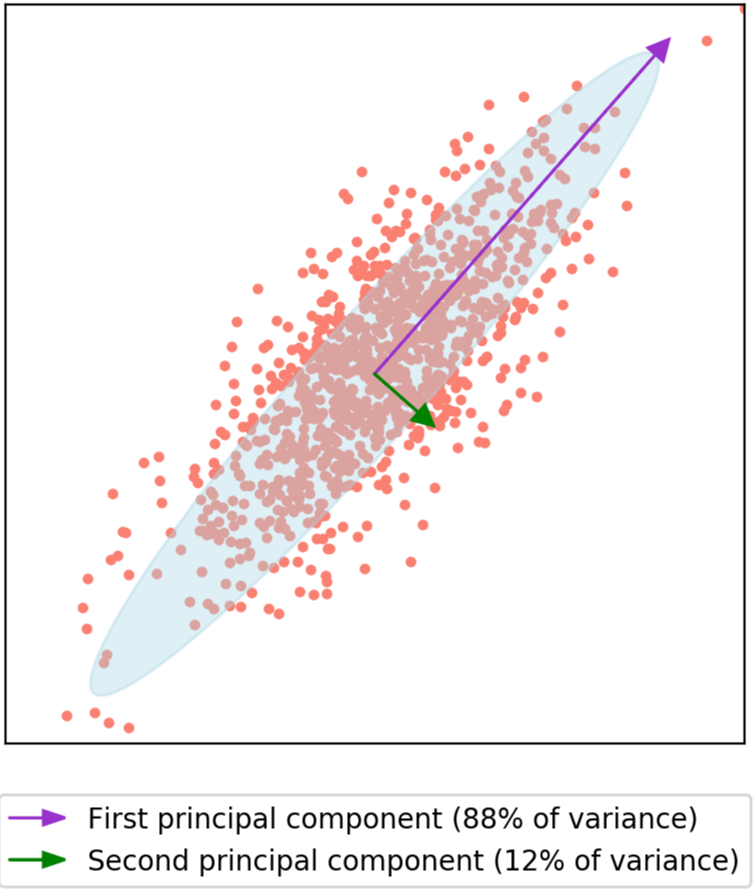
HHL



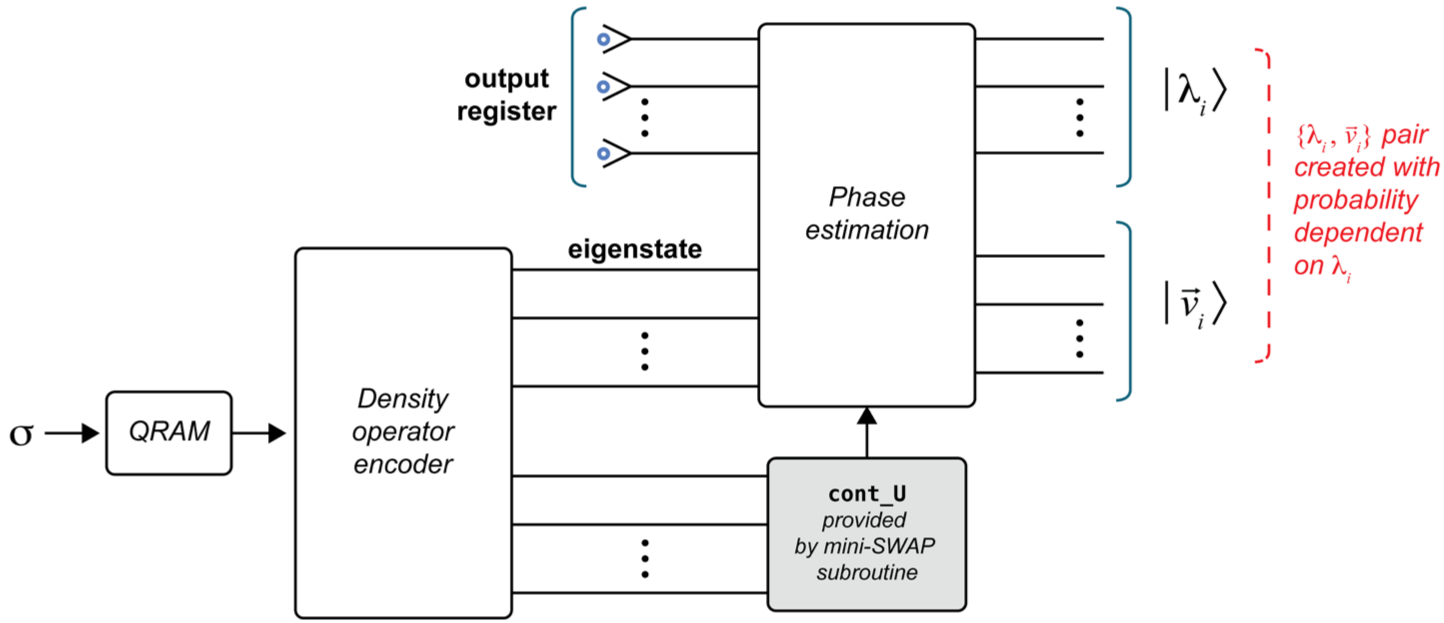
$\vec{b}$
QRAM
Amplitude encoding of $\vec{b}$
Scratch register
ROTY scratch
Phase estimation
Invert values
Calculate arccos
C-ROTY
Amplitude amplification
Uncompute
A
Quantum Simulation
Move inverted values to amplitudes


c-A
output
Phase
estimation
eigenstate
$|\lambda\rangle$
$|\vec{v}\rangle$

QSVM

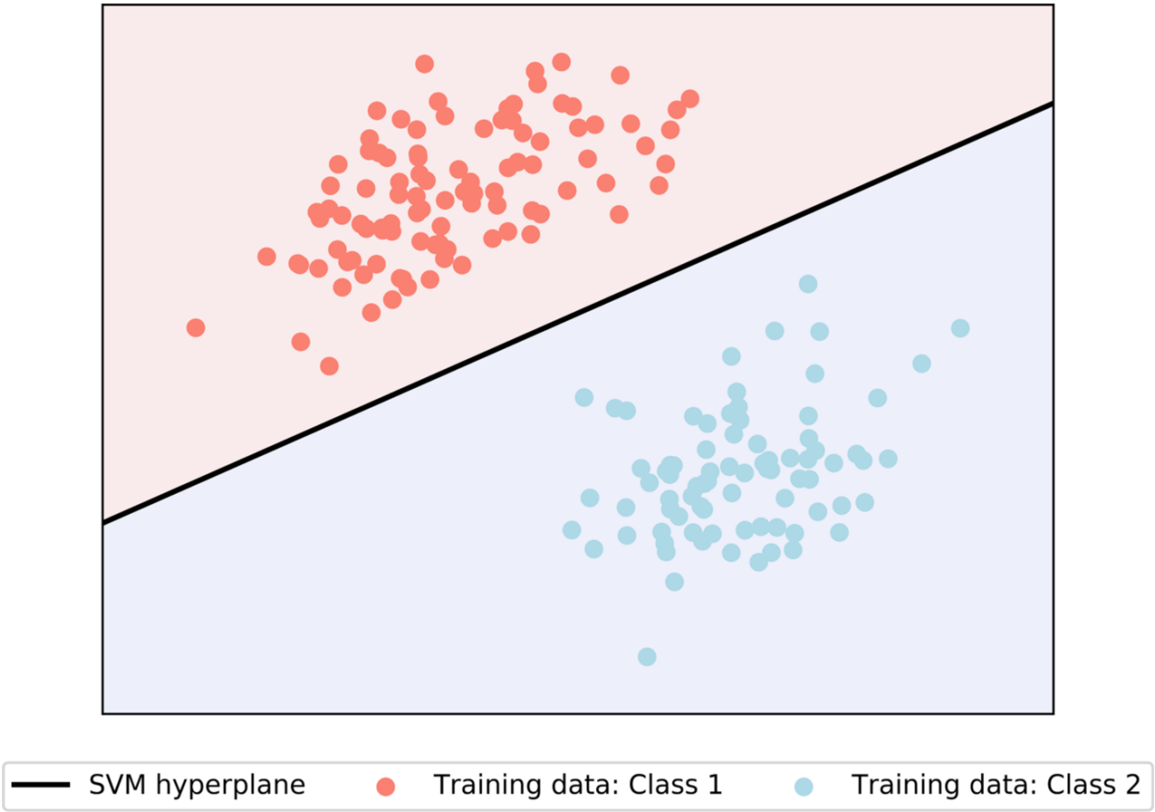

First principal component (88% of variance)
Second principal component (12% of variance)

QSVM

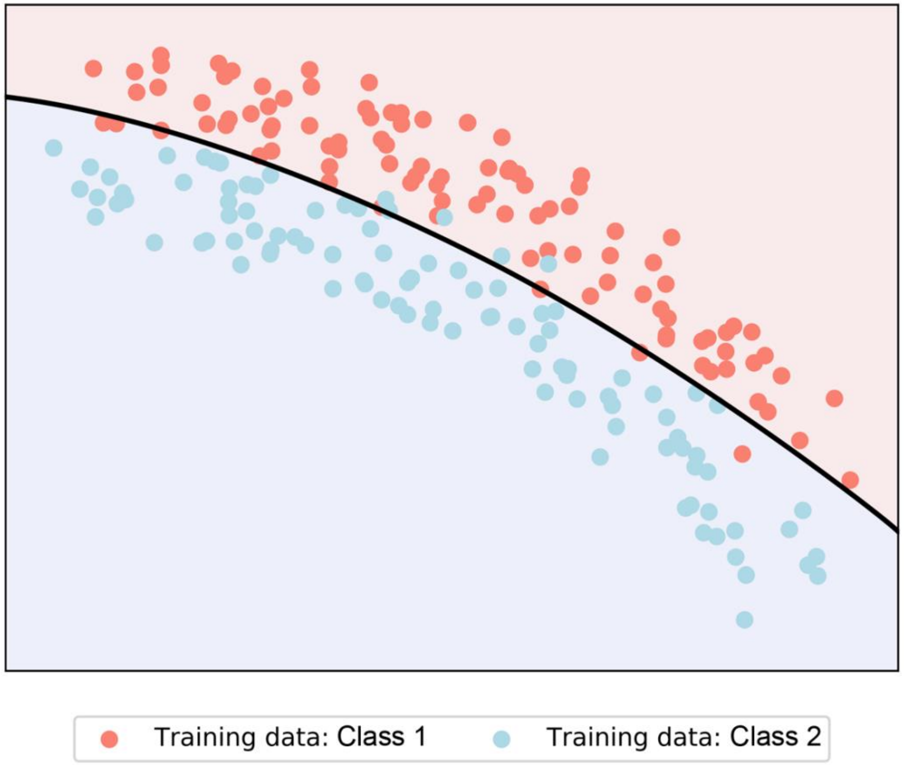


output register
eigenstate
Phase estimation
$|\lambda_i\rangle$
$|\vec{v}_i\rangle$
$\{\lambda_i, \vec{v}_i\}$ pair created with probability dependent on λ_i
σ
QRAM
Density operator encoder
cont_U provided by mini-SWAP subroutine

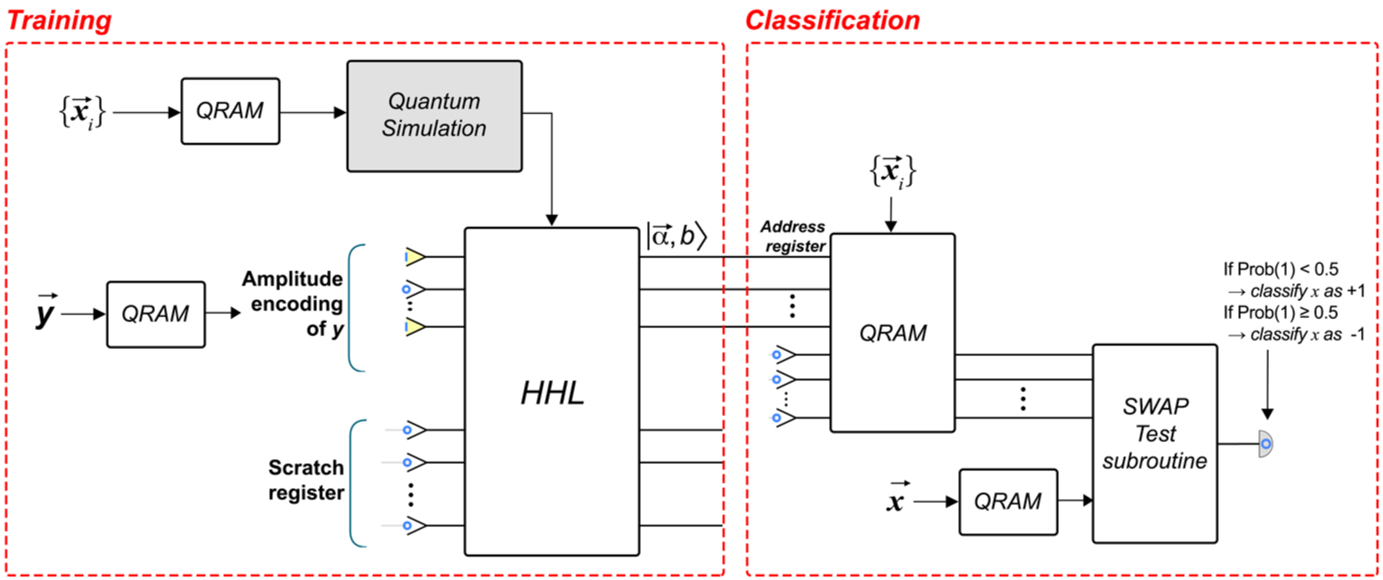
QSVM



QSVM



QSVM



Training
Classification
{x_i}
QRAM
Quantum Simulation
y
QRAM
Amplitude encoding of y
Scratch register
HHL
|α, b⟩
Address register
{x_i}
QRAM
If Prob(1) < 0.5 → classify x as +1
If Prob(1) ≥ 0.5 → classify x as -1
SWAP Test subroutine
x
QRAM

Projects

- QML:
  - Implement HHL in QCEngine, QX, Q#, Cirq or Quirk.
    - Or in some other real quantum system or quantum simulator.
  - Implement QPCA and QSVM.

- Quantum Cryptography:
  - Implement Shor’s factoring algorithm.
  - Break (a tiny version of) RSA!
  - Implement QKD.

- Quantum Image Processing (QIP):
  - Implement QSS.

QC/QP: What Next?

- Hope you enjoyed the course!
  - Send me your feedback:

    `moez@alexu.edu.eg`.

- ‘Staying on Top’:
  - Summary of Ch.14.
  - QPLs:
    - QML Overview.

QML Overview

- QML: A functional quantum programming language.
  - Ref: ‘An Algebra of Pure Quantum Programming’, Altenkirch et al. (& GVS), ENTCS, 2007.

- QML Code Example:

$$
\begin{aligned}
H\ x = {} & \mathbf{if}^{\circ}\ x \\
& \mathbf{then}\ (\mathit{false} + (-1) * \mathit{true}) \\
& \mathbf{else}\ \ (\mathit{false} + \mathit{true})
\end{aligned}
$$

- Wanted: Prove that $H\ (H\ x)$ is observationally equivalent to x. (i.e., $H^2 = id$).

QML Overview

- QML definition (P.4 in QPLRef1):

$$
\begin{array}{llll}
p, q & ::= x \mid (x, y) & (\mathit{Patterns}) & (\mathit{Variables}) \quad x, y, \ldots \in \mathit{Vars} \\
t, u, e & ::= x \mid () \mid (t, u) & (\mathit{Terms}) & (\mathit{Prob.amplitudes}) \quad \kappa, \iota, \ldots \in \mathbb{C} \\
& \mid \quad \mathbf{let}\ p = t\ \mathbf{in}\ u & & \\
& \mid \quad \mathbf{if}^{\circ}\ t\ \mathbf{then}\ u\ \mathbf{else}\ u' & & \\
& \mid \quad \mathit{false} \mid \mathit{true} \mid \overrightarrow{0} \mid \kappa * t \mid t + u & &
\end{array}
$$

The classical sub-language consists of variables, **let**-expressions, unit, pairs, booleans, and conditionals. Quantum data is modelled using the constructs $\kappa * t$, $\overrightarrow{0}$, and $t + u$. The term $\kappa * t$, where κ is a complex number, associates the *probability amplitude* κ with the term t. It is convenient to have a special constant $\overrightarrow{0}$ for terms with probability amplitude zero. The term $t + u$ is a quantum *superposition* of t and u. Quantum superpositions are first-class values: when used as the first subexpression of a conditional, they turn the conditional into a *quantum control* construct. For example, $\mathbf{if}^{\circ}\ (\mathit{true} + \mathit{false})\ \mathbf{then}\ t\ \mathbf{else}\ u$ evaluates both t and u and combines their results in a quantum superposition.

QML Overview

- The following three functions correspond to simple rotations on qubits:

$$\begin{aligned} qnot\ x &= \mathbf{if}^{\circ}\ x\ \mathbf{then}\ \mathit{false}\ \mathbf{else}\ \mathit{true} \\ had\ x &= \mathbf{if}^{\circ}\ x\ \mathbf{then}\ ((-1) * \mathit{true} + \mathit{false})\ \mathbf{else}\ (\mathit{true} + \mathit{false}) \\ z\ x &= \mathbf{if}^{\circ}\ x\ \mathbf{then}\ ((-1) * \mathit{true})\ \mathbf{else}\ \mathit{false} \end{aligned}$$

- The first is the quantum version of boolean negation; it behaves as usual when applied to classical values but it also applies to quantum data.

- Evaluating

$$qnot\ (\kappa\ *\ false\ +\ \iota\ *\ true)$$

swaps the probability amplitudes associated with $false$ and $true$.

- The second function represents the fundamental $Hadamard$ matrix, and the third represents the $Pauli - Z$ operator.

QML Overview

The function:

$$\begin{aligned} \mathit{cnot}\ c\ x = \mathbf{if}^{\circ}\ c \\ \mathbf{then}\ (\mathit{true},\ \mathit{qnot}\ x) \\ \mathbf{else}\ \ (\mathit{false}, x) \end{aligned}$$

is the conditional-not operation, which behaves as follows: if the control qubit c is *true* it negates the second qubit x; otherwise it leaves it unchanged. When the control qubit is in some superposition of *true* and *false*, the result is a superposition of the two pairs resulting from the evaluation of each branch of the conditional. For example, evaluating *cnot* (*false* + *true*) *false* produces the *entangled* pair (*false*, *false*) + (*true*, *true*).

- Handling the no-cloning property; defining a type system for reasoning about programs.

- QPL Refs:
  - 'An Algebra of Pure Quantum Programming', Altenkirch et al., ENTCS, 2007.
  - 'Foundations of Quantum Programming', Ying, Elsevier (MK), 2016.

QC/QP: What Next … Even More!

- URLs: quantamagazine.com? zoo? … others.
- Research group at FoE?
- QNLP?
  - Oxford, others.
- Q…<<techs>>???

> The Future:
> What happens next is all at your hands,
> and it is all up to
> YOU!

Discussion

Q & A

Next Lecture Appetizer!

- In next lecture (isA):
  - No next lecture! ☺
  - No necessary textbook reading. ☺
  - Prepare for the FINAL!!! 😐 😐 😐

Course Webpage

http://eng.staff.alexu.edu.eg/staff/moez/teaching/pqc-f19

- Where you can:
  - Download lecture slides (incl. exercises and homework).
  - Check links to other useful material.

Thank You