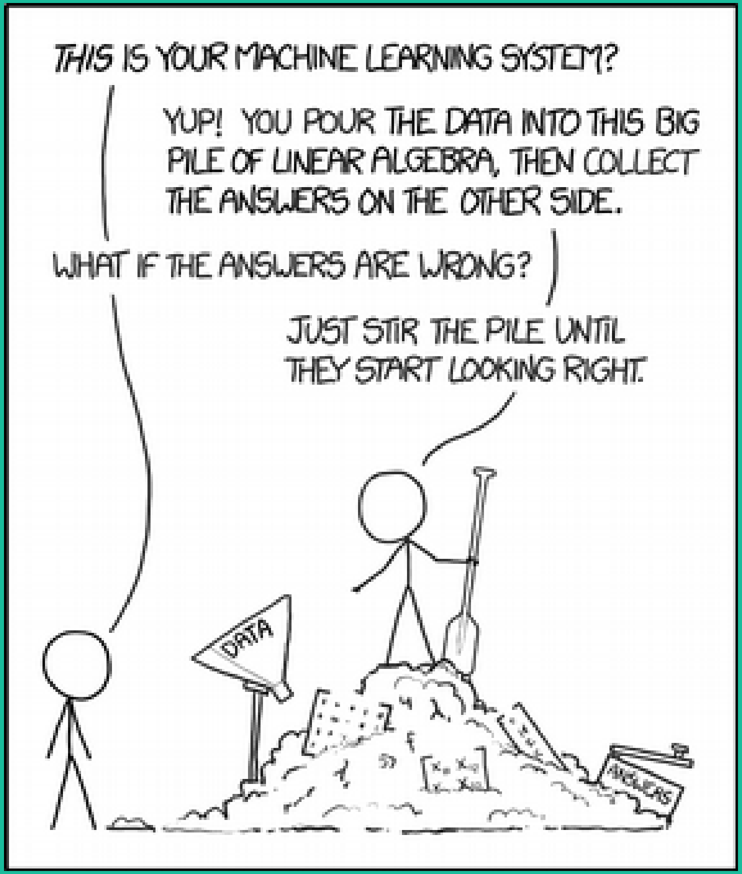


Quantum Machine Learning

Speaker: Chahan Kropf

Website: www.dmf.unicatt.it/kropf

19 Dec. 2019

Who are you?

Master

PostDoc

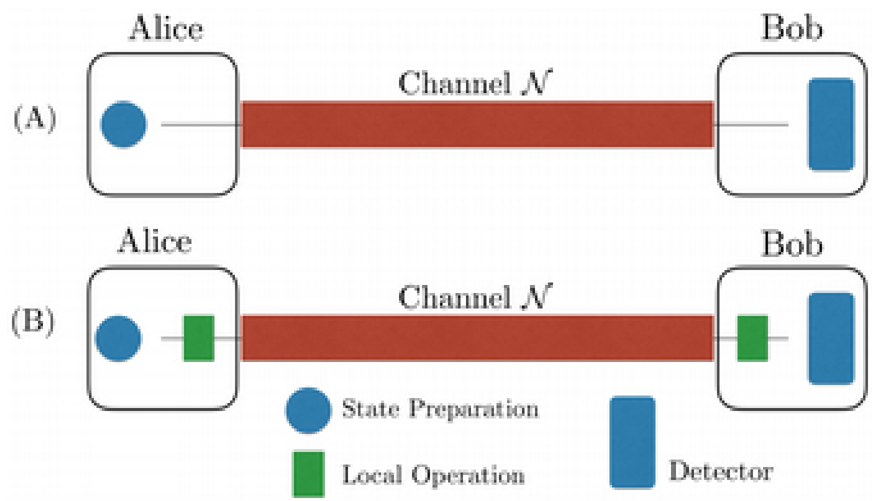
Bachelor

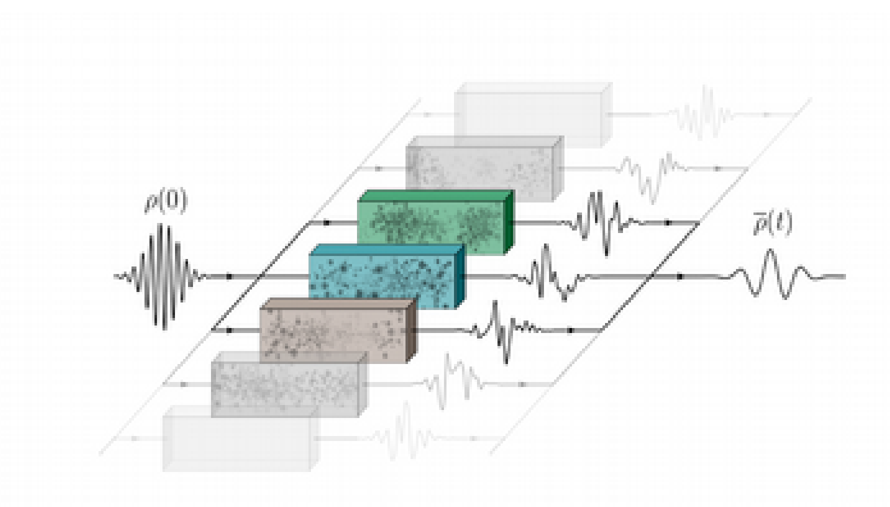
PhD

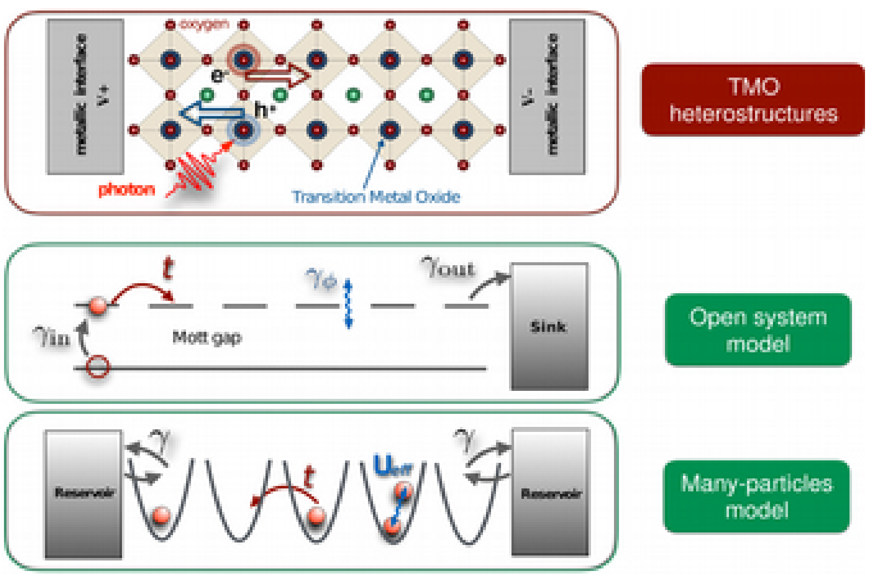
- **Neuchâtel – Heidelberg – Freiburg - Brescia**

What do you do ? Quantum physics!

- **Optimal quantum transport**
- **Dynamics of disordered quantum systems**
- **Quantum information and cryptography (channel coding of a measurement)**
- **Quantum machine learning**
- **Foundations of quantum mechanics**







Outline

- **Quantum machine learning**
  - What?
  - Why?
- **Quantum mechanics**
  - What is it?
  - Quantum computing
- **Example of a quantum machine learning algorithm**
  - Quantum square distance classifier
  - Python code for IBM Qcomputer

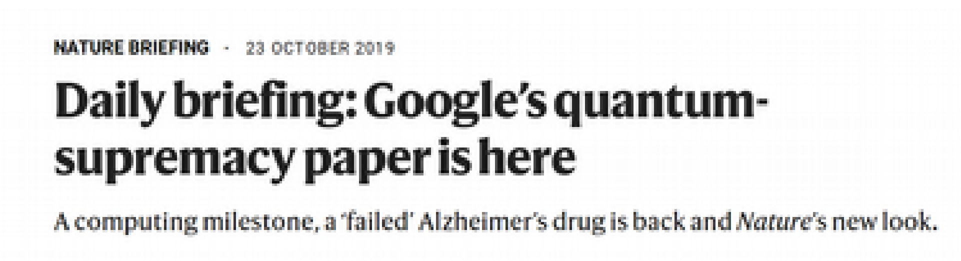
Quantum Machine Learning

- **Quantum communication and cryptography**

QUANTUM
INTERNET
ALLIANCE

Quantum Machine Learning

- **Quantum Computing**

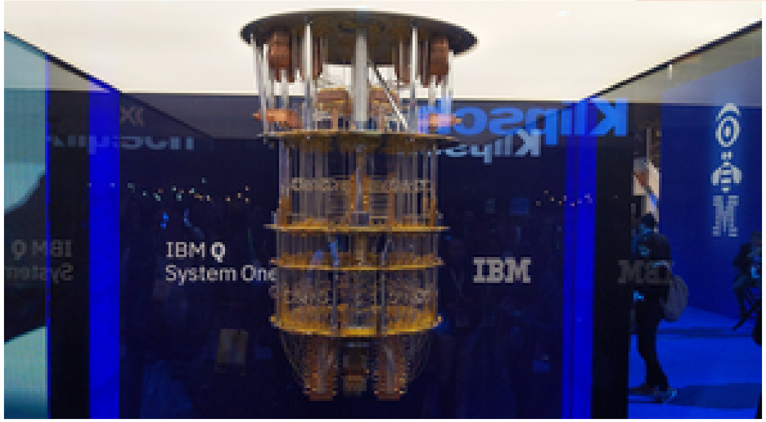

NATURE BRIEFING · 23 OCTOBER 2019

Daily briefing: Google's quantum-supremacy paper is here

A computing milestone, a 'failed' Alzheimer's drug is back and *Nature*'s new look.

4000 qubits

53 qubits

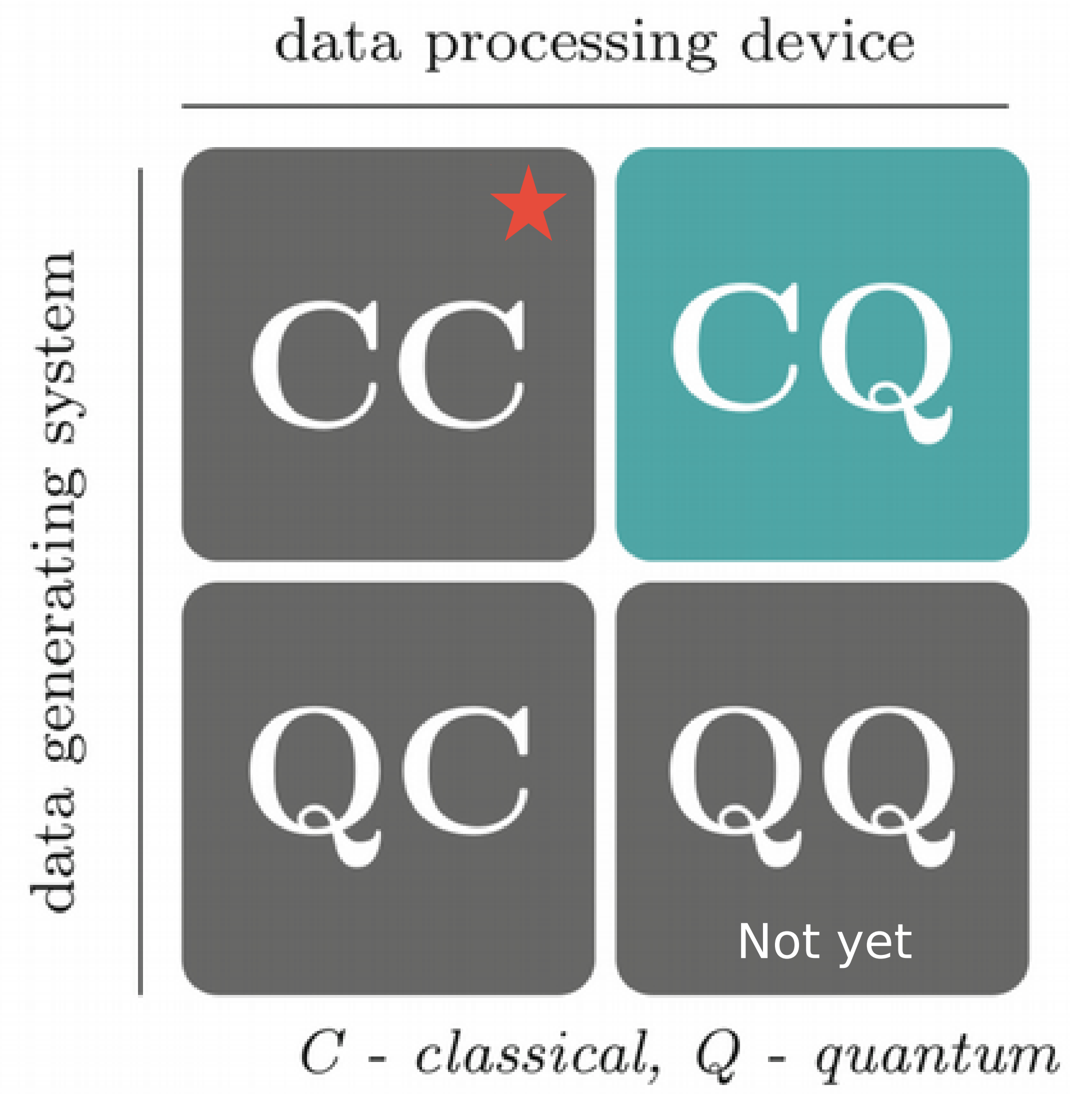


https://www.dwavesys.com

extremetech.com

Quantum Machine Learning

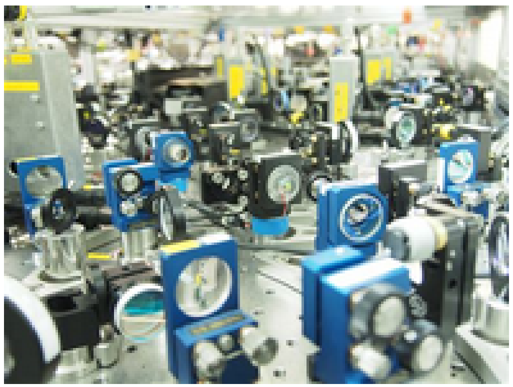
Quantum Machine Learning



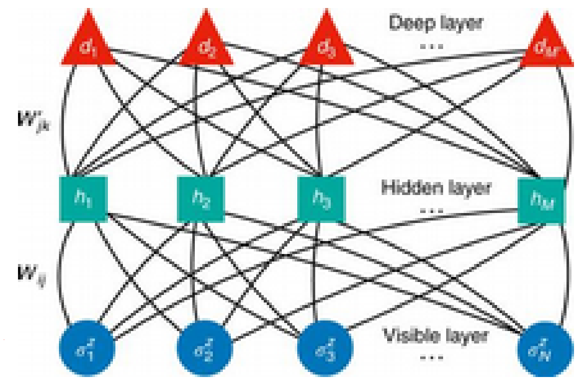
Thanks to Maria Schuld from Xanadu

Quantum – Classical (QC)

Experimental data



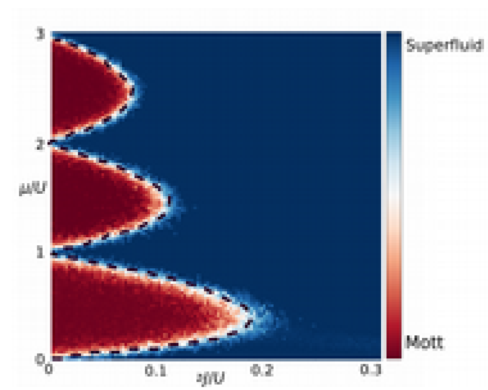
ML algorithm



Theoretical Model

$$i\hbar\frac{d}{dt}\psi(\vec{r},t) = \left[\frac{-\hbar^2}{2m}\nabla^2 + V(\vec{r},t)\right]\psi(\vec{r},t)$$

Result



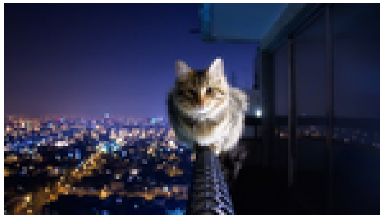
- **Quantum processor/experiment optimal design**
- **Material physics - phase transitions (high-temperature superconductivity)**

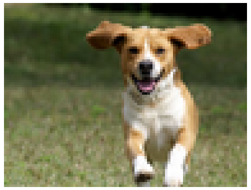
eurekalert.org

PhysRevB.97.134109

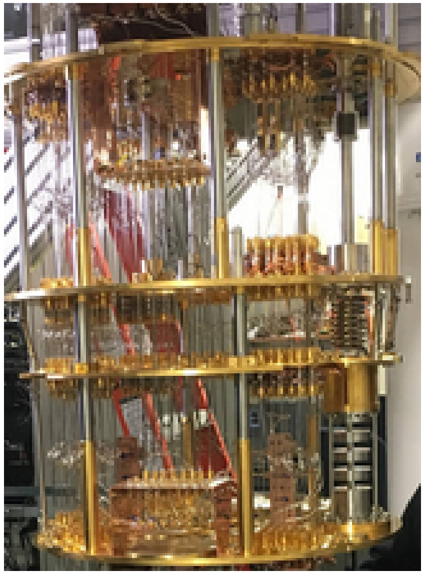
Classical – Quantum (CQ)

Normal data





QML algorithm



- **Optimization (annealing)**
- **QPCA, QSVM, QNN, Qclustering, QBM, ...**

Result

Cat

Or

Dog

coleandmarmalade.com animals-inthe-world.blogspot.com businessinsider.com.au top-tenpics.blogspot.com electronicsweekly.com

Quantum inspired Classical

- **Quantum information theory can help to understand classical machine learning and improve it.**
- **Example: learn correlations between two datasets and learn relevant features**

Learning relevant features for statistical inference

Cédric Bény[1]

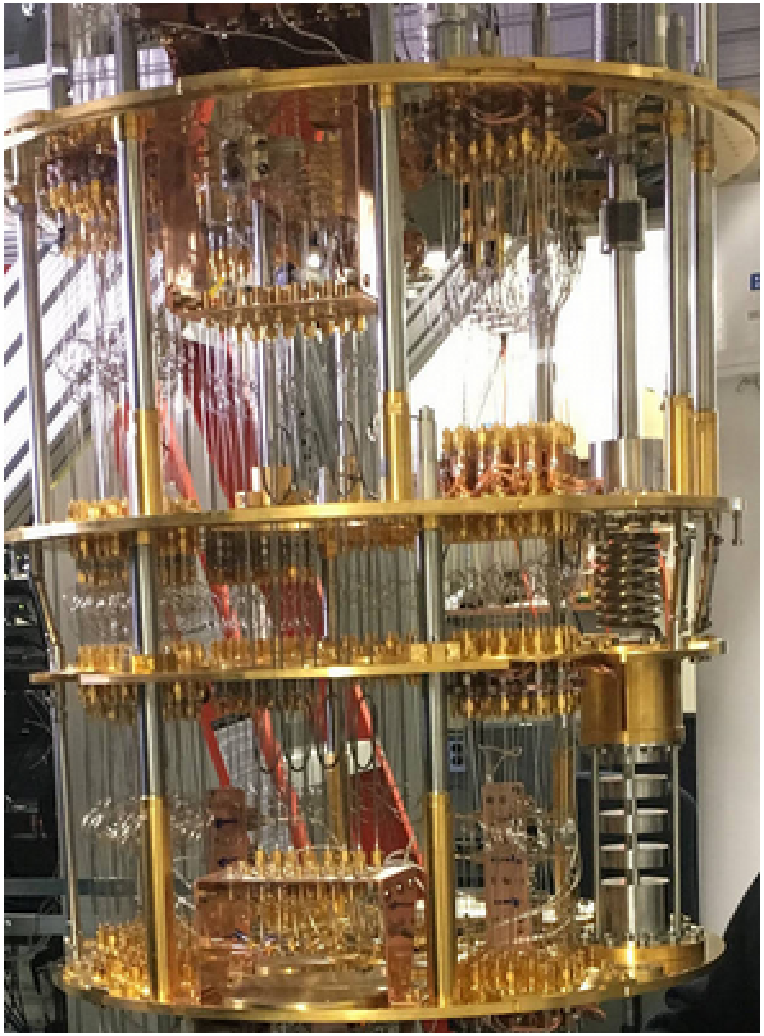
[1] *Department of Applied Mathematics, Hanyang University (ERICA), 55 Hanyangdaehak-ro, Ansan, Gyeonggi-do, 426-791, Korea.*

We introduce an algorithm that learns correlations between two datasets, in a way which can be used to infer one type of data given the other. The approach allows for the computation of expectation values over the inferred conditional distributions, such as Bayesian estimators and their standard deviations. This is done by learning feature maps which span hyperplanes in the spaces of probabilities for both types of data, optimized to optimally represent correlations. When applied to supervised learning, this yields a new objective function which automatically provides regularization and results in faster convergence. We propose that, in addition to many applications where two correlated variables appear naturally, this approach could also be used to identify dominant independent features of a single dataset in an unsupervised fashion: in this scenario, the second variables should be produced from the original data by adding noise in a manner which defines an appropriate information metric.

Why QML?

- Solve complex quantum problems with ML
- Better ML from quantum information theory
- Faster full ML algorithms on quantum computer
- Faster optimization routines
- Better sampling methods
- Entirely new type of AI and algorithms
- QML may be the current best reason to actually develop quantum computers

CQ



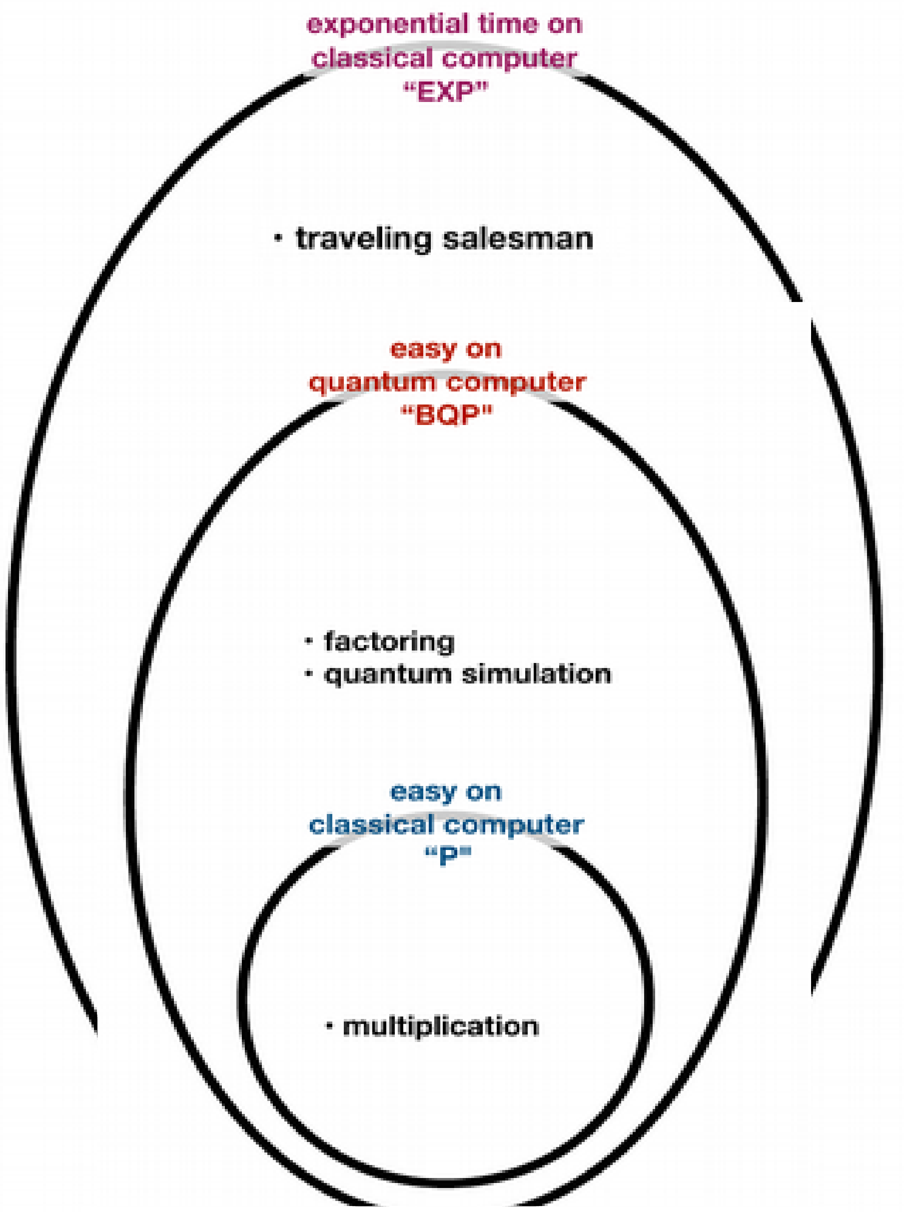
The power of quantum computers

- **Potential:**
  - Algorithm speed-up
  - Quantum sampling
  - Direct quantum analysis

- **Ingredients**
  - Coherence
  - Entanglement
  - Interference
  - Stochasticity
  - Linear dynamics
  - Non-linear measurement

- **Difficulties**
  - Coherence is destroyed by measurement
  - No cloning makes error correction hard
  - Classical: 10^-24 errors per gate, quantum 10^-2 errors per gate

The power of quantum computer



- **BQP contains P: a quantum computer can (quickly) do everything a classical computer can (quickly) do**
- **P does not contain BQP: a quantum computer can do some things quickly that a classical computer cannot do quickly**
- **EXP contains BQP : a classical computer can eventually do everything a quantum computer can do**
- **BQP does not contain EXP: a quantum computer does not give generic exponential speed-up**

From Adam Brown, HEP-AI

Timeline of quantum algorithms

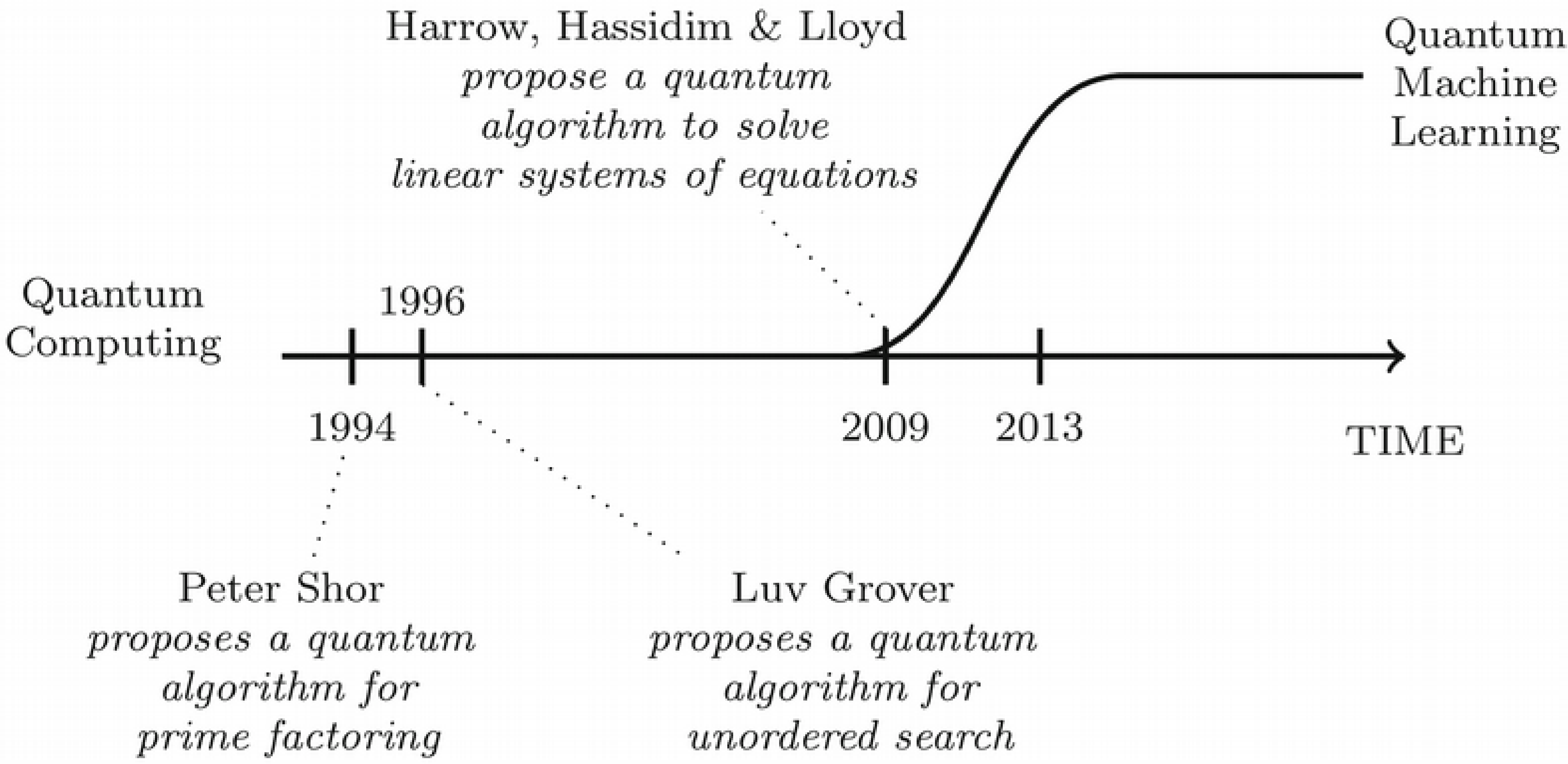
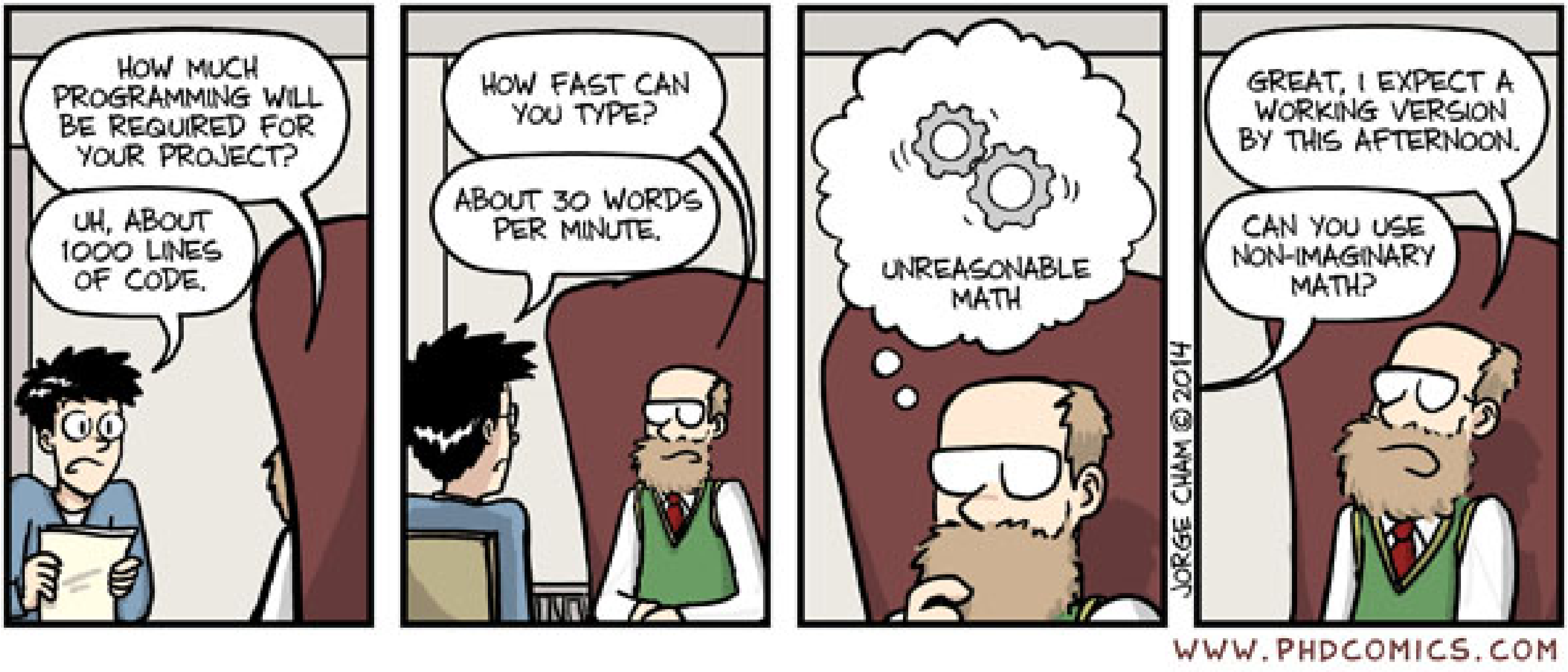


Fig. 3.3 Timeline of quantum computing and quantum machine learning

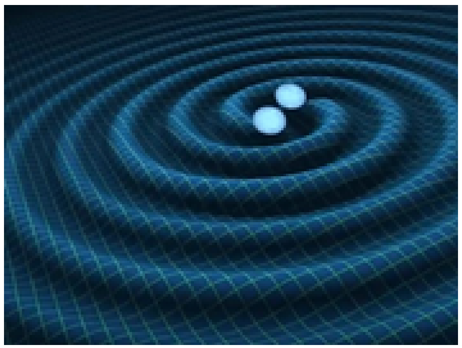
List of QML algorithms

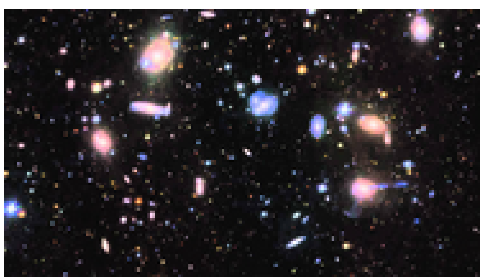
- qSearch [Grover 1996]
- qPrincipal Component Analysis [Lloyd, Mohseni, Rebentrost 2013]
- qSupport Vector Machine [Rebentrost, Mohseni, Lloyd, 2013]
- qBayesian Network [Tucci 2012, Moreira, Wichert 2018]
- qHidden Markov Model [Clark, Huang, Barlow, Beige 2014, Cholewa, Gawron, Glomb, Kurzyk 2015, Srinivasan, Gordon, Boots 2017]
- qMatrix Operations, qBLAS [Harrow, Hassidim, Lloyd 2009, Le Gall 2012, Zhang, Zhang, Xue 2018, Shao 2018]
- qNeural Network [Schuld, Sinayskiy, Petruccione 2014, Neukart 2013, Ricks, Ventura 2004, Dorozhinsky, Pavlovsky 2018, Farhi, Neven 2018]
- qAnnealing [Behrman, Steck, Moustafa 2016, Li, Felice, Rohs, Lidar 2018]

www.phdcomics.com

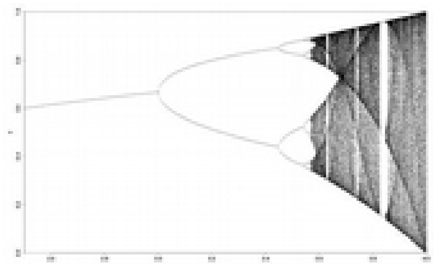


But what is quantum?

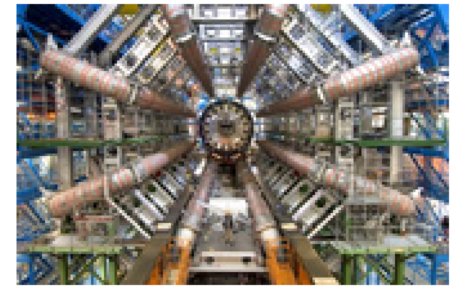


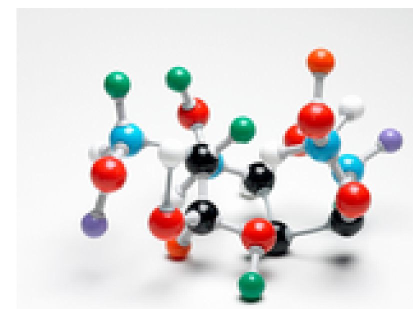


- **General relativity**
  - Curved space-time
  - Time dilation



- **Classical physics**
  - Newton
  - Chaos





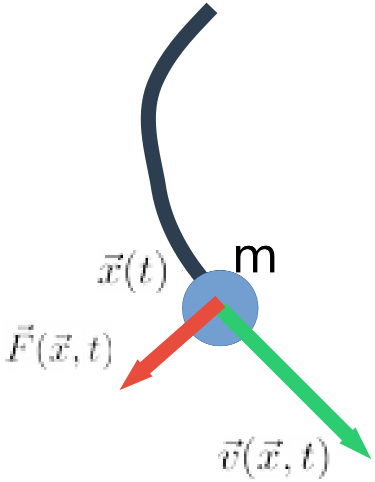
- **Quantum mechanics**
  - Non-locality
  - Tunneling

Classical mechanics

- **‘points’ with masses and velocity**
- **Forces acting on ‘points’**
- **Newton’s law of motion**

$$\vec{x}(t) \qquad \vec{v}(\vec{x},t)$$

$$\vec{F} = m\vec{a}$$



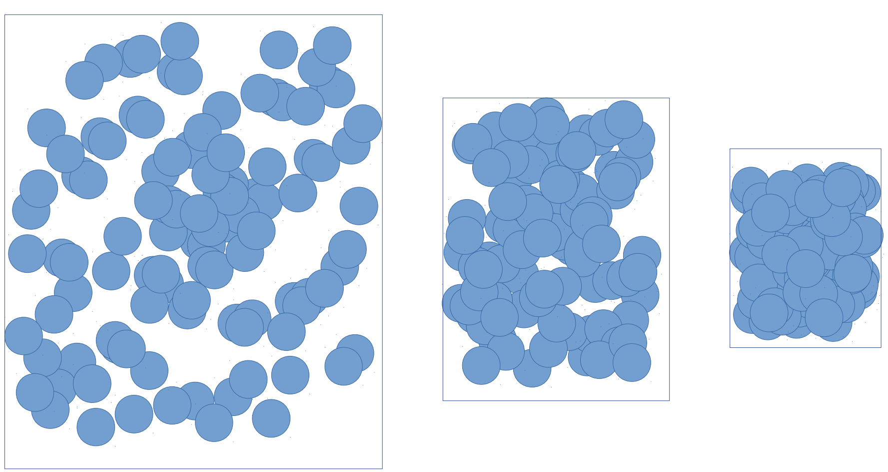
Statistical physics

- **Probability distribution of points**
- **Liouville equation for the distribution W (equivalent of Newton)**

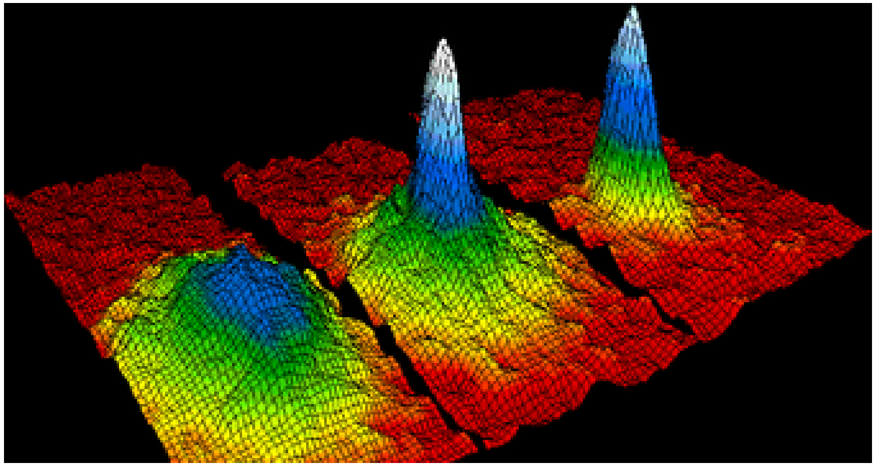
$$W(\vec{q}, \vec{p}, t)$$

$$\frac{\partial}{\partial t} W(\vec{q}, \vec{p}, t) = \{W(\vec{q}, \vec{p}, t), H(\vec{q}, t)\}$$

Particles moving around



Density probability distribution



$$W(\vec{q}, \vec{p}, t)$$

thoughtco.com

Quantum mechanics

- **Wavefunctions (complex object!)**

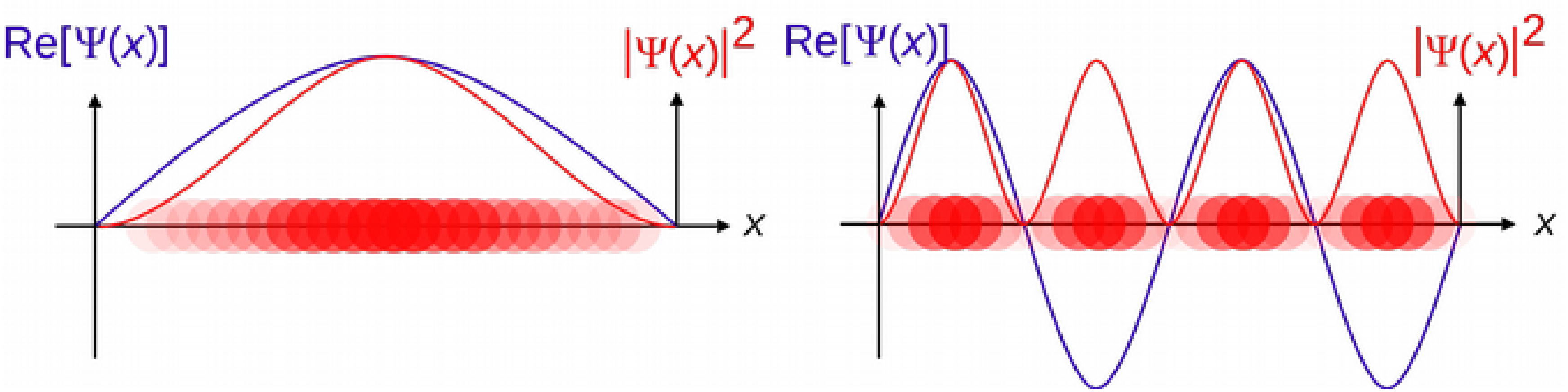
$$\psi(\vec{x}, t) \in \mathbb{C}$$

- **Schrödinger equation (linear!)**

$$i\hbar \frac{d}{dt}\psi(\vec{x}, t) = [-\frac{i\hbar}{2m}\nabla^2 + V(\vec{x})]\psi(\vec{x}, t)$$

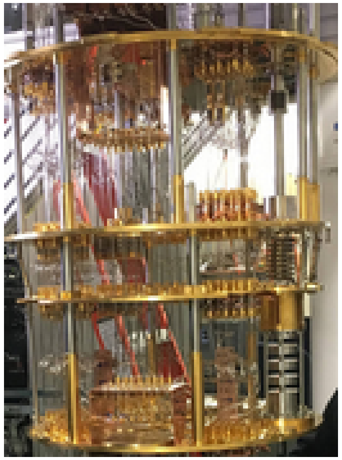
- **Outcome: probabilities**

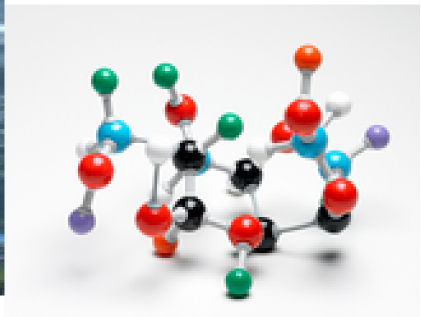
$$p(\vec{x}, t) = |\psi(\vec{x}, t)|^2 = \psi(\vec{x}, t)\psi^*(\vec{x}, t)$$



When does quantum start?

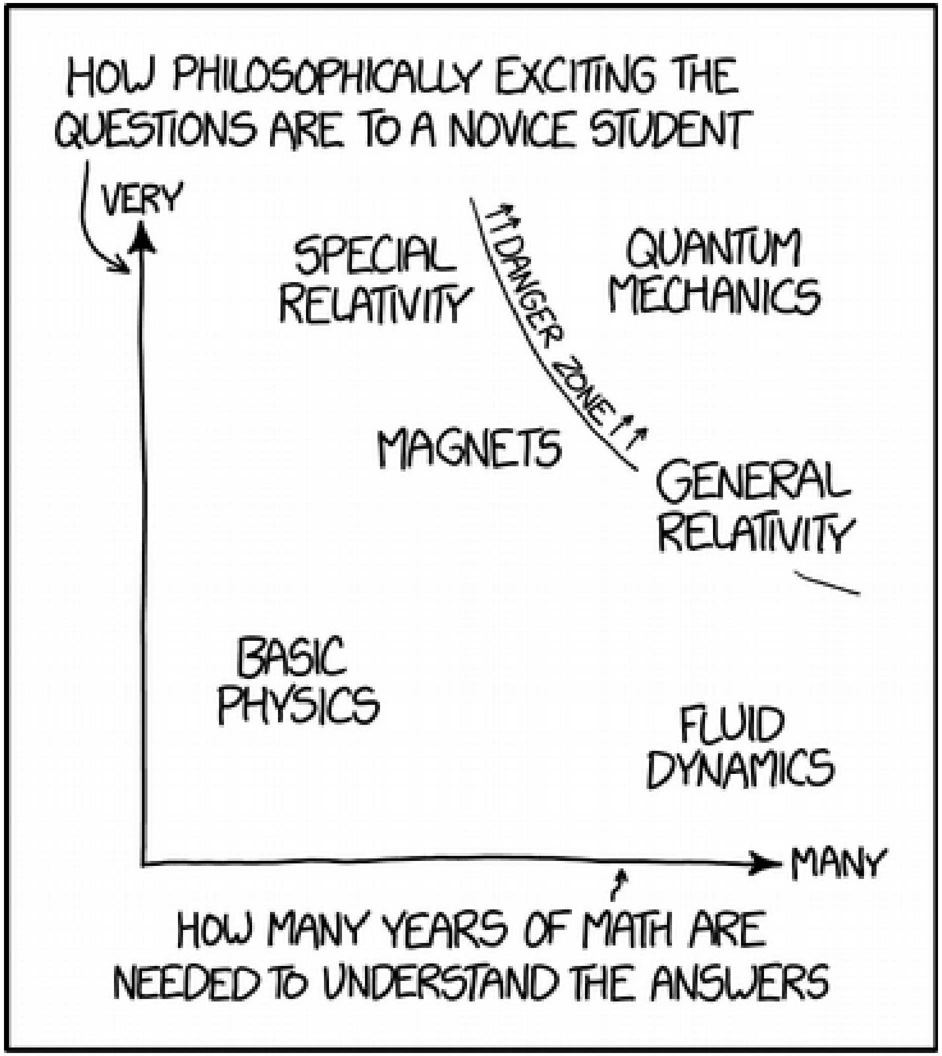
- **Where does the beach start/end?**





- **Ultra-cold**
- **Very-high Vacuum**
- **Ultra-short**

www.xkcd.com



WHY SO MANY PEOPLE HAVE WEIRD IDEAS ABOUT QUANTUM MECHANICS

Qubit

- **Binary equivalent of bit**

$$0 \longrightarrow |0\rangle \qquad 1 \longrightarrow |1\rangle \qquad |0\rangle, |1\rangle \in \mathcal{H}$$

- **All superposition also exist**

$$|q\rangle = \alpha |0\rangle + \beta |1\rangle \qquad \alpha, \beta \in \mathbb{C} \qquad \alpha^2 + \beta^2 = 1$$

- **Probability to obtain 0 or 1**

$$p(0) = \alpha^2 \qquad p(1) = \beta^2$$

- **After measurement**

$$|q\rangle \to |0\rangle \qquad |q\rangle \to |1\rangle$$

Qubits

- **Two qubits:**

$$|\psi\rangle = \alpha_1 |00\rangle + \alpha_2 |01\rangle + \alpha_3 |10\rangle + \alpha_4 |11\rangle$$

- **Many qubits:**

$$|\psi\rangle = \alpha_1 |0\ldots00\rangle + \alpha_2 |0\ldots01\rangle + \ldots + \alpha_{2^n} |1\ldots11\rangle$$

$$\Downarrow$$

$$n \text{ qubits} = 2^n \text{ complex amplitudes } \alpha_i$$

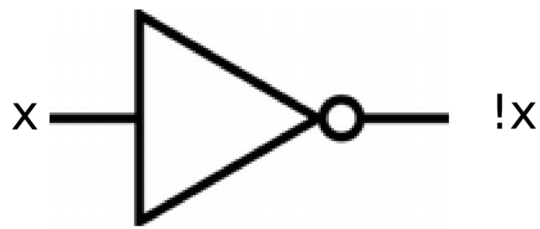
- **Superposition → Quantum parallelism**

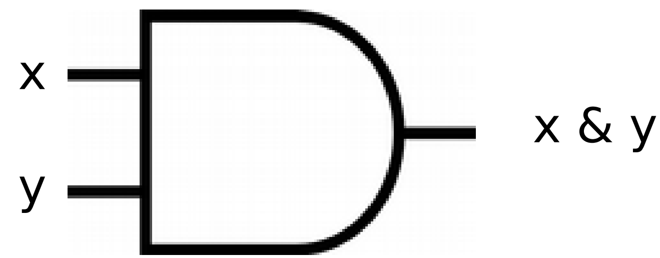
- **Normalization, but otherwise free → entanglement (non-classical correlations)**

$$\sum_{i=1}^{n} |\alpha_i|^2 = 1$$

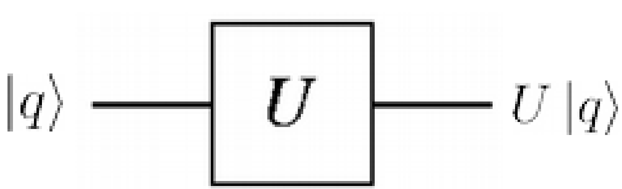
Universal quantum computer – universal gate algebra

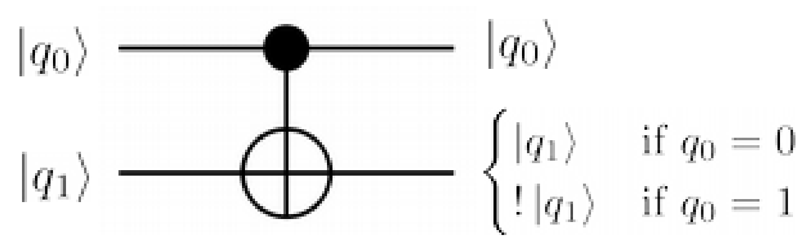
- **Not**
- **And (2 bits operation)**





- **Single qubit operation (unitary)**
- **2 qubits gate CNOT (unitary)**





Convenient representation

- **Qubit as complex unit vector**

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \in \mathbb{C}^2 \qquad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix} \in \mathbb{C}^2 \qquad |\psi\rangle = \frac{1}{\sqrt{2}}(1\,|0\rangle + i\,|1\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\i\end{pmatrix}$$

- **Unitary Gate as Hermitian matrix**

  - **Not** $X = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}$
  - **Phase not** $Y = \begin{pmatrix}0 & -i\\i & 0\end{pmatrix}$

$$X\,|0\rangle = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix} = |1\rangle$$
$$X\,|1\rangle = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix} = |0\rangle$$

$$Y\,|0\rangle = \begin{pmatrix}0 & -i\\i & 0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\i\end{pmatrix} = i\,|1\rangle$$
$$Y\,|1\rangle = \begin{pmatrix}0 & -i\\i & 0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}-i\\0\end{pmatrix} = -i\,|0\rangle$$

Hadamard Gate

- **One Qubit**

$$H = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad H\left|0\right\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \left|0\right\rangle \quad H\left|1\right\rangle = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix} = -\left|1\right\rangle$$

- **Two qubits**

$$H_2 = \frac{1}{\sqrt{2}} H \otimes \mathbb{I}_{2\times 2} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \qquad \left|\psi\right\rangle = \alpha_1\left|00\right\rangle + \alpha_2\left|01\right\rangle + \alpha_3\left|10\right\rangle \alpha_4\left|11\right\rangle = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix}$$

$$H\left|\psi\right\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} \alpha_1 + \alpha_3 \\ \alpha_2 + \alpha_4 \\ \alpha_1 - \alpha_3 \\ \alpha_2 - \alpha_4 \end{pmatrix}$$
$$= (\alpha_1 + \alpha_2)\left|00\right\rangle + (\alpha_2 + \alpha_4)\left|01\right\rangle + (\alpha_1 - \alpha_3)\left|10\right\rangle + (\alpha_2 - \alpha_4)\left|11\right\rangle$$

Hadamard Gate

- **n qubits: apply Hadamard to first one only**

$$|\psi\rangle = \alpha_1 |0\ldots00\rangle + \alpha_2 |0\ldots01\rangle + \ldots + \alpha_N |1\ldots11\rangle \qquad N = 2^n$$

$$H = \frac{1}{\sqrt{2}} H_{2\times2} \otimes \mathbb{I}_{2\times2} \ldots \mathbb{I}_{2\times2} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & \ldots & 1 & 0 & \ldots \\ 0 & 1 & \ldots & 0 & 1 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & 0 & \ldots & -1 & 0 & \ldots \\ 0 & 1 & \ldots & 0 & -1 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbb{I} & \mathbb{I} \\ \mathbb{I} & -\mathbb{I} \end{pmatrix} \qquad \mathbb{I} : \frac{N}{2} \times \frac{N}{2}$$

$$\begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_{\frac{N}{2}} \\ \alpha_{\frac{N}{2}+1} \\ \vdots \\ \alpha_N \end{pmatrix} \rightarrow \frac{1}{\sqrt{2}} \begin{pmatrix} \alpha_1 + \alpha_{\frac{N}{2}+1} \\ \vdots \\ \alpha_{\frac{N}{2}} + \alpha_N \\ \alpha_1 - \alpha_{\frac{N}{2}+1} \\ \vdots \\ \alpha_{\frac{N}{2}} - \alpha_N \end{pmatrix}$$

Measurement vs Evolution (gate)

- **Measure first qubit**

$$|\psi\rangle = \alpha_1 |00\rangle + \alpha_2 |01\rangle + \alpha_3 |10\rangle \alpha_4 |11\rangle$$

$$p(q_0 = 0) = \alpha_1^2 + \alpha_2^2 \qquad p(q_0 = 1) = \alpha_3^2 + \alpha_4^2$$

$$|\psi\rangle = \frac{1}{\alpha_1^2 + \alpha_2^2}(\alpha_1 |00\rangle + \alpha_2 |01\rangle) \qquad |\psi\rangle = \frac{1}{\alpha_3^2 + \alpha_4^2}(\alpha_3 |10\rangle + \alpha_4 |11\rangle)$$

$$\alpha_3 = 0, \alpha_4 = 0 \qquad \alpha_1 = 0, \alpha_2 = 0$$

Non-Linear!

- **Gate (quantum evolution): matrix multiplication**

$$X|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$$

Linear!

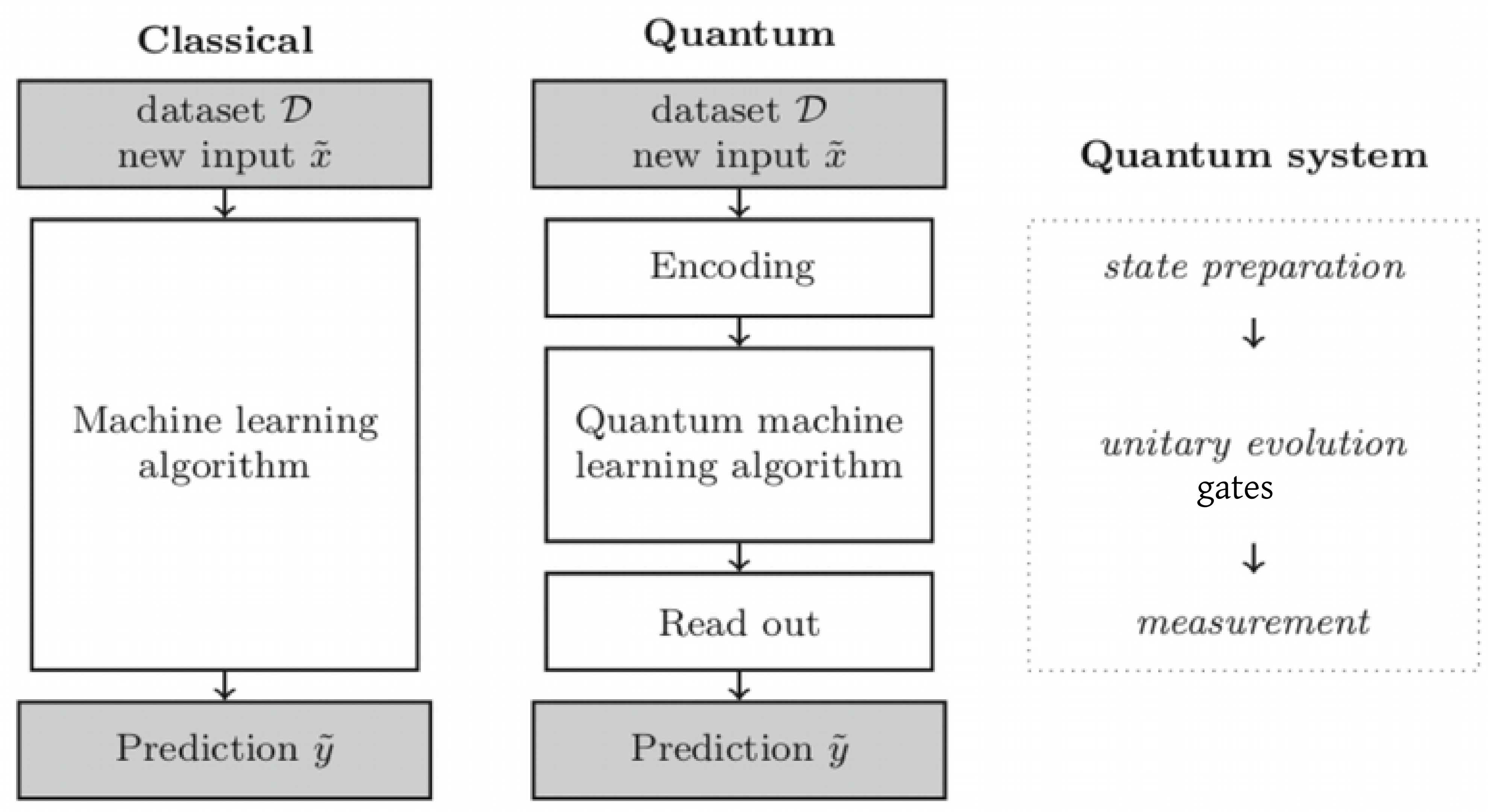
Key ingredients

- **Complex wavefunction**
  - Interferences
  - Entanglement
  - Superposition → large space of possibilities
- **Linear evolution equation**
- **Measurements**
  - Destroys superpositions
  - Only probabilistic predictions
- **Constraints**
  - No cloning theorem
  - One Qubit = One bit of information

www.smbc-comics.com

It was surprisingly easy to get $100 million from NASA.

QML design


Classical
dataset $\mathcal{D}$
new input $\tilde{x}$
Machine learning algorithm
Prediction $\tilde{y}$
Quantum
dataset $\mathcal{D}$
new input $\tilde{x}$
Encoding
Quantum machine learning algorithm
Read out
Prediction $\tilde{y}$
Quantum system
state preparation
unitary evolution
gates
measurement

Quantum (ML) algorithm - example

- **Square-distance classifier**
- **Titanic dataset**

historic-uk.com

Thanks to Maria Schuld from Xanadu

Quantum (ML) algorithm - example

- **Square-distance classifier**
- **Titanic dataset**

	raw data		
	price	cabin	survival
Passenger 1	8,500	0910	1 (yes)
Passenger 2	1,200	2105	0 (no)
Passenger 3	7,800	1121	?

Quantum (ML) algorithm - example

- **Square-distance classifier**
- **Titanic dataset**

	raw data		preprocessed data		
	price	cabin	price	cabin	survival
Passenger 1	8,500	0910	0.85	0.36	1 (yes)
Passenger 2	1,200	2105	0.12	0.84	0 (no)
Passenger 3	7,800	1121	0.78	0.45	?

$\vec{x}$ $\qquad$ y

Thanks to Maria Schuld from Xanadu

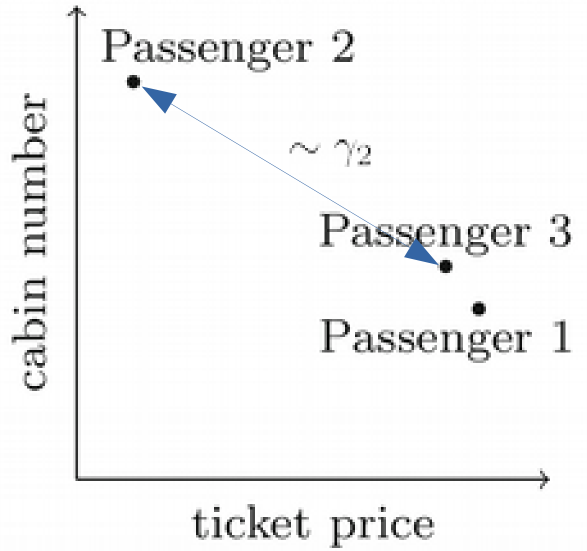
Classical square distance classifier

	raw data		preprocessed data		
	price	cabin	price	cabin	survival
Passenger 1	8,500	0910	0.85	0.36	1 (yes)
Passenger 2	1,200	2105	0.12	0.84	0 (no)
Passenger 3	7,800	1121	0.78	0.45	?

$\vec{x}_1 = (0.85, 0.36)$ $\quad y_1 = 1$

$\vec{x}_2 = (0.12, 0.84)$ $\quad y_2 = 0$

$\vec{\tilde{x}} = (0.78, 0.45)$ $\quad \tilde{y} = ?$



$$\gamma_m = 1 - \frac{1}{c}|\tilde{x} - x_m|^2$$

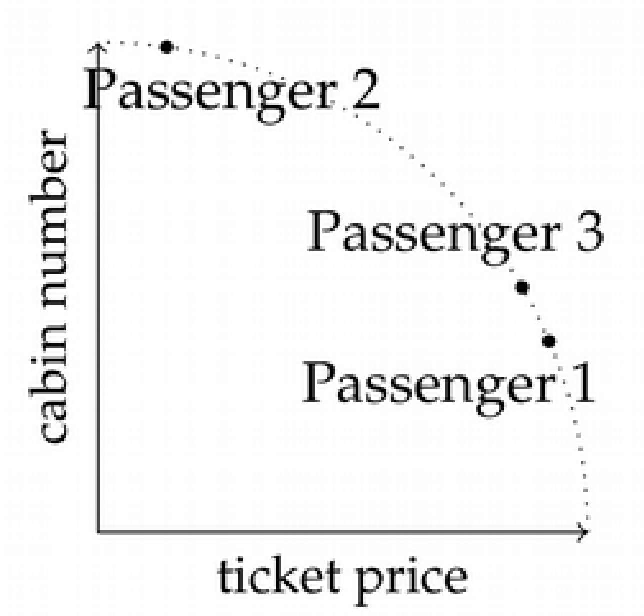
$$p(\tilde{y} = 1) = \frac{1}{\chi}\frac{1}{M_1}\sum_{m|y_m=1}^{M_1} \gamma_m$$

$$p(\tilde{y} = 0) = 1 - p(\tilde{y} = 1)$$

Quantum square-distance classifier

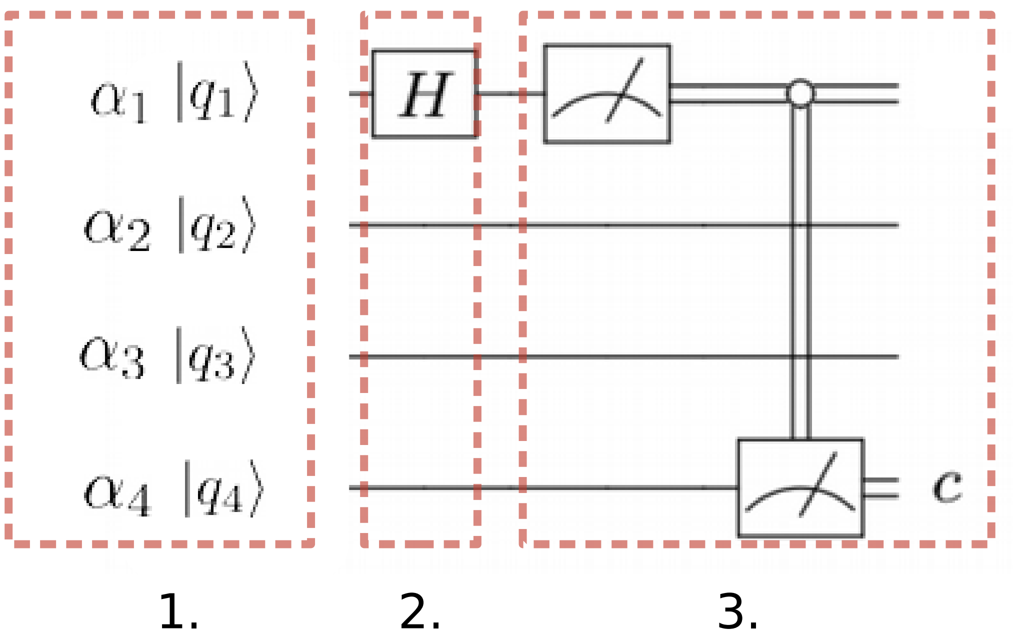
- **Data preprocessing normalized vectors (0-1)**

	price	room	survival
Passenger 1	0.921	0.390	yes (1)
Passenger 2	0.141	0.990	no (0)
Passenger 3	0.866	0.500	?



Algorithm

- **(1. Amplitude encoding on four qubits)**
- **2. Apply one Hadamard to the first qubit**
- **3. Measure first qubit**
  - If 0, measure qubit fourth qubit
  - If 1, discard
- **Repeat to obtain statistics**



Algorithm results

				Step 1
0	0	0	0	0
0	0	0	1	$\frac{1}{\sqrt{4}}0.921$
0	0	1	0	0
0	0	1	1	$\frac{1}{\sqrt{4}}0.390$
0	1	0	0	$\frac{1}{\sqrt{4}}0.141$
0	1	0	1	0
0	1	1	0	$\frac{1}{\sqrt{4}}0.990$
0	1	1	1	0
1	0	0	0	0
1	0	0	1	$\frac{1}{\sqrt{4}}0.866$
1	0	1	0	0
1	0	1	1	$\frac{1}{\sqrt{4}}0.500$
1	1	0	0	$\frac{1}{\sqrt{4}}0.866$
1	1	0	1	0
1	1	1	0	$\frac{1}{\sqrt{4}}0.500$
1	1	1	1	0

	price	room	survival
Passenger 1	0.921	0.390	yes (1)
Passenger 2	0.141	0.990	no (0)
Passenger 3	0.866	0.500	?

$$|\psi\rangle = \frac{1}{\sqrt{4}}\big(0.921\,|0001\rangle + 0.390\,|0011\rangle + 0.141\,|0100\rangle + 0.990\,|0110\rangle + 0.866\,|1001\rangle + 0.500\,|1011\rangle + 0.866\,|1100\rangle + 0.500\,|1110\rangle\big)$$

(1. Amplitude encoding) 2. Apply one Hadamard to the first qubit
3. Measure first qubit

Algorithm results

				Step 1	Step 2
0	0	0	0	0	0
0	0	0	1	$\frac{1}{\sqrt{4}}0.921$	$\frac{1}{\sqrt{4}}(0.921 + 0.866)$
0	0	1	0	0	0
0	0	1	1	$\frac{1}{\sqrt{4}}0.390$	$\frac{1}{\sqrt{4}}(0.390 + 0.500)$
0	1	0	0	$\frac{1}{\sqrt{4}}0.141$	$\frac{1}{\sqrt{4}}(0.141 + 0.866)$
0	1	0	1	0	0
0	1	1	0	$\frac{1}{\sqrt{4}}0.990$	$\frac{1}{\sqrt{4}}(0.990 + 0.500)$
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	1	$\frac{1}{\sqrt{4}}0.866$	$\frac{1}{\sqrt{4}}(0.921 - 0.866)$
1	0	1	0	0	0
1	0	1	1	$\frac{1}{\sqrt{4}}0.500$	$\frac{1}{\sqrt{4}}(0.390 - 0.500)$
1	1	0	0	$\frac{1}{\sqrt{4}}0.866$	$\frac{1}{\sqrt{4}}(0.141 - 0.866)$
1	1	0	1	0	0
1	1	1	0	$\frac{1}{\sqrt{4}}0.500$	$\frac{1}{\sqrt{4}}(0.990 - 0.500)$
1	1	1	1	0	0

(1. Amplitude encoding) 2. Apply one Hadamard to the first qubit
3. Measure first qubit

Algorithm results

				Step 1	Step 2	Step 3
0	0	0	0	0	0	0
0	0	0	1	$\frac{1}{\sqrt{4}}0.921$	$\frac{1}{\sqrt{4}}(0.921 + 0.866)$	$\frac{1}{\sqrt{4x}}(0.921 + 0.866)$
0	0	1	0	0	0	0
0	0	1	1	$\frac{1}{\sqrt{4}}0.390$	$\frac{1}{\sqrt{4}}(0.390 + 0.500)$	$\frac{1}{\sqrt{4x}}(0.390 + 0.500)$
0	1	0	0	$\frac{1}{\sqrt{4}}0.141$	$\frac{1}{\sqrt{4}}(0.141 + 0.866)$	$\frac{1}{\sqrt{4x}}(0.141 + 0.866)$
0	1	0	1	0	0	0
0	1	1	0	$\frac{1}{\sqrt{4}}0.990$	$\frac{1}{\sqrt{4}}(0.990 + 0.500)$	$\frac{1}{\sqrt{4x}}(0.990 + 0.500)$
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	0	1	$\frac{1}{\sqrt{4}}0.866$	$\frac{1}{\sqrt{4}}(0.921 - 0.866)$	0
1	0	1	0	0	0	0
1	0	1	1	$\frac{1}{\sqrt{4}}0.500$	$\frac{1}{\sqrt{4}}(0.390 - 0.500)$	0
1	1	0	0	$\frac{1}{\sqrt{4}}0.866$	$\frac{1}{\sqrt{4}}(0.141 - 0.866)$	0
1	1	0	1	0	0	0
1	1	1	0	$\frac{1}{\sqrt{4}}0.500$	$\frac{1}{\sqrt{4}}(0.990 - 0.500)$	0
1	1	1	1	0	0	0

(1. Amplitude encoding) 2. Apply one Hadamard to the first qubit
3. Measure first qubit

Result

				Step 1	Step 2	Step 3
0	0	0	0	0	0	0
0	0	0	1	$\frac{1}{\sqrt{4}}0.921$	$\frac{1}{\sqrt{4}}(0.921 + 0.866)$	$\frac{1}{\sqrt{4\chi}}(0.921 + 0.866)$
0	0	1	0	0	0	0
0	0	1	1	$\frac{1}{\sqrt{4}}0.390$	$\frac{1}{\sqrt{4}}(0.390 + 0.500)$	$\frac{1}{\sqrt{4\chi}}(0.390 + 0.500)$
0	1	0	0	$\frac{1}{\sqrt{4}}0.141$	$\frac{1}{\sqrt{4}}(0.141 + 0.866)$	$\frac{1}{\sqrt{4\chi}}(0.141 + 0.866)$
0	1	0	1	0	0	0
0	1	1	0	$\frac{1}{\sqrt{4}}0.990$	$\frac{1}{\sqrt{4}}(0.990 + 0.500)$	$\frac{1}{\sqrt{4\chi}}(0.990 + 0.500)$
0	1	1	1	0	0	0

Probability to measure the last qubit in state 0

$$p(q_4 = 0) = \frac{1}{4\chi}\left(|0.141 + 0.866|^2 + |0.990 + 0.500|^2\right) \approx 0.448$$

This is the same as our goal

$$p(q_4 = 0) = \frac{1}{\chi}\left(1 - \frac{1}{4}|0.141 - 0.866|^2 + |0.990 - 0.500|^2\right) = \frac{1}{\chi}\frac{1}{M_1}\sum_{m|y_m=0}^{M_0} \gamma_m \approx 0.448$$

Result

Passenger 3 survived!

Insights

- **Only one Hadamard and two simple measurements!**
- **Remains the same for any number of points**
- **But:**
  - Data encoding is a crucial step (often the runtime bottleneck)
  - Constraints on the preprocessing (unit length vectors)
  - Result is a probabilistic measurement
  - Not actually faster than classical one
- **Remark:**
  - Use of interferences
  - QM often inspired by particular variations of ML models which suite quantum computing

Python code

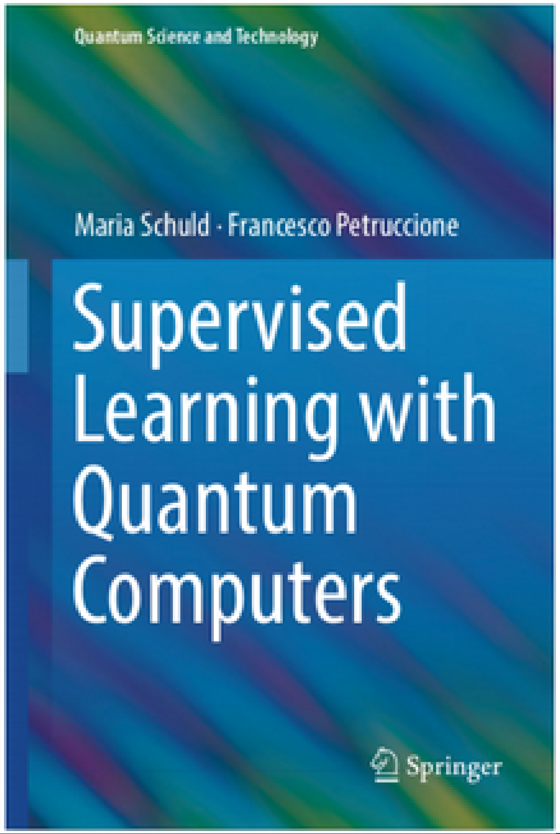
Code does not work as is. New Qiskit convention for connection to the IBM devices. See jupyter notebook.

```
import numpy as np
from qiskit import QuantumCircuit, ClassicalRegister,
QuantumRegister
from qiskit import execute
import math
import numpy
#
# State and circuit creation
#
# Create a 4 qubit quantum register
q = QuantumRegister(4)
# Create a classical register for measurements
c = ClassicalRegister(2)
# Create a quantum circuit with input q & output c
circ = QuantumCircuit(q,c)
# Initialize the register content
v = numpy.zeros(16)
v[ 1] = 0.920843009649141/2 # |0001>
v[ 3] = 0.389933522001265/2 # |0011>
v[ 4] = 0.141001339519088/2 # |0100>
v[ 6] = 0.990009405134023/2 # |0110>
v[ 9] = 0.866019052628739/2 # |1001>
v[11] = 0.500011000363013/2 # |1011>
v[12] = 0.866019052628739/2 # |1100>
v[14] = 0.500011000363013/2 # |1110>
circ.initialize(v, q)
# Add a H gate on qubit 3 (leftmost)
circ.h(q[3])
# Measure leftmost qubit
circ.measure(q[3], c[1])
# Measure rightmost qubit
circ.measure(q[0], c[0])
```

```
# Running on IBM Q
#
from qiskit import IBMQ
# Load credentials
IBMQ.load_accounts()
# Choose device
from qiskit.providers.ibmq import least_busy
large_enough_devices = IBMQ.backends(filters=lambda x:
x.configuration().n_qubits > 3 and not x.configuration().simulator)
backend = least_busy(large_enough_devices)
print("The best backend is " + backend.name())
# Execute the circuit
from qiskit.tools.monitor import job_monitor
# Number of shots to run the program
shots = 1024
# Maximum number of credits to spend on executions
max_credits = 3
job = execute(circ, backend=backend, shots=shots,
max_credits=max_credits)
job_monitor(job)
# Wait until the job has finished
result = job.result()
#
# Print the results
#
counts = result.get_counts(circ)
# Calculate probability of label 1
# q4=0 & q0=1 over q4=0 & (q0=0 or q0=1)
p1 = counts.get('01',0) / (counts.get('00',0)+counts.get('01'))
print("Probability of label 0 (1) is %.3f (%.3f)" % ((1-p1),p1))
```


https://github.com/michelbarbeau/Quantum-Machine-Learning/blob/master/Schuld_Petruccion_2018_sim.py

Conclusion



Thank You!