Correlations in time series (some preliminaries) :

Univariate time series :

$$C(\tau) = \langle (x(t) - \bar{x}) \ (x(t+\tau) - \bar{x}) \rangle$$

Bi-variate time series :

$$C_{x,y}(\tau) = \langle (x(t) - \bar{x}) \ (y(t) - \bar{y}) \rangle$$

Multivariate time series : $x_1(t), x_2(t), x_3(t) \ldots\ldots x_N(t)$

$$t = 1, 2, 3....T$$

Pair wise correlations : $\dfrac{N(N-1)}{2}$

Correlation matrix : $C = D^{\mathrm{T}} D$

- D is T x N data matrix, with each column representing a time series.

  Then, C is a square matrix of order N.

- Spectra of correlation matrix C.

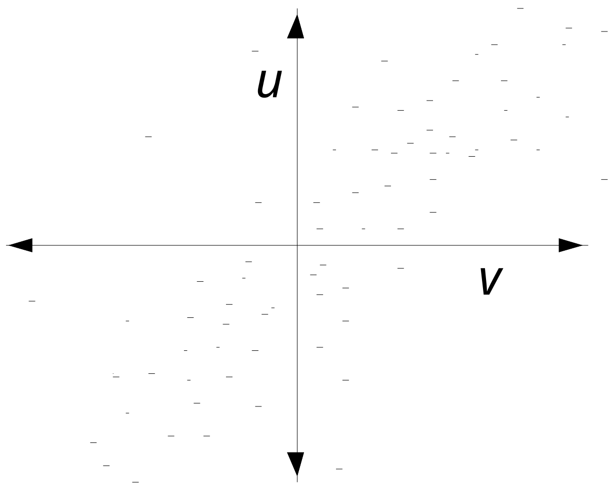
$$C \; u_i = \lambda_i \; u_i , \qquad i = 1, 2, \ldots N$$

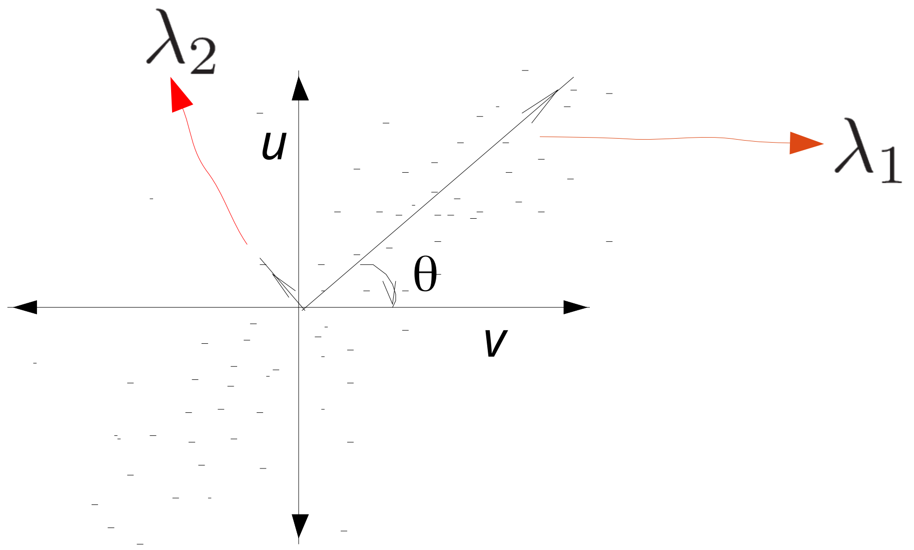
- Positive semi-definite eigenvalues : $\lambda_i \geq 0$

  Real, symmetric : $C = C^{\mathrm{T}}$

Bi-variate example : $u(t),\ v(t)$

$$\mathbf{S} = \begin{pmatrix} c_{uu} & c_{uv} \\ c_{vu} & c_{vv} \end{pmatrix}$$





$$\boxed{\lambda_1 \geq \lambda_2 \geq \lambda_3 \ldots} \boxed{\ldots \ldots \lambda_{N-1} \geq \lambda_N}$$

- Which of these modes represent random correlations ?

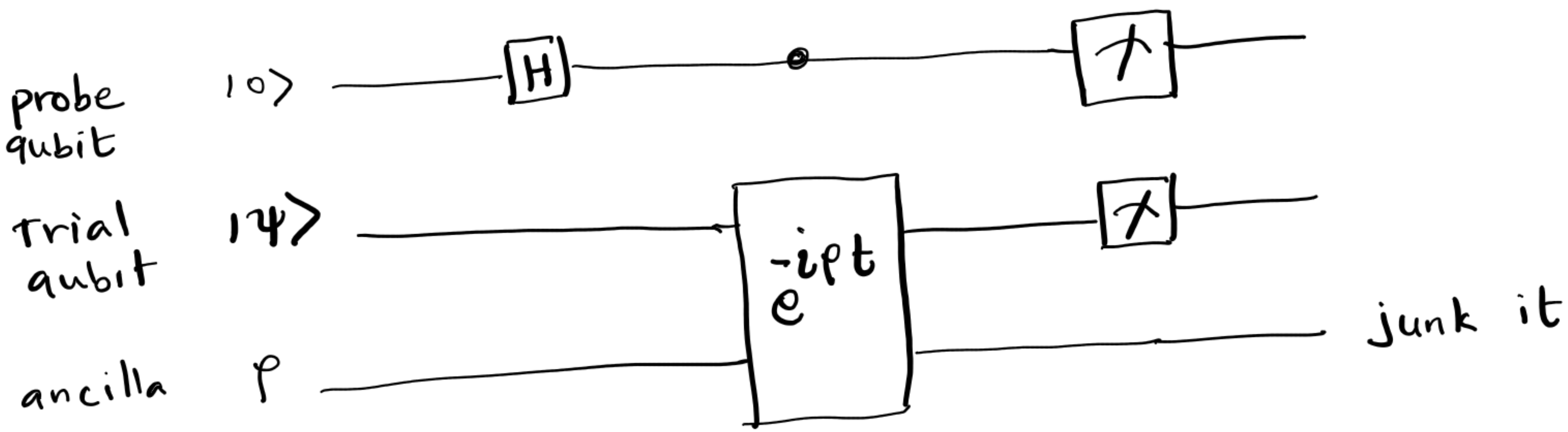
And which ones contain significant information ?

- Principal Component Analysis (PCA)

$$C\, u_i = \lambda_i\, u_i \qquad i = 1, 2, \cdots N$$

- Find all eigenvalues and eigenvectors?
- In practice, only the dominant eigenvalues are useful.

Top-level view of qPCA



Nature Physics 10, 631 (2014)

- $\rho \rightarrow$ prepare classical data as a quantum density matrix.

- Low-rank preferred.
- Assumption: Many copies of ρ available. Needed to perform DME.

- $C \rightarrow$ classical covariance matrix

$$\rho = \frac{C}{\mathrm{Tr}\, C}$$

- QRAM style encoding data $\sim O(\log N)$
  or a dedicated encoding technique $\sim O(n)$

- **Key idea:** Density matrix exponentialtion using a series of SWAP gates

$$\rho \longrightarrow e^{-i\rho t}$$

- $\rho = \sum_j \lambda_j |u_j\rangle\langle u_j|$ Spectral decomposition

$$\xi_t(\sigma) = e^{-i\rho\Delta t}\,\sigma\, e^{i\rho\Delta t} \simeq \sigma - i\,\Delta t\,[\rho,\sigma]$$

if $\sigma = |\psi\rangle\langle\psi|$ pure state projector

$$\xi_t(|\psi\rangle\langle\psi|) = e^{-i\rho t}|\psi\rangle\langle\psi|\, e^{i\rho t} = |\psi(t)\rangle\langle\psi(t)|$$

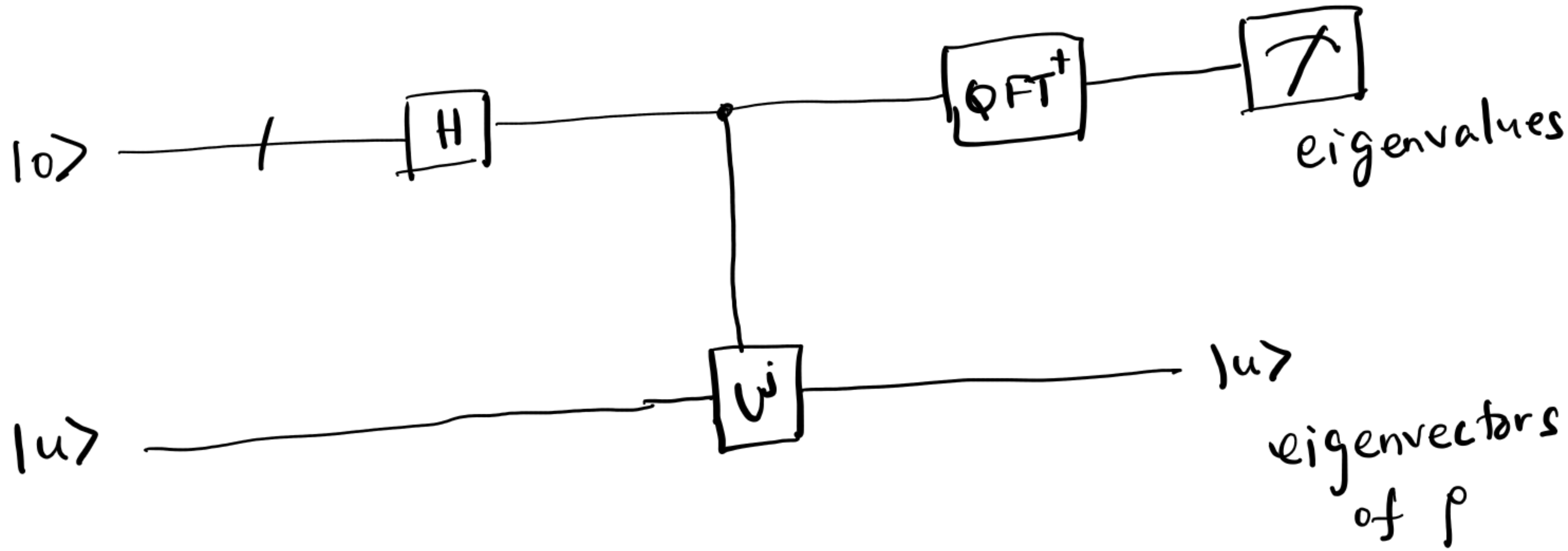
$$|\psi(t)\rangle = e^{-i\rho t}\,\psi(t)$$

- Apply quantum phase estimation

$$e^{-i\rho t}\,|u\rangle = e^{i2\pi\varphi}\,|u\rangle$$

determine φ.

- DME feeds directly into QPE



- Compare with classical

$$C_{d\times d} = X_{d\times n} X^T_{n\times d}$$

<table>
<tr><th>classical PCA</th><th>Quantum PCA</th></tr>
<tr><td>Generating covariance matrix $O(nd^2)$</td><td>Same as classical</td></tr>
<tr><td>Eigen decomposition $O(d^3)$</td><td>DME + QPE
$O(\log d)$</td></tr>
<tr><td>$= O(nd^2 + d^3)$</td><td>$\approx O(\log d)$ (assuming qRAM)</td></tr>
<tr><td>or $O(dn^2 + n^3)$</td><td>$\approx O(d + \log d)$ (otherwise)</td></tr>
</table>

assuming low-rank ρ

quantum Support Vector Machine

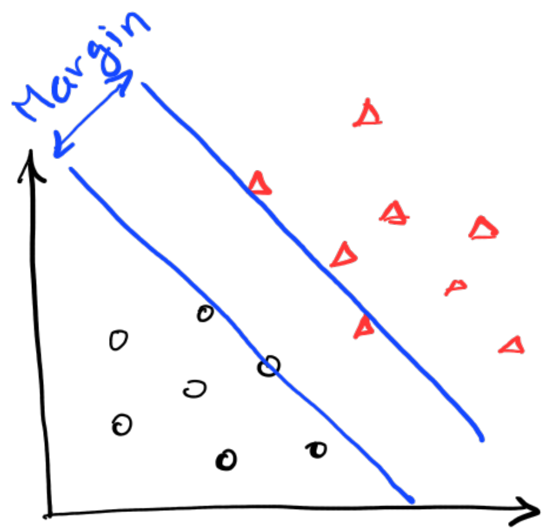
Phys. Rev. Lett. 113, 130503 (2014)

- Given M training data points

$$(\vec{x}_j\ ,\ y_j) \qquad j = 1, 2, 3 \text{ ----- } M$$

$\vec{x}_j$ → vectors with N components

y_j → labels ± 1



Linear SVM

- Recollect eqn for linear hyperplane

$$w^T x + b = 0$$

$w \rightarrow$ normal vector to hyperplane

$b \rightarrow$ dist from origin along the direction of w

- Hyperplanes are

$$\vec{\omega} \cdot \vec{x}_j + b \leq -1 \quad \text{for} \quad x_j \in -1 \text{ class}$$

$$\vec{\omega} \cdot \vec{x}_j + b \geq 1 \quad \text{for} \quad x_j \in 1 \text{ class}$$

- Here's the optimisation problem

- $$\underset{\omega, b}{\text{minimise}} \quad \frac{1}{2} \|\omega\|^2$$

subject to $\quad y_j \left(\omega^T x_i + b \right) \geq 1 \quad$ for $\quad i = 1, \dots M$

- classifier: $\quad f(x) = \omega^T x + b$
- Predicted label $\quad \hat{y} = +1 \quad$ if $\quad f(x) > 0$

$\qquad\qquad\qquad\qquad -1 \quad$ if $\quad f(x) < 0$

Dual formulation of SVM

- maximise objective function or Karush-Kuhn-Tucker multipliers

$$\vec{\alpha} = (\alpha_1, \alpha_2 \,---\, \alpha_M)^T$$

- $$L(\vec{\alpha}) = \sum_{j=1}^{M} y_j \alpha_j - \frac{1}{2} \sum_{j=1}^{M} \sum_{k=1}^{M} \alpha_j K_{jk} \alpha_k$$

Subject to constraint $\sum_{j=1}^{M} \alpha_j = 0$ and $y_j \alpha_j \geq 0$.

- Hyperplanes are: $$\vec{\omega} = \sum_{j=1}^{M} \alpha_j \vec{x}_j$$

$$b = y_j - \vec{\omega}_j \cdot \vec{x}_j$$

$$K_{jk} = k(\vec{x}_j, \vec{x}_k) = \underbrace{\vec{x}_j \cdot x_k}_{\text{defines kernel function}}$$

$\downarrow$

kernel matrix

Solving classically:

- Evaluate $\frac{M(M-1)}{2}$ dot products in kernel matrix $\sim O(N)$
- Then, find optimal α_j by quadratic programming $\sim O(M^3)$

$$\text{classical SVM}: \quad \frac{M(M-1)}{2} N + M^3$$

$$\approx O\left(M^2 (M+N)\right)$$

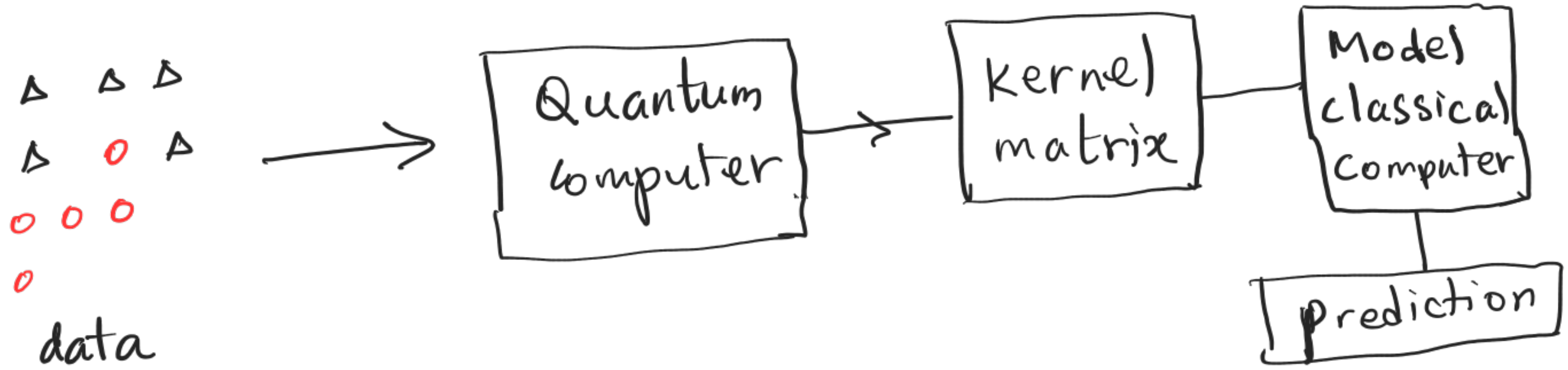
Two quantum approaches to SVM problem

a) Hybrid "implicit" approach

Nature Physics 567, 209 (2019)

Phys Rev Lett 122, 040504 (2019)

- Estimate kernel matrix using QC
- Training the model and classification done using classical optimiser and simulations



b) Full quantum solution

- **Key idea:** Convert the dual problem into least squares SVM (LS-SVM)

- introduce slack variables and replace inequality constraints by equality constraints

$$y_j(\vec{w}\cdot\vec{x}_j + b) \geq 1 \longmapsto (\vec{\omega}\cdot\vec{x}_j + b) = y_j - y_j e_j$$

- Minimise $\frac{1}{2}\|\omega\|^2 + \frac{\gamma}{2}\sum_j e_j^2$

Subject to $\vec{\omega}\cdot\vec{x} + b = y_j(1 - e_j)$

$\gamma \rightarrow$ value that determines relative weight of training error and SVM objective

- This can be reformulated as linear system

$$\begin{pmatrix} 0 & \vec{\mathbb{1}}^T \\ \vec{\mathbb{1}} & K + \gamma^{-1} I \end{pmatrix} \begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ \vec{y} \end{pmatrix}$$

$$\vec{\mathbb{1}} = (1 \; 1 \; 1 \; \ldots \; 1)^T$$

$$\vec{y} = (y_1 \; y_2 \; \ldots \; y_M)^T$$

$$F \begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ \vec{y} \end{pmatrix}$$

- This is solved using HHL algorithm

Phys Rev Lett 103, 150502 (2009)

Hybrid implicit approach

Only the Kernel matrix is to be estimated by quantum algorithm

- Can be done via QRAM $\rightarrow$ $O(\log MN)$
- Can be done via QRAM-like data loading $\rightarrow$ $O(M \log N)$

(diff. between loading entire $\vec{x}_j$ as a superposition state vs. loading individual x_j vectors)

- Optimiser is classical (QP) $\rightarrow$ $O(M^3)$.

classical SVM $\Big\}$ $O(M^2(N+M))$

Quantum Implicit SVM $O(M^2(\log N + M))$

b) Full quantum SVM

Employs HHL (phase estimation algorithm + some small manipulations)

$$O(\log MN)$$

classical SVM $\Big\}$ $O\left(M^2(N+M)\right)$

full quantum SVM $O(\log MN)$

Gives exponential advantage!