Quantum Machine Learning

CS682A

Aadil Hayat
17111001

Proposal

Proposed Papers

- ✔ Quantum Machine Learning https://arxiv.org/pdf/1611.09347.pdf

  J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe & S. Lloyd

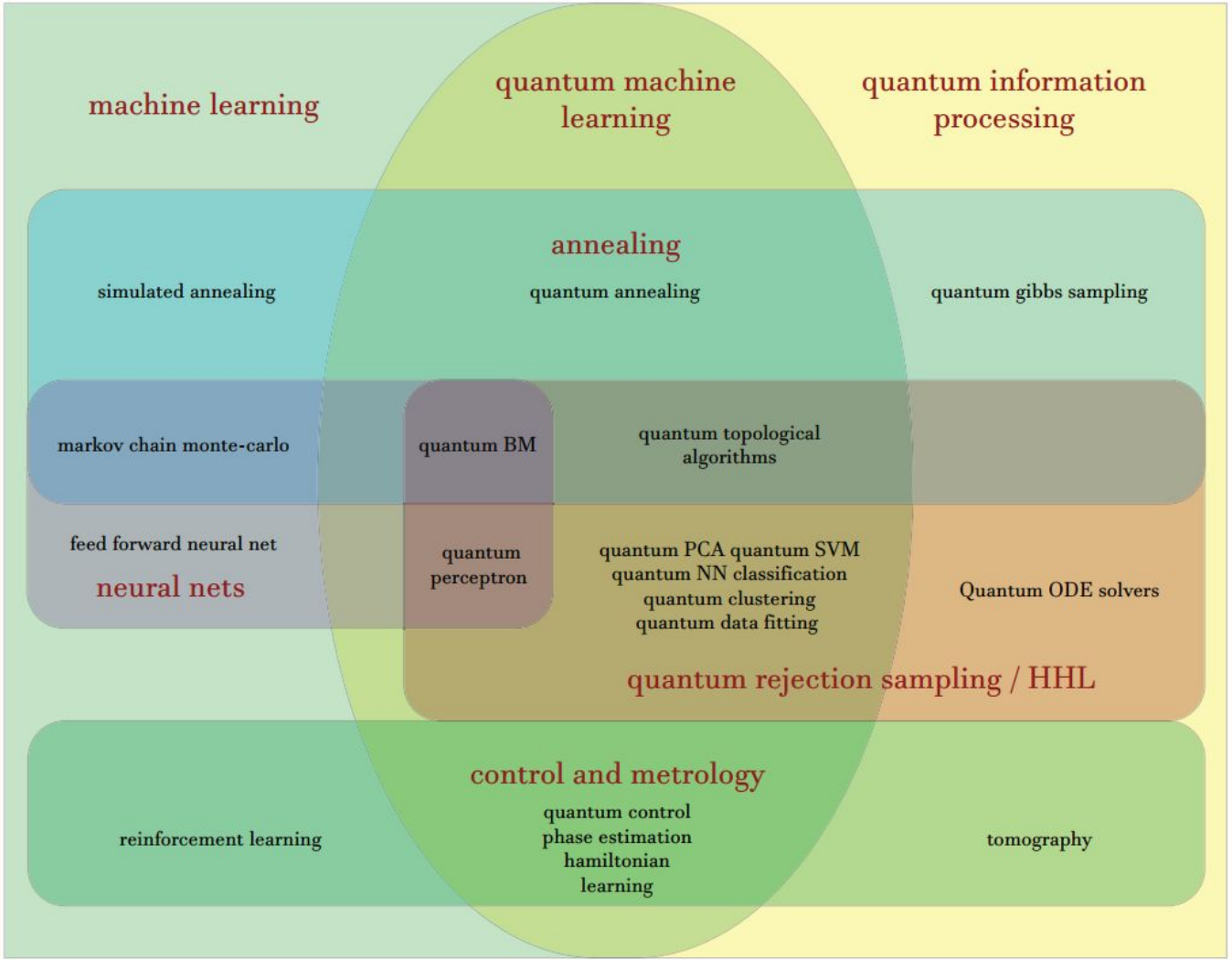
- ✔ Quantum Principal Component Analysis https://arxiv.org/pdf/1307.0401.pdf

  S. Lloyd, M. Mohseni & P. Rebentrost

- ● Quantum Support Vector Machines https://arxiv.org/pdf/1307.0471.pdf

  P. Rebentrost, M. Mohseni & S. Lloyd

Quantum Machine Learning


machine learning
quantum machine learning
quantum information processing
annealing
simulated annealing
quantum annealing
quantum gibbs sampling
markov chain monte-carlo
quantum BM
quantum topological algorithms
feed forward neural net
neural nets
quantum perceptron
quantum PCA quantum SVM
quantum NN classification
quantum clustering
quantum data fitting
Quantum ODE solvers
quantum rejection sampling / HHL
control and metrology
reinforcement learning
quantum control
phase estimation
hamiltonian learning
tomography

Quantum PCA

Principal Component Analysis

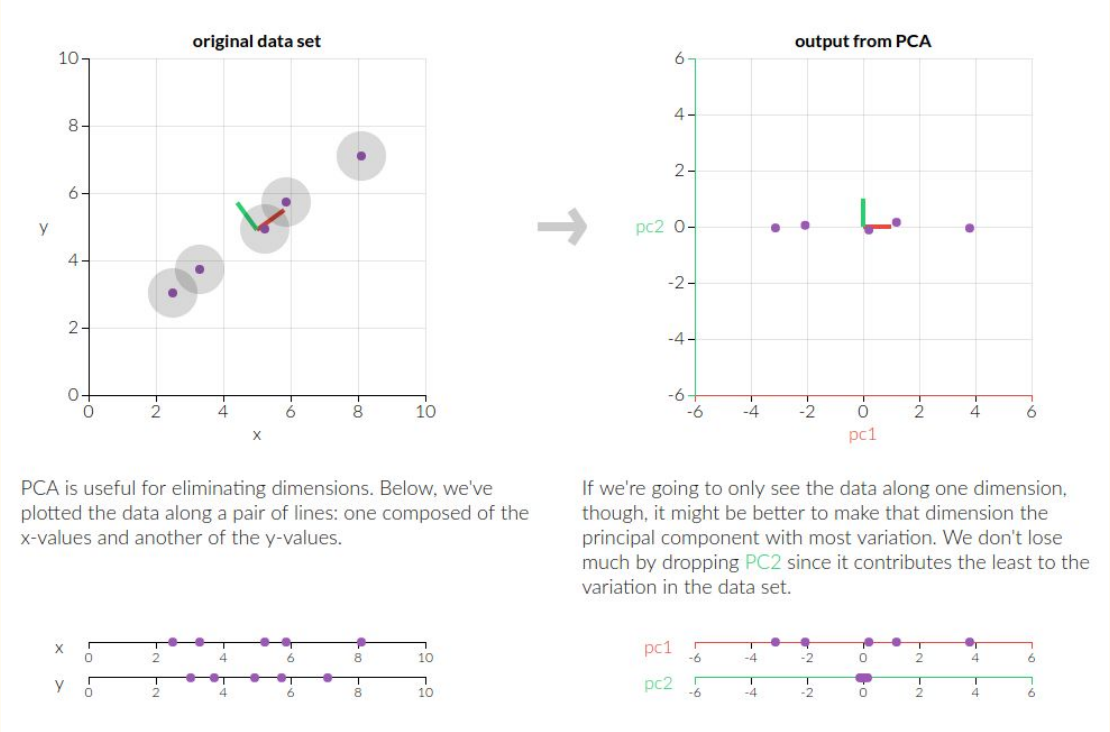


Figure 1. Principal Component Analysis on 2D data. Source: setosa.io

Mathematical Formulation

Data Matrix $\mathbf{Y}_{nxm}$ $\xrightarrow{PCA}$ Principal Components Representation $\mathbf{Z}_{nxm}$

$$\mathbf{Z} = \mathbf{YW}$$

$\mathbf{W}_{mxm}$ is a matrix of eigenvectors of $\mathbf{X}_{mxm} = \mathbf{Y}^{\mathbf{T}}\mathbf{Y}$ in decreasing order of eigenvalues

$\mathbf{X}_{mxm}$ is a positive semi-definite Hermitian matrix

Quantum PCA

Input: Multiple copies of a density matrix X in a d-dimensional Hilbert space.

(X is positive semi-definite Hermitian matrix)

Output: Eigenvectors and Eigenvalues of matrix X with accuracy $\in$

Quantum PCA Steps

1. Efficient access to the input $\mathbf{X}$

2. Implement unitary operator $\mathbf{U} = \mathbf{e}^{-iXt}$

3. Using $\mathbf{U}$ to obtain eigenvectors and eigenvalues of $\mathbf{X}$

Step 1: Input

$\mathbf{X}$ is a p.s.d. Hermitian matrix

$\mathbf{X} = \mathbf{A}^*\mathbf{A}$ where $\mathbf{A} = \sum_i \mathbf{b}_i \, |\, \mathbf{a}_i \rangle \langle\, \mathbf{e}_i \,|$,

$\mathbf{b}_i \, |\, \mathbf{a}_i \rangle$ is the ith column of $\mathbf{A}$

$\langle\, \mathbf{e}_i \,|$ is an orthonormal basis

$|\, \mathbf{a}_i \rangle$ is normalized to 1

$\mathbf{b}_i$ is positive

Step 1: Input

We have following Quantum Oracle also known as qRAM:

$$Q_x : |i\rangle |0\rangle |0\rangle \longrightarrow |i\rangle |a_i\rangle |b_i\rangle$$

Using this oracle we can construct the state $\sum_i b_i |a_i\rangle |e_i\rangle$ (which is a representation of **X**) with **O(log d)** operations. [1]

Step 2: Implementing Unitary Operator

$$\mathrm{tr}_1\{e^{-iS\Delta t}\ X\otimes K\ e^{iS\Delta t}\} = (\cos^2\Delta t)K + (\sin^2\Delta t)X - i\sin\Delta t[X,K]$$

$$= K - i\Delta t[X,K] + O(\Delta t^2)$$

$$= e^{-iX\Delta t}K\ e^{iX\Delta t} + O(\Delta t^2)$$

where S is a swap operator and tr_1 means partial trace of first variable.

Using Suzuki-Trotter expansion theorem[2] we can simulate e^{-iXt} with accuracy ϵ with $n = O(t^2\epsilon)$ where $t = n\Delta t$

Step 3: Analysing X

We use quantum phase algorithm using **U** unitary operator for varying times.

U: $|\psi\rangle |0\rangle \longrightarrow \sum_i \psi_i |v_i\rangle |r^e_i\rangle$

where $|\psi\rangle$ is some arbitrary initial state,

$|v_i\rangle$ are the eigenvectors of **X**

r^e_i are the estimates of corresponding eigenvalues r_i

Using improved phase-estimation[3] techniques eigenvectors and eigenvalues can be measured with accuracy ϵ for time $t = O(\epsilon^{-1})$ and so requires $O(\epsilon^{-3})$ copies of **X**

Step 3: Analysing X

U: $\rceil \psi \rangle \rceil 0 \rangle \longrightarrow \Sigma_i \psi_i \rceil v_i \rangle \rceil r^e_i \rangle$

We put initial state as **X** itself which gives:

$$\Sigma_i r_i \rceil v_i \rangle \langle v_i \rceil \otimes \rceil r^e_i \rangle \langle r^e_i \rceil$$

This algorithm works well for low rank matrices as this final state will be dominated by the eigenvectors with largest eigenvalues.

Time Complexity

Quantum PCA: **O(R*log m*)**

PCA: **O(*poly(m)*)**

Application to Cluster Assignment

Suppose we have 2 sets of density matrices:

$$\rho = (1/m)\Sigma_i \, | \Phi_i \rangle \langle \Phi_i |$$

$$\sigma = (1/m)\Sigma_i \, | \Psi_i \rangle \langle \Psi_i |$$

We are given a new state $| x \rangle$: (we decompose it in terms of ρ-σ)

$$| x \rangle | 0 \rangle \longrightarrow \Sigma_i \, x_i \, | v_i \rangle | w_i \rangle$$

Quantum Machine Learning

CS682A

Aadil Hayat
17111001

Proposal

Proposed Papers

✔ Quantum Machine Learning https://arxiv.org/pdf/1611.09347.pdf

J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe & S. Lloyd

✔ Quantum Principal Component Analysis https://arxiv.org/pdf/1307.0401.pdf

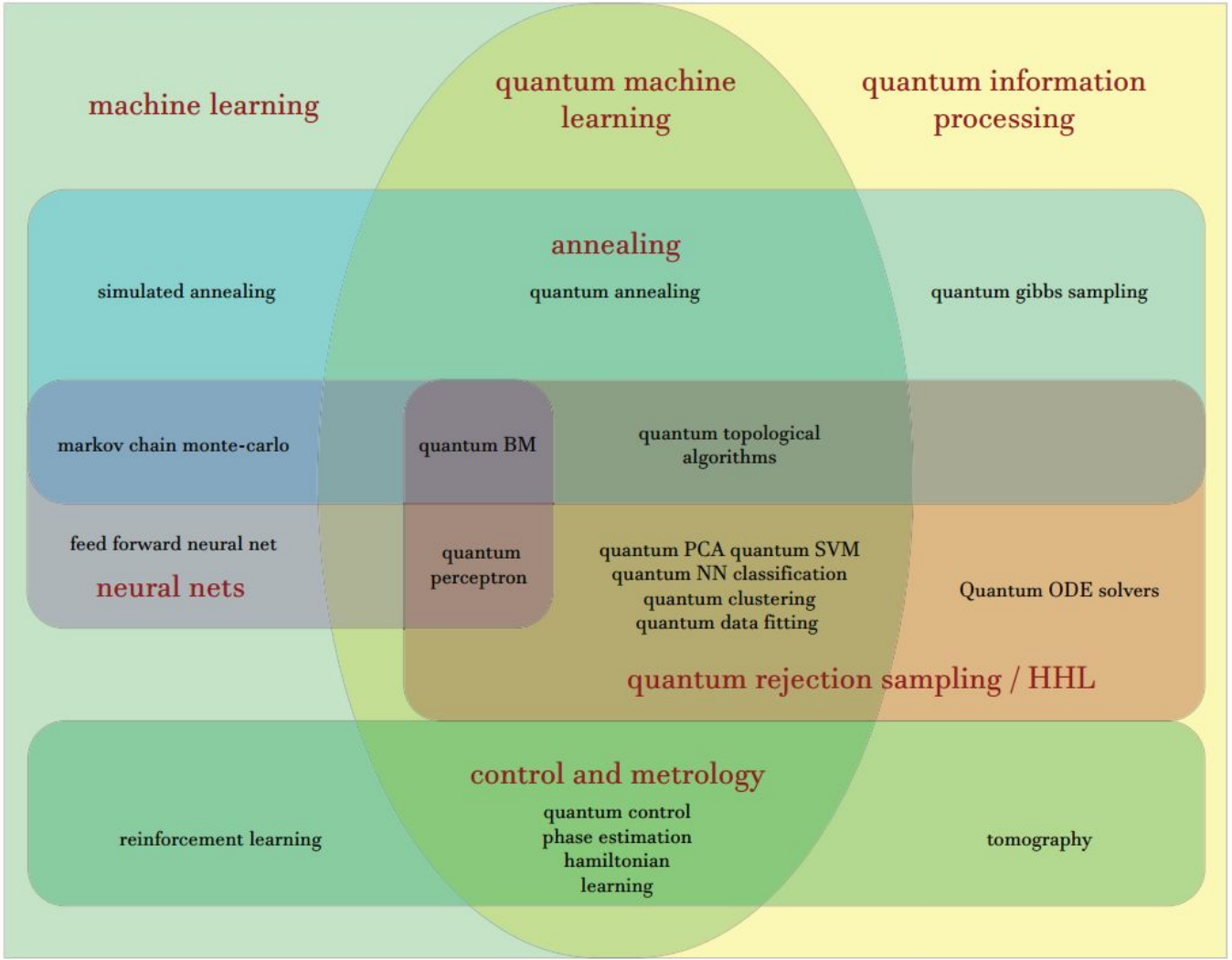
S. Lloyd, M. Mohseni & P. Rebentrost

↪ Quantum Support Vector Machines https://arxiv.org/pdf/1307.0471.pdf

P. Rebentrost, M. Mohseni & S. Lloyd

Previous Work

Quantum Machine Learning


machine learning
quantum machine learning
quantum information processing
annealing
simulated annealing
quantum annealing
quantum gibbs sampling
markov chain monte-carlo
quantum BM
quantum topological algorithms
feed forward neural net
neural nets
quantum perceptron
quantum PCA quantum SVM
quantum NN classification
quantum clustering
quantum data fitting
Quantum ODE solvers
quantum rejection sampling / HHL
control and metrology
reinforcement learning
quantum control
phase estimation
hamiltonian learning
tomography

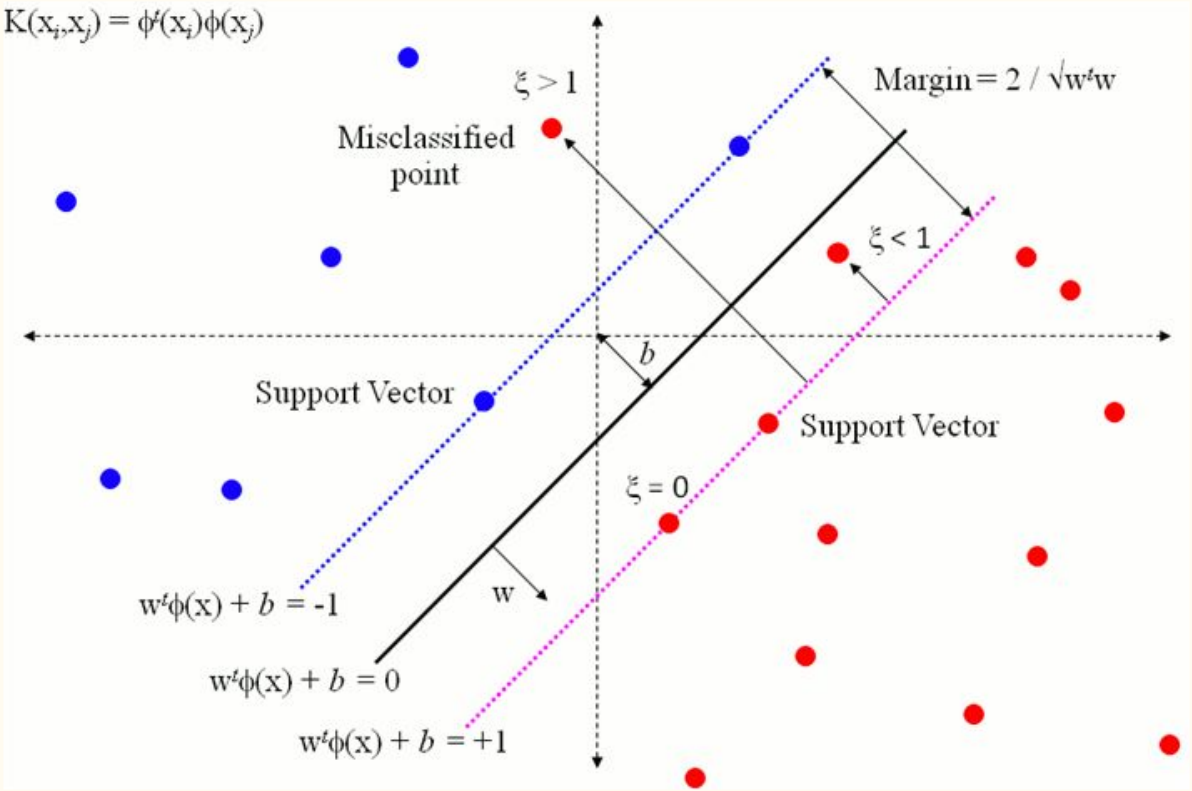
Quantum PCA

Overview

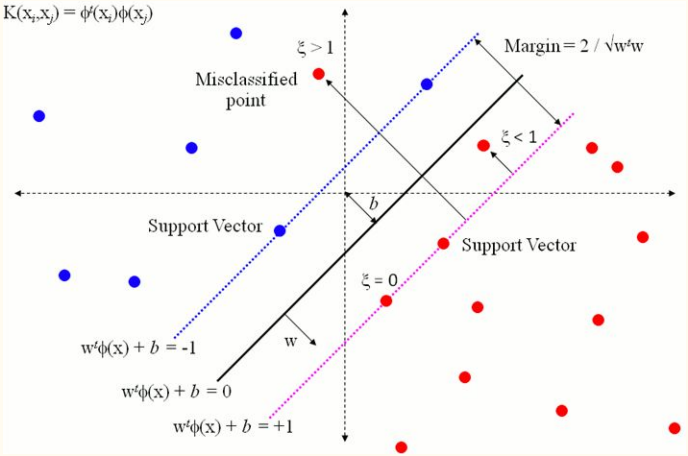
- Mathematical formulation of PCA in terms of Eigenvalue problem
- Using QRAM for efficient access to the input X
- Implementing Unitary Operator $\mathbf{e}^{-iXt}$ using multiple copies of X
- Using above unitary operator to obtain eigenvectors and eigenvalues of X
- Exponential speedup
- Application to Cluster Assignment

Quantum SVM

Support Vector Machine



Mathematical Formulation



Training data: $\{x_i, y_i\}$ for i = 1 … N and $x_i \in R^n$ and $y_i \in \{-1,1\}$

When data is separable:

$$y_i [w^T \Phi(x_i) + b] \geq 1$$

When data is not separable:

$$y_i [w^T \Phi(x_i) + b] \geq 1 - e_i \quad \text{where } e_i \geq 0$$

and $\Phi(x_i)$ is the non-linear map from original space to the high dimensional space a.k.a. Kernel function.

Optimization Problem:

$$\textit{min} \ \tfrac{1}{2} w^T w + c \Sigma e_i$$

Least Square Formulation:

$$\textit{min} \ \tfrac{1}{2} w^T w + \gamma/2 \Sigma e_i^2$$

Solution

$$
\begin{aligned}
L_2(w,b,e,\alpha) &= J_2(w,e) - \sum_{i=1}^{N} \alpha_i \left\{ \left[w^T \phi(x_i) + b \right] + e_i - y_i \right\}, \\
&= \tfrac{1}{2} w^T w + \tfrac{\gamma}{2} \sum_{i=1}^{N} e_i^2 - \sum_{i=1}^{N} \alpha_i \left\{ \left[w^T \phi(x_i) + b \right] + e_i - y_i \right\}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial L_2}{\partial w} = 0 \quad &\rightarrow \quad w = \sum_{i=1}^{N} \alpha_i \phi(x_i), \\
\frac{\partial L_2}{\partial b} = 0 \quad &\rightarrow \quad \sum_{i=1}^{N} \alpha_i = 0, \\
\frac{\partial L_2}{\partial e_i} = 0 \quad &\rightarrow \quad \alpha_i = \gamma e_i, \; i = 1, \ldots, N, \\
\frac{\partial L_2}{\partial \alpha_i} = 0 \quad &\rightarrow \quad y_i = w^T \phi(x_i) + b + e_i, \; i = 1, \ldots, N.
\end{aligned}
$$

$$
\begin{bmatrix} 0 & 1_N^T \\ 1_N & \Omega + \gamma^{-1} I_N \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ Y \end{bmatrix}
$$

Solution

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

$$F = J + K_\gamma$$

$$J = \begin{pmatrix} 0 & 1^\top \\ 1 & 0 \end{pmatrix}$$

$$\hat{F}|b, \vec{\alpha}\rangle = |\vec{y}\rangle$$

$$K_\gamma = \begin{pmatrix} 0 & 0 \\ 0 & K + \gamma^{-1} I \end{pmatrix}$$

$$\hat{F} = \frac{F}{\mathrm{tr}(F)} = \frac{F}{\mathrm{tr}(K_\gamma)}$$

Quantum SVM

Input: Multiple copies of a Kernel matrix K in a d-dimensional Hilbert space.
(K is positive semi-definite Hermitian matrix)

Output: Parameters of trained SVM model $(b,\alpha_1,...,\alpha_N)$

Quantum SVM Steps

1. Efficient access to the input
2. Implement unitary operator $\mathbf{U} = \mathbf{e}^{-iFt}$
3. Using **U** obtain the best estimate of model parameters
4. Classifying query states

Step 2: Implementing Unitary Operator

$$e^{-i\hat{F}\Delta t} = e^{-i\, J\Delta t/\mathrm{tr}(K_\gamma)} e^{-i\gamma^{-1} I\Delta t/\mathrm{tr}(K_\gamma)} e^{-iK\Delta t/\mathrm{tr}(K_\gamma)} + O(\Delta t^2)$$

Lie Product Formula

$$F = J + K_\gamma$$

$$J = \begin{pmatrix} 0 & 1^\top \\ 1 & 0 \end{pmatrix}$$

$$K_\gamma = \begin{pmatrix} 0 & 0 \\ 0 & K + \gamma^{-1} I \end{pmatrix}$$

Step 2: Implementing Unitary Operator

$$\mathrm{tr}_1\{e^{-iS\Delta t}\ K\otimes X\ e^{iS\Delta t}\} = (\cos^2\Delta t)X + (\sin^2\Delta t)K - i\sin\Delta t[K,X]$$

$$= X - i\Delta t[K,X] + O(\Delta t^2)$$

$$= e^{-iK\Delta t}X\ e^{iK\Delta t} + O(\Delta t^2)$$

where S is a swap operator and tr_1 means partial trace of first variable.

Using Suzuki-Trotter expansion theorem[2] we can simulate e^{-iKt} with accuracy ϵ with $n = O(t^2\epsilon)$ where $t = n\Delta t$

Step 3: Estimating Parameters

We use quantum phase algorithm using **U** unitary operator for varying times.

$$U: \; |\psi\rangle |0\rangle \longrightarrow \sum_i \psi_i \; |v_i\rangle |r^e_i\rangle$$

where $|\psi\rangle$ is some arbitrary initial state,

$|v_i\rangle$ are the eigenvectors of **X**

r^e_i are the estimates of corresponding eigenvalues r_i

Using improved phase-estimation[3] techniques eigenvectors and eigenvalues can be measured with accuracy ϵ for time $t = O(\epsilon^{-1})$ and so requires $O(\epsilon^{-3})$ copies of **X**

Step 3: Estimating Parameters

$$|\tilde{y}\rangle|0\rangle \rightarrow \sum_{j=1}^{M+1} \langle u_j|\tilde{y}\rangle|u_j\rangle|\lambda_j\rangle \rightarrow \sum_{j=1}^{M+1} \frac{\langle u_j|\tilde{y}\rangle}{\lambda_j}|u_j\rangle$$

Second step is done using matrix inversion techniques of Quantum algorithm for linear systems of equations[2] with runtime of $r^2 \log(N)/\varepsilon$

Step 4: Classification

$$|b,\vec{\alpha}\rangle = \frac{1}{\sqrt{C}}\left(b|0\rangle + \sum_{k=1}^{M}\alpha_k|k\rangle\right)$$

$$|\tilde{u}\rangle = \frac{1}{\sqrt{N_{\tilde{u}}}}\left(b|0\rangle|0\rangle + \sum_{k=1}^{M}\alpha_k|\vec{x}_k||k\rangle|\vec{x}_k\rangle\right)$$

$$|\tilde{x}\rangle = \frac{1}{\sqrt{N_{\tilde{x}}}}\left(|0\rangle|0\rangle + \sum_{k=1}^{M}|\vec{x}||k\rangle|\vec{x}\rangle\right)$$

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle|\tilde{u}\rangle + |1\rangle|\tilde{x}\rangle)$$

$$|\phi\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$$P = |\langle\psi|\phi\rangle|^2 = \frac{1}{2}(1 - \langle\tilde{u}|\tilde{x}\rangle)$$

$$\langle\tilde{u}|\tilde{x}\rangle = \frac{1}{\sqrt{N_{\tilde{x}}N_{\tilde{u}}}}\left(b + \sum_{k=1}^{M}\alpha_k|\vec{x}_k||\vec{x}|\langle\vec{x}_k|\vec{x}\rangle\right)$$

Step 4: Classification

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle|\tilde{u}\rangle + |1\rangle|\tilde{x}\rangle)$$

$$|\phi\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$$P = |\langle\psi|\phi\rangle|^2 = \frac{1}{2}(1 - \langle\tilde{u}|\tilde{x}\rangle)$$

$$\langle\tilde{u}|\tilde{x}\rangle = \frac{1}{\sqrt{N_{\tilde{x}} N_{\tilde{u}}}} \left(b + \sum_{k=1}^{M} \alpha_k |\vec{x}_k||\vec{x}|\langle\vec{x}_k|\vec{x}\rangle\right)$$

Time Complexity

Quantum SVM: $\mathbf{O(\mathit{log(MN)})}$

SVM: $\mathbf{O(\mathit{M^3})}$

Thanks