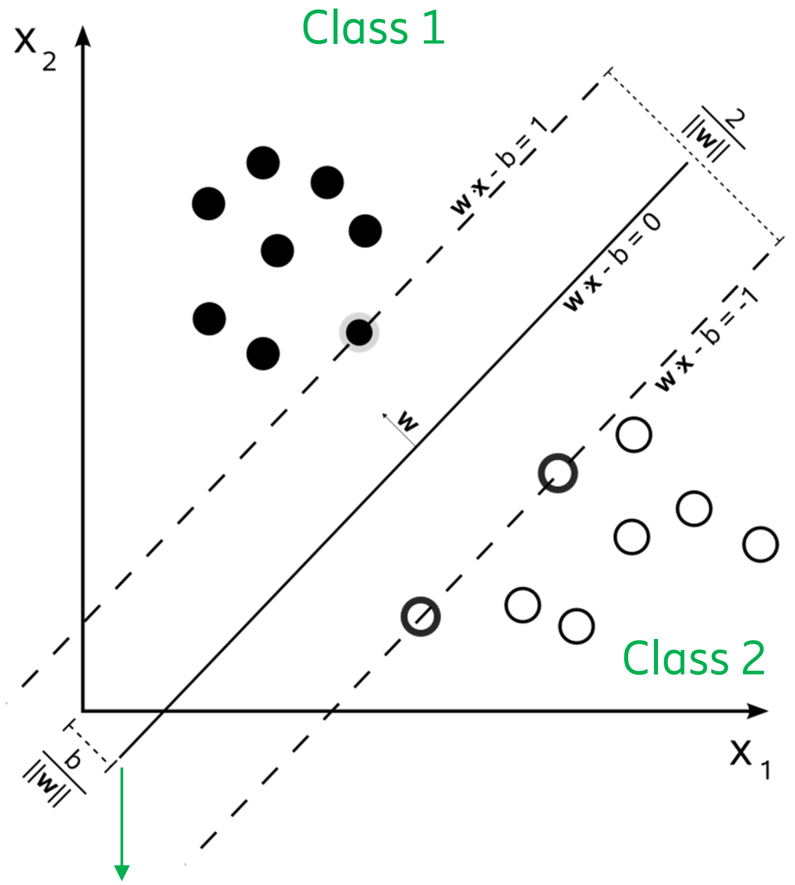
Quantum Support Vector Machine

Claire Yang
2019-03-14

Classical SVM



M training data points: $\{(\vec{x_j}, y_j): \vec{x_j} \in \mathbb{R}^N, y_j = \pm 1\}, j = 1 \dots M$

Goal → find the maximum-margin hyperplane → maximize $\frac{2}{|\vec{\omega}|}$ → minimize $\frac{|\vec{\omega}|^2}{2}$

The constraint:

$$\begin{cases} \vec{w} \cdot \vec{x_j} + b \geq 1 & \text{if } y_j = +1 \ (y_j \ belongs\ to\ class\ 1) \\ \vec{w} \cdot \vec{x_j} + b \leq -1 & \text{if } y_j = -1 \ (y_j \ belongs\ to\ class\ 2) \end{cases} \Longleftrightarrow y_i(\vec{w} \cdot \vec{x_j} + b) \geq 1$$

Computational Complexity: $O(log(\epsilon^{-1})poly(N, M))$

Dimension of feature space (input data)

Number of training data points

Image: https://zh.wikipedia.org/wiki/File:Svm_max_sep_hyperplane_with_margin.png

SVM structure

Goal → find the maximum-margin hyperplane → maximize $\frac{2}{|\vec{\omega}|}$ s.t. $y_j(\vec{w} \cdot \vec{x_j} + b) \geq 1$ → minimize $\frac{|\vec{\omega}|^2}{2}$ s.t. $y_j(\vec{w} \cdot \vec{x_j} + b) \geq 1$ → Fulfils KKT condition

↓ introduce the Lagrange multipliers $\vec{\alpha}$

$$\max_{\vec{\alpha}} \min_{\vec{\omega}, b} L(\vec{\omega}, b, \vec{\alpha}) = \frac{|\vec{\omega}|^2}{2} - \sum_{j=1}^{M} \alpha_j y_j (y_j(\vec{\omega} \cdot \vec{x_j} + b) - 1), \text{s.t.} \sum_{j=1}^{M} \alpha_j = 0 \text{ and } \alpha_j y_j \geq 0$$

Partial derivative

$$\begin{cases} \frac{\partial L}{\partial \vec{\omega}} = 0 \Rightarrow \vec{\omega} = \sum_{j=1}^{M} \alpha_j y_j \vec{x_j} \\ \frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{j=1}^{M} \alpha_j y_j^2 = 0 \end{cases}$$

↓

Kernel, Inner product ←Problem 1

$$\max_{\vec{\alpha}} L(\vec{\alpha}) = \sum_{j=1}^{M} \alpha_j y_j - \frac{1}{2} \sum_{j,k=1}^{M} \alpha_j \vec{x_j} \cdot \vec{x_k} \alpha_k$$

↓

Find optimal $\vec{\alpha}^*$ ←Problem 2

↓

Optimal hyperplane with parameters:

$$\begin{cases} \vec{\omega} = \sum_{j,k=1}^{M} \alpha_j^* \vec{x_j} \\ b = y_i - \vec{\omega} \end{cases}$$

Quantum SVM

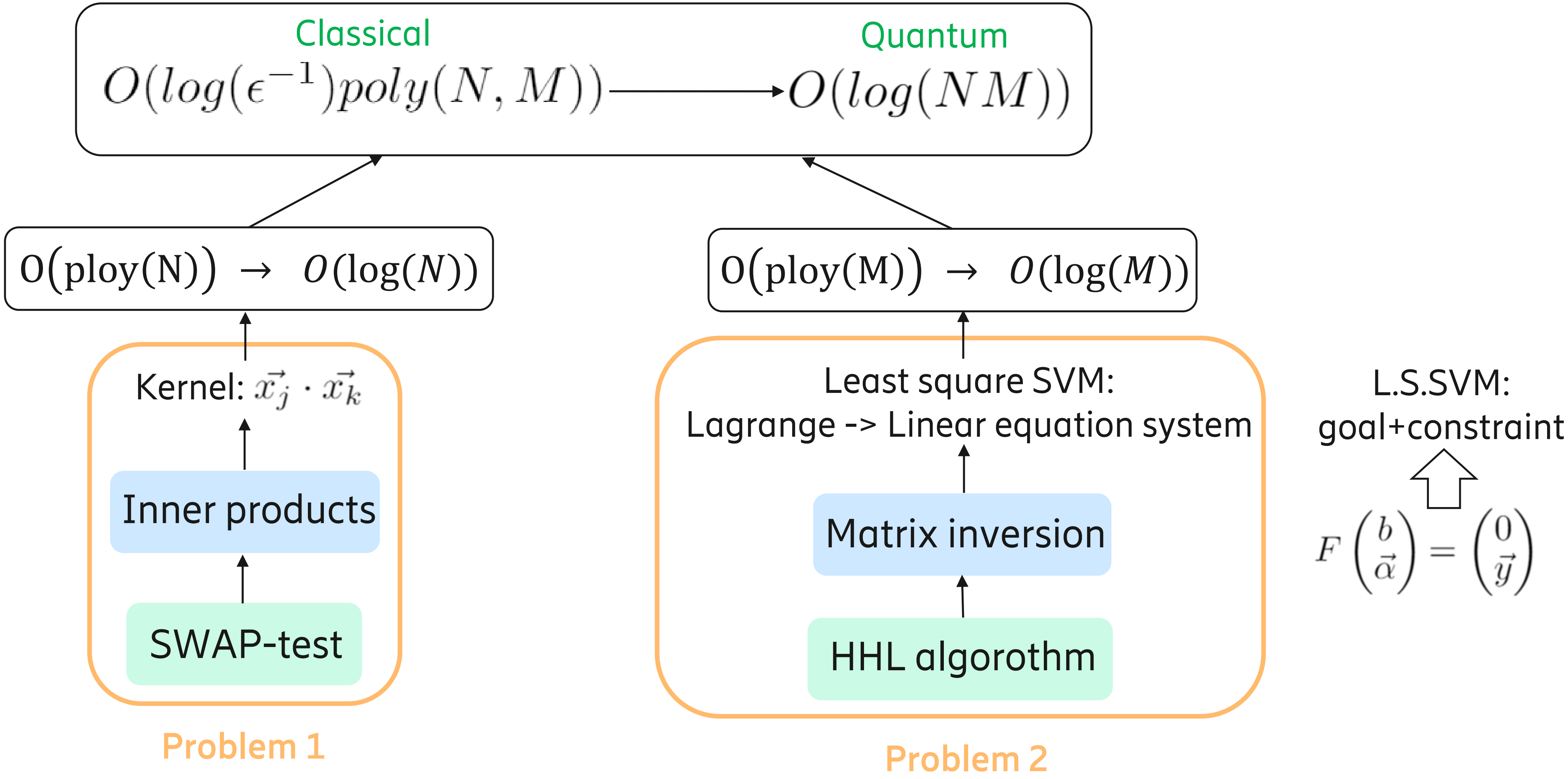
Computational Complexity:

Classical: $O(log(\epsilon^{-1})poly(N, M))$ $\longrightarrow$ Quantum: $O(log(NM))$

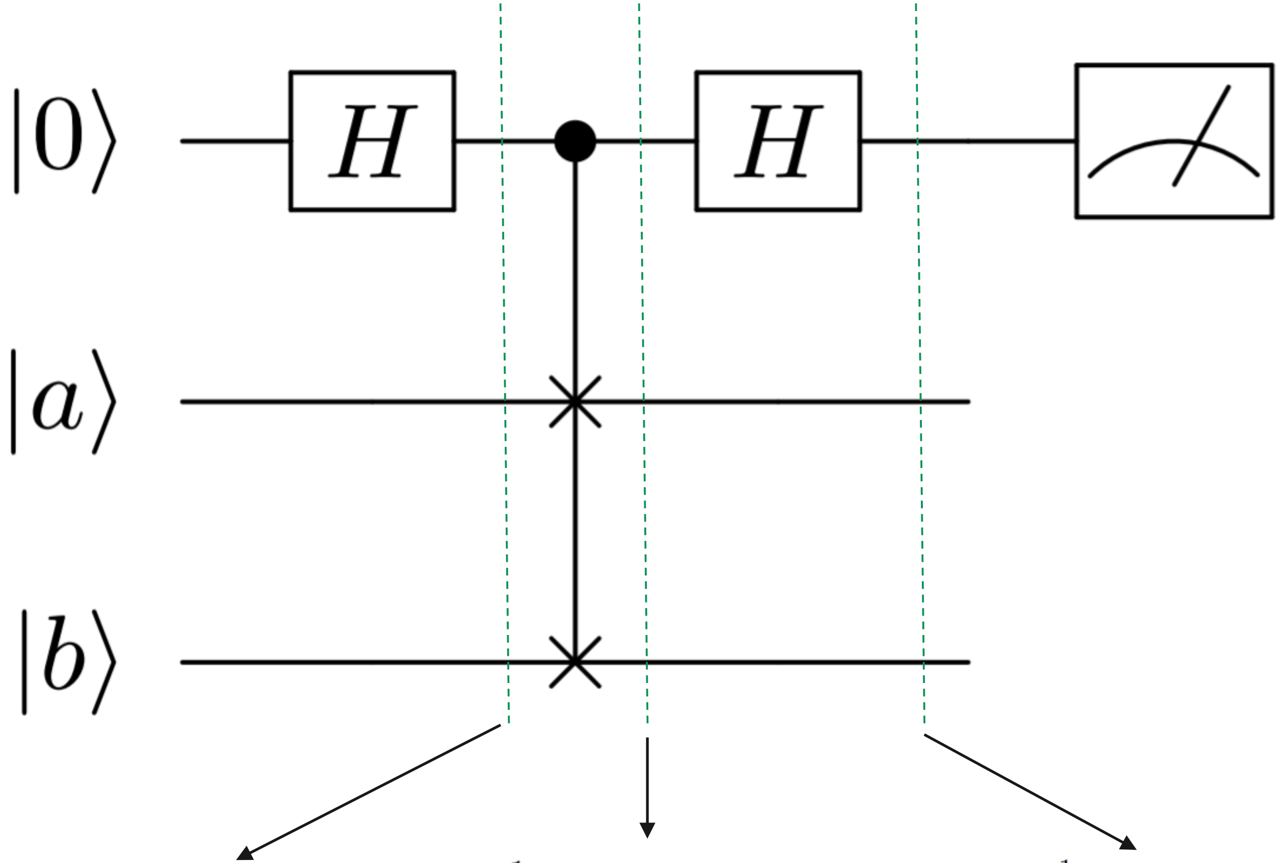
training

testing

HHL based qSVM



Problem 1 – Inner product & SWAP-test



$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|0,a,b\rangle + |1,a,b\rangle)$$

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|0,a,b\rangle + |1,b,a\rangle)$$

$$|\psi_3\rangle = \frac{1}{2}|0\rangle(|a,b\rangle + |b,a\rangle) + \frac{1}{2}|1\rangle(|a,b\rangle - |b,a\rangle)$$

$$\begin{aligned}
P(|0\rangle) &= |\langle 0|\psi_3\rangle|^2 \\
&= |\frac{1}{2}\langle 0|0\rangle(|a,b\rangle + |b,a\rangle) + \frac{1}{2}\langle 0|1\rangle(|a,b\rangle - |b,a\rangle)|^2 \\
&= \frac{1}{4}|(|a,b\rangle + |b,a\rangle)|^2 \\
&= \frac{1}{4}(\langle b|b\rangle\langle a|a\rangle + \langle b|a\rangle\langle a|b\rangle + \langle a|b\rangle\langle b|a\rangle + \langle a|a\rangle\langle b|b\rangle) \\
&= \frac{1}{2} + \frac{1}{2}|\langle a|b\rangle|^2
\end{aligned}$$

Inner product

Image: https://arxiv.org/pdf/1804.10068.pdf

Problem 2 – Least Square SVM

Slack variable

Constraint: $y_j(\vec{\omega}\cdot\vec{x_j}+b)\geqslant 1 \overset{y_j^2=1}{\longrightarrow} \vec{\omega}\cdot\vec{x_j}+b=y_j-y_j e_j$

New Lagrange function: $L(\vec{\omega},b,\vec{e},\vec{\alpha})=\frac{|\vec{\omega}|^2}{2}+\frac{\gamma}{2}\sum_{j=1}^{M}e_j^2-\sum_{j=1}^{M}\alpha_j y_j(\vec{\omega}\cdot\vec{x_j}+b-y_j+y_j e_j)$

Penalty term

⇩

Linear equation system: $F\begin{pmatrix} b \\ \vec{\alpha}\end{pmatrix}\equiv\begin{pmatrix} 0 & -\vec{1}^T \\ \vec{1} & K+\gamma^{-1}I\end{pmatrix}\begin{pmatrix} b \\ \vec{\alpha}\end{pmatrix}=\begin{pmatrix} 0 \\ \vec{y}\end{pmatrix}$

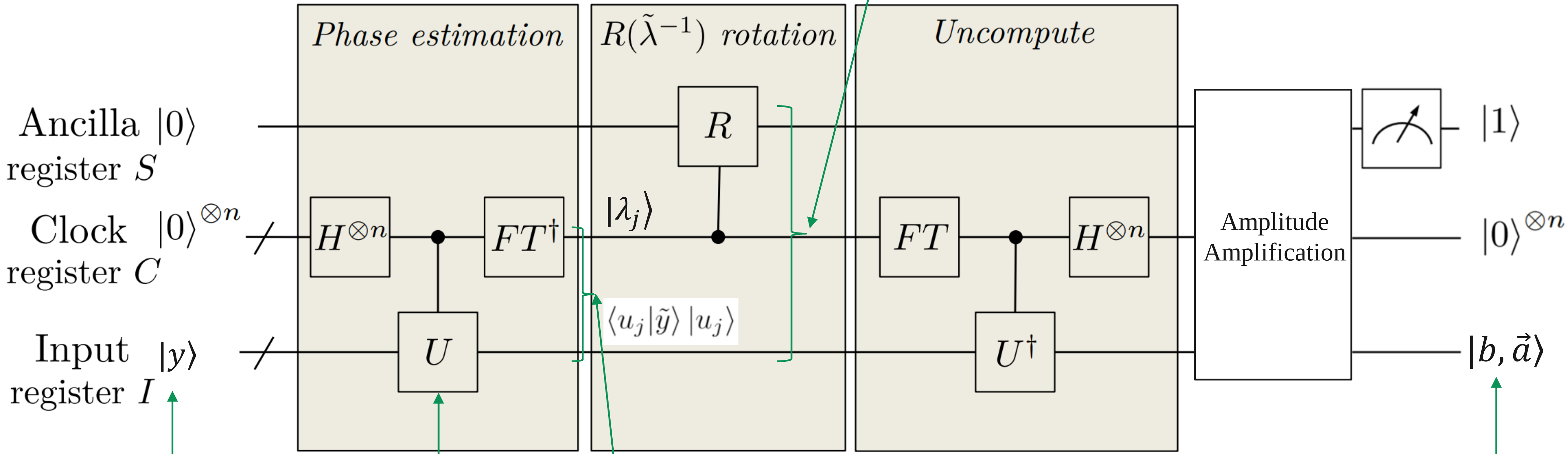
⇩

HHL

Training process

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

$\sum_{j=1}^{M+1}\left(\sqrt{1-\frac{C^2}{\lambda_j^2}}|0\rangle + \frac{C}{\lambda_j}|1\rangle\right)\langle u_j|\tilde{y}\rangle\,|u_j\rangle|\lambda_j\rangle$



$|\tilde{y}\rangle = \sum_{j=1}^{M+1}\langle u_j|\tilde{y}\rangle\,|u_j\rangle$
(1 qubit longer than $|\vec{y}\rangle$)

$F \rightarrow \mathrm{U} = e^{-i\hat{F}t}$

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

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

Difficult point: how to enact this exponentiation?

Image: https://arxiv.org/pdf/1802.08227.pdf

Difficult point in training process - Enact $e^{-i\hat{F}t}$

$e^{-i\hat{F}\Delta t}$

$\downarrow$

$\hat{K} = K/trK$

$\downarrow$

$e^{-i\hat{K}\Delta t}$

$$F = \begin{pmatrix} 0 & \vec{1}^T \\ \vec{1} & K + \gamma^{-1} I \end{pmatrix}, \quad \hat{F} = (J + K + \gamma^{-1} I)/trF, \quad J = \begin{pmatrix} 0 & \vec{1}^T \\ \vec{1} & 0 \end{pmatrix}$$

Eliminate γ^{-1}

$$e^{-i\hat{F}\Delta t} = e^{-i\Delta t I/trF} e^{-i\Delta t J/trF} e^{-i\Delta t K/trF} + O(\Delta t^2)$$

$$= e^{-i\Delta t' I/trK} e^{-i\Delta t' J/trK} e^{-i\Delta t' K/trK} + O(\Delta t^2)$$

Rescale time by a factor $\frac{trK}{trF}$

$\gamma^{-1} I$: is trivial

J: eigenvalues: $\lambda_{\pm}^{star} = \pm\sqrt{M}$, eigenstates: $|\lambda_{\pm}^{star}\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle \pm \frac{1}{\sqrt{M}} \sum_{k=1}^{M} |k\rangle\right)$

K/trK : density operator

Image:https://mp.weixin.qq.com/s?__biz=MzUxMDQzNzEzNA==&mid=2247483963&idx=1&sn=7ca3a577f3f02ed8e8b3aeff407cb5bb&chksm=f903b05ace74394c33c2cdae2d7989cf841c4fbab6095cbb224bfd3892de1f9c08c8a38d1a2b&scene=21#wechat_redirect

Difficult point in training process – Enact $\frac{K}{trK}$

1. Use **training data oracle** to prepare state $|\mathcal{X}\rangle$
2. Obtain $\frac{K}{trK}$ by calculating the **partial trace** of $|\mathcal{X}\rangle$

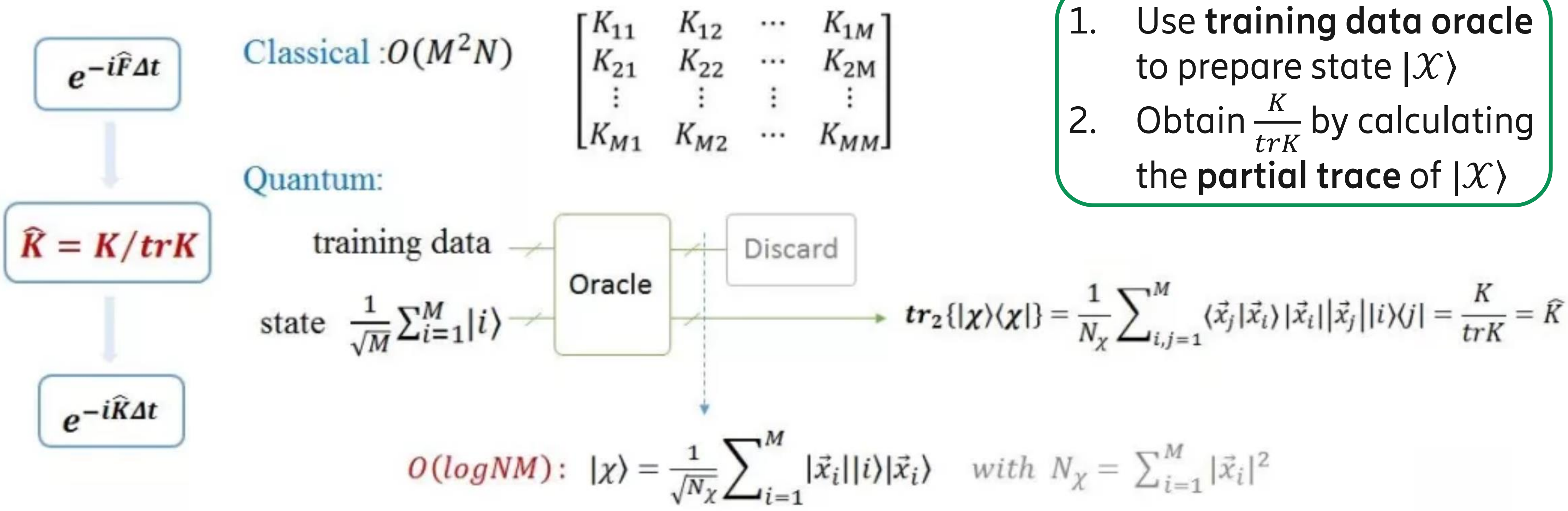


Image:https://mp.weixin.qq.com/s?__biz=MzUxMDQzNzEzNA==&mid=2247483963&idx=1&sn=7ca3a577f3f02ed8e8b3aeff407cb5bb&chksm=f903b05ace74394c33c2cdae2d7989cf841c4fbab6095cbb224bfd3892de1f9c08c8a38d1a2b&scene=21#wechat_redirect

Difficult point in training process - Enact $e^{-i\hat{K}t}$

$e^{-i\hat{F}\Delta t}$

$\downarrow$

$\hat{K} = K/trK$

$\downarrow$

$e^{-i\hat{K}\Delta t}$

Given by the qPCA articale

To simulate non-sparse symmetric or Hermitian matrics:

Density matrix exponentiation: n copies of $\hat{K}$

For some quantum state ρ, $e^{-i\hat{K}\Delta t}\rho e^{i\hat{K}\Delta t} \equiv e^{-iL_{\hat{K}}\Delta t}(\rho)$, $L_{\hat{K}} = [\hat{K}, \rho]$ $\left(L_{\hat{K}} = [\hat{K}, \cdot]\right)$

$$e^{-iL_{\hat{K}}\Delta t}(\rho) \approx tr_1\{e^{-iS\Delta t}\hat{K} \otimes \rho e^{iS\Delta t}\} = \rho - i\Delta t[\hat{K}, \rho] + O(\Delta t^2)$$

$$S = \sum_{m,n=1}^{M} |m\rangle\langle n| \otimes |n\rangle\langle m|$$

M^2 by M^2 matrix, the SWAP matrix

Image:https://mp.weixin.qq.com/s?__biz=MzUxMDQzNzEzNA==&mid=2247483963&idx=1&sn=7ca3a577f3f02ed8e8b3aeff407cb5bb&chksm=f903b05ace74394c33c2cdae2d7989cf841c4fbab6095cbb224bfd3892de1f9c08c8a38d1a2b&scene=21#wechat_redirect

Classification process

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

Output: $y \in \{-1, +1\}$

Algorithm:

1. By calling the training data oracle, construct $|\tilde{u}\rangle$ and the query state $|\tilde{x}\rangle$:

$$|\tilde{u}\rangle = \frac{1}{\sqrt{N_{\tilde{u}}}}\left(b|0\rangle|0\rangle + \sum_{k=1}^{M} \alpha_k |\vec{x}_k||k\rangle|\vec{x}_k\rangle\right), \ N_{\tilde{u}} = b^2 + \sum_{k=1}^{M} \alpha_k^2 |\vec{x}_k|^2$$

$$|\tilde{x}\rangle = \frac{1}{\sqrt{N_{\tilde{x}}}}\left(|0\rangle|0\rangle + \sum_{k=1}^{M} |\vec{x}||k\rangle|\vec{x}\rangle\right), \ N_{\tilde{x}} = M|\vec{x}|^2 + 1$$

2. Perform a swap test.

   Using an ancilla, construct : $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle|\tilde{u}\rangle + |1\rangle|\tilde{x}\rangle)$, measure the ancilla in $|\phi\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

   Success $P = |\langle\psi|\phi\rangle|^2 = \frac{1}{2}(1 - \langle\tilde{u}|\tilde{x}\rangle)$, $\langle\tilde{u}|\tilde{x}\rangle = 1/\sqrt{N_{\tilde{x}} N_{\tilde{u}}}\left(b + \sum_{k=1}^{M} \alpha_k |\vec{x}_k||\vec{x}|\langle\vec{x}_k|\vec{x}\rangle\right)$

3. If $P < \frac{1}{2}$, we classify $|\vec{x}\rangle$ as +1; otherwise, -1.

Image:https://mp.weixin.qq.com/s?__biz=MzUxMDQzNzEzNA==&mid=2247483963&idx=1&sn=7ca3a577f3f02ed8e8b3aeff407cb5bb&chksm=f903b05ace74394c33c2cdae2d7989cf841c4fbab6095cbb224bfd3892de1f9c08c8a38d1a2b&scene=21#wechat_redirect

References

— Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. Physical review letters, 113(13):130503, 2014.

— Bojia Duan. Wechat article (link)

— Dawid Kopczyk. Quantum machine learning for data scientists. arXiv preprint. arXiv:1804.10068, 2018.