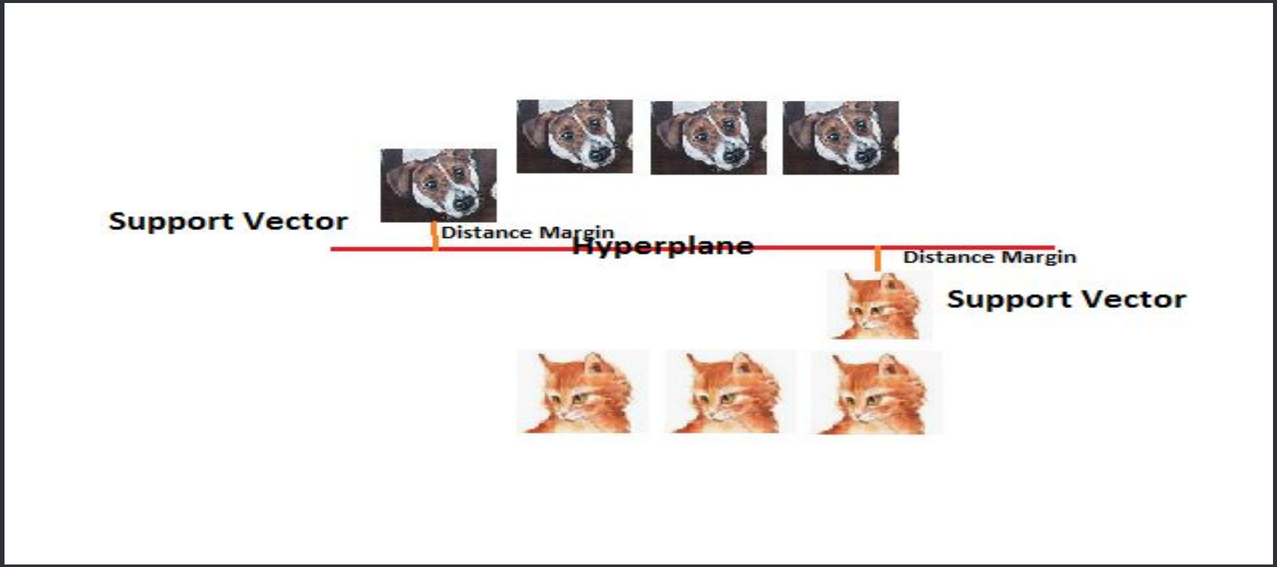
Quantum-enhanced Support Vector Machine (QSVM)

Support Vector Machines (SVM)

SVM is a classification algorithm and a supervised learning model:

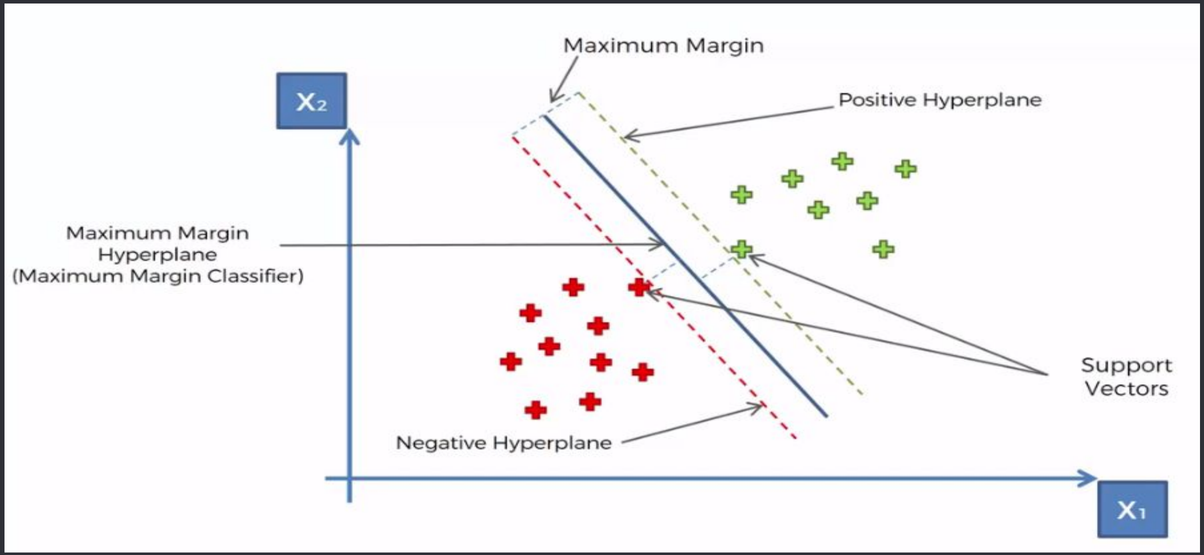
- Pattern recognition
- Data mining

The goal is to design a hyperplane that classifies all training vectors into two classes



KEY CONCEPTS

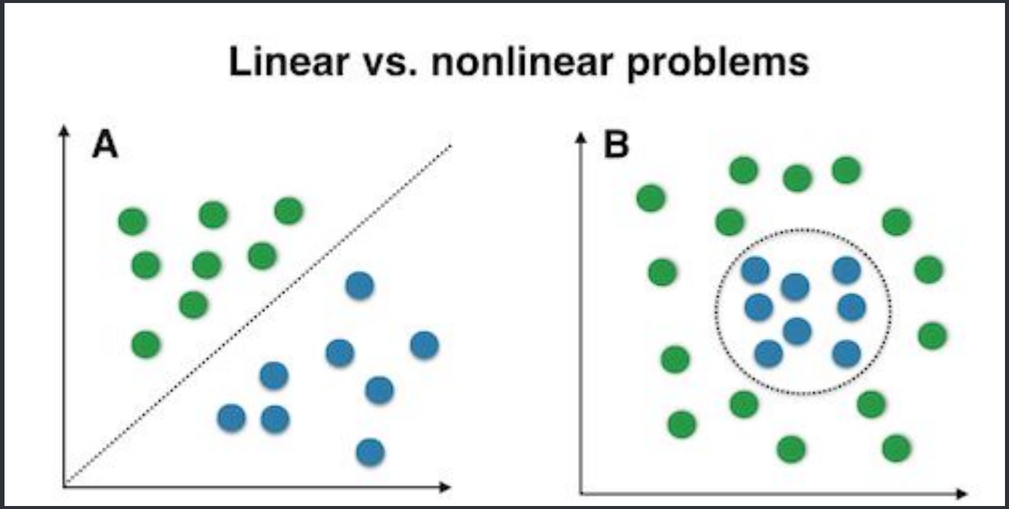
- The line that separates the two classes is called **hyperplane**
- The **margin** is the distance between the hyperplane and the closest elements from this hyperplane
- **Support vectors** are vector points closest to the hyperplane



Example of a linear classification

Non-linear classifications

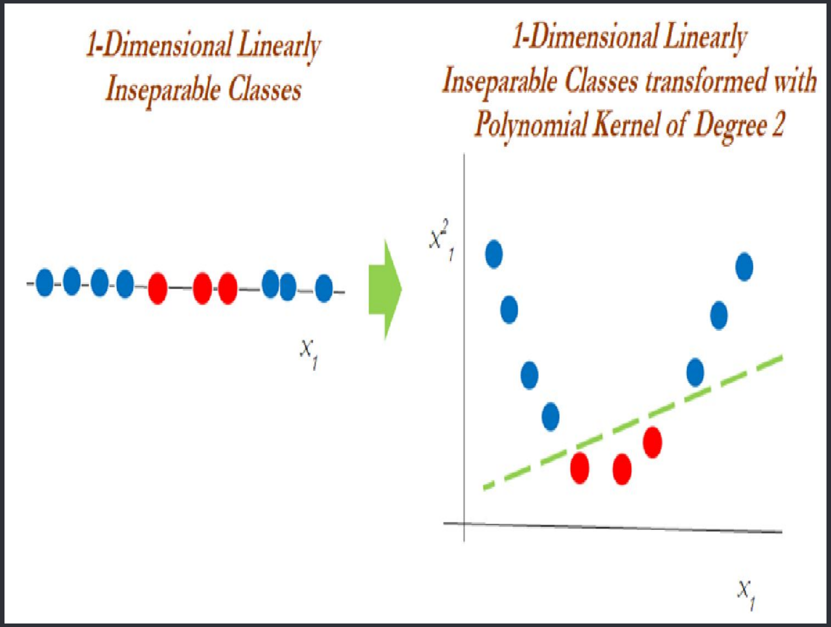
- Some data cannot be separated by a hyperplane in its original space
- We can transform a non-linear classification into a linear classification using the **kernel function**

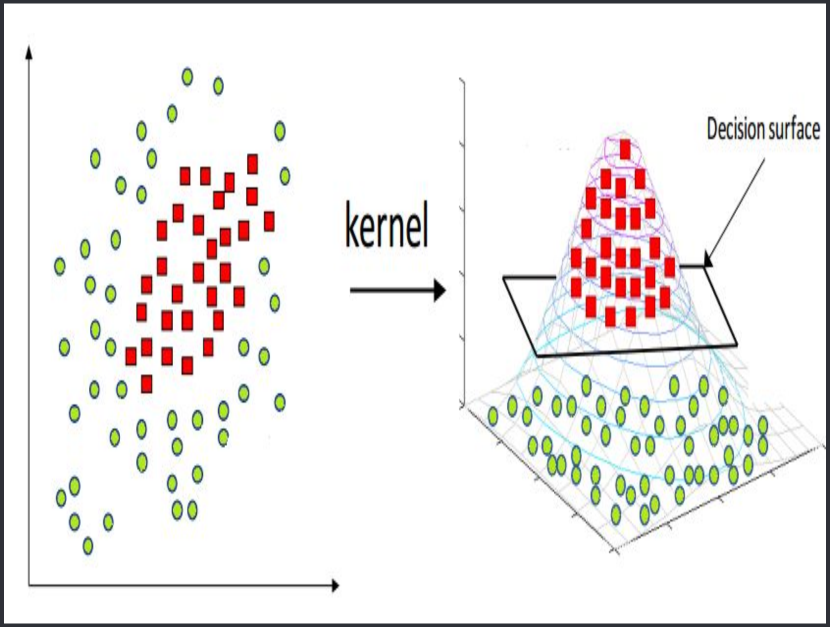


Kernel Function

SVM can perform non-linear classifications by using the kernel function or kernel trick, which maps the inputs into high-dimensional features (feature map)

1D to 2D **2D to 3D**


1-Dimensional Linearly Inseparable Classes
x_1
1-Dimensional Linearly Inseparable Classes transformed with Polynomial Kernel of Degree 2
x_1^2
x_1



Kernel Function cont.

- Higher dimensional space is computationally expensive
- The kernel function reduces the computational cost by avoiding the direct transformations of data points and calculating the distance of two points in a higher dimension directly
- For example, applying the dot product between two vectors so every point is mapped into a higher dimensional space and thus transforming a non-linear space into a linear space

Quantum Support Vector Machines (QSVM)

- Classification problems for computing the kernel is not efficient classically
- Computational resources will exponentially increase with the size of the problem
- This can be solved in a quantum processor by a direct estimation of the kernel in the feature space
- It consist of a **training phase** (where the kernel is calculated and the support vectors are obtained)
- **Test or classification phase** (where new unlabeled data is classified according to the solution found in the training phase)
- Quantum kernel is equivalent to the classical kernel function

Quantum Classification

To perform a quantum classification, it requires 3 steps.

1. Convert classical data to quantum data
2. Process the data
3. Apply measurements to read the output

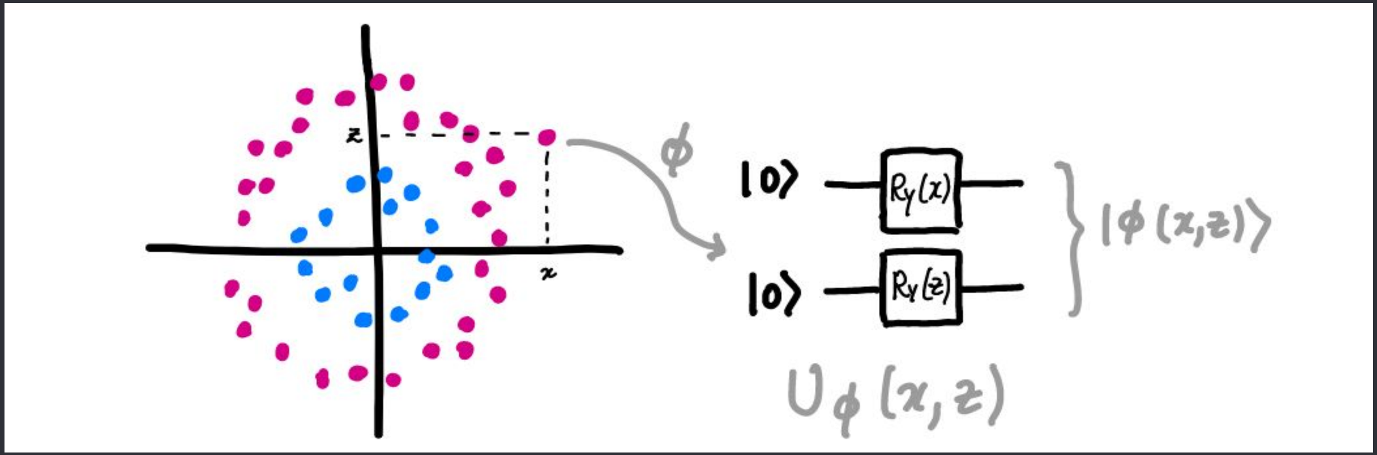
Quantum Feature Maps

To convert the classical data into quantum data, we can use a **Quantum feature map**

Let $\mathcal{X}$ be a set of input data

- Let $\mathcal{F}$ be the vector space
- Let $\boldsymbol{\phi}$(phi) be the feature map
- $\boldsymbol{\phi}$ is a function that acts as $\boldsymbol{\phi} : \mathcal{X} \rightarrow \mathcal{F}$

The map transforms $x \rightarrow |\varphi(x)\rangle$ by way of a unitary transformation $U_\varphi(x)$



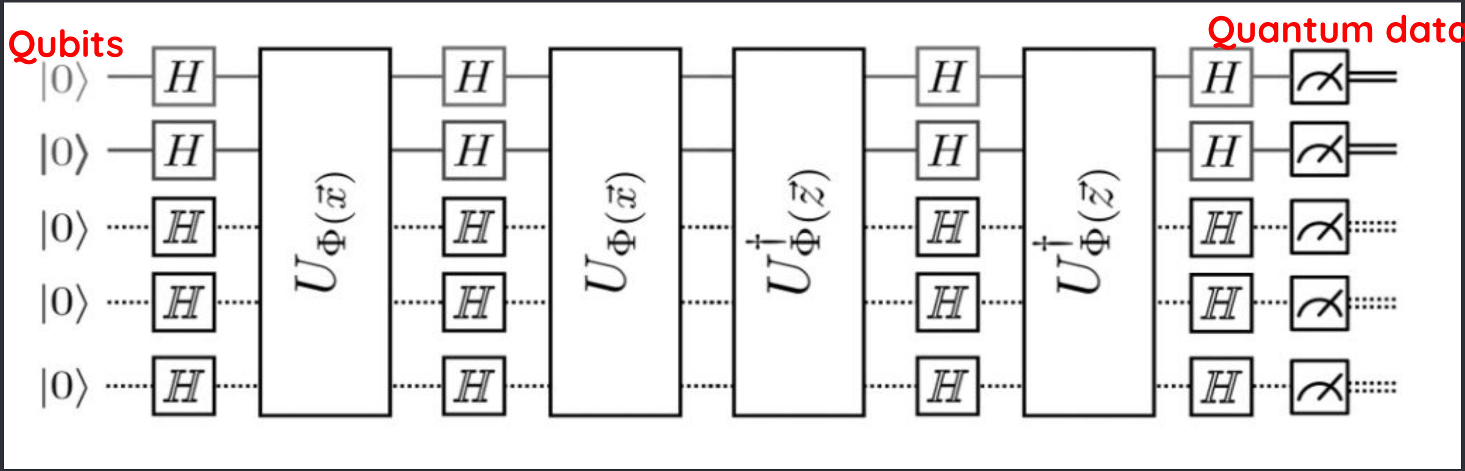
Quantum Feature Maps cont.

We want to represent the classical data into a Hilbert Space: the feature map takes inputs and does some gate operations.

U is a matrix $U1(\lambda) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix}$ and H is a matrix $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

The **output** is represented in quantum data.

The quantum data are then used to build the kernel of the SVM



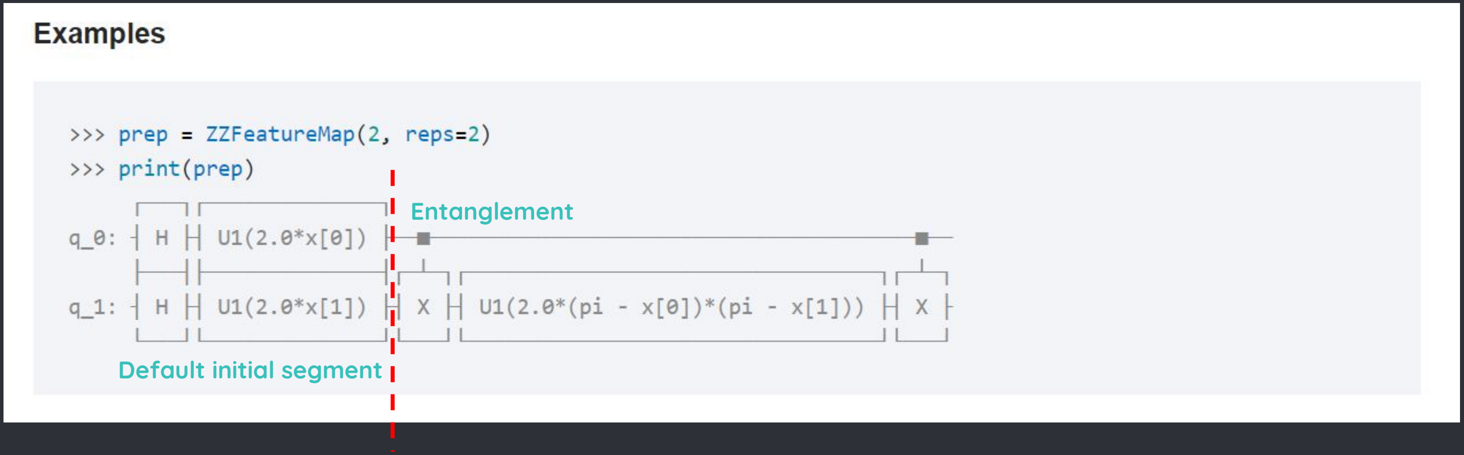
ZZFeatureMap

ZZFeatureMap is used in our code.

The initial segment of the ZZFeatureMap is a default setting and the next segment creates the entanglement state.

The number of qubits is equivalent to the number of features. For example, in a stock market with 10 features such as the opening price, closing price, time, etc. would require 10 qubits.

Processing the data and measuring the output are done by the QSVM package in Qiskit.



Examples

```
>>> prep = ZZFeatureMap(2, reps=2)
>>> print(prep)
     ┌───┐┌─────────────────┐
q_0: ┤ H ├┤ U1(2.0*x[0]) ├──■────────────────────────────────────■──
     ├───┤├─────────────────┤┌─┴─┐┌──────────────────────────────┐┌─┴─┐
q_1: ┤ H ├┤ U1(2.0*x[1]) ├┤ X ├┤ U1(2.0*(pi - x[0])*(pi - x[1])) ├┤ X ├
     └───┘└─────────────────┘└───┘└──────────────────────────────┘└───┘
```

Comparing SVM Results and QSVM

The SVM Time Complexity:

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

$$O(\log(1/\epsilon) M^2(N + M))$$

N: the dimension of the feature space

M: Number of training vectors

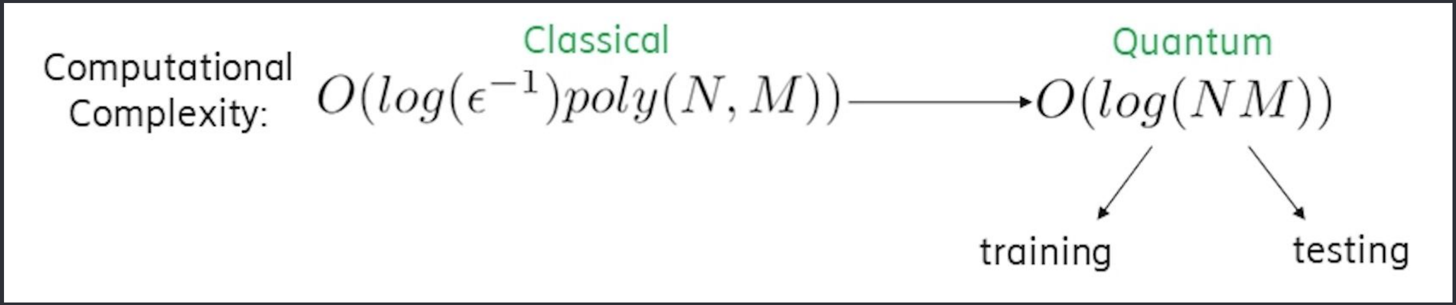
ϵ: The accuracy

Comparing SVM Results and QSVM

The QSVM Time Complexity:

Grover’s Search based(2003) -----Quadratic speedup

HHL based(2013) -----Exponential speedup



Computational Complexity:

$$\begin{tikzcd}[row sep=small] \text{Classical} & & \text{Quantum} \\ O(log(\epsilon^{-1})poly(N,M)) \arrow[rr] & & O(log(NM)) \arrow[dl] \arrow[dr] \\ & \text{training} & & \text{testing} \end{tikzcd}$$

Comparing SVM Results and QSVM

Solve SVM

Points: --------> $\{(\vec{x}_j, y_j) : \vec{x}_j \in \mathbb{R}^N, y_j = \pm 1\}_{j=1\ldots M}$

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

$k(\vec{x}_j, \vec{x}_k) = \vec{x}_j \cdot \vec{x}_k$

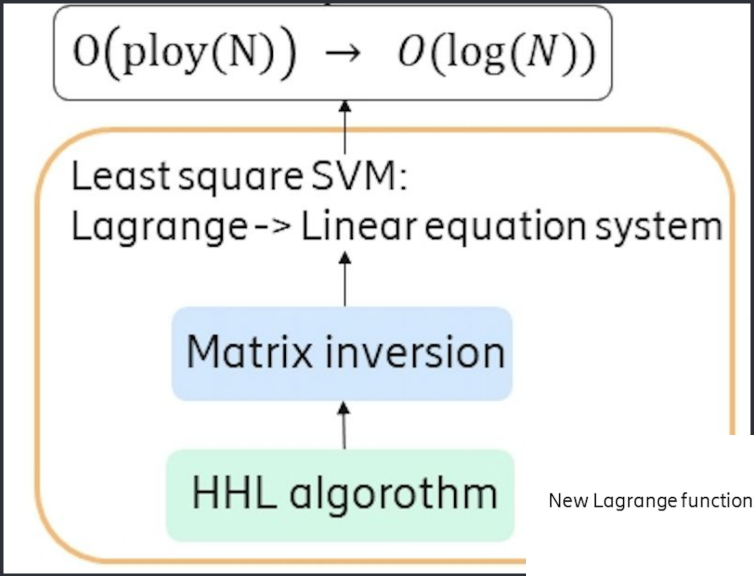
Classically inner product take $O(\epsilon^{-2}\mathrm{poly}(N))$

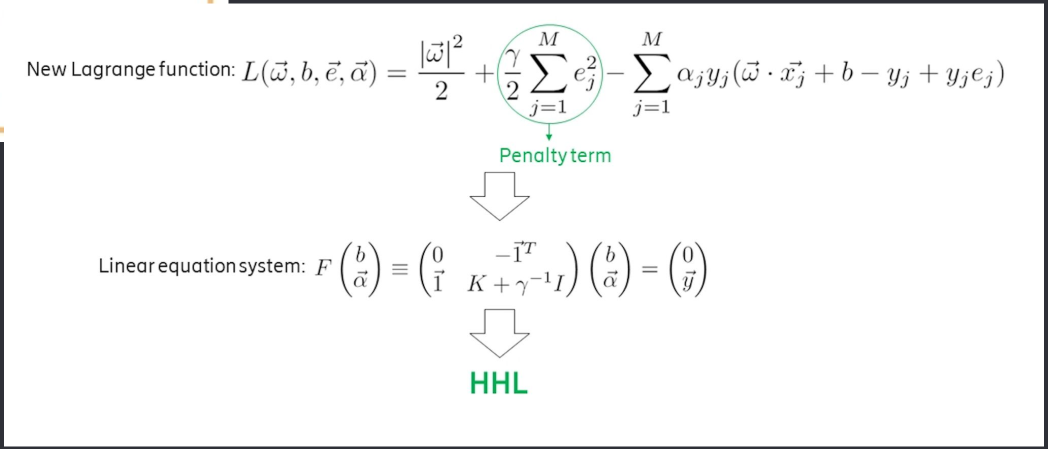
SWAP-test:---------> $O(\mathrm{ploy}(N)) \rightarrow O(\log(N))$

$O(\log(1/\epsilon)M^2(N+M))$ $O(\log(1/\epsilon)M^3 + M^2 \log N/\epsilon)$

Comparing SVM Results and QSVM

Least Square SVM:





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

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

Comparing SVM Results and QSVM

Result:

$$O(\kappa_{\text{eff}}^{3}\epsilon^{-3}\log MN)$$

$$\kappa_{\text{eff}} = 1/\epsilon_K$$

https://arxiv.org/pdf/1307.0471.pdf

https://blog.csdn.net/m0_37622530/article/details/88851497