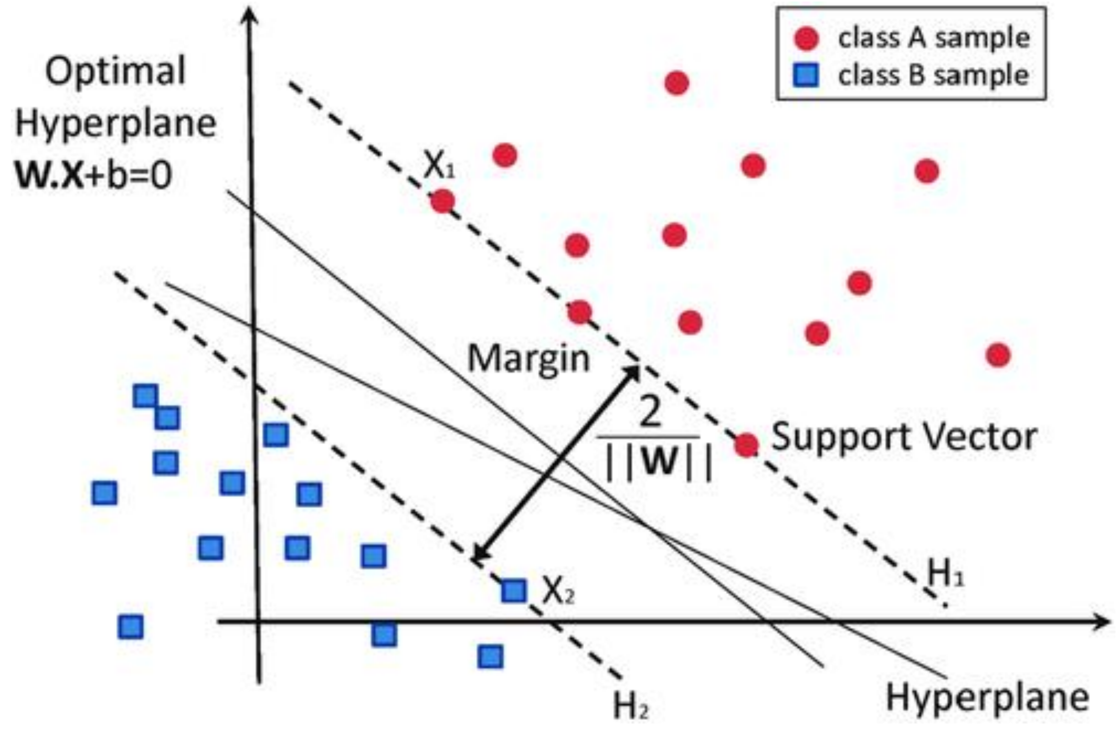
Delft University of Technology

Variational quantum support vector machine based on Γ matrix expansion and variational universal-quantum-state generator

Applied Quantum algorithm

Little reminder on SVM : Getting a quantum friendly equation



Objective: Finding the hyperplane that maximize the distance between two groups of points

1. Choice of the Kernel function :

   To separate the data points, it is necessary to choose a kernel function adapted to the situation (are the point linearly separable or not?):

2. Construction of the optimization problem :

   We want to find a **w** and $\mathbf{w_0}$ such that $\mathbf{w.x} + \mathbf{w_0}$ separates the points. It leads to the linear problem

$$F\begin{pmatrix}\omega_0\\ \alpha_1\\ \vdots\\ \alpha_M\end{pmatrix} = \begin{pmatrix}0\\ y_1\\ \vdots\\ y_M\end{pmatrix}$$

$$F = \begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & & & \\ \vdots & & K + I_M/\gamma & \\ 1 & & & \end{pmatrix}$$

Solving the linear problem through quantum gates:

- Usually, to solve that type of problem with a quantum computer, the **HHL algorithm** is used. However, it requires an exponential number of quantum gates.

$$H = \begin{pmatrix} 0 & F \\ F^\dagger & 0 \end{pmatrix}$$

  - Applying e^{iHt} is a heavy operation
  - Requires many quantum gates
  - Solution is precise but not easy to use

- **Variational Quantum Linear solver** allows prepare $|\psi_{\text{in}}\rangle$ such that $F|\psi_{\text{in}}\rangle = c|\psi_{\text{out}}\rangle$ where $|\psi_{\text{out}}\rangle$ F and are known. This however requires that F admits an unitary decomposition. This is the method used in this presentation.

The Γ matrix expansion

- First, we need F to be extended to a « Quantum-friendly size », i.e to be of size 2^N by adding trivial equations.

$$F = \begin{pmatrix} 0 & 1 & \cdots & 1 \\ 1 & & & \\ \vdots & K + I_M/\gamma & & \\ 1 & & & \end{pmatrix} \longrightarrow F = \begin{pmatrix} 0 & 1 & \cdots & 1 & 0 & \cdots & 0 \\ 1 & & & & & & \\ \vdots & K + I_M/\gamma & & & & & \\ 1 & & & & & & \\ 0 & & & & & & \\ \vdots & & & & & & \\ 0 & & & \cdots & & & 0 \end{pmatrix}$$

- We want F expressed as a linear combination of unitary Γ matrices.

$$F = \sum_{j=0}^{2^N-1} c_j \Gamma_j \qquad \text{where} \qquad \Gamma_j = \bigotimes_{\beta=1}^{N} \sigma_{\alpha}^{(\beta)}$$

Our goal hence is to compute the c_j .

- Accordingly to the decomposition, we have:

$$c_j = \mathrm{Tr}\left[\Gamma_j F\right]$$

We can expend the trace under the following equation:

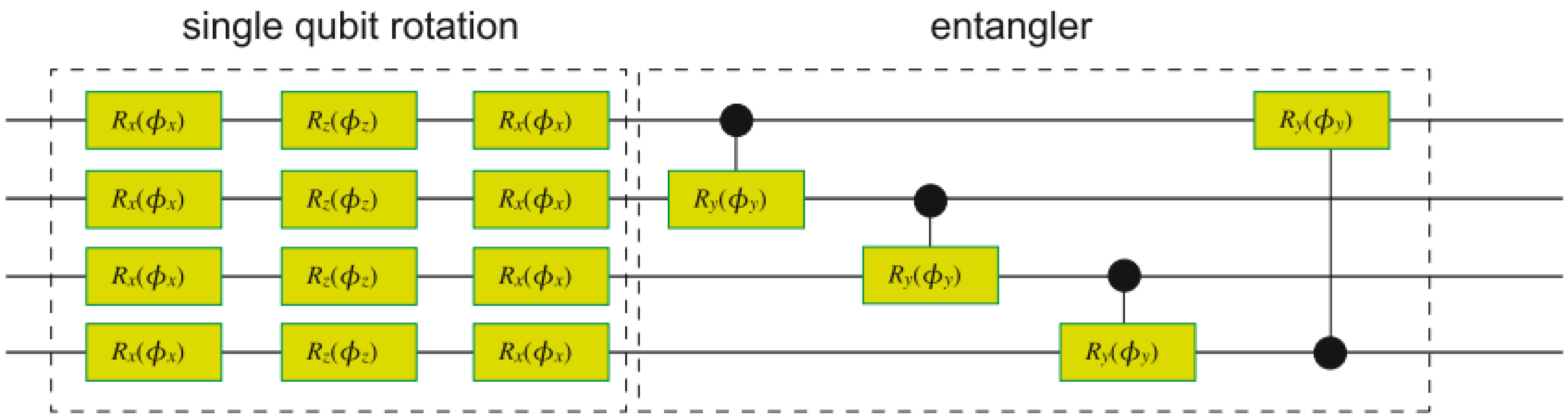
$$c_j = \sum_{q=0}^{2^N-1} \left(\Gamma_j \left| f_q \right\rangle\right)_q = \sum_{q=0}^{2^N-1} \left\langle\!\left\langle q \right| \Gamma_j \left| f_q \right\rangle$$

The vectors $\left\langle\!\left\langle q \right|$ can easily be obtained by unitary transformations :

$$U_X^{(q)} = \bigotimes_{n_i=1} \sigma_x^{(i)}$$

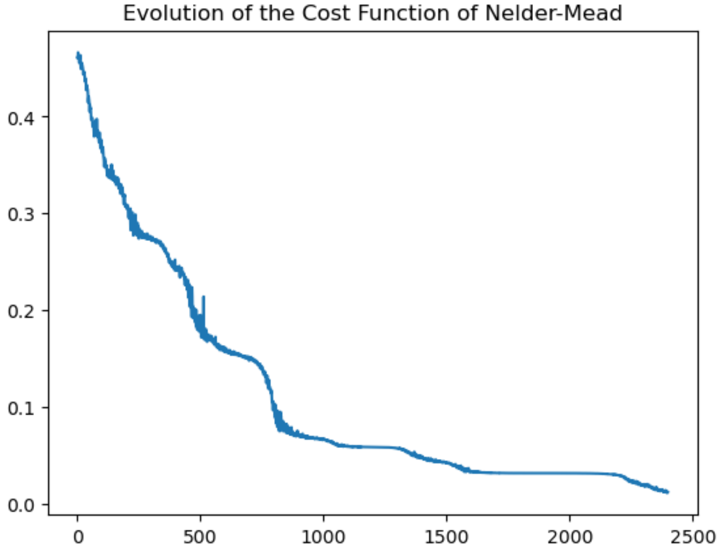
However, we also want the columns of F, $\left| f_q \right\rangle$ to be obtained by unitary transformations.
For this purpose, we need a variational universal quantum-state generator.

Variational universal quantum-state generator:

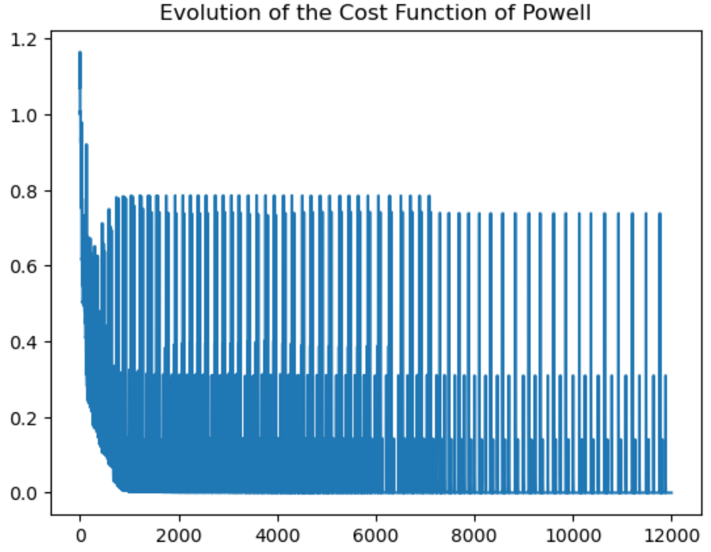


$$U(\theta_i)|0\rangle\rangle = |\psi(\theta_i)\rangle$$

$$E_{\text{cost}}(\theta_i) \equiv 1 - |\langle\psi(\theta_i)|\psi\rangle|^2$$



Final cost function value: 0.012371606017701259
Distance norm wrt original: 17.43497483590689

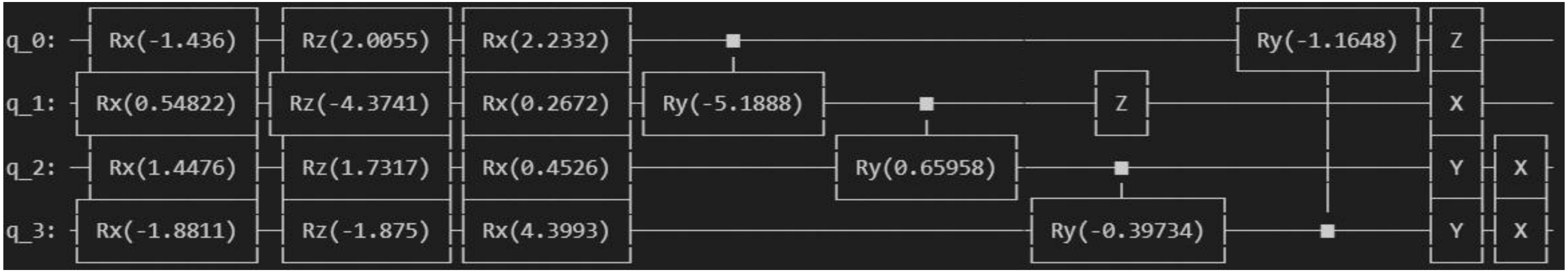


Final cost function value: 1.4855884407503004e-07
Distance norm wrt original: 0.06268952763225173

Computation of the c_j : total quantum circuit

$$c_j = \sum_{q=0}^{2^N-1} \langle\langle 0|\, U_X^{(q)}\, \Gamma_j U_{f_q} |0\rangle\rangle$$

Throughout the different computation, we achieved a mean norm distance of 13 for 5 qubits between the original matrix and its reconstruction:



Now that we have a unitary expansion of F, we can focus on the linear solving

Steepest descent method

- To solve the equation $F|\psi_{\text{in}}\rangle = c|\psi_{\text{out}}\rangle$ we use steepest descent method with update:

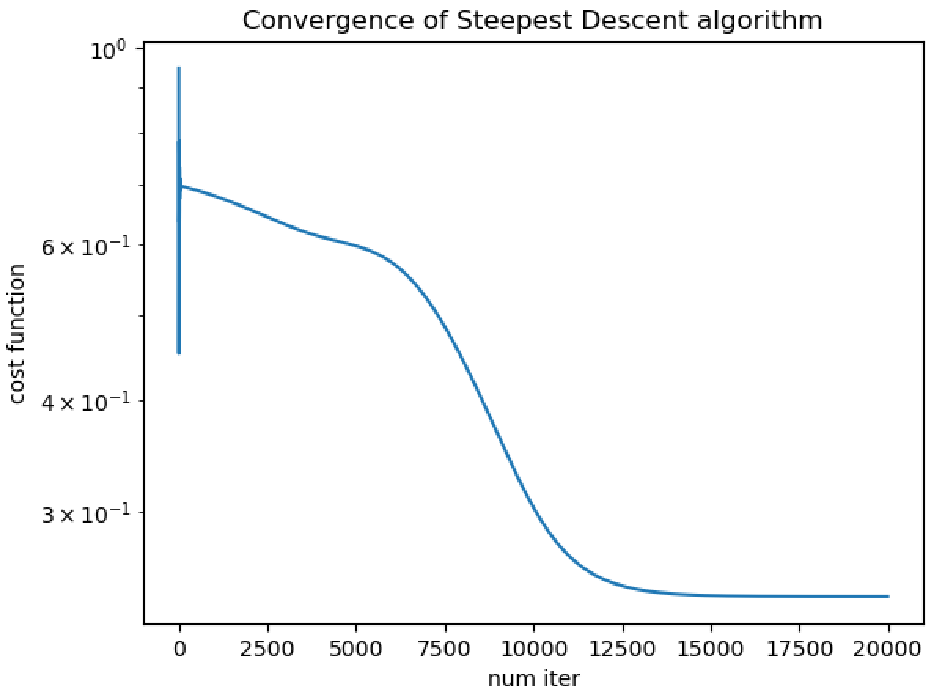
$$|\tilde{\psi}_{\text{in}}(t)\rangle \rightarrow |\tilde{\psi}_{\text{in}}(t)\rangle - \eta(t)\frac{\Delta E_{\text{cost}}}{\Delta|\tilde{\psi}_{\text{in}}(t)\rangle}\Delta|\tilde{\psi}_{\text{in}}(t)\rangle$$

And cost function :

$$E_{\text{cost}} \equiv 1 - \left|\langle\tilde{\psi}_{\text{out}}|\psi_{\text{out}}\rangle\right|^2$$

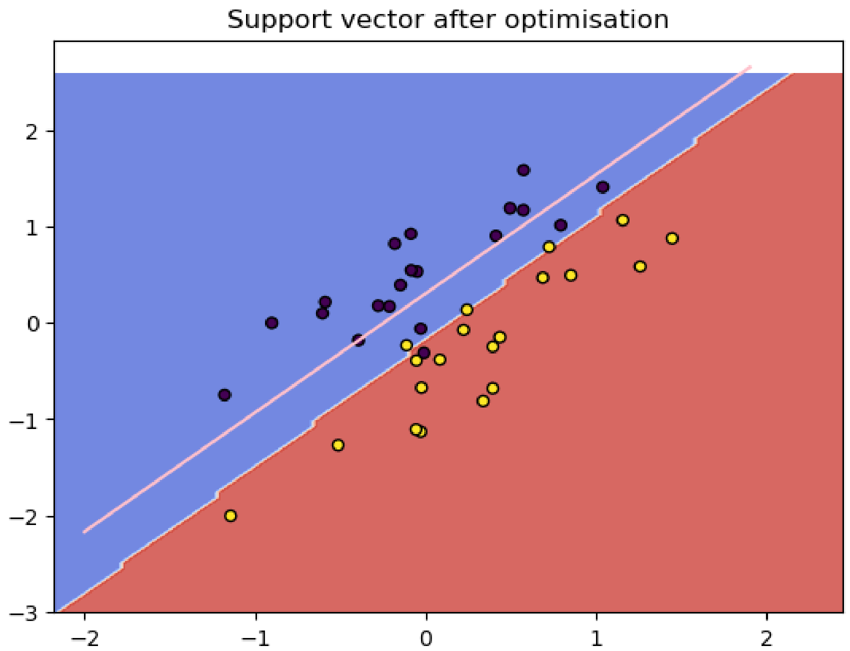
- We used an optimizer to compute the optimal values for the hyperparameters of

$$\eta(t) = \xi_1 e^{-\xi_2 t}$$

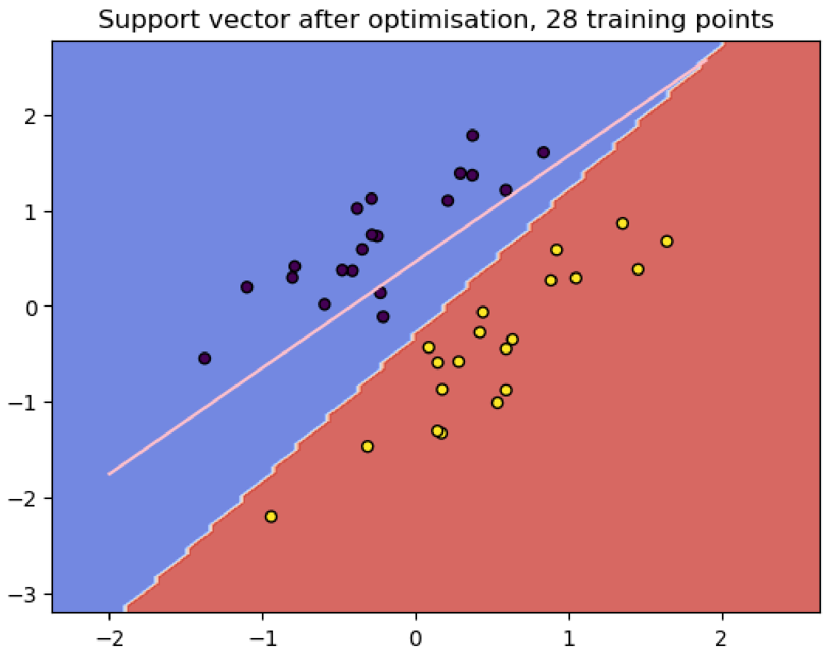


Convergence for 5 qubits

Results and comparison to classical SVM



Results for 28 points of training over 40 points in total with the Powell optimizer. Data separation = 0.3. Accuracy =88%



Results for 28 points of training over 40 points in total with the Nelder-Mead optimizer. Data separation = 0.5. Accuracy = 77%

Quantum svm hyperplane

Time comparison

	4 qubits	5 qubits	6 qubits
Quantum solver	7min30s*(N-M)* – 10min*(P)*	13m26s*(N-M)* – 20min*(P)*	crash
Classical	0.1s	3 s	

Discussion:

- The method is limited by the classical optimizer, and way slower than classical methods

- The precision is satsifying (Accuracy >=70%)
- We can maybe use Quantum Annealing for the optimization

Questions

Issues

$$\Delta E_{\text{cost}} \equiv E_{\text{cost}}\left(|\tilde{\psi}_{\text{in}}(t)\rangle + \Delta|\tilde{\psi}_{\text{in}}(t)\rangle\right) - E_{\text{cost}}\left(|\tilde{\psi}_{\text{in}}(t)\rangle\right) \simeq \frac{\Delta E_{\text{cost}}}{\Delta|\tilde{\psi}_{\text{in}}(t)\rangle}\Delta|\tilde{\psi}_{\text{in}}(t)\rangle. \quad (25)$$

We explain how to construct $|\tilde{\psi}_{\text{in}}(t)\rangle$ by a quantum circuit soon later; See Eq. (33). Then, we renew the state as

$$|\tilde{\psi}_{\text{in}}(t)\rangle \to |\tilde{\psi}_{\text{in}}(t)\rangle - \eta(t)\frac{\Delta E_{\text{cost}}}{\Delta|\tilde{\psi}_{\text{in}}(t)\rangle}\Delta|\tilde{\psi}_{\text{in}}(t)\rangle, \quad (26)$$

$$E_{\text{cost}}^{(p)}((n+1)\Delta t) \equiv 1 - \left|\langle\tilde{\psi}_{\text{in}}^{(p)}((n+1)\Delta t)\rangle|\psi_{\text{out}}\rangle\right|^2. \quad (30)$$

By running p from 1 to 2^N, we obtain a vector $E_{\text{cost}}^{(p)}((n+1)\Delta t)$, whose p-th component is $E_{\text{cost}}^{(p)}((n+1)\Delta t)$. Then, the gradient is numerically obtained as

$$|\tilde{\psi}_{\text{in}}^{(p)}((n+1)\Delta t)\rangle = |\tilde{\psi}_{\text{in}}^{(p)}(n\Delta t)\rangle + \Delta E_{\text{cost}}(n+1)$$

$$2^{N-1} < D \leq 2^N$$

$$F = \sum_{j=0}^{2^N-1} c_j \Gamma_j$$

Appendix A : Derivating the linear equation from the optimization problem

We minimize the distance d_j between a data point $\boldsymbol{x}_j$ and the hyperplane given by

$$d_j = \frac{|\boldsymbol{\omega} \cdot \boldsymbol{x}_j + \omega_0|}{|\boldsymbol{\omega}|}.$$

$$(\boldsymbol{\omega} \cdot \boldsymbol{x}_j + \omega_0) y_j \geq 1$$

$$L(\boldsymbol{\omega}, \omega_0, \boldsymbol{\alpha}) = \frac{1}{2}|\boldsymbol{\omega}|^2 - \sum_j \beta_j [(\boldsymbol{\omega} \cdot \boldsymbol{x}_j + \omega_0) y_j - 1]$$

$$(\boldsymbol{\omega} \cdot \boldsymbol{x}_j + \omega_0) y_j \geq 1 - \xi_j, \qquad \xi_j \geq 0$$

$$E_{\text{cost}} = \frac{1}{2}|\boldsymbol{\omega}|^2 + \gamma \sum_{j=1}^{M} \xi_j^2.$$

$$L(\boldsymbol{\omega}, \omega_0, \xi_i, \boldsymbol{\beta}) = \frac{1}{2}|\boldsymbol{\omega}|^2 + \gamma \sum_{j=1}^{M} \xi_j^2 - \sum_{j=1}^{M} [(\boldsymbol{\omega} \cdot \boldsymbol{x}_j + \omega_0) \beta_j y_j - (1 - \xi_i)].$$

The stationary points are determined by

$$\frac{\partial L}{\partial \boldsymbol{\omega}} = \boldsymbol{\omega} - \sum_{j=1}^{M} \beta_j y_j \boldsymbol{x}_j = 0,$$

$$\frac{\partial L}{\partial \omega_0} = -\sum_{j=1}^{M} \beta_j y_j = 0,$$

$$\frac{\partial L}{\partial \xi_j} = \gamma \xi_j - \beta_j = 0,$$

$$\frac{\partial L}{\partial \beta_j} = (\boldsymbol{\omega} \cdot \boldsymbol{x}_j + \omega_0) y_j - (1 - \xi_i) = 0.$$

We may solve these equations to determine $\boldsymbol{\omega}$ and υ_j as

$$\boldsymbol{\omega} = \sum_{j=1}^{M} \beta_j y_j \boldsymbol{x}_j,$$

from (48), and

$$\xi_j = \beta_j / \gamma$$