The objective of this question is to give you an idea about the Support Vector Machine (SVM), which is one of the most successful binary classification techniques in machine learning. Suppose we are given m labeled data points $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)$, where $\boldsymbol{x}_j \in \mathbb{R}^n$ and $y_j \in \{1, -1\}$ for $j = 1, \ldots, m$, which are linearly separable. The SVM training-problem is to find the widest *margin* that separates

$$S_1 := \{\boldsymbol{x}_j \ : \ \text{the corresponding label } y_j = 1\} \quad \text{and} \quad S_2 := \{\boldsymbol{x}_j \ : \ \text{the corresponding label } y_j = -1\}.$$

In other words, the objective is to find the hyperplane $\mathcal{H}_0 := \{\boldsymbol{x} \in \mathbb{R}^n \ : \ \boldsymbol{w}_0^\top \boldsymbol{x} + b_0 = 0\}$ with the widest *margin* that separates S_1 and S_2.

The Margin of a Separating Hyperplane

Suppose a hyperplane $\mathcal{H} := \{\boldsymbol{x} \in \mathbb{R}^n \ : \ \boldsymbol{w}^\top \boldsymbol{x} + b = 0\}$ separates S_1 and S_2, i.e.,

$$\boldsymbol{w}^\top \boldsymbol{x}_j + b > 0 \qquad \forall\, \boldsymbol{x}_j \in S_1, \tag{4a}$$

$$\boldsymbol{w}^\top \boldsymbol{x}_j + b < 0 \qquad \forall\, \boldsymbol{x}_j \in S_2. \tag{4b}$$

Then, the margin of $\mathcal{H}$ is defined as twice the minimum distance between $\mathcal{H}$ and data points, i.e.,

$$\text{margin}\,(\mathcal{H}) := 2 \min_{1 \le j \le m} \frac{|\boldsymbol{w}^\top \boldsymbol{x}_j + b|}{\|\boldsymbol{w}\|_2} = 2 \min_{1 \le j \le m} \frac{y_j \left(\boldsymbol{w}^\top \boldsymbol{x}_j + b\right)}{\|\boldsymbol{w}\|_2}.$$

Note that even if we change $\boldsymbol{w} \mapsto \alpha\, \boldsymbol{w}$ and $b \mapsto \alpha\, b$, where $\alpha > 0$, the hyperplane $\mathcal{H}$ and the binary classification rules given by (4) do not change; thus, $\text{dist}(\boldsymbol{x}_j, \mathcal{H})$ remains unchanged for every $j \in \{1, \ldots, m\}$. Therefore, without loss of generality, we can assume that $\boldsymbol{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are chosen such that

$$\min_{1 \le j \le m} \ y_j \left(\boldsymbol{w}^\top \boldsymbol{x}_j + b\right) = 1;$$

as a result, $\text{margin}\,(\mathcal{H}) = \dfrac{2}{\|\boldsymbol{w}\|_2}$.

Remark. *(a) The SVM training-problem can be formulated as the following CP:*

$$\begin{aligned} &\textit{Minimize:} && \frac{1}{2}\, \|\boldsymbol{w}\|_2^2 \\ &\textit{Subject to:} && 1 - y_j(\boldsymbol{w}^\top \boldsymbol{x}_j + b) \le 0, \quad j = 1, \ldots, m. \end{aligned} \tag{5}$$

(b) If $\mathcal{H}_0$ is the hyperplane with the widest margin that separates S_1 and S_2, then the classification rule (see Eq. (4)) on a test data-point $\boldsymbol{v} \in \mathbb{R}^n$ is given by:

- *If $\boldsymbol{w}_0^\top \boldsymbol{v} + b_0 > 0$, then $\boldsymbol{v} \in S_1$.*
- *If $\boldsymbol{w}_0^\top \boldsymbol{v} + b_0 < 0$, then $\boldsymbol{v} \in S_2$.*
- *If $\boldsymbol{w}_0^\top \boldsymbol{v} + b_0 = 0$, then a random label is assigned.*

(c) Read about a linear separation of training data-points in a higher dimensional feature space (which could be even infinite dimensional) using the kernel trick; note that this corresponds to an exact separation of training data-points in the original space with possibly a nonlinear decision boundary. ♦

(i) Show that the Lagrange dual of (5) is given by:

$$\begin{aligned} \text{Maximize:} \quad & \boldsymbol{\lambda}^\top \mathbf{1} - \frac{1}{2}\, \boldsymbol{\lambda}^\top (Y \Sigma Y)\, \boldsymbol{\lambda} \\ \text{Subject to:} \quad & \boldsymbol{\lambda} \succeq \mathbf{0}, \\ & \boldsymbol{y}^\top \boldsymbol{\lambda} = 0, \end{aligned} \tag{6}$$

where $\mathbf{1} \in \mathbb{R}^m$ is the vector of all ones, $Y := \operatorname{diag}(y_1, \ldots, y_m) \in \mathbb{S}^m$, $\Sigma := X^\top X \in \mathbb{S}^m_+$ and $X := [\boldsymbol{x}_1 \ldots \boldsymbol{x}_m] \in \mathbb{R}^{n \times m}$. Show that if $\boldsymbol{\lambda}^\star$ is an optimal solution of (6), then an optimal solution of (5) is given by

$$\boldsymbol{w}_0 = \sum_{i=1}^{m} \lambda_i^\star y_i \, \boldsymbol{x}_i.$$

(ii) Run "svm_gendata.m" to generate the training data $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)$. Solve (6) on this dataset using CVX. Plot the training error at each iteration.

(iii) Consider the Gaussian kernel given by

$$K(\boldsymbol{x}, \boldsymbol{y}) := \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2\right) \qquad \forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n. \tag{7}$$

Define the matrix Σ in (6) as $\Sigma_{i,j} := K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Solve (6) for $\sigma \in \{10^{-2}, 10^{-1}, 0.5, 10, 10^2\}$ and report the training error in each case; comment on the results.

Remark. *If $\boldsymbol{\lambda}^\star$ is the solution of* (6)*, then the kernel-based classification rule is given by the sign of the following expression:*

$$\sum_{i=1}^{m} \lambda_i^\star y_i \, K(\boldsymbol{x}_i, \boldsymbol{x}) + b.$$

But how do we set b? (Hint: the inequality constraints in (5) *corresponding to the "support vectors" are active).* ♦

(iv) The dataset "data.mat" contains raw features of handwritten digits 0-9, and "label.mat" contains the corresponding labels. Both files contain a training set and a test set. Use the Matlab command *reshape(data, 784, size(data, 3)')/255* to preprocess the training and test data.

(a) Solve the SVM model in (5) on the training data corresponding to the digits '6' and '8'. Use the trained model for classifying '6' and '8' from the test data; report the test accuracy.

(b) Perform the same experiment using the model in (6), both with and without the Gaussian kernel in (7). This time use the digits '1' and '7'.