The Support Vector Machine (SVM) is an algorithm for learning linear classifiers. Given a linearly separable training set \((x_1, y_1), \ldots, (x_m, y_m) \in \mathbb{R}^d \times \{-1, 1\}\), SVM outputs the linear classifier corresponding to the unique solution \(w^* \in \mathbb{R}^d\) of the following convex optimization problem with linear constraints

\[
\min_{w \in \mathbb{R}^d} \frac{1}{2} \|w\|^2 \\
\text{s.t.} \quad y_t w^\top x_t \geq 1 \quad t = 1, \ldots, m.
\] (1)

Geometrically, \(w^*\) corresponds to the maximum margin separating hyperplane. For every linearly separable set \((x_1, y_1), \ldots, (x_m, y_m) \in \mathbb{R}^d \times \{-1, 1\}\), the maximum margin is defined by

\[
\gamma^* = \max_{u : \|u\|=1} \min_{t=1,\ldots,m} y_t u^\top x_t
\]

and the vector \(u^*\) achieving the maximum margin is the maximum margin separator.

**Theorem 1.** For every linearly separable set \((x_1, y_1), \ldots, (x_m, y_m) \in \mathbb{R}^d \times \{-1, 1\}\), the maximum margin separator \(u^*\) satisfies \(u^* = \gamma^* w^*\), where \(w^*\) is the unique solution of (1).

**Proof.** Note that \(u^*\) is the solution of the following optimization problem

\[
\max_{u \in \mathbb{R}^d, \gamma > 0} \gamma^2 \\
\text{s.t.} \quad \|u\|^2 = 1 \\
\quad y_t u^\top x_t \geq \gamma \quad t = 1, \ldots, m.
\]

Indeed, \(u\) maximizing the margin \(\gamma\) is the same \(u\) maximizing \(\gamma^2\) because the function \(f(\gamma) = \gamma^2\), is monotone for \(\gamma > 0\). Dividing by \(\gamma > 0\) both sides of each constraint \(y_t u^\top x_t \geq \gamma\), we obtain the equivalent constraint \(y_t (u^\top x_t) / \gamma \geq 1\). Introducing \(w = u / \gamma\), and noting that \(\|w\|^2 = 1 / \gamma^2\) because of the constraint \(\|u\|^2 = 1\), we obtain the equivalent problem

\[
\min_{w \in \mathbb{R}^d, \gamma > 0} \|w\|^2 \\
\text{s.t.} \quad \gamma^2 \|w\|^2 = 1 \\
\quad y_t w^\top x_t \geq 1 \quad t = 1, \ldots, m.
\]

Now observe that the constraint \(\gamma^2 \|w\|^2 = 1\) is redundant and can be eliminated. Indeed, for all \(w \in \mathbb{R}^d\) we can find \(\gamma > 0\) such that the constraint is satisfied. Multiplying the objective function by \(1 / \gamma^2\), we obtain

\[
\min_{w \in \mathbb{R}^d} \frac{1}{\gamma^2} \|w\|^2 \\
\text{s.t.} \quad y_t w^\top x_t \geq 1 \quad t = 1, \ldots, m
\]

concluding the proof.
We have thus shown the equivalence between the problem of maximizing the margin of \( \mathbf{u} \) while keeping the norm \( \| \mathbf{u} \| \) constant, and the problem of minimizing the norm \( \| \mathbf{w} \| \) while keeping the margin of \( \mathbf{w} \) constant.

The following result helps us compute the form of the optimal solution \( \mathbf{w}^* \).

**Lemma 2** (Fritz John optimality condition). Consider the problem

\[
\min_{\mathbf{w} \in \mathbb{R}^d} f(\mathbf{w}) \\
\text{s.t.} \quad g_t(\mathbf{w}) \leq 0 \quad t = 1, \ldots, m
\]

where the functions \( f, g_1, \ldots, g_m \) are all differentiable. If \( \mathbf{w}_0 \) is an optimal solution, then there exists a nonnegative vector \( \mathbf{\alpha} \in \mathbb{R}^m \) such that

\[
\nabla f(\mathbf{w}_0) + \sum_{t \in I} \alpha_t \nabla g_t(\mathbf{w}_0) = 0
\]

where \( I = \{1 \leq t \leq m : g_t(\mathbf{w}_0) = 0\} \).

By applying the Fritz John optimality condition to the SVM objective with \( f(\mathbf{w}) = \frac{1}{2} \| \mathbf{w} \|^2 \) and \( g_t(\mathbf{w}) = 1 - y_t \mathbf{w}^\top \mathbf{x}_t \) we obtain

\[
\mathbf{w}^* - \sum_{t \in I} \alpha_t y_t \mathbf{x}_t = \mathbf{0}.
\]

Hence, the optimal solution has form

\[
\mathbf{w}^* = \sum_{t \in I} \alpha_t y_t \mathbf{x}_t
\]

where \( I \) denotes the set of training examples \((\mathbf{x}_t, y_t)\) such that \( y_t (\mathbf{w}^*)^\top \mathbf{x}_t = 1 \). These \( \mathbf{x}_t \) are called **support vectors**, and are all those training points for which the margin of \( \mathbf{w}^* \) is exactly 1. If we removed all training examples except for the support vectors, the SVM solution would not change.

We now move on to consider the case of a training set that is not linearly separable. How should we change the SVM objective? Consider the following formulation

\[
\min_{(\mathbf{w}, \xi) \in \mathbb{R}^{d+m}} \lambda \frac{1}{2} \| \mathbf{w} \|^2 + \frac{1}{m} \sum_{t=1}^m \xi_t \\
\text{s.t.} \quad y_t \mathbf{w}^\top \mathbf{x}_t \geq 1 - \xi_t \quad t = 1, \ldots, m \\
\xi_t \geq 0 \quad t = 1, \ldots, m.
\]

The components of \( \xi = (\xi_1, \ldots, \xi_m) \) are called **slack variables** and measure how much each margin constraint is violated by a potential solution \( \mathbf{w} \). The average of these violations is then added to the objective function. Finally, a regularization parameter \( \lambda > 0 \) is introduced to balance the two terms.

We now consider the constraints involving the slack variables \( \xi_t \). That is, \( \xi_t \geq 1 - y_t \mathbf{w}^\top \mathbf{x}_t \) and \( \xi_t \geq 0 \). In order to minimize each \( \xi_t \), we can set

\[
\xi_t = \begin{cases} 
1 - y_t \mathbf{w}^\top \mathbf{x}_t & \text{if } y_t \mathbf{w}^\top \mathbf{x}_t < 1 \\
0 & \text{otherwise}.
\end{cases}
\]
To see this, fix \( w \in \mathbb{R}^d \). If the constraint \( y_t w^\top x_t \geq 1 \) is satisfied by \( w \), then \( \xi_t \) can be set to zero. Otherwise, if the constraint is not satisfied by \( w \), then we set \( \xi_t \) to the smallest value such that the constraint becomes satisfied, namely \( 1 - y_t w^\top x_t \). Summarizing, \( \xi_t = [1 - y_t w^\top x_t]_+ \), which is exactly the hinge loss \( h_t(w) \) of \( w \).

The SVM problem can then be re-formulated as \( \min_{w \in \mathbb{R}^d} F(w) \), where
\[
F(w) = \frac{1}{m} \sum_{t=1}^{m} h_t(w) + \frac{\lambda}{2} \|w\|^2 .
\]

We now show that, even when the training set is not linearly separable, the solution \( w^* \) belongs to the subspace defined by linear combinations of training points multiplied by their labels.

**Theorem 3.** The minimizer \( w^* \) of \( F \) can be written as a linear combination of \( y_1 x_1, \ldots, y_m x_m \).

**Proof.** By contradiction, assume
\[
w^* = \sum_{t=1}^{m} \alpha_t y_t x_t + u
\]
where \( u \in \mathbb{R}^d \) is the component of \( w^* \) orthogonal to the subspace spanned by \( x_1, \ldots, x_m \). Therefore,
\[
y_t u^\top x_t = 0 \quad t = 1, \ldots, m.
\]
Now, let \( v = w^* - u \). First, \( \|v\|^2 \leq \|w^*\|^2 \) because in (2) we wrote \( w^* \) as a sum of two orthogonal components and we removed one of them, and so its length decreased. Second,
\[
h_t(v) = [1 - y_t v^\top x_t]_+ = [1 - y_t (w^* - u)^\top x_t]_+ = [1 - y_t (w^*)^\top x_t + y_t u^\top x_t]_+ = h_t(w^*)
\]
using (3). Therefore \( F(v) \leq F(w^*) \), contradicting the optimality of \( w^* \). Hence \( u = 0 \) and the proof is concluded. \( \square \)

Note that, as in the linearly separable case, \( w^* \) generally depends on a subset of support vectors. However, unlike the linearly separable case, these support vectors also include the training points associated with positive slack variables.

We proceed by showing how \( F \) can be minimized using Online Gradient Descent (OGD). First, observe that
\[
F(w) = \frac{1}{m} \sum_{t=1}^{m} \ell_t(w)
\]
where \( \ell_t(w) = h_t(w) + \frac{\lambda}{2} \|w\|^2 \) is a strongly convex function. Indeed, \( \frac{\lambda}{2} \|w\|^2 \) is \( \lambda \)-strongly convex, and \( h_t \) is convex (and also piecewise linear). This implies that their sum is \( \lambda \)-strongly convex. We can then apply the OGD algorithm for strongly convex functions to the set of losses \( \ell_1, \ldots, \ell_m \).

This instance of OGD, which is known as Pegasos, can be described as follows.
**Parameters:** number $T$ of rounds, regularization coefficient $\lambda > 0$

**Input:** Training set $(x_1, y_1), \ldots, (x_m, y_m) \in \mathbb{R}^d \times \{-1, 1\}$

Set $w_1 = 0$

For $t = 1, \ldots, T$

1. Draw uniformly at random an element $(x_{Z_t}, y_{Z_t})$ from the training set
2. Set $w_{t+1} = w_t - \eta_t \nabla \ell_{Z_t}(w_t)$

Output: $\overline{w} = \frac{1}{T}(w_1 + \cdots + w_T)$.

Pegasos is an example of a class of algorithms known as **stochastic gradient descent**. These are OGD-like algorithms that are run over a sequence of examples randomly drawn from the training set.

We now move on to analyze Pegasos. Let $(x_{Z_1}, y_{Z_1}), \ldots, (x_{Z_T}, y_{Z_T})$ the sequence of training set examples that were drawn at random in step 1 of the algorithm, and let $\ell_{Z_1}, \ldots, \ell_{Z_T}$ the corresponding sequence of loss functions. Namely, $\ell_{Z_t}(w) = h_{Z_t}(w) + \frac{\lambda}{2} \|w\|^2$ where $h_{Z_t}(w) = [1 - y_{Z_t} w^\top x_{Z_t}]_+$. Let $w^*$ be the optimal SVM solution,

$$w^* = \arg\min_{w \in \mathbb{R}^d} \left( \frac{1}{m} \sum_{t=1}^m h_t(w) + \frac{\lambda}{2} \|w\|^2 \right).$$

(4)

For every realization $s_1, \ldots, s_T$ of the random variables $Z_1, \ldots, Z_T$, OGD analysis for strongly convex losses immediately gives

$$\frac{1}{T} \sum_{t=1}^T \ell_{s_t}(w_t) \leq \frac{1}{T} \sum_{t=1}^T \ell_{s_t}(w^*) + \frac{G^2}{2\lambda T} (\ln T + 1)$$

(5)

where $G = \max_{t=1,\ldots,T} \|\nabla \ell_{s_t}(w_t)\|$ is also a random variable.

In order to show how this result can be used to bound $F(\overline{w})$, we use the following fact

$$\mathbb{E}[\ell_{Z_t}(w_t) | Z_1, \ldots, Z_{t-1}] = \frac{1}{m} \sum_{s=1}^m \ell_s(w_t) = F(w_t).$$

(6)

In other words, conditioned on the first $t - 1$ random draws (which determine $w_t$), the expected value of $\ell_{Z_t}(w_t)$ is equal to $F(w_t)$. We also use the fact that for every pair of random variables
We now bound the expectation but also in probability. In order to determine the coefficients of this linear combination, we fix \( \lambda > 0 \). As one can easily show by induction, \( \eta_t \) has rate \( \frac{\ln T}{\lambda T} \). When \( \eta_t = 1/(\lambda t) \), the update rule for \( \mathbf{w}_t \) takes the following simple form, 

\[
\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla \ell_t(\mathbf{w}_t) = \mathbf{w}_t + \eta_t \mathbf{v}_t - \eta_t \lambda \mathbf{w}_t = \left( 1 - \frac{1}{t} \right) \mathbf{w}_t + \frac{1}{\lambda t} \mathbf{v}_t.
\]

Let \( X = \max_{s=1, \ldots, m} \| \mathbf{x}_s \| \). Since \( \| \nabla \ell_s(\mathbf{w}_t) \| \leq \| \mathbf{v}_t \| + \lambda \| \mathbf{w}_t \| \leq X + \lambda \| \mathbf{w}_t \| \), we are left with the task of computing an upper bound for \( \| \mathbf{w}_t \| \). In order to do so, we look at the recurrence

\[
\mathbf{w}_{t+1} = \left( 1 - \frac{1}{t} \right) \mathbf{w}_t + \frac{1}{\lambda t} \mathbf{v}_t.
\]

As one can easily show by induction, \( \mathbf{w}_{t+1} \) can be written as a linear combination of \( \mathbf{v}_1, \ldots, \mathbf{v}_t \). In order to determine the coefficients of this linear combination, we fix \( s \leq t \) and observe that \( \mathbf{v}_s \) is added to the sum with coefficient \( 1/(\lambda s) \). When \( \mathbf{w}_{t+1} \), is computed, the coefficient of \( \mathbf{v}_s \) has become

\[
\frac{1}{\lambda s} \prod_{r=s+1}^{t} \left( 1 - \frac{1}{r} \right) = \frac{1}{\lambda s} \prod_{r=s+1}^{t} \frac{r-1}{r} = \frac{1}{\lambda t}.
\]
We thus obtain a simple expression for \( \mathbf{w}_{t+1} \),

\[
\mathbf{w}_{t+1} = \frac{1}{\lambda t} \sum_{s=1}^{t} \mathbf{v}_s .
\]

Because \( \mathbf{w}_{t+1} \) is an average of \( \mathbf{v}_s \) divided by \( \lambda \), we finally have \( \| \mathbf{w}_{t+1} \| \leq \frac{1}{\lambda} \max_s \| \mathbf{v}_s \| \leq \frac{1}{\lambda} X \). This allows us to conclude that \( \| \nabla \ell_t (\mathbf{w}_t) \| \leq X + \lambda \| \mathbf{w}_t \| \leq 2X \). Substituting this bound for \( G \) in (7) we get

\[
\mathbb{E}[F(\mathbf{w})] \leq F(\mathbf{w}^*) + \frac{2X^2}{\lambda T} (\ln T + 1) .
\]

Theorem 3 states that the solution \( \mathbf{w}^* \) to the SVM problem can be written as

\[
\mathbf{w}^* = \sum_{s \in S} y_s \alpha_s \mathbf{x}_s,
\]

where \( \alpha_s > 0 \) and \( S \equiv \{ t = 1, \ldots, m : h_t (\mathbf{w}^*) > 0 \} \). An important consequence of this result is that we can solve the problem (4) in a RKHS \( \mathcal{H}_K \), where the objective function \( F \) becomes

\[
F_K (g) = \frac{1}{m} \sum_{t=1}^{m} h_t (g) + \frac{\lambda}{2} \| g \|^2_K \quad g \in \mathcal{H}_K
\]

with \( h_t (g) = [1 - y_t g(x_t)]_+ \). In \( \mathcal{H}_K \), the SVM solution can therefore be written as

\[
\sum_{s \in S} y_s \alpha_s K(\mathbf{x}_s, \cdot)
\]

which is clearly an element of the RKHS

\[
\mathcal{H}_K \equiv \left\{ \sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i, \cdot) : \mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^d, \alpha_1, \ldots, \alpha_N \in \mathbb{R}, N \in \mathbb{N} \right\}
\]

As we did for the Perceptron, we can run Pegasos in the RKHS \( \mathcal{H}_K \). The gradient update in kernel Pegasos on some training example \((\mathbf{x}_{s_t}, y_{s_t})\) can be written as

\[
g_{t+1} = \left( 1 - \frac{1}{t} \right) g_t + \frac{y_{s_t}}{\lambda t} \mathbb{I}\{ h_{s_t} (g_t) > 0 \} K(\mathbf{x}_{s_t}, \cdot)
\]

where \( h_{s_t} (g_t) = [1 - y_{s_t} g_t (\mathbf{x}_{s_t})]_+ \).