Support Vector Machines – SVM

Andrei Dugăeșescu

27 October, 2024

1 Introduction

1.1 SVM for linearly separable input spaces

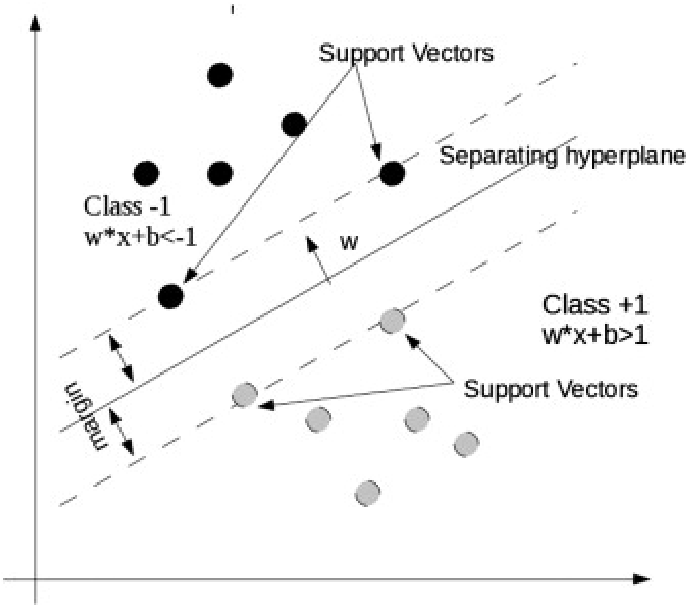


Figure 1: Linear SVM hyperplan and support vectors visualisation

Important: the target labels are not $\{0, 1\}$, but rather $\{-1, 1\}$. For the linear case, the classifier equation can be expressed as:

$$y(x) = w^T x + b$$

and therefore:

$$\begin{aligned} \mathcal{H} : w^T x + b = 0 &- \text{the separating hyperplane} \\ w^T x + b \geq 1 &\Rightarrow \text{positive class} \\ w^T x + b \leq -1 &\Rightarrow \text{negative class} \end{aligned}$$

However, it is seldom the case that the input space is linearly separable (between positive and negative examples). In this case, one widely adopted technique is to map the input space to a new space, called feature space, which has a higher dimensionality. The idea is that the higher number of dimensions from feature space allows the existence of a separation hyperplane that is linear in nature. A visual intuition is illustrated in Fig. 2.

1.2 SVM for non-linearly separable input spaces

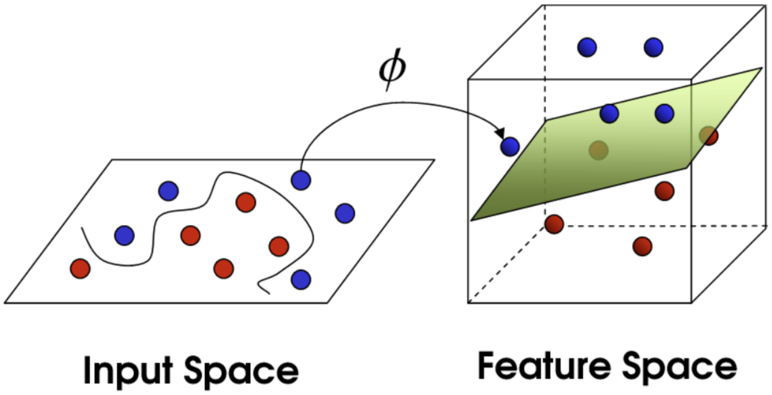


Figure 2: Projection of the Input Space using the mapping function ϕ into a linearly separable, higher dimensionality, Feature Space

Formally, the mapping can be expressed as follows:

$$x \in \mathbb{R}^D, x = (x_1, x_2, \ldots, x_D)$$

$$\phi : \mathbb{R}^D \to \mathbb{R}^M, \phi(x) \in \mathbb{R}^M$$

And therefore, the SVM formulation becomes:

$$y(x) = w^T \phi(x) + b$$

1.3 Distance measure

In Euclidean geometry, the distance from a point to a line can be computed in the following manner. For a point (x_0, y_0) and a line $ax + by + c = 0$, the distance is:

$$d = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}$$

Similarly, in the case of SVMs, the distance from one point to the hyperplane is:

$$d_{\mathcal{H}}\left(\phi(x)\right) = \frac{|w^T \phi(x) + b|}{\|w\|_2}$$

2 SVM math

2.1 Optimization problem

The goal of SVMs is to maximize the **margin** – the minimum distance to the closest points (the support vectors).

$$\mathcal{H}: \; w^T x + b = 0 - \text{the separating hyperplane}$$

$$w^* = \arg\max_w \left[\min_n d_{\mathcal{H}}\left(\phi(x_n)\right)\right]$$

Remember that each label y_n is either -1 or 1 and that

$$w^T x + b \geq 1 \Rightarrow \text{positive class}$$
$$w^T x + b \leq -1 \Rightarrow \text{negative class}$$

Thus

$$y_n \left[w^T \phi(x) + b \right] = \begin{cases} \geq 0 & \text{, correct classification} \\ < 0 & \text{, incorrect classification} \end{cases}$$

Substituting back into the equation of maximizing the minimum distance:

$$w^* = \arg\max_w \left[\min_n d_{\mathcal{H}} \left(\phi(x_n) \right) \right]$$
$$w^* = \arg\max_w \left[\min_n \frac{|w^T \phi(x_n) + b|}{\|w\|_2} \right]$$
$$w^* = \arg\max_w \left[\min_n \frac{y_n \left[w^T \phi(x_n) + b \right]}{\|w\|_2} \right] - \text{perfect separation}$$
$$w^* = \arg\max_w \frac{1}{\|w\|_2} \left[\min_n y_n \left[w^T \phi(x_n) + b \right] \right] - \text{closest distance to } \mathcal{H}$$

The margin is the distance to the closest point from the training data, which is called a support vector. In order to simplify the equation above, the distance to the support vectors will be normalized (i.e. scaled) such that it has unit value (scaling all the values does not change the minimum/maximum). Formally, this is expressed as follows:

$$w^* = \arg\max_w \frac{1}{\|w\|_2} \left[\min_n y_n \left[w^T \phi(x_n) + b \right] \right]$$

$$\text{Let } \min_n y_n \left[w^T \phi(x_n) + b \right] = 1$$

$$w \to cw$$
$$b \to cb$$
$$(cw)^T \phi(x_n) + (cb) = c \left(w^T \phi(x_n) + b \right) = 0$$

2.2 Primal form of SVM

From the equations above regarding the optimal set of weights, the objective is:

$$w^* = \arg\max_w \frac{1}{\|w\|_2}$$

$$\text{s.t. } \min_n y_n \left[w^T \phi(x_n) + b \right] = 1$$

It is hard to solve a maximization problem in this case, but is much easier to solve a minimization one. Converting the problem above we obtain the *Primal Form of SVM.*

$$\min_w \frac{1}{2} \|w\|_2^2$$

$$\text{s.t. } \min_n y_n \left[w^T \phi(x_n) + b \right] \geq 1 \; \forall n$$

If x is on the decision boundary:

$$\begin{aligned}
&w^T x + b = 0 \\
&w^T \left(x + k\frac{w}{\|w\|} \right) + b = 1 \\
&w^T x + k\frac{w^T w}{\|w\|} + b = 1 \\
&w^T x + b + k\frac{w^T w}{\|w\|} = 1
\end{aligned}$$

Therefore:

$$\begin{aligned}
&k\frac{w^T w}{\|w\|} = 1 \\
&k\frac{\|w\|^2}{\|w\|} = 1 \\
&k\,\|w\| = 1 \\
&k = \frac{1}{\|w\|}
\end{aligned}$$

The distance from the separation hyperplane to the support vector is $\frac{1}{\|w\|}$ so the size of the margin will be twice that size: $\frac{2}{\|w\|}$. That being said, as we will see shortly, it is easier to maximize $\frac{2}{\|w\|^2}$ which is equivalent to minimizing $\frac{1}{2}\|w\|^2$.

2.3 New primal form of SVM

Unfortunately, the previous formulation for this optimization problem assumes that the data is perfectly separable. In order to account for misclassified samples, we can introduce a slack variable ξ_n that will allow for a number of incorrect classification to occur within a set threshold. That being said, the new primal form for SVM is:

$$\min_{w,b,\{\xi_n\}} \frac{1}{2}\|w\|_2^2 + C\sum_n \xi_n$$

$$\text{s.t. } \min_n y_n \left[w^T \phi(x_n) + b \right] \geq 1 - \xi_n \;\forall n$$

$$\xi_n \geq 0 \;\forall n$$

where C is a hyperparameter that controls how much freedom the model has

$$C = \begin{cases} 0 & \text{, linear classifier (least complex, might underfit)} \\ \dots & \\ \text{inf} & \text{, most complex decision boundary (will overfit)} \end{cases}$$

This is a convex quadratic optimization problem because the objective function itself is quadratic in w whilst the constraints are linear in w (and ξ_n).

2.4 The dual formulation of SVM

The last big problem is that $\phi(x)$ is very hard or even impossible to compute and therefore we need to find a way of solving the minimization problem from before without having to compute $\phi(x)$ – the projection of each point into the feature space. This is accomplished through the use of kernels.

2.4.1 Kernel Trick

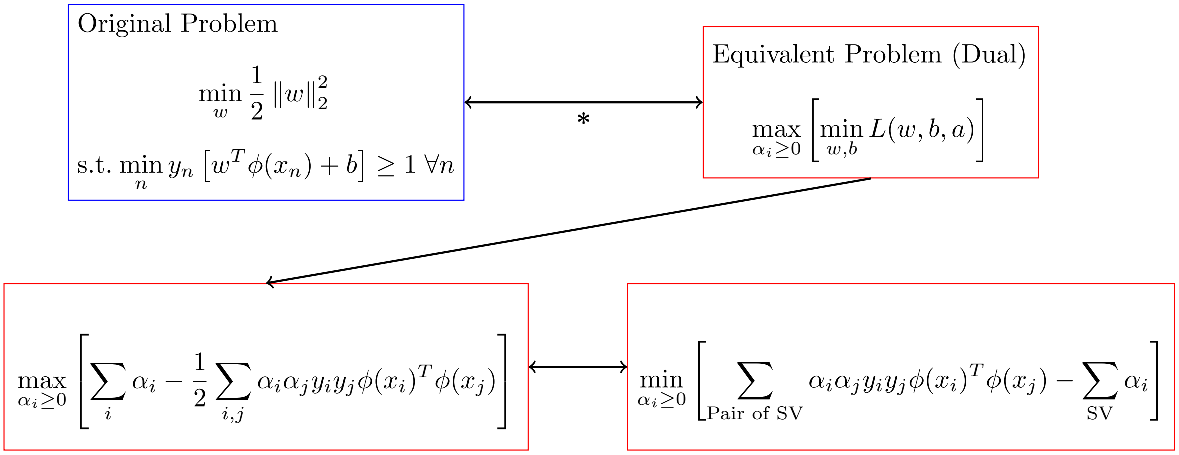
Kernel definition. A kernel is a function that takes as input vectors from the original input space and outputs the dot product of the vectors from the feature space. Formally, a kernel is defined as:

$$
\begin{gathered}
x, z \in \mathbb{R}^D,\ \phi : \mathbb{R}^D \to \mathbb{R}^M \\
k(x, z) = \langle \phi(x), \phi(z) \rangle
\end{gathered}
$$

One of the great aspects of kernels is that they completely circumvent the need to project the points from the input space into the feature space. With kernels, all we need to compute is how each point compares to each other data point as if we were to apply the non-linear transformation. The dot product between all the pair of points is then stored in a matrix called the Gram Matrix/Kernel Matrix, which can be expressed as follows:

$$
\phi\phi^T = \begin{bmatrix} \phi(x_1)^T\phi(x_1) & \phi(x_1)^T\phi(x_2) & . & \phi(x_1)^T\phi(x_n) \\ \phi(x_2)^T\phi(x_1) & \phi(x_2)^T\phi(x_2) & . & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \phi(x_n)^T\phi(x_1) & \phi(x_n)^T\phi(x_2) & . & \phi(x_n)^T\phi(x_n) \end{bmatrix} = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & . & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & . & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & . & k(x_n, x_n) \end{bmatrix}
$$

2.4.2 Determining the Dual Form for SVM



Original Problem

$$\min_w \frac{1}{2} \|w\|_2^2$$

$$\text{s.t. } \min_n y_n \left[w^T\phi(x_n) + b\right] \geq 1\ \forall n$$

*

Equivalent Problem (Dual)

$$\max_{\alpha_i \geq 0} \left[\min_{w,b} L(w, b, a)\right]$$

$$\max_{\alpha_i \geq 0} \left[\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \phi(x_i)^T\phi(x_j)\right]$$

$$\min_{\alpha_i \geq 0} \left[\sum_{\text{Pair of SV}} \alpha_i\alpha_j y_i y_j \phi(x_i)^T\phi(x_j) - \sum_{\text{SV}} \alpha_i\right]$$

Lagrangian

$$L(w, b, \alpha) = \frac{1}{2}\|w\|_2^2 - \sum_n \alpha_n \left(y_n \left[w^T\phi(x_n) + b\right] - 1\right)$$

Stationarity

$$\frac{\partial L}{\partial w} = w - \sum_n \alpha_n y_n \phi(x_n) = 0$$

$$\frac{\partial L}{\partial b} = \sum_n \alpha_n y_n = 0$$

The steps involved in determining the dual formulation for SVMs, so that they no longer require the computation of $\phi(x)$ are the following:

1. Obtain the Primal Form (already done)

   $\min_x f(x)$
   $g_i(x) \leq 0$

2. Determine the Lagrangian

   $L(x, \{\lambda_i\}) = f(x) + \sum_{i=1}^{n} \lambda_i g_i(x)$
   $\lambda_i(x) \geq 0$

3. Solve for the primal variables (calculating the Lagrange multipliers)

   $\frac{\partial L}{\partial x} = 0$
   $x = h(\{\lambda_i\})$

4. Substitute back in the Lagrangian

   $L(\{\lambda_i\}) = f(h(\{\lambda_i\})) + \sum_{i=1}^{n} \lambda_i g_i(h(\{\lambda_i\})$

5. Rewrite the constraints

   $\max_{\{\lambda_i\} \geq 0} \min_x L(x, \{\lambda_i\})$

1. Obtaining the Primal Form

$$\min_{w,b,\{\xi_n\}} \frac{1}{2} \|w\|_2^2 + C \sum_n \xi_n$$

$$\text{s.t. } \min_n y_n \left[w^T \phi(x_n) + b\right] \geq 1 - \xi_n \; \forall n$$

$$\xi_n \geq 0 \, \forall n$$

2. Determining the Lagrangian

$$L(w, b, \{\xi_n\}, \{\lambda_n\}, \{\alpha_n\}) = \underbrace{\left[\frac{1}{2} \|w\|_2^2 + C \sum_n \xi_n\right]}_{\text{the objective function}} + \underbrace{\sum_n \left\{\alpha_n \left[1 - \xi_n - y_n \left[w^T \phi(x_n) + b\right]\right]\right\}}_{\text{first set of constraints}} + \underbrace{\sum_n \lambda_n(-\xi_n)}_{\text{second set of constraints}}$$

3. Solve for primal variables

$$
\begin{aligned}
\frac{\partial L}{\partial w} &= w - \sum_n \alpha_n y_n \phi(x_n) = 0 & \qquad\qquad w &= \sum_n \alpha_n y_n \phi(x_n) \\
\frac{\partial L}{\partial b} &= \sum_n \alpha_n y_n = 0 & \sum_n \alpha_n y_n &= 0 \\
\frac{\partial L}{\partial \xi_n} &= C - \alpha_n - \lambda_n = 0 & C - \alpha_n - \lambda_n &= 0
\end{aligned}
$$

4. Substitute back in the Lagrangian

$$
\begin{aligned}
L(w, b, \{\xi_n\}, \{\lambda_n\}, \{\alpha_n\}) = &\left[\frac{1}{2}\left\|w\right\|_2^2 + C\sum_n \xi_n\right] + \\
&\sum_n \left\{\alpha_n \left[1 - \xi_n - y_n \left[w^T\phi(x_n) + b\right]\right]\right\} + \\
&\sum_n \lambda_n(-\xi_n) \\
L(w, b, \{\xi_n\}, \{\lambda_n\}, \{\alpha_n\}) = &\left[\frac{1}{2}\left(\sum_m \alpha_m y_m \phi(x_m)\right)^T \left(\sum_n \alpha_n y_n \phi(x_n)\right) + C\sum_n \xi_n\right] + \\
&\sum_n \left\{\alpha_n \left[1 - \xi_n - y_n \left[\left(\sum_m \alpha_m y_m \phi(x_m)\right)^T \phi(x_n) + b\right]\right]\right\} + \\
&\sum_n \lambda_n(-\xi_n) \\
L(w, b, \{\xi_n\}, \{\lambda_n\}, \{\alpha_n\}) = &\left[\frac{1}{2}\sum_n \alpha_m \alpha_n y_m y_n \phi(x_m)^T \phi(x_n) + C\sum_n \xi_n\right] + \\
&\sum_n \alpha_n - \sum_n \alpha_n \xi_n + \sum_n \lambda_n(-\xi_n) \\
L(w, b, \{\xi_n\}, \{\lambda_n\}, \{\alpha_n\}) = &\sum_n \alpha_n + \frac{1}{2}\sum_n \alpha_m \alpha_n y_m y_n \phi(x_m)^T \phi(x_n) + \sum_n \alpha_n(C - \xi_n - \lambda_n) \\
L(w, b, \{\xi_n\}, \{\lambda_n\}, \{\alpha_n\}) = &\sum_n \alpha_n + \frac{1}{2}\sum_n \alpha_m \alpha_n y_m y_n \phi(x_m)^T \phi(x_n) \\
L(\{\alpha_n\}) = &\sum_n \alpha_n + \frac{1}{2}\sum_n \alpha_m \alpha_n y_m y_n \phi(x_m)^T \phi(x_n)
\end{aligned}
$$

5. Rewrite the constraints

$$\max_{\{\alpha_i\}} L(\{\alpha_n\}) = \sum_n \alpha_n + \frac{1}{2}\sum_n \alpha_m \alpha_n y_m y_n \phi(x_m)^T \phi(x_n)$$

$$\alpha_n, \lambda_n \geq 0 \, \forall n$$

$$\sum_n \alpha_n y_n = 0$$

$$C - \alpha_n - \lambda_n = 0$$

2.4.3 Making predictions

Considering that the SVM is trained – the Lagrange multipliers are computed, the prediction can be obtained as follows:

$$\begin{aligned} y(x) &= w^T\phi(x) + b \\ &= \left(\sum_n \alpha_n y_n \phi(x)\right)^T \phi(x) + b \\ &= \sum_n \alpha_n y_n \phi(x)^T \phi(x) + b \\ &= \underbrace{\sum_n \alpha_n y_n k(x_n, x) + b}_{\text{no } \phi \text{ terms}} \end{aligned}$$

$$y(x) = \sum_n \alpha_n y_n k(x_n, x) + b$$

where

$$b = \frac{1}{N_S} \sum_{i \in S} \left(y_i - \sum_{j \in S} a_j y_j k(x_i, x_j) \right)$$

S being the set of all the support vectors – those samples x_i with an associated Lagrange multiplier $a_i = 0$.