Simplified Mathematics behind Support Vector Machines (SVM)

Abdallah Benkadja

1 Introduction

The SVM approach originally stems from the work of Cortes et Vapnik [1]. It aims to solve a binary classification problem. The mathematical formulation consists in identifying the decision function : $f : \mathbb{R}^p \to \{+1, -1\}$.

The input value consists of a data vector x_i with p dimensions. The output value is binary, either $+1$ for belonging to a specific class, or -1 for not belonging, due to the nature of binary classification. The meaning of these classes may vary depending on the problem studied, for example : True, False, malignant tumor, benign tumor, etc.

Since SVM is a solution to a binary classification problem, we can consider the training dataset as a set of vectors $x_i \in \mathbb{R}^p$ along with their class values $y_i \in \{+1, -1\}$. Thus, the training dataset corresponds to the set $\{(x_1, y_1), ..., (x_n, y_n)\}$.

The goal of SVM is to determine the parameters of the hyperplane that separates the data into two classes ($+1$ and -1). For example, in a two-dimensional space $\mathbb{R}^2$, as illustrated in Figure 1, the hyperplane is a line that separates the two groups of samples. In a three-dimensional space $\mathbb{R}^3$, the hyperplane is a two-dimensional plane, as illustrated in Figure 2. More generally, in a p-dimensional space ($\mathbb{R}^p$), the hyperplane is an affine subspace of dimension $p - 1$ that separates the two groups of data that we seek to discriminate.

A visual representation of the separating hyperplane is presented in Figures 1 and 2, where

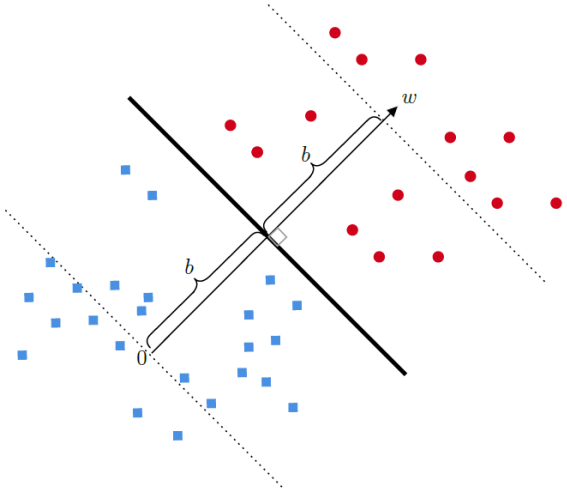

FIGURE 1 – The projection onto the separating hyperplane

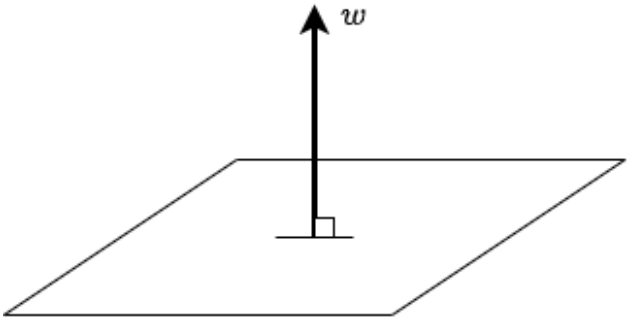

FIGURE 2 – The separating hyperplane in $\mathbb{R}^3$

the vector $w \in \mathbb{R}^p$ represents the normal to the hyperplane and b is the intercept. The parameters searched to define the hyperplane are therefore $w \in \mathbb{R}^p$ and $b \in \mathbb{R}$.

The dot product $\langle x_i, x_j \rangle$ is a measure of the similarity between the vectors x_i and x_j, widely used in key concepts of analytic geometry such as projection and orthogonality. In the context of SVM classification, the dot product is essential in the decision function $f(x)$, which evaluates the similarity between a data vector $x \in \mathbb{R}^p$ and the normal vector w of the separating hyperplane, shifted by the intercept b.

$$\begin{aligned} f : \mathbb{R}^p &\longrightarrow \mathbb{R} \\ x &\longmapsto \langle w, x \rangle + b \end{aligned}$$

In other words, if the value of the decision function $f(x)$ is large and positive, it means that

the vector x is in the direction of the normal vector w of the hyperplane, and thus on the side of the class $y = +1$. Conversely, if the value of $f(x)$ is large and negative, it indicates that the vector x is in the direction opposite to that of w, and thus on the side of the class $y = -1$. When $f(x)$ is equal to 0, it means that the vector x is on the boundary of the hyperplane, which is the border between the two classes.

The boundary between the two classes is established by the parameter b, which determines the shift of the hyperplane relative to the origin. The goal of SVM classification is therefore to find the optimal values of the parameters w and b to define the hyperplane that effectively separates the two classes in the data space.

The data vectors x_i that belong to the class $y_i = +1$ must satisfy the following equation (1) :

$$\langle w, x_i \rangle + b \geq 0 \text{ for each } y_i = +1 \tag{1}$$

On the other hand, data vectors belonging to the class $y_i = -1$ must conform to the following equation (2) :

$$\langle w, x_i \rangle + b < 0 \text{ for each } y_i = -1 \tag{2}$$

The two equations (1) and (2) can be combined :

$$y_i(\langle w, x_i \rangle + b) \geq 0 \tag{3}$$

The basic constraint of the SVM approach, as expressed in equation (3), must be satisfied by the training dataset. However, this constraint alone does not lead to a unique solution. Intuitively, this can lead to an infinite number of solutions, as illustrated in figure 3. The mathematical formulation with a single constraint as presented in equation (3) leads to infinite possible solutions ; hence the interest of the notion of margin, which adds an additional constraint allowing the system to converge to a unique solution.

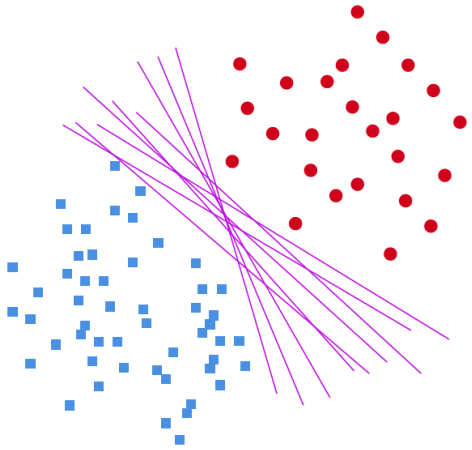

FIGURE 3 – Infinite possible solutions for the separating hyperplane.

2 The SVM margin

For a linearly separable training dataset $(x_1, y_1), ..., (x_n, y_n)$, the constraint in equation (3) alone does not ensure convergence to a unique solution for obtaining the parameters w and b of the separating hyperplane. This is why the concept of margin is introduced to impose an additional constraint related to the maximization of this margin, in order to determine a unique solution.

The concept of margin is illustrated in Figure 4. It represents the distance between the vector x_a and the sought hyperplane, given :

— x_a is the point in the training dataset that is closest to the sought hyperplane.

— x'_a is the orthogonal projection of point x_a onto the hyperplane.

— r is the orthogonal distance between the hyperplane and the point x_a.

For data vectors belonging to class $y_i = +1$, their distance from the hyperplane in the direction of w must be greater than r. Similarly, for data vectors belonging to class $y_i = -1$, their distance from the hyperplane in the negative direction of w must be greater than r. In summary, for all data vectors x_i, the following constraint must be satisfied :

$$y_i(\langle w, x_i \rangle + b) \geq r \tag{4}$$

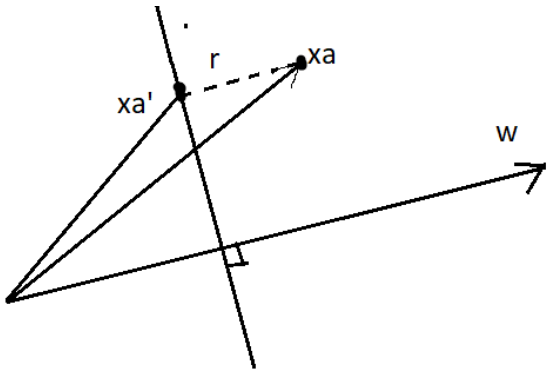

FIGURE 4 – The SVM marin.

Now, the optimization problem boils down to solving :

$$\underset{w,b,r}{\text{maximize}} \quad r \tag{5}$$

$$\text{subject to :} \quad y_i(\langle w, x_i \rangle + b) \geq r, \tag{6}$$

$$r > 0, \tag{7}$$

$$\|w\| = 1 \tag{8}$$

If we relax the constraint $\|w\| = 1$ and choose $r = 1$, it will have no impact on the optimization function. Choosing this scale of $r = 1$ ensures that the value of $\langle w, x_a \rangle + b = 1$ for the vector x_a closest to the sought-after hyperplane. Thus, the orthogonal projection of x_a, i.e., x'_a, will be exactly on the margin.

$$\langle w, x'_a \rangle + b = 0 \tag{9}$$

Knowing the length of w, we can use the scaling factor r to calculate the absolute distance between x_a and x'_a, which is the orthogonal projection of x_a onto the hyperplane. By adding the vectors, as shown in Figure 4, we can deduce the following equation :

$$x_a = x'_a + r\frac{w}{\|w\|} \Rightarrow x'_a = x_a - r\frac{w}{\|w\|} \tag{10}$$

By substituting the term x'_a from equation (10) into equation (9), we obtain :

$$\langle w, x_a - r\frac{w}{\|w\|} \rangle + b = 0 \Rightarrow \langle w, x_a \rangle + b - r\frac{\langle w, w \rangle}{\|w\|} = 0 \tag{11}$$

Since x_a is the closest vector to the sought-after hyperplane, it implies that $\langle w, x_a \rangle + b = 1$ (since $r = 1$). After substitution in equation (11), the margin to be maximized takes the following form

$$r = \frac{1}{\|w\|}$$

The optimization problem for maximizing the margin while relaxing the constraint $\|w\| = 1$ takes the following form :

$$\underset{w,b}{\text{maximize}} \quad \frac{1}{\|w\|} \tag{12}$$

$$\text{subject to :} \quad y_i(\langle w, x_i \rangle + b) \geq 1 \tag{13}$$

The optimization problem related to the SVM approach with the $\frac{1}{2}$ factor for optimization purposes through gradient descent can be formulated as follows :

$$\underset{w,b}{\text{minimize}} \quad \frac{1}{2}\|w\|^2$$

$$\text{subject to :} \quad y_i(\langle w, x_i \rangle + b) \geq 1$$

3 Soft Margin SVM

The model that allows for some classification errors is known as the *Soft-margin SVM*. This concept is illustrated in Figure 5. The key idea is to introduce a "slack" or "error" variable ζ_i associated with each data pair (x_i, y_i), allowing the vectors x_i to not necessarily belong strictly to their class y_i.

By adding the constraint related to the tolerance margin, the optimization problem takes

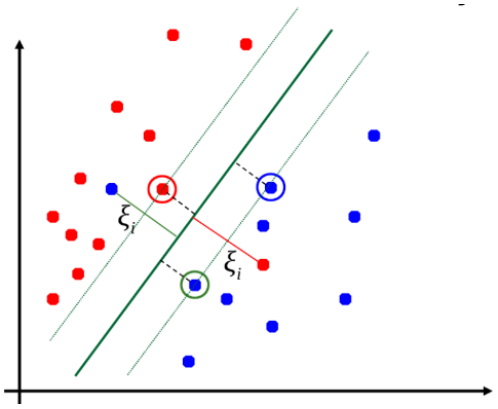

FIGURE 5 – Soft Margin SVM.

the following formula for Soft margin SVM :

$$\underset{w,b,\zeta}{\text{minimize}} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\zeta_i \tag{14}$$

$$\text{subject to :} \quad y_i(\langle w, x_i\rangle + b) \geq 1 - \zeta_i, \tag{15}$$

$$\zeta_i \geq 0 \tag{16}$$

n represents the total number of training vectors. The parameter C is a hyperparameter that controls the trade-off between the tolerance margin and incorrect classification of training data. A larger value of C leads to a more rigorous classification of data, minimizing classification errors but possibly resulting in a narrower margin. A smaller value of C allows for a wider margin, but tolerates more classification errors.

In other words, a higher C parameter places more weight on classification errors, which can lead to a model that is more closely fitted to the training data, while a lower C parameter places less weight on classification errors, allowing for a wider margin and greater tolerance for errors. The appropriate choice of C value depends on the specific problem and characteristics of the training data and may require experimentation to find the best value.

4 Lagrange duality

To solve optimization problems with constraints, there are various numerical approaches, among which optimization using Lagrange multipliers is one of the most commonly used.

The primal optimization problem of SVM, as illustrated in equation (14) for Soft Margin SVM, has primary variables of w, b, and ζ. The objective function of the associated dual to the primal problem of equation (14) for Soft Margin SVM takes the following form :

$$\mathcal{L}(w, b, \zeta, \alpha, \gamma) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\zeta_i - \sum_{i=1}^{n}\alpha_i(y_i(\langle w, x_i\rangle + b) - 1 + \zeta_i) - \sum_{i=1}^{n}\gamma_i\zeta_i \tag{17}$$

Given that α_i and γ_i are the Lagrange multipliers.

To find the values of the primal variables that maximize the Lagrangian, we look for the points where the partial derivatives with respect to w, b, and ζ are zero. By taking the derivative of equation (17) with respect to these variables, we obtain :

$$\frac{\partial\mathcal{L}}{\partial w} = w^T - \sum_{i=1}^{n}\alpha_i y_i x_i^T \tag{18}$$

$$\frac{\partial\mathcal{L}}{\partial b} = \sum_{i=0}^{n}\alpha_i y_i \tag{19}$$

$$\frac{\partial\mathcal{L}}{\partial \zeta_i} = C - \alpha_i - \gamma_i \tag{20}$$

By setting the derivatives of equations (18) and (19) to zero, we obtain the following equations :

$$\frac{\partial\mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n}\alpha_i y_i x_i \tag{21}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} y_i \alpha_i = 0 \tag{22}$$

By replacing equation (21) in the dual function (17), we obtain the following dual :

$$\begin{aligned} \mathfrak{D}(\zeta, \alpha, \gamma) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle + C \sum_{i=1}^{n} \zeta_i - \sum_{i=1}^{n} y_i \alpha_i \langle \sum_{j=1}^{n} y_j \alpha_j x_j, x_i \rangle \\ -b \sum_{i=1}^{n} y_i \alpha_i + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i \zeta_i - \sum_{i=1}^{n} \gamma_i \zeta_i \end{aligned} \tag{23}$$

$$\begin{aligned} \Rightarrow \mathfrak{D}(\zeta, \alpha, \gamma) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle - \sum_{i=1}^{n} y_i \alpha_i \langle \sum_{j=1}^{n} y_j \alpha_j x_j, x_i \rangle + C \sum_{i=1}^{n} \zeta_i \\ -b \sum_{i=1}^{n} y_i \alpha_i + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i \zeta_i - \sum_{i=1}^{n} \gamma_i \zeta_i \end{aligned} \tag{24}$$

$$\begin{aligned} \Rightarrow \mathfrak{D}(\zeta, \alpha, \gamma) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle - \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle + \sum_{i=1}^{n} \alpha_i \\ - \sum_{i=1}^{n} \alpha_i \zeta_i - \sum_{i=1}^{n} \gamma_i \zeta_i + C \sum_{i=1}^{n} \zeta_i - b \sum_{i=1}^{n} y_i \alpha_i \end{aligned} \tag{25}$$

Furthermore, we have from equation (22) : $\sum_{i=1}^{n} y_i \alpha_i = 0$. By substituting equations (18) and (19) into equation (25), the Lagrangian dual takes the following form :

$$\mathfrak{D}(\zeta, \alpha, \gamma) = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle + \sum_{i=1}^{n} \alpha_i + \sum_{i=1}^{n} (C - \alpha_i - \gamma_i) \zeta_i \tag{26}$$

5 Kernels

It is true that SVM is initially designed to deal with linearly separable cases. However, it is possible to handle non-linearly separable cases by introducing the concept of kernels.

It should be noted that in equation (26), the only dot product that appears is between the vectors x_i and x_j. Therefore, if we consider a function $\phi(x_i)$ to represent x_i and thus apply a (linear or non-linear) transformation on the x_i, the only change in the SVM objective function will be to replace the dot product function $\langle x_i, x_j \rangle$ by another function k. Thus, the concept of kernel allows for indirectly applying a transformation on $(x_1, y_1), ..., (x_n, y_n)$ through a function k that substitutes for the dot product $\langle x_i, x_j \rangle$. This option allows the SVM to be more flexible and to find non-linear decision boundaries in spaces transformed by $\phi(x_i)$.

In other words, using SVM, it is possible to construct nonlinear classifiers for the data $x_1, x_2, ..., x_n$ by replacing the dot product with a kernel function $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$, where $\phi(x_i)$ can be a nonlinear transformation function. The idea behind the kernel function is that it can be computed more efficiently than the dot product $\langle \phi(x_i), \phi(x_j) \rangle$. The kernel function k is used as a parameter to train the model in a more flexible way, allowing SVM to construct nonlinear decision boundaries in the transformed space.

The most commonly used kernel functions are :

— linear : $k(x_i, x_j) = x_i^\top x_j + c$.

— polynomial : $k(x_i, x_j) = (\gamma x_i^\top x_j + c)^d$.

— Radial Basis Function (RBF) : $k(x_i, x_j) = e^{(-\gamma \|x_i - x_j\|^2)}$.

where c is a hyperparameter that applies to both linear and polynomial kernels, d is a hyperparameter specific to the polynomial kernel and corresponds to the degree of the polynomial. γ is a hyperparameter in the RBF kernel function that controls the shape and scope of the kernel function. More specifically, γ is a regularization parameter that determines the "width" of the radial basis functions around the training examples. A larger value of γ leads

to narrower and more "pointed" radial basis functions, while a smaller value of γ leads to wider and more "smooth" radial basis functions.

Références

[1] C. Cortes et V. Vapnik : Support-vector networks. *Machine learning*, 20(3):273–297, 1995.