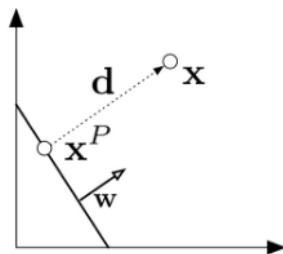


SVM notes

Matteo Lugli

November 16, 2023

1 Margins



Considering the above figure, $x^p = x - d$ and $d = w\alpha$. But x^p is also a point on the hyperplane, so $w^t x^p + b = 0$. By substituting we get $w^t(x - \alpha w) + b = 0$.

$$w^t - w^t \alpha w + b = 0$$

$$\frac{w^t x + b}{w^t w} = \alpha$$

$$d = \frac{w^t x + b}{w^t w} w$$

Now we want to calculate the magnitude of the distance d , which is

$$\begin{aligned} \|d\|_2 &= \sqrt{d^t d} = \sqrt{\alpha^2 w^t w} = \sqrt{\alpha^2 w^t w} = \alpha \sqrt{w^t w} \\ &= \frac{w^t x + b}{w^t w} \sqrt{w^t w} = \frac{w^t x + b}{\sqrt{w^t w}} = \frac{w^t x + b}{\|w\|_2} \end{aligned}$$

So the margin is defined as follows:

$$\gamma(w, b) = \min_{x \in D} \left(\frac{w^t x + b}{\|w\|_2} \right) \quad (1)$$

In the SVM problem we want to find the separating hyperplane that maximizes the margin:

$$\begin{aligned} \max_{w, b} (\gamma(w, b)) &= \max_{w, b} \left(\min_{x \in D} \left(\frac{w^t x + b}{\|w\|_2} \right) \right) \\ &= \max_{w, b} \left(\frac{1}{\|w\|_2} \min_{x \in D} (w^t x + b) \right) \end{aligned}$$

If we consider the hyperplane $w^t x + b$ it is scale invariant, meaning that we can multiply w and b for whatever number β and nothing will change. Intuitively, you can think of it this way: imagine that the weights of your classifier are $[0.1, 0.2, 0.3]$. It means that feature 1 of your vector has importance 0.1 on the overall classification, feature 2 has importance 0.2, etc. . .

The important thing here is that feature 2 is twice more important than feature 1, so if you multiply everything by the same amount nothing will change. It means that we can set β so that $w^t + b = 1$, obtaining the following objective function:

$$\max_{w, b} \frac{1}{\|w\|_2} (1) = \min_{w, b} \|w\| \quad (2)$$

Given that $f(z) = z^2$ is a monotonic function, we can write our full optimization problem like this:

$$\begin{aligned} &\min_{w, b} \|w\|^2 \\ \text{s.t.} \quad &\forall_i \quad y_i(w^t x_i + b) \geq 0 \\ &\min_i \quad w^t x_i + b = 1 \end{aligned}$$

We can intuitively combine the two constraints in a single one and get the final formulation:

$$\begin{aligned} &\min_{w, b} \|w\|^2 \\ \text{s.t.} \quad &\forall_i \quad y_i(w^t x_i + b) \geq 1 \end{aligned} \quad (3)$$

This problem can be solved by using a quadratic programming solver, and it's totally fine. Moreover there are many ways to re-formulate and solve the problem, which are particularly useful when our data is not linearly separable.

2 Lagrangian

In machine learning we usually deal with problems in which points are not linearly separable. The previous problem can be re-written in a way that allows us to use the so called **Kernel Trick**. To do that we need to introduce the *Lagrangian formulation*. It is basically a way to reduce a constrained optimization problem into a single equation.

$$\begin{aligned} & \min f(x) \\ & s.t \quad h_i(x) \geq 0 \quad \forall i = 1 \dots m \\ & \Downarrow \\ & L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i h_i(x) \end{aligned} \tag{4}$$

where λ_i are called **lagrangian multipliers** and are all ≥ 0 .

The lagrangian dual function is defined like this:

$$g(\lambda) = \min_x (L(x, \lambda)) \tag{5}$$

According to the **strong duality** we can write that

$$d^* = \max_{\lambda \geq 0} g(\lambda) = \min f(x) = p^* \tag{6}$$

where d^* is the solution of the dual and p^* is the solution of the primal. So to find the equation of the dual, we must set $\frac{dL}{dx_1} = 0, \frac{dL}{dx_2} = 0, \dots, \frac{dL}{dx_n} = 0$. If we plug what we obtain in L we get a function of λ only, which is what we need. So let's try to do that for our specific minimization problem and see what we get. Remember that we have a constraint for each one of the points in our dataset, so $x^1 \dots x^i$

$$L(w, b, \lambda) = \frac{1}{2} w^t w + \sum_{i=1}^m \lambda_i (1 - y_i (w^t x^{(i)} + b)) \tag{7}$$

So if we simply compute the derivatives $\frac{dL}{dw} = 0$ and $\frac{dL}{db} = 0$ we get:

$$\begin{aligned} \frac{dL}{db} = 0 & \Rightarrow \sum_{i=1}^m \lambda_i y_i = 0 \\ \frac{dL}{dw} = 0 & \Rightarrow w = \sum_{i=1}^m \lambda_i y_i x_i \end{aligned} \tag{8}$$

Let's plug the easiest of the two conditions (8) in the lagrangian and see what we get:

$$\frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j x^{(i)t} x^{(j)} + \sum_i \lambda_i - \sum_{i,j} \lambda_i y_i \lambda_j y_j x^{(i)t} x^{(j)} \quad (9)$$

This sum is composed by three terms: if you look carefully the first and the last one are the same (except for the $\frac{1}{2}$), so we can simplify:

$$\sum_i \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j x^{(i)t} x^{(j)}$$

let's write the formulation of the new problem:

$$\begin{aligned} \max_{\lambda \geq 0} \quad & \sum_i \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \langle x^{(i)} x^{(j)} \rangle \\ \text{s.t.} \quad & \sum_{i=1}^m \lambda_i y_i = 0 \end{aligned} \quad (10)$$

Now we have a new objective function that only depends on λ . It also contains a inner product on which we can apply a **Kernel** (explained later).

2.1 Lagrangian in non linear case

When we deal with non-linearly separable data, we introduce the slack variables ξ_i , needed to insert some amount of tolerance when separating the data. The original problem is the following:

$$\begin{aligned} \min \quad & \frac{1}{2} w^t w + C \sum_i \xi_i \\ \text{s.t.} \quad & y_i (w x_i + b) \geq 1 - \xi_i \quad i = 1 \dots N \end{aligned} \quad (11)$$

Let's write the formulation of the lagrangian:

$$L(w, b, \xi, \alpha, \beta, C) = \frac{1}{2} w^t w + C \sum_i \xi_i - \sum_i \alpha_i [y^{(i)} (x^{(i)t} w + b) - 1 + \xi_i] - \sum_i \beta_i \xi_i \quad (12)$$

remembering that:

- $x^{(i)}$ are the vectors corresponding to the samples (each sample is a vector indeed);
- $y^{(i)}$ are the labels;

- α and β are the lagrangian multipliers. We want the new dual problem to have only these as variables;

As usual, to write the dual formulation we need to compute the derivative with respect to each one of the variables that we don't want in the new problem, ξ, w, b in this case. We don't derive with respect to C because we'll see that it will cancel out.

$$\frac{dL}{dw} = 0 \Rightarrow w^* = \sum_i \alpha_i^* y^{(i)} x^{(i)} \quad (13)$$

$$\frac{dL}{db} = 0 \Rightarrow \sum_i \alpha_i^* y^{(i)} = 0 \quad (14)$$

$$\begin{aligned} \frac{dL}{d\xi} = 0 &\Rightarrow C - \alpha_i - \beta_i = 0 \\ &\Rightarrow C = \alpha_i + \beta_i \\ &\Rightarrow 0 \leq \alpha_i \leq C \end{aligned} \quad (15)$$

the last relation has to be true because the lagrangian multipliers need to be positive. Let's plug 13 in the lagrangian (using y_i instead of $y^{(i)}$ to have a lighter notation):

$$\frac{1}{2} \sum_{i,j} a_i a_j y_i y_j x_i^t x_j + C \sum_i \xi_i - \sum_{i,j} a_i a_j y_i y_j x_i^t x_j - b \sum_i \alpha_i y_i + \sum_i \alpha_i - \sum_i \alpha_i \xi_i - \sum_i \beta_i \xi_i \quad (16)$$

- The 1^o and the 2^o term are the same except for the $\frac{1}{2}$;
- The 4^o term is null because of 14;

$$-\frac{1}{2} \sum_{i,j} a_i a_j y_i y_j x_i^t x_j + C \sum_i \xi_i + \sum_i \alpha_i - \sum_i \alpha_i \xi_i - \sum_i \beta_i \xi_i \quad (17)$$

Let's use equation 15 and re-write the C:

$$\begin{aligned} &-\frac{1}{2} \sum_{i,j} a_i a_j y_i y_j x_i^t x_j + (\alpha_i + \beta_i) \sum_i \xi_i + \sum_i \alpha_i - \sum_i \alpha_i \xi_i - \sum_i \beta_i \xi_i \\ &-\frac{1}{2} \sum_{i,j} a_i a_j y_i y_j x_i^t x_j + \sum_i \xi_i \alpha_i + \sum_i \xi_i \beta_i + \sum_i \alpha_i - \sum_i \alpha_i \xi_i - \sum_i \beta_i \xi_i \\ &-\frac{1}{2} \sum_{i,j} a_i a_j y_i y_j x_i^t x_j + \sum_i \alpha_i \end{aligned} \quad (18)$$

s.t $0 \leq \alpha_i \leq C$

s.t $\sum_i a_i y_i = 0$

If we carefully look at equation 18 we notice that the formulation is basically the same as the one in the linearly separable case, we just have an extra bound on the value of α_i (quite an easy bound to consider), that now is limited by C .

3 KKT conditions

There is a series of equations called KKT conditions that summarize the relation of the primal and the dual problem. If x^* is a solution for the primal problem, f is convex and all of the $h_i(x)$ are linear constraints, then there exist some multipliers $\lambda_1 \dots \lambda_m$ such that:

$$\nabla f(x^*) - \lambda^t \nabla h(x^*) = 0 \tag{19}$$

$$h(x^*) \geq 0 \tag{20}$$

$$\lambda_i \geq 0 \quad \forall i = 1 \dots m \tag{21}$$

$$\lambda^t h(x^*) = 0 \tag{22}$$

where λ^t is the solution of the dual! The first 3 conditions are quite intuitive, the last one is the tricky one. Let's write the demonstration for that. Let x^* be the solution of the primal and λ^* be the solution of the dual. Because of the strong duality, we can write the following:

$$\begin{aligned}
f(x^*) &= g(\lambda^*) = \min_x \left(f(x) + \sum_{i=1}^m \lambda_i^* h_i(x) \right) \\
&\geq f(x^*) + \sum_{i=1}^m \underbrace{\lambda_i^*}_{\geq 0} \underbrace{h_i(x^*)}_{\geq 0} \\
&\geq f(x^*)
\end{aligned}$$

which means that $\sum_{i=1}^m \lambda_i^* h_i(x^*) = 0 \Rightarrow \lambda^{*T} h(x^*) = 0$.

This means that if we have a lagrangian multiplier $\lambda_i \geq 0$, then the relative constraint should be active (it is satisfied with equality), so $h_i(x^*) = 0$.

4 Pegasos

Pegasos is a training algorithm that solves the primal formulation of the problem by applying gradient descent. We will call L the "loss" function.

$$\min_w L(w) = \frac{\lambda}{2} \|w\|^2 + \underbrace{\frac{1}{N} \sum_{i=1}^N \max(0, 1 - y_i \langle w, x_i \rangle)}_{\text{Hinge loss}} \quad (23)$$

- λ is a parameter that controls the tradeoff between maximizing the margin and classifying the points correctly.
- The loss function increases if we missclassify a point, so if the inner product is > 0 . If it is negative, the 0 gets taken.

For each step of the gradient descent algorithm we update the weights following this rule:

$$w^{t+1} = w^t - \eta \nabla L(w^t) \quad (24)$$

where η is the lenght of the step size. It decreases at each iteration:

$$\eta^t = \frac{1}{t\lambda} \quad (25)$$

Computing the gradient of the Loss is easy in this case:

$$\nabla L(w^t) = \lambda w^t - \frac{1}{N} \sum_{y_i \langle w^t, x_i \rangle < 1} y_i x_i \quad (26)$$

note that the loss takes into account only missclassified points!