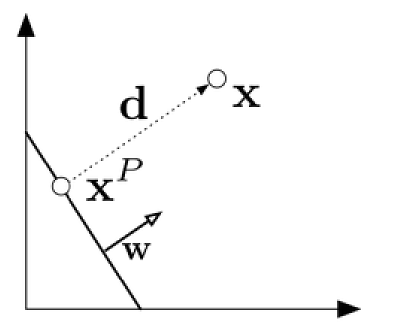
SVM notes

Matteo Lugli

October 22, 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 $w^t(x - \alpha w) + b = 0$.

$$
\begin{aligned}
&w^t - w^t \alpha w + b = 0 \\
&\frac{w^t x + b}{w^t w} = \alpha \\
&d = \frac{w^t x + b}{w^t w} w
\end{aligned}
$$

Now we want to calculate the magnitute 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:

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

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

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

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 thing 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, ecc...
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:

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

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

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

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

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

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 usefull 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.

$$minf(x)$$
$$s.t \quad h_i(x) \geq 0 \quad \forall i = 1...m$$
$$\Downarrow$$

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) \tag{4}$$

where λ_i are called **lagrangian multipliers** and are all ≥ 0.
The lagrangian dual function is defined like this:

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

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

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

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 $\frac{dL}{dx_1} = 0$, $\frac{dL}{dx_2} = 0$, $\ldots$, $\frac{dL}{dx_n} = 0$. 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 $x^1 \ldots x^i$

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

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

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

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

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

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:

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

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).

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 $\lambda_i \dots \lambda_m$ such that:

$$\nabla f(x^*) - \lambda^t \nabla h(x^*) = 0 \tag{11}$$

$$h(x^*) \geq 0 \tag{12}$$

$$\lambda_i \geq 0 \quad \forall i = 1 \dots m \tag{13}$$

$$\lambda^t h(x^*) = 0 \tag{14}$$

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:

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

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

$$min_w L(w) = \frac{\lambda}{2}||w||^2 + \underbrace{\frac{1}{N}\sum_{i=1}^{N} max(0, 1 - y_i\langle w, x_i\rangle)}_{\text{Hinge loss}} \tag{15}$$

- 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:

$$w^{t+1} = w^t - \eta \nabla L(w^t) \tag{16}$$

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

$$\eta^t = \frac{1}{t\lambda} \tag{17}$$

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

$$\nabla L(w^t) = \lambda w^t - \frac{1}{N} \sum_{y_i\langle w^t, x^i\rangle < 1} y_i x_i \tag{18}$$

note that the loss takes into account only missclassified points!