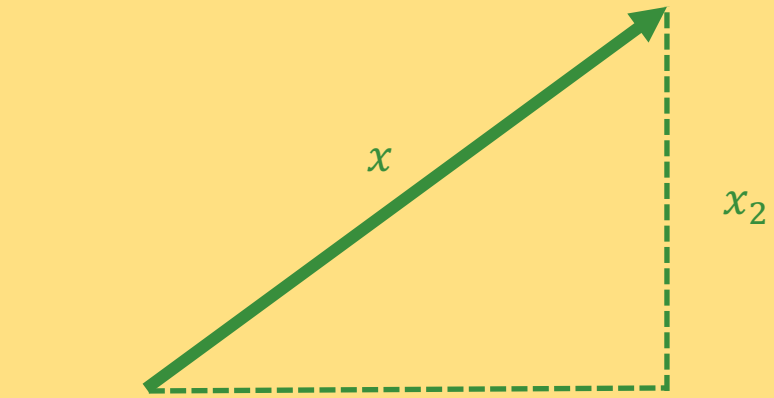
Support Vector Machines

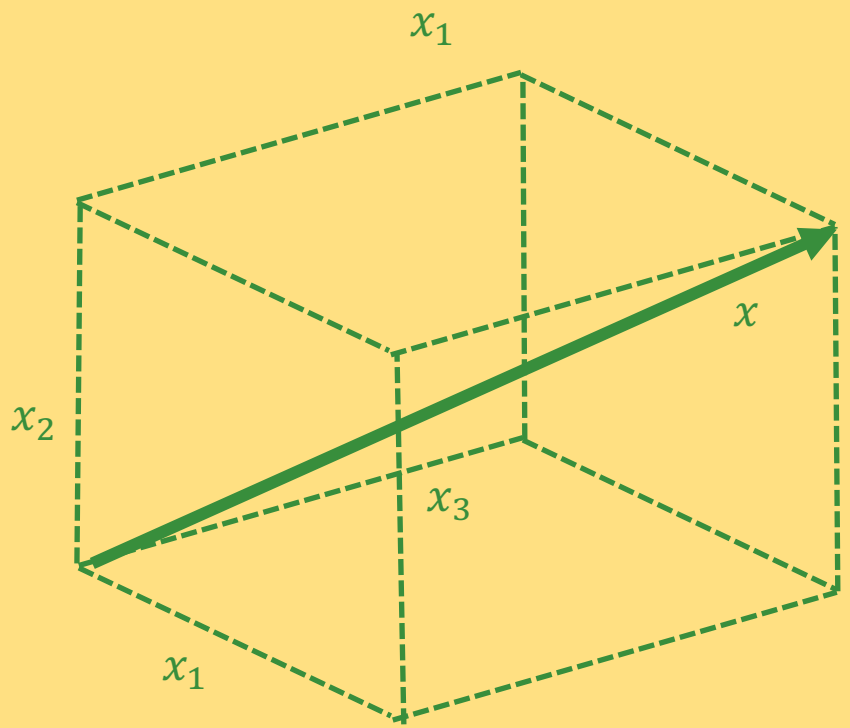
Linear Algebra Concepts

A vector is a set of n numbers in the form:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

A vector can be represented as an arrow, a metric with both direction and magnitude



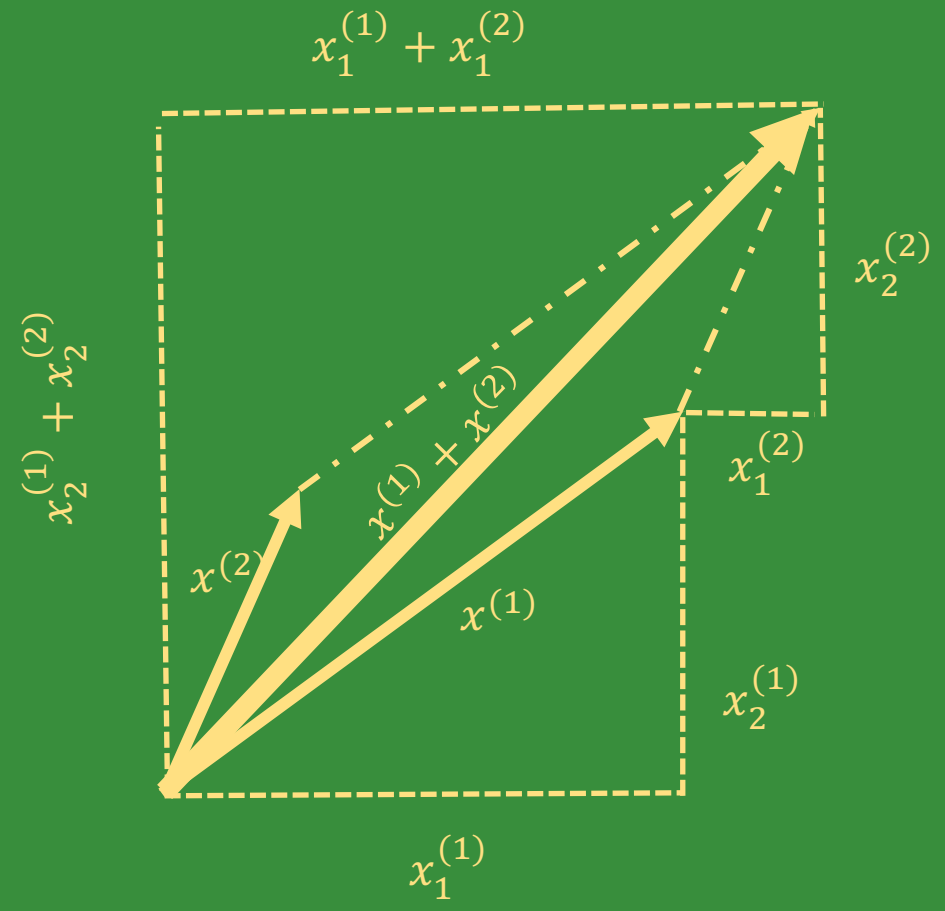


The length of a vector x in an n dimensional space (denoted by $\|x\|$,read as the norm of x) can be calculated as:

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

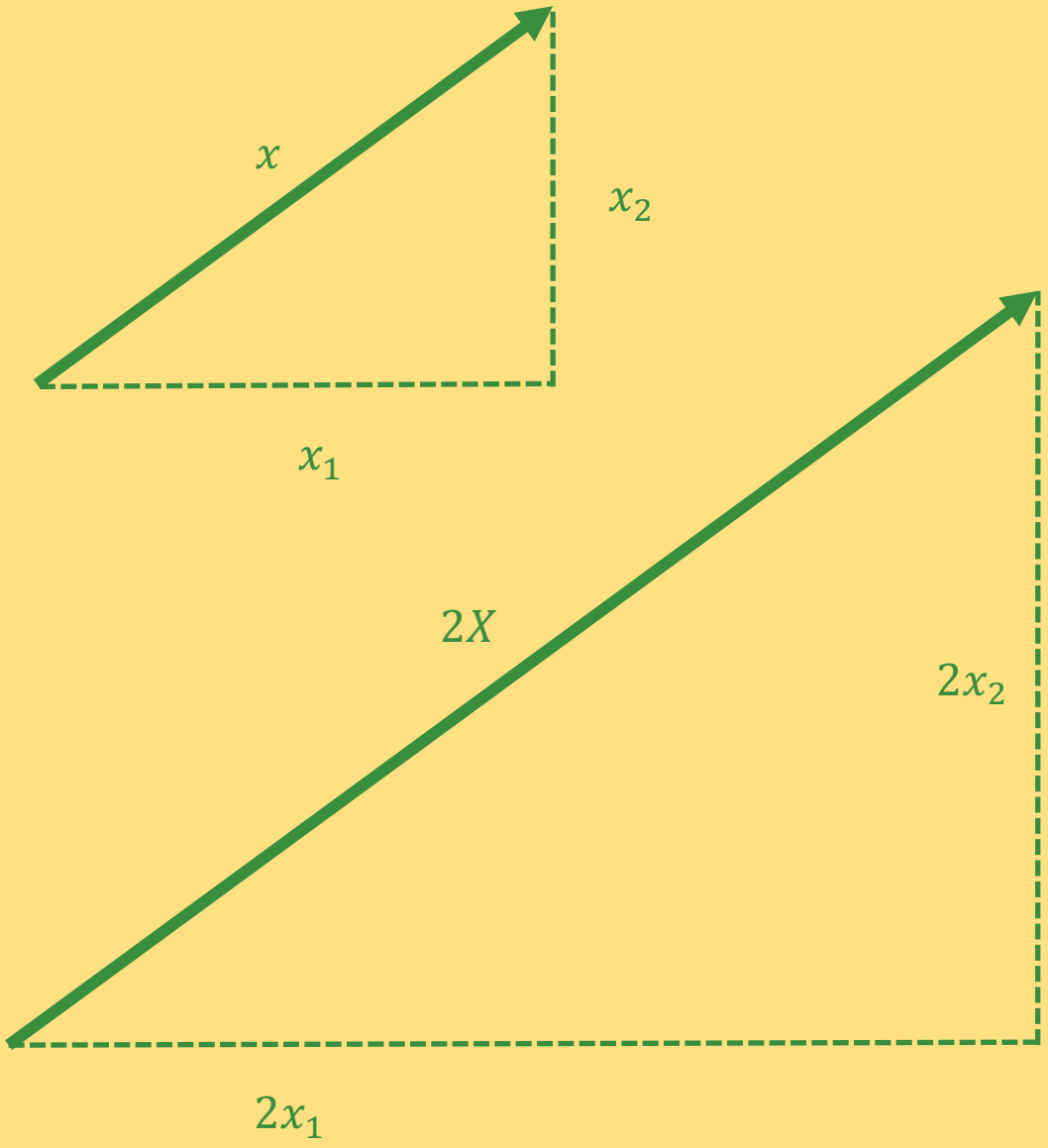
Vectors can be added

$$\begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \end{bmatrix} + \begin{bmatrix} x_1^{(2)} \\ x_2^{(2)} \end{bmatrix} = \begin{bmatrix} x_1^{(1)} + x_1^{(2)} \\ x_2^{(1)} + x_2^{(2)} \end{bmatrix}$$



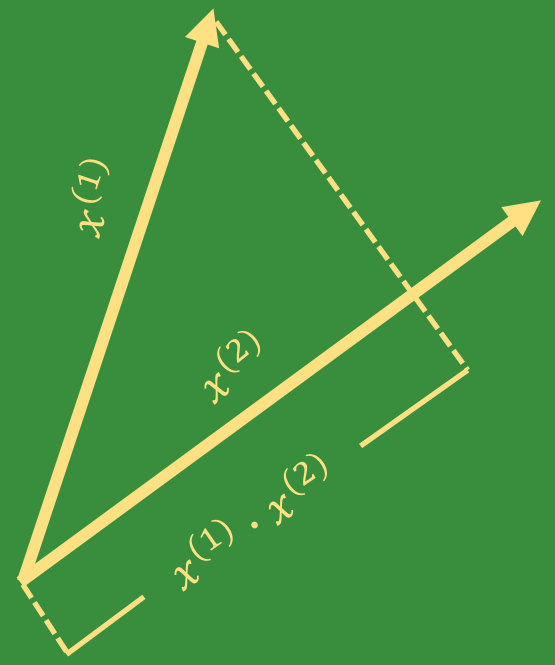
Vectors can be multiplied to scalar numbers

$$2\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix}$$



Vectors can be multiplied to other vectors

$$\begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \end{bmatrix} \cdot \begin{bmatrix} x_1^{(2)} \\ x_2^{(2)} \end{bmatrix} = x_1^{(1)} x_1^{(2)} + x_2^{(1)} x_2^{(2)}$$



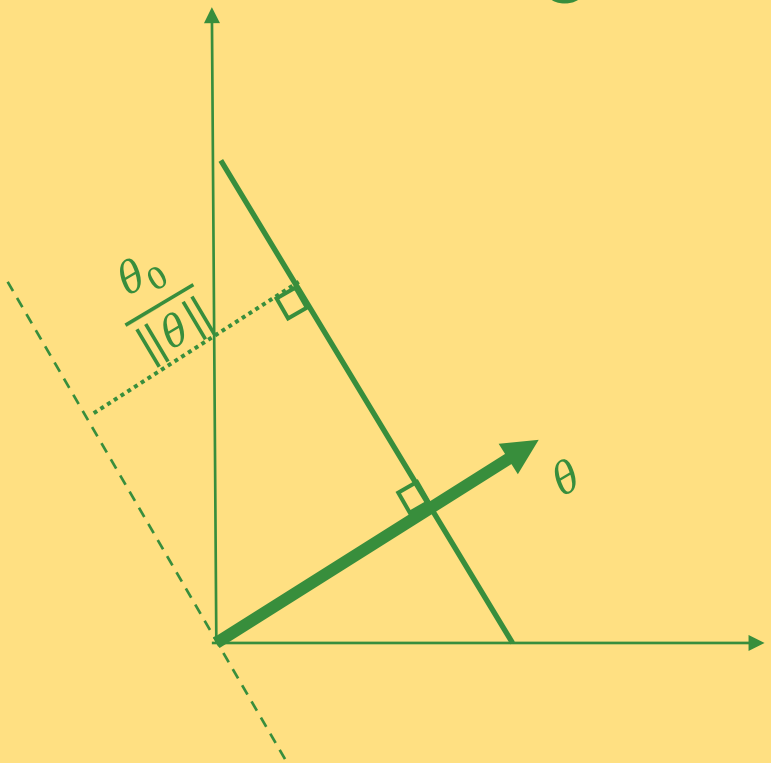
The dot product of $x^{(1)} \cdot x^{(2)}$ is also called the scalar projection of $x^{(1)}$ into the direction of $x^{(2)}$

$$x^{(1)} \cdot x^{(2)} = \left(x^{(1)}\right)^T x^{(2)}$$
$$= \|x^{(1)}\| \|x^{(2)}\| \cos\theta$$

Maximum Margin Classifier

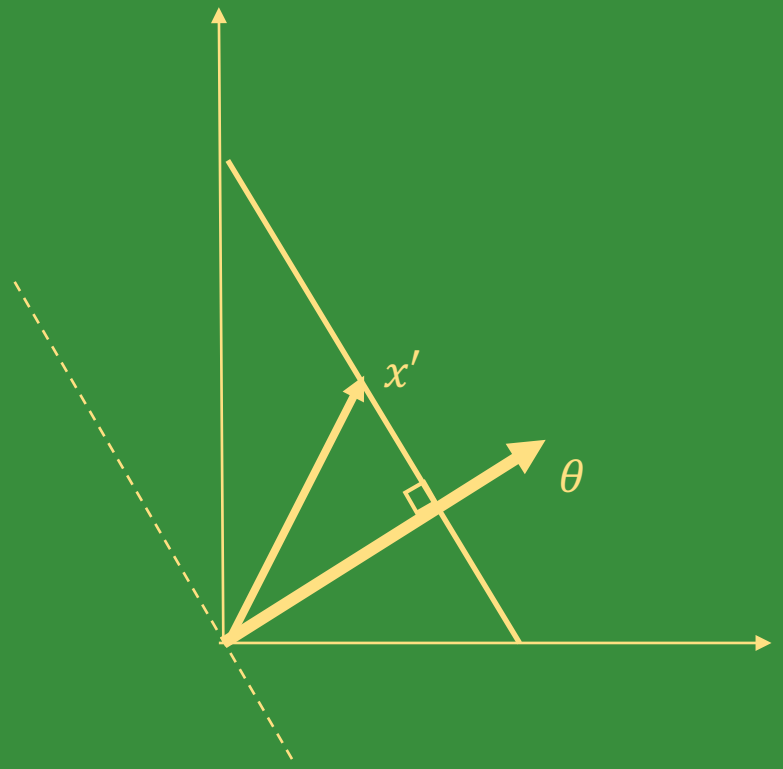
Given a dataset with 2 classes, $\{-1,1\}$ and n features, suppose there exists an n-1 dimensional **SEPARATING HYPERPLANE** that perfectly divides the two classes

A hyperplane can be described using a vector θ and bias θ_0. The separating hyperplane is a line orthogonal to θ with the an orthogonal distance $-\frac{\theta_0}{\|\theta\|}$ from the origin



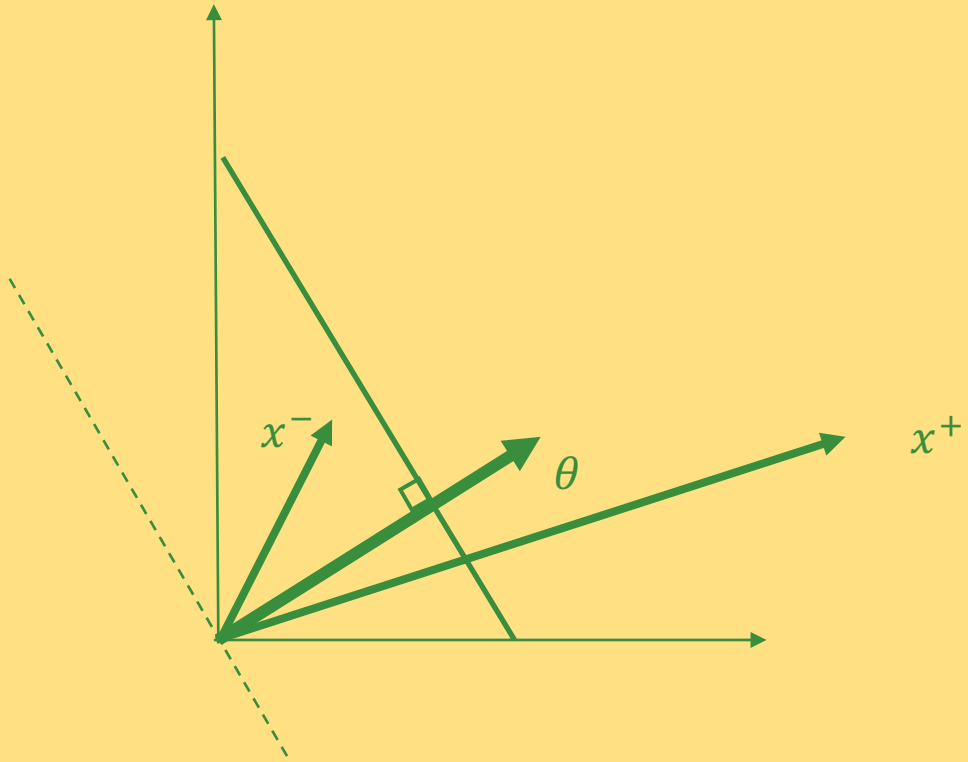
Any vector x' touching the separating hyperplane satisfies the condition

$$h(x) = \theta^T x' + \theta_0 = 0$$



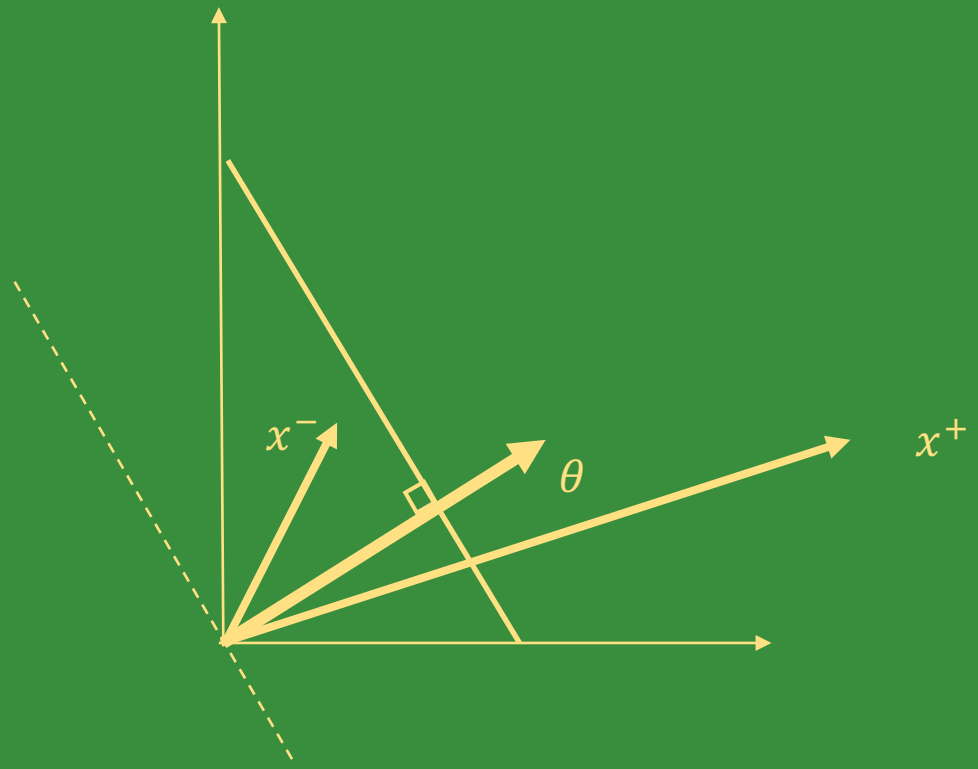
Any vector x^- to the left of the separating hyperplane satisfies the condition

$$h(x) = \theta^T x^- + \theta_0 \leq 0$$



Any vector x^+ to the right of the separating hyperplane satisfies the condition

$$h(x) = \theta^T x^+ + \theta_0 \geq 0$$



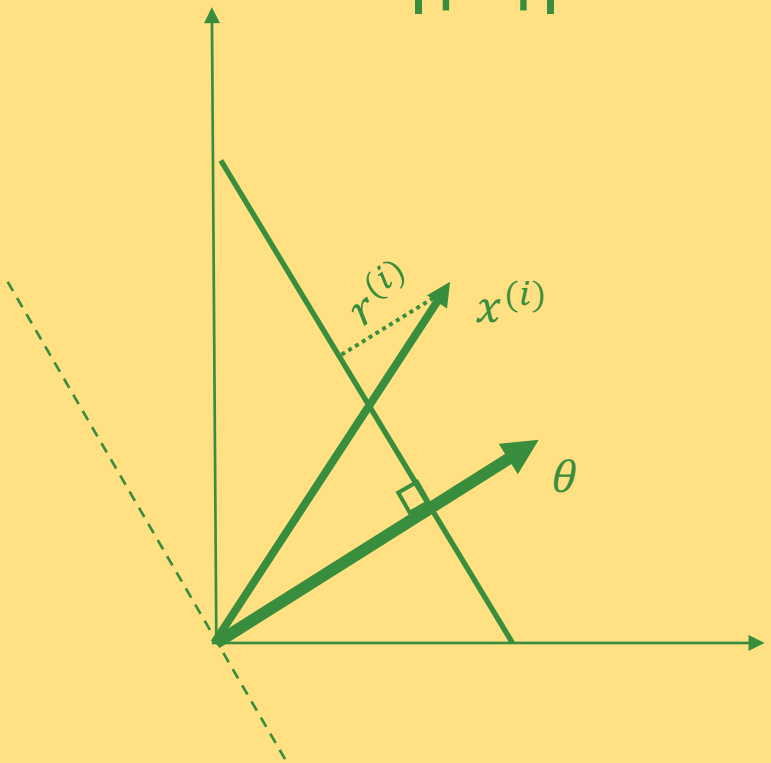
correct predictions of any example $x^{(i)}$ satisfies the condition:

$$y^{(i)}h\left(x^{(i)}\right) > 0$$

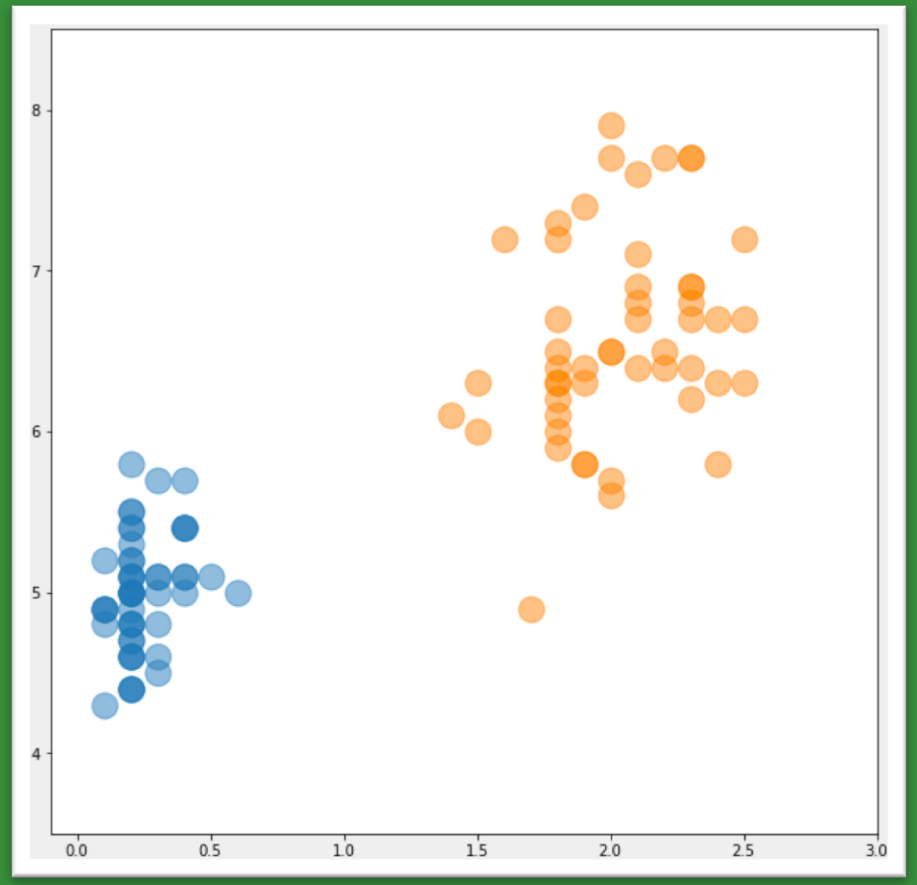
this is because if the signs of $y^{(i)}$ and $h\left(x^{(i)}\right)$ are the same (both negative or both positive) the product is positive, otherwise the product is negative

The distance of some point $x^{(i)}$ to the separating hyperplane is calculated to be

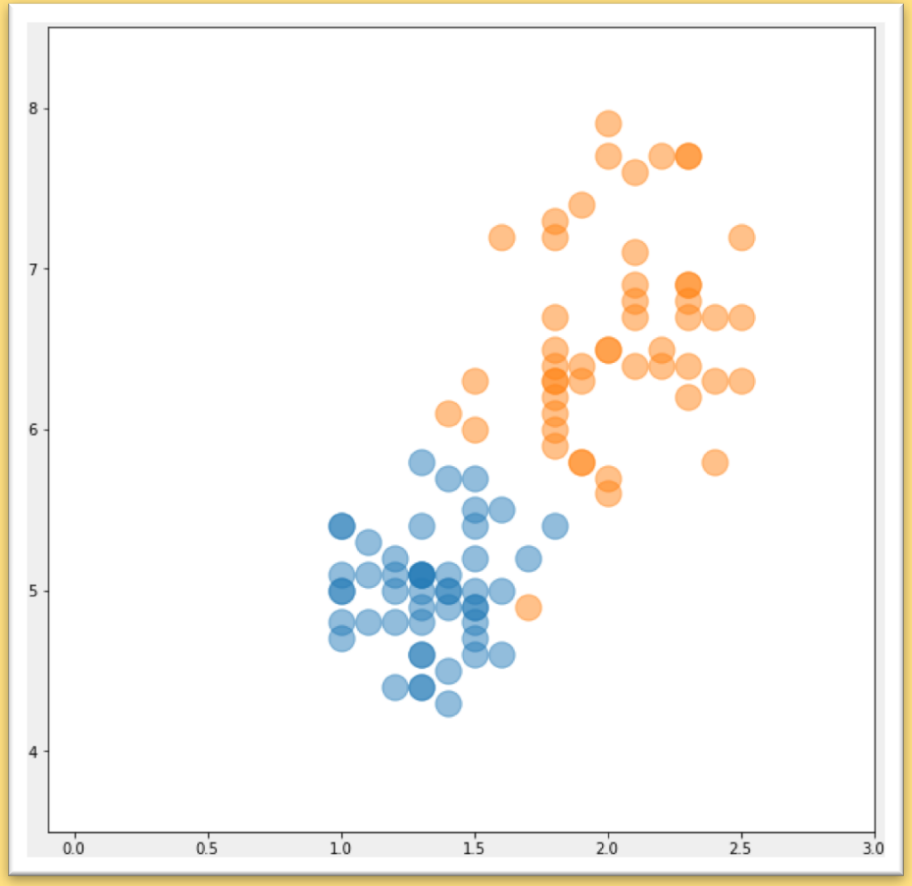
$$r^{(i)} = \frac{h(x^{(i)})}{||\theta||}$$



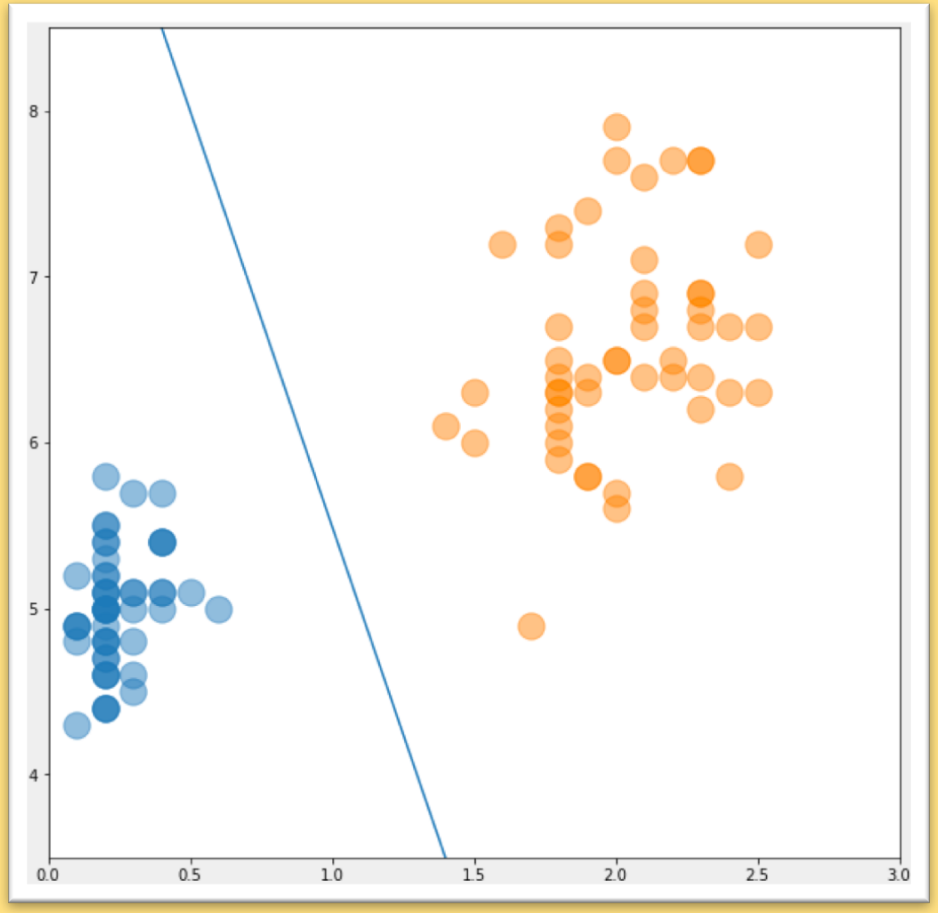
A dataset where there exists infinitely many perfectly separating hyperplane is said to be **LINEARLY SEPARABLE**

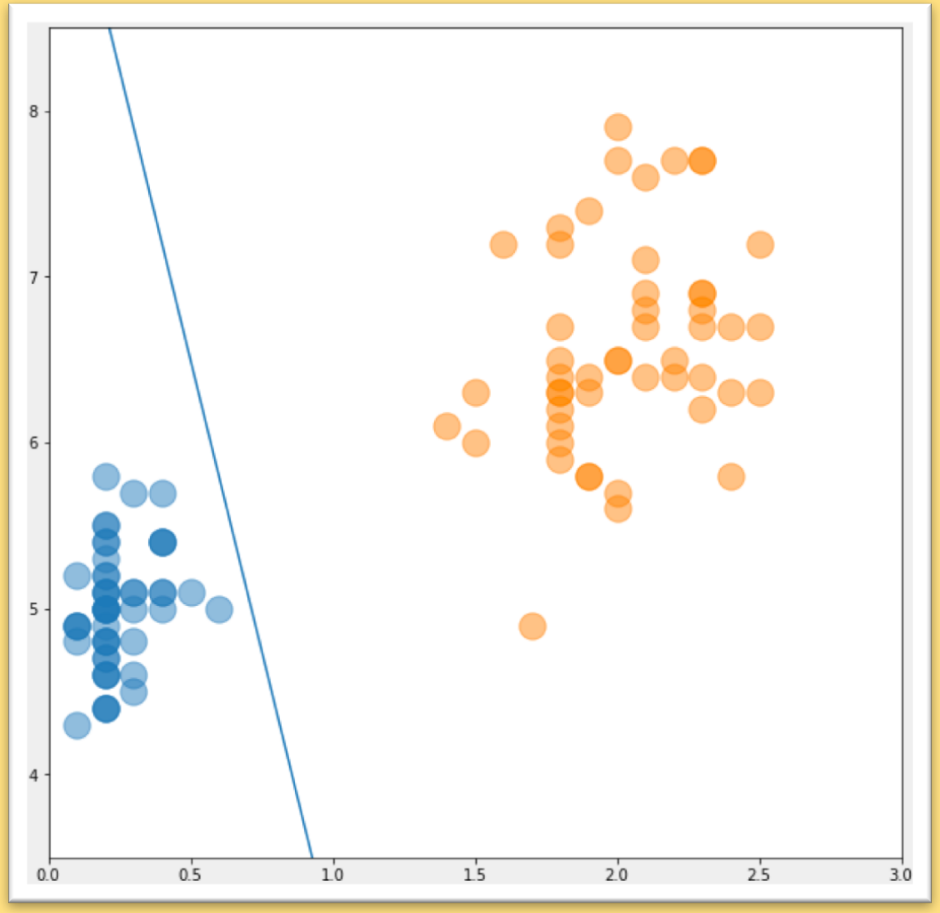


A dataset where there exists exactly zero perfectly separating hyperplanes is said to be **NON-LINEARLY SEPARABLE**



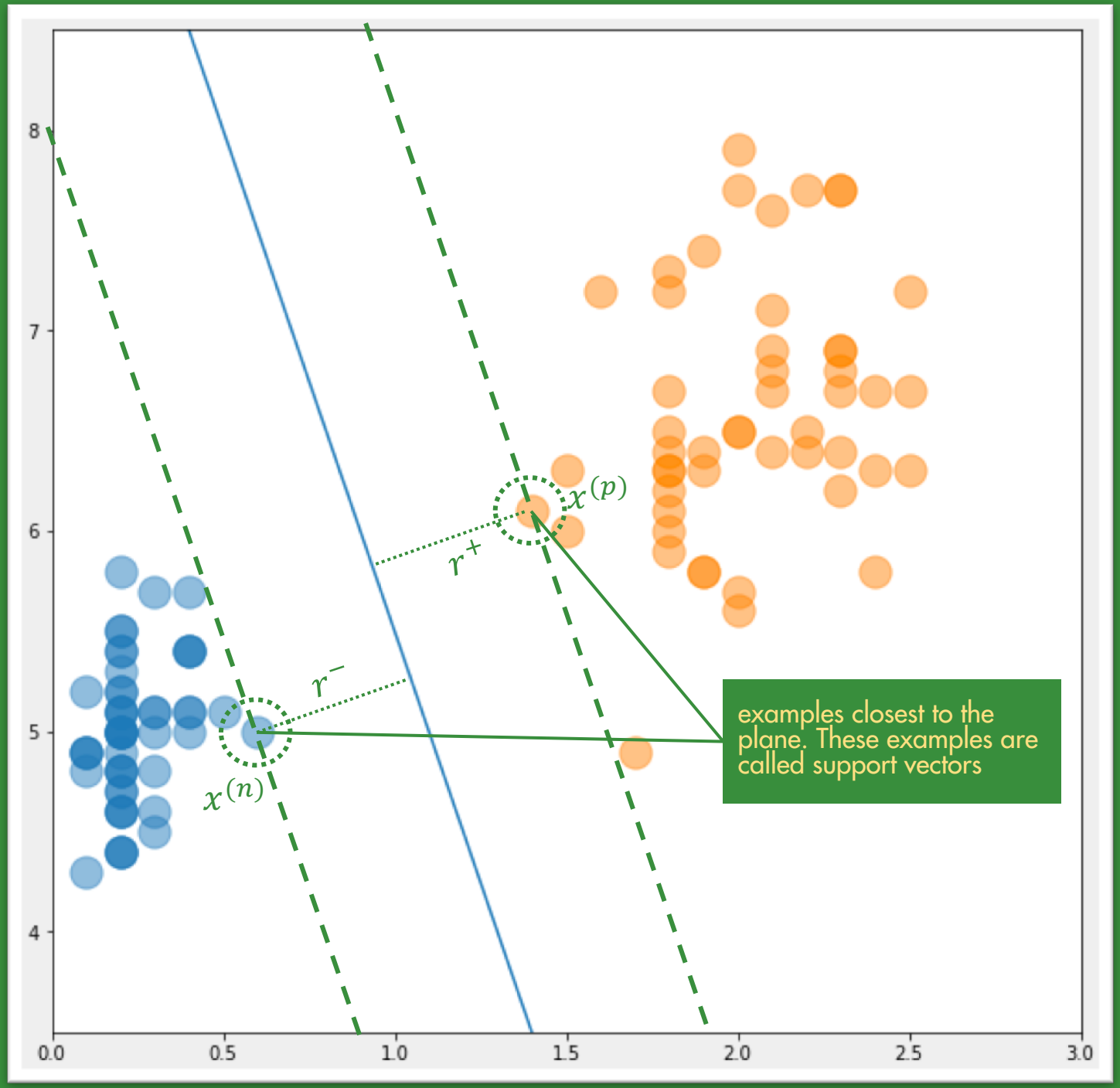
Consider these separating hyperplanes, which one of them is better?





The left hyperplane is better than right hyperplane since it offers more room between the examples and the hyperplane. The right hyperplane will prefer predictions of the orange class which could lead to more erroneous predictions.

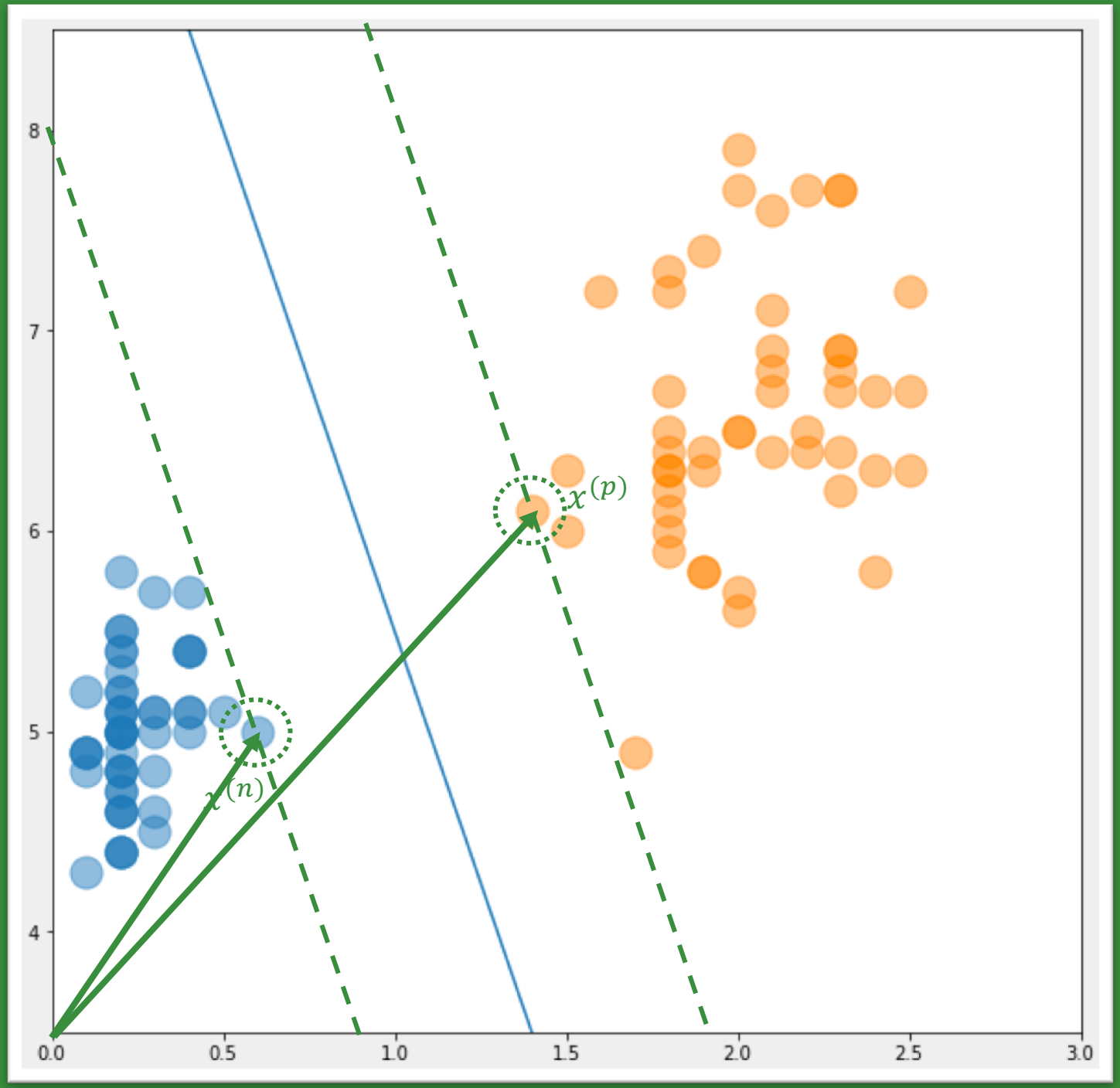
This means that the best separating hyperplane can be found by searching for the hyperplane where the separation between the closest positive example and closest negative example is the greatest.


8
7
6
5
4
0.0
0.5
1.0
1.5
2.0
2.5
3.0
$x^{(p)}$
r^+
r^-
$x^{(n)}$
examples closest to the plane. These examples are called support vectors

The values of x^p and x^n can be found for any given hyperplane by evaluating these minimizations:

$$x^{(p)} = \min_{i,\, y^{(i)}=1} y^{(i)} h\left(x^{(i)}\right)$$

$$x^{(n)} = \min_{i,\, y^{(i)}=-1} y^{(i)} h\left(x^{(i)}\right)$$


8
7
6
5
4
0.0
0.5
1.0
1.5
2.0
2.5
3.0
$x^{(p)}$
$x^{(n)}$

Finding the hyperplane can be done by finding the values of θ and θ_0 that evaluates the following maximization

$$\max_{\theta,\theta_0} \left(x^{(p)} - x^{(n)}\right)^T \frac{\theta}{||\theta||}$$

Instead of working with this maximization objective, we can simplify the problem by introducing some constraints

$$y^{(i)}h(x^{(i)}) \geq 1$$
$$h(x^{(+)}) \geq 1$$
$$h(x^{(-)}) \leq -1$$

This constraint directly enforces margins in the decision boundary. This constraint will force the examples to be defined as positive or negative beyond the margins only.

Because of this we can find the values of $x^{(p)}$ and $x^{(n)}$ as the closest to zero possible values for $h(x)$, which are actually the values $h(x^{(p)}) = 1$ and $h(x^{(n)}) = -1$.

$$\theta^T x^{(p)} + \theta_0 = 1$$
$$\theta^T x^{(n)} + \theta_0 = -1$$

Using this constraint, we can simplify the maximization into:

$$\max_{\theta,\theta_0}\left(\frac{\theta^T x^{(p)}}{||\theta||} - \frac{\theta^T x^{(n)}}{||\theta||}\right) =$$

$$\max_{\theta,\theta_0}\frac{2}{||\theta||}, \text{where: } y^{(i)} h(x^{(i)}) = 1$$

we can change our optimization objective into a more mathematically convenient optimization:

$$\min_{\theta,\theta_0} = \frac{1}{2}||\theta||^2, \text{where: } y^{(i)}h(x^{(i)}) - 1 = 0$$

We can now derive this constrained optimization as a Lagrangian:

$$\mathcal{L} = \frac{1}{2}||\theta||^2 - \sum_{i=1}^{m} \ell_i(y^{(i)}(\theta^T x^{(i)} + \theta_0) - 1)$$

Solving for $\frac{\partial \mathcal{L}}{\partial \theta} = 0$ and $\frac{\partial \mathcal{L}}{\partial \theta_0} = 0$ will lead us to the the value of θ ($\theta = \sum_{i=0}^{n} \ell_i y^{(i)} x^{(i)}$). Substituting these values to the Lagrangian will lead us to our final optimization objective:

$$\mathcal{L} = \sum_{i=0}^{n} \ell_i - \sum_{i=0}^{n} \sum_{k=0}^{n} \ell_i \ell_k y^{(i)} y^{(k)} {x^{(i)}}^T x^{(k)}$$

Kernel Trick

How do we solve non-linearly separable cases?

We can map each of our examples, x into some transformation, $\phi(x)$ that makes the dataset linearly separable, changing our objective into:

$$\mathcal{L} = \sum_{i=0}^{n} \ell_i - \sum_{i=0}^{n} \sum_{k=0}^{n} \ell_i \ell_k y^{(i)} y^{(k)} \phi(x^{(i)})^T \phi(x^{(k)})$$

How do we find the transformation of x, $\phi(x)$?

HINT: We actually don't need to.

We only need to define as some scalar similarity function $\mathcal{K}(x^{(i)}, x^{(k)})$. This is possible because $\mathcal{K}(x^{(i)}, x^{(k)})$ yields a scalar value just like the dot product $\phi(x^{(i)})^T \phi(x^{(k)})$.

Therefore, we only need to define a similarity measure $\mathcal{K}(x^{(i)}, x^{(k)})$ which yields large values for similar $x^{(i)}$ and $x^{(k)}$ and small values for different $x^{(i)}$ and $x^{(k)}$. These similarity functions are called **KERNELS** and are the basis for the so-called **KERNEL TRICK**.

Choices for $\mathcal{K}(x^{(i)}, x^{(k)})$

Polynomial Kernel: $\mathcal{K}(x^{(i)}, x^{(k)}) = (x^{(i)^T} x^{(k)} + 1)^p$

Calculates the polynomial similarity of two examples

Radial Basis Kernel: $\mathcal{K}(x^{(i)}, x^{(k)}) = e^{\frac{||x^{(i)} - x^{(k)}||}{\sigma}}$

Calculates the proximity of two examples