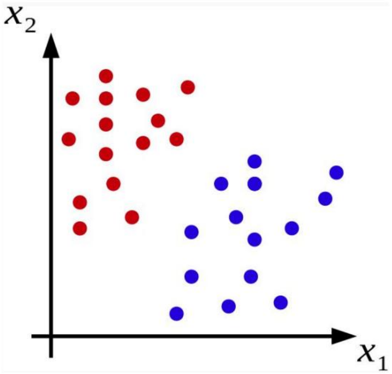
Machine Learning

Classifiers based on support vectors - Part 1

2023-2Q

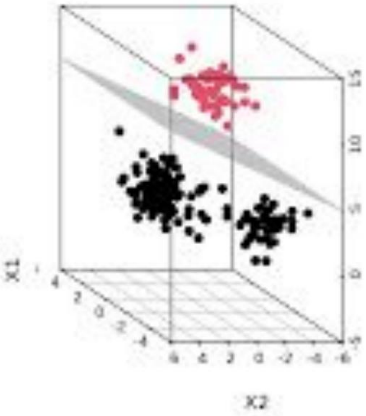
Support Vector Machines (SVM)

It is a **supervised learning** algorithm used for both **classification** and regression problems .



Its **main objective** is to find the **hyperplane optimal** that best **separates the different classes.**

Hyperplane concept



Equation of a hyperplane

In a **p-dimensional** space a hyperplane is defined by

b0 + b1 x1 + b2 x2 +·· + b = 0 x pp

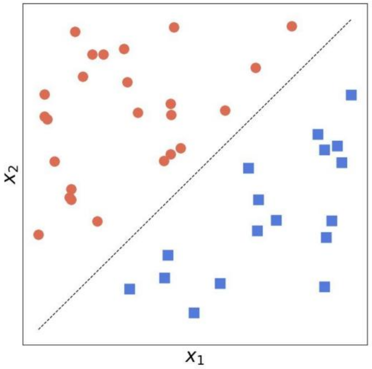
Equation of a hyperplane in R2

In **R2,** the line is defined by **b0 + b1 x1 + b2 x2 = 0** • **x =**

(x1 , x2) ÿ on the **line** • **xÿ = (xÿ1 ,**

xÿ2) such that **b0 + b1 xÿ1 + b2 xÿÿ2 > 0** •

xÿÿ = (xÿÿ1 , xÿÿ2) such that **b0 + b1 xÿÿ1 + b2 xÿÿ2 < 0**



Linear separability

Suppose we have a set of **n examples** of **p attributes xi** and **class yi** (1 ÿ i ÿ n):

$$x1 = (x1,1, x1,2, .., x1,p),$$

$$y1\ x2 = (x2,1, x2,2, .., x2,p), y2$$

$$\dots$$

$$x_n = (xn,1, xn,2, .., xn,p), yn$$

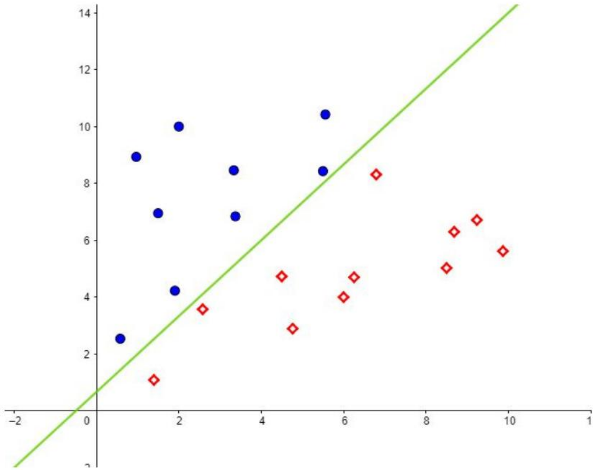
where each **yi belongs to {1, ÿ1}.**

The goal is to find a classifier that correctly classifies each of these examples according to their class.

Linear separability

Separation Hyperplane

If we obtain a hyperplane such that all the examples whose class is -1 are on one side of the hyperplane and all examples whose class is +1 remain from the other we will have achieved the objective.



Linear separability

So ...

Given xi , i = 1, . . . n if it happens that

b0 + b1 xi,1 + b2 xi,2 + · · · + b p xi,p > 0 when **yi = 1**

and

b0 + b1 xi,1 + b2 xi,2 + · · · + b p xip < 0 when **yi = ÿ1**

then we can guarantee that **yi (b0 + b1 xi,1 + b2 xi,2 + · · · + b p xi,p) > 0**

Linear separability

yi (b0 + b1 xi,1 + b2 xi,2)

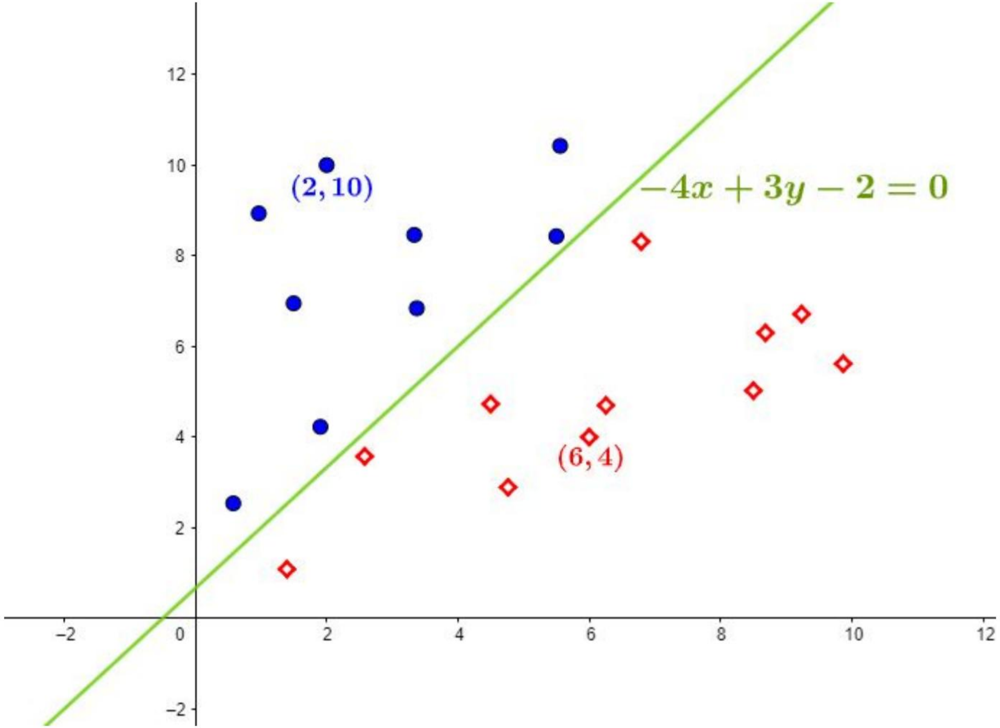
> 0 yi (-2 -4xi,1 + 3xi,2) > 0

1 * (-2 -4*2 + 3*10) > 0

* (20) > 0 1

-1 * (-2 -4*6 + 3*4) > 0

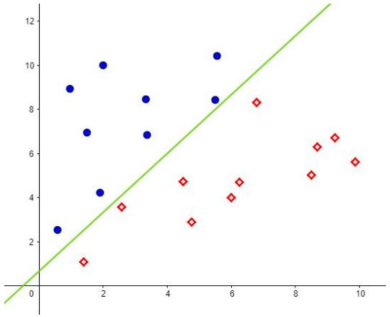
-1 * (-14) > 0



Linear separability

Given a test observation **xÿ** with p attributes we can classify it according to:

- **xÿ** will be of **class 1** if b0 + b1 xÿ1 + b2 xÿ2 + · · + bxÿ **>** 0 pp



- **xÿ** will be of **class -1** if b0 + b1 xÿ1 + b2 xÿ2 + · · + b p xÿ <0 p

Linear separability

Let **f (xÿ) = b0 + b1 xÿ1 + b2 xÿ2 +·· + bxÿ pp** Furthermore, we can say that:

- If f (xÿ) is a value close to 0, xÿ will be close to the hyperplane

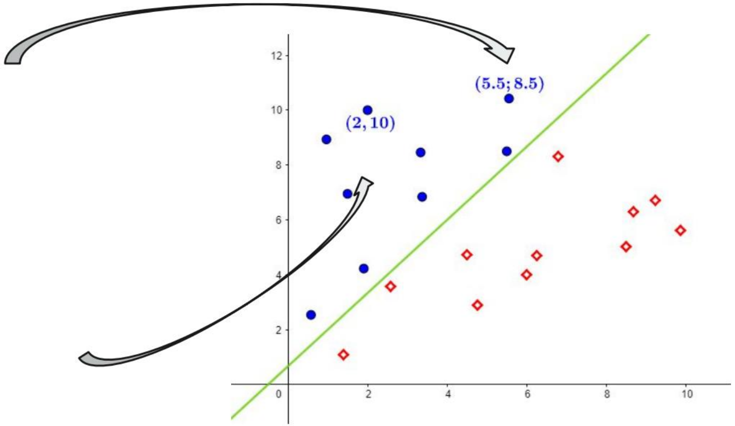
- If f (xÿ) is a value far from 0, xÿ will be far from the hyperplane.

$$-4.5, 5 + 3.8, 5 - 2 = -1, 5$$

-1.5 near 0 ÿ (5.5; 8.5) near the hyperplane

$$-4.2 + 3.10 - 2 = 20$$

20 far from 0 ÿ (2.10) far from the hyperplane

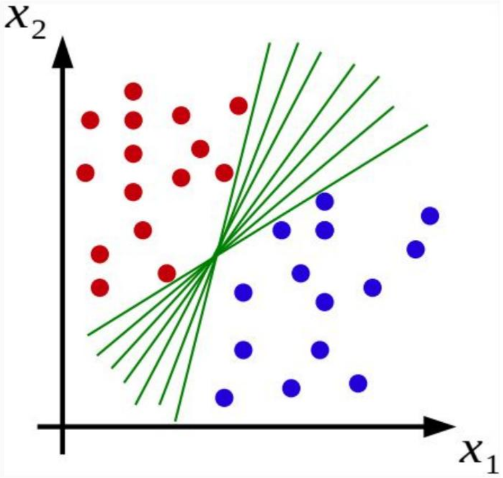


The maximum margin classifier

How do we separate the classes?

There may be more than one hyperplane separating them, in

In general, infinite planes that separate both classes.

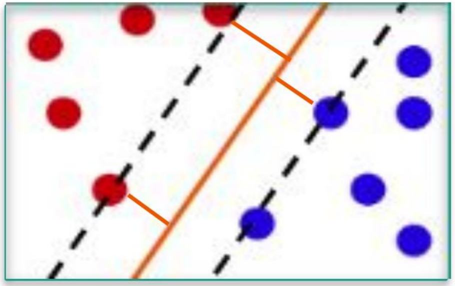


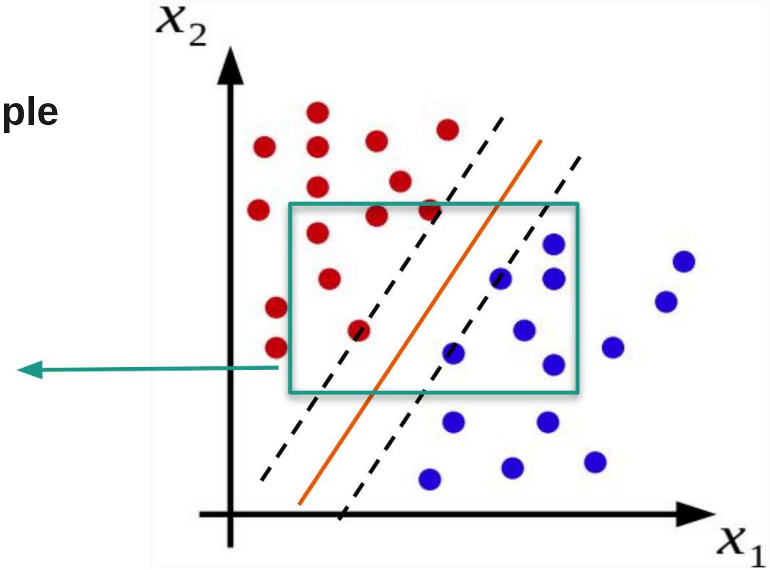
However, there is a hyperplane that has a particular property.

The maximum margin classifier

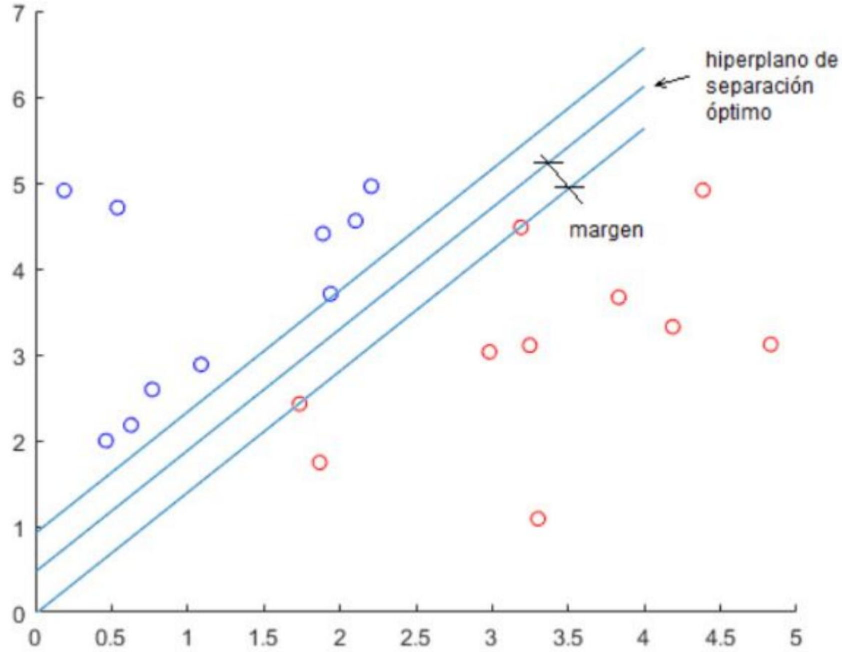
Let **H** be a hyperplane that separates both classes, let us consider the distances of each one of the examples to H.

We will define **margin** as the **distance of the example closest to H.**



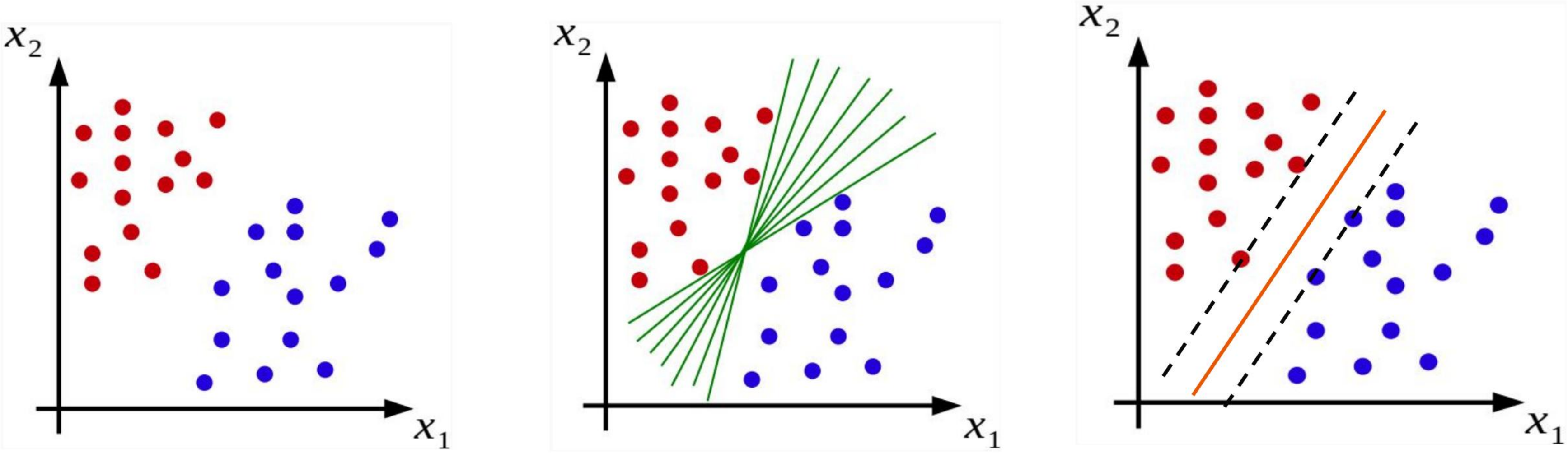


The maximum margin classifier



The maximum margin classifier

We are interested in the hyperplane that has a maximum margin, that is, **Hyperplane with maximum margin** o **Optimal separation hyperplane.**

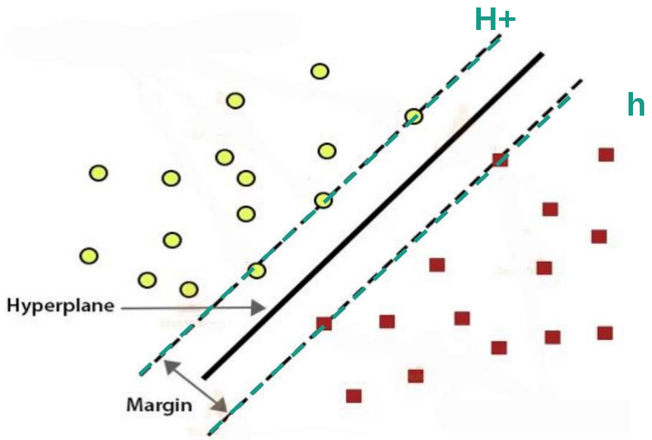


The maximum margin classifier

If we use the **maximal margin hyperplane** to separate both classes, the classifier is called **Maximal Margin Classifier.**

In addition, two hyperplanes are defined

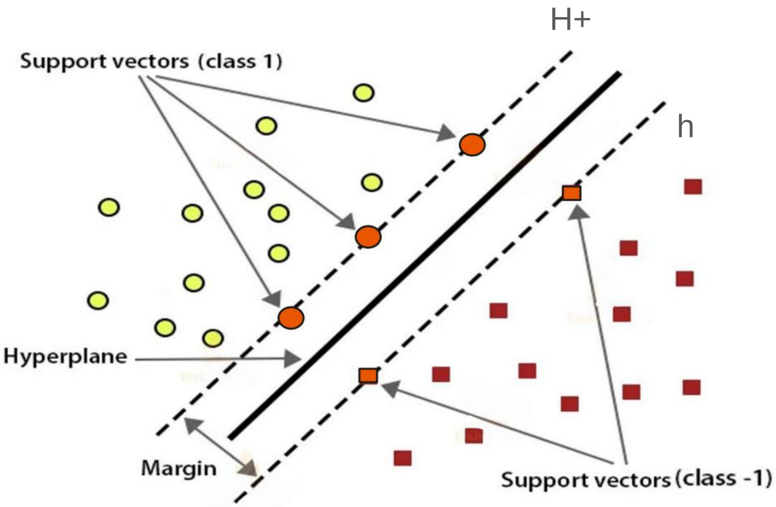
more, equidistant from H, **H+ and Hÿ.**



The distance between H+ and H, and between Hÿ and H, is the same: the margin.

The maximum margin classifier

On H+ and Hÿ you can see examples of both classes that are above them are called **support vectors .**



The **optimal separation hyperplane depends only on the support vectors** and not on the rest of the class examples.

The maximum margin classifier

Construction of the Maximum Margin Classifier

Let M be the margin (which we want to maximize) and since b defines the optimal separation hyperplane then what we want to do is find the values of b such that they maximize M subject to

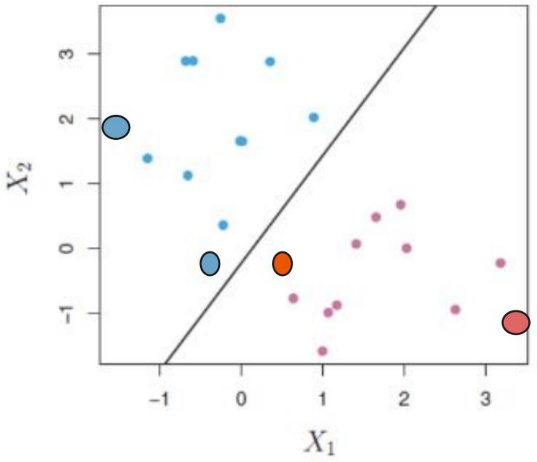
- $y_i * (b_0 + b_1 x_{i,1} + b_2 x_{i,2} + ... + b_p x_{i,p}) \geq M, \forall i, 1 \leq i \leq n,$
- $\sum_{j=1}^{p} b_j^2 = 1.$

Support Vector Classifier

The distance of a test observation from the optimal separation hyperplane gives us an idea of the **confidence** we can have in the classification.

- If the **distance is great** we will have **more confidence** (the observation is quite inside the class).

- If the **distance is small,** close to 0, we will have **less confidence** (the observation is close to the class boundary and therefore close to the other class).



Is the optimal separation hyperplane always optimal?

Suppose two classes that are well separated, that is

That is, they have a considerable margin.

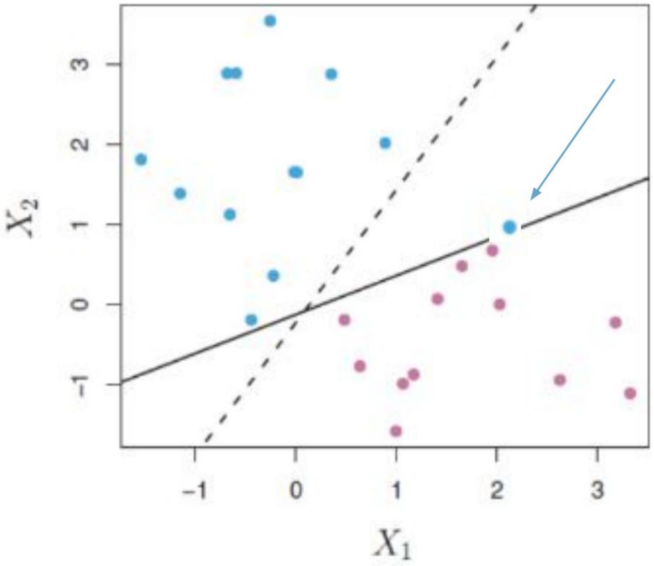
We added an example that makes the margin

reduce considerably.

Examples that with the initial hyperplane would have been

ranked with greater confidence will now be

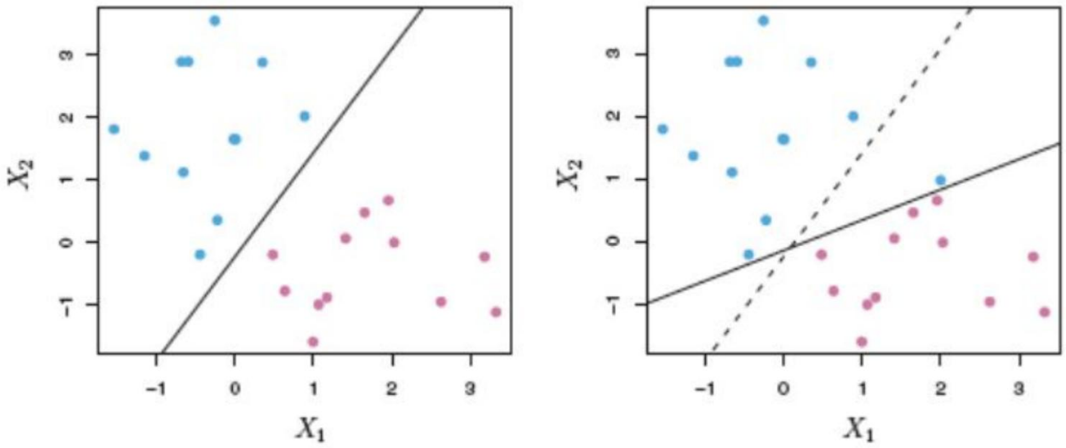
classified with less confidence.



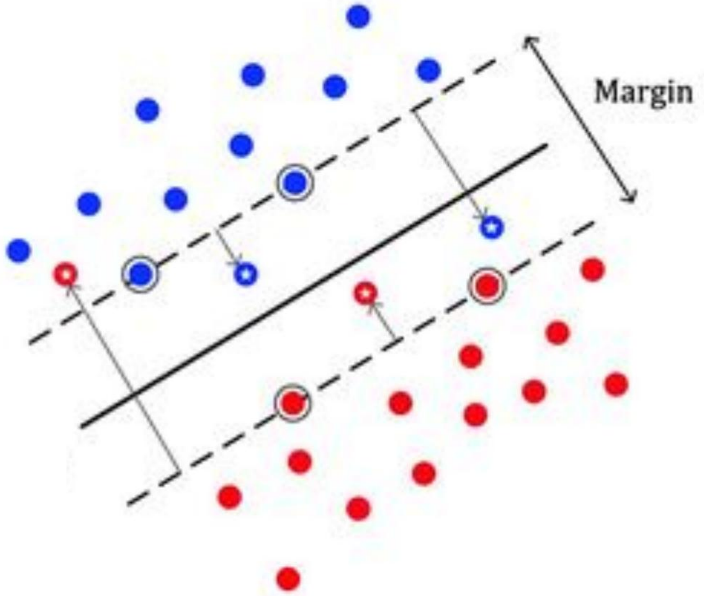
Is the optimal separation hyperplane always optimal?

Wouldn't it be worth using the initial hyperplane (the one that didn't take into account the example that makes the margin very small) instead of using the new hyperplane that divides but

Does it lead to a less reliable test?



Classifier with tolerant margin



Classifier with tolerant margin

Goals:

- That the individual observations are classified with robustness (that the distance to the hyperplane is not critical)

- Better classification of most training examples (assuming non-linearly separable classes).

Classifier with tolerant margin

Find the values of b and ÿ $_1$, ..., ÿn such that they maximize M subject to

- $y_i * (b_0 + b_1 x_{i,1} + b_2 x_{i,2} + \ldots + b_p x_{i,p}) \geq M * (1 - \epsilon_i), \ \forall i, 1 \leq i \leq n,$
- $\sum_{j=1}^{p} b_j^2 = 1.$
- $\forall_i, \epsilon_i \geq 0 \wedge \sum_{i=1}^{n} \epsilon_i \leq C.$

where C is a method tuning parameter.

Each ÿ allows you to classify example xi in the wrong place if necessary.

It could go through:

- be within the class margin
- being on the wrong side of the hyperplane.

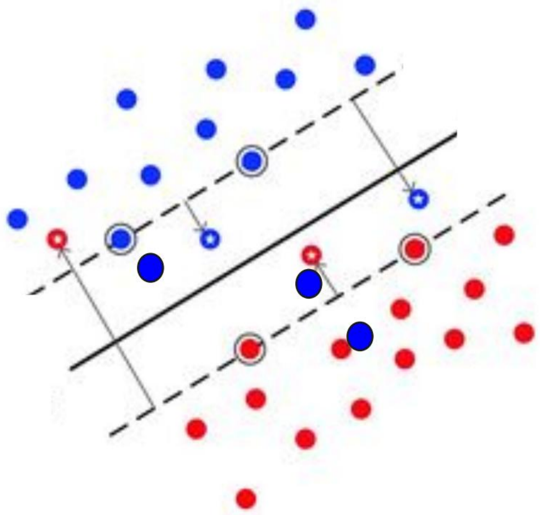
Classifier with tolerant margin

Given an observation xi if

$$y_i * (b_0 + b_1 x_{i,1} + b_2 x_{i,2} + \dots + b_p x_{i,p}) \geq M * (1 - \epsilon_i), \forall i, 1 \leq i \leq n,$$

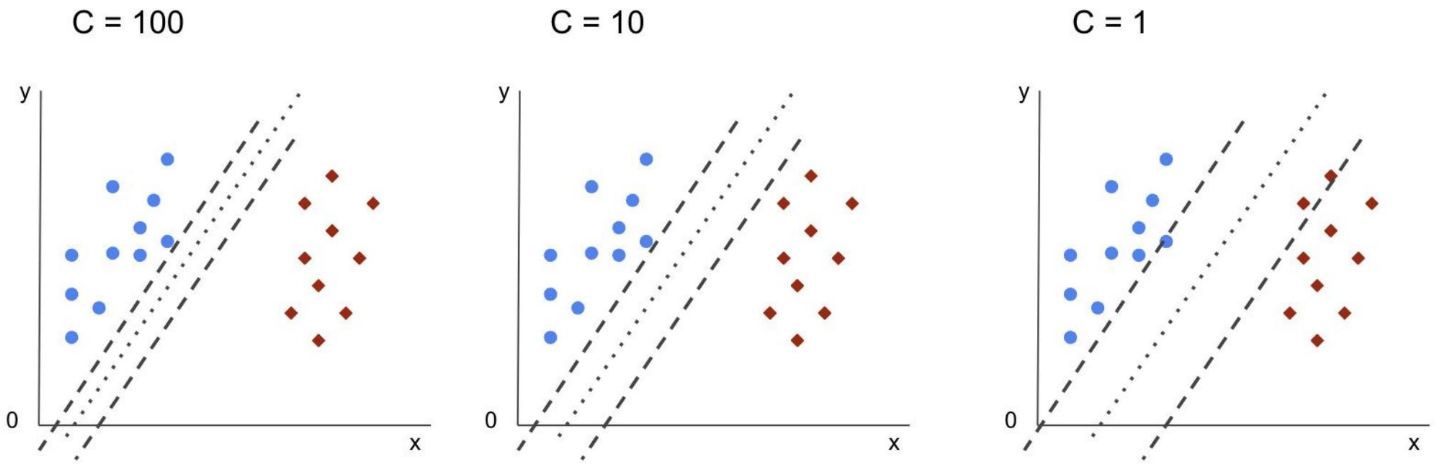
is satisfied for

- ÿ i = 0 ÿ the observation is on the **correct side** of the **margin**
- ÿ i > 0 ÿ the observation is on the **wrong side** of the **margin**
- ÿ i > 1 ÿ the observation is on the **wrong side** of the **hyperplane**



Classifier with tolerant margin

The value of **C** is a parameter that says how much the observations (as a whole) will be allowed to **violate the margin or hyperplane.**



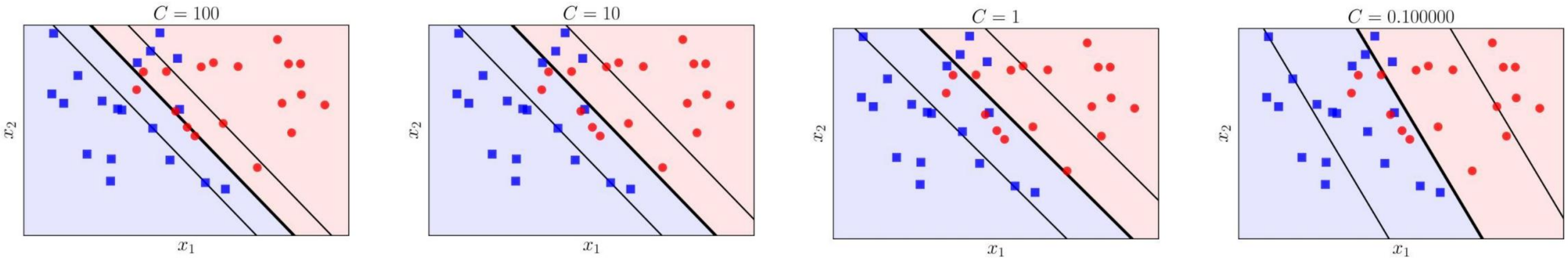
Classifier with tolerant margin

- If C = 0, the Tolerant Margin Classifier becomes a Margin Classifier

  maximal.

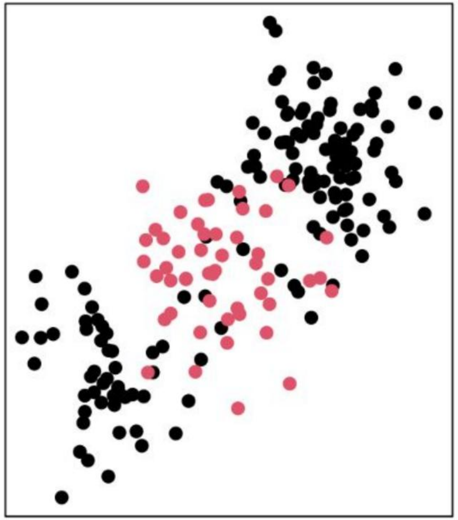
- One way to find C is with cross validation.

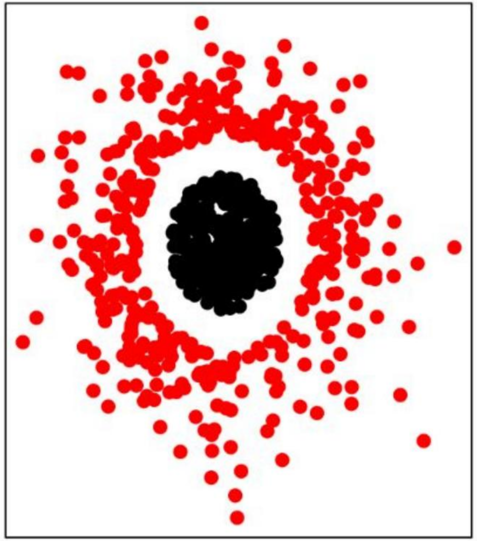
.



Classification with nonlinear decision boundaries

What we have seen so far is effective when the separation between classes is linear, but it doesn't work well in non-linear cases.





Classification with nonlinear decision boundaries

Let's consider a training set where

its examples xi are of dimension p = 2 and its class is

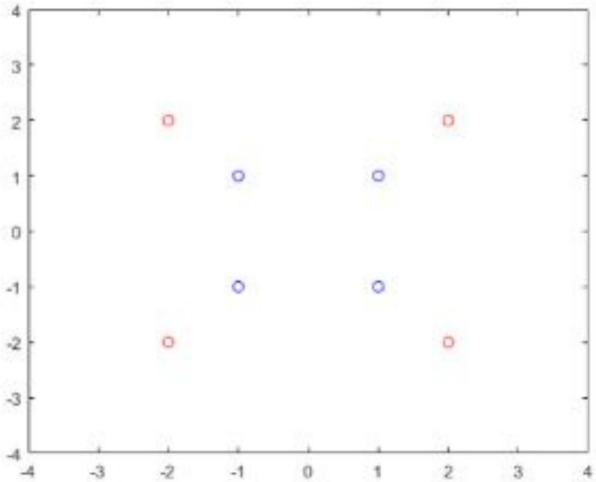
yi ÿ {ÿ1, 1} (-1 in red and 1 in blue):

X = {(ÿ2, ÿ2), (- 2, 2), (2, ÿ2), (2, 2), (- 1, ÿ1), (- 1, 1), (1, - - 1),(1, 1)}

Y = {ÿ1, ÿ1, ÿ1, ÿ1, 1, 1, 1, 1}

where a hyperplane cannot be established

linear separation between one class and the other.

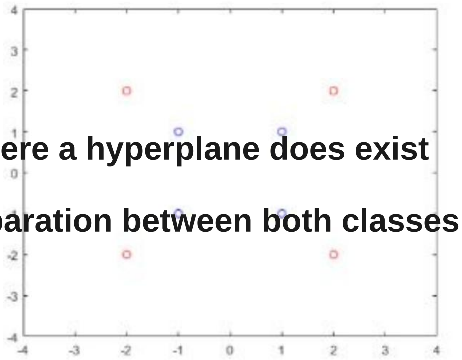


Classification with nonlinear decision boundaries

But if the examples xi = (xi1 , xi2) of _ _ _ _

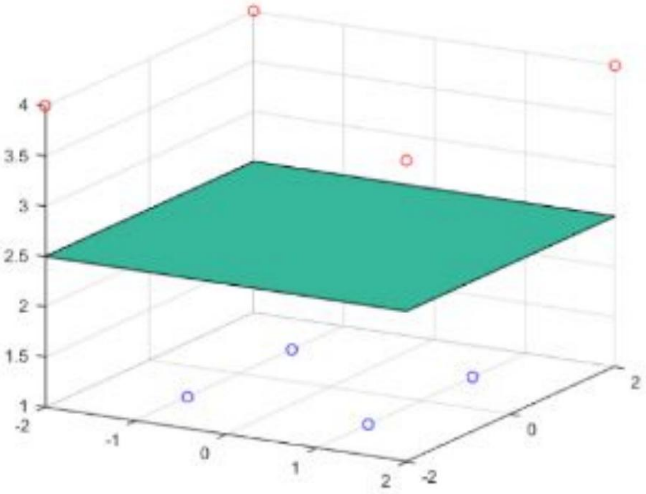
X = {(ÿ2, ÿ2, 4), (- 2, 2, 4), (2, ÿ2, 4), (2, 2, 4), (- 1, ÿ1, 1), (ÿ1, 1, 1),(1, ÿ1, 1),(1, 1, 1)}

Y = {ÿ1, ÿ1, ÿ1, ÿ1, 1, 1, 1, 1}



Where a hyperplane does exist

separation between both classes.



Classification with nonlinear decision boundaries

For example

If we have examples in a p dimension xi1, xi2, .., xip we could represent them in a 2p dimension according to

$x_{i1}, x^2_{i1}, x_{i2}, x^2_{i2}, .., x_{ip}, x^2_{ip}$ where there could be a hyperplane of dimension 2p-1 that will separate them.

Classification with nonlinear decision boundaries

In this case, the problem to be solved is to find the values of b = b0 , b11, b12, .., bp1, bp2 and ÿ1, .., ÿn such that they maximize M subject to

- $y_i * (b_0 + \sum_{j=1}^{p} b_{j1} * x_{ij} + \sum_{j=1}^{p} b_{j2} * x_{ij}^2) \geq (M - \epsilon_i)$, $\forall i, 1 \leq i \leq n$
- $\sum_{j=1}^{p} \sum_{k=1}^{2} b_{jk}^2 = 1.$
- $\epsilon_i \geq 0, \forall_i, 1 \leq i \leq n \wedge \sum_{i=1}^{n} \epsilon_i \leq C.$

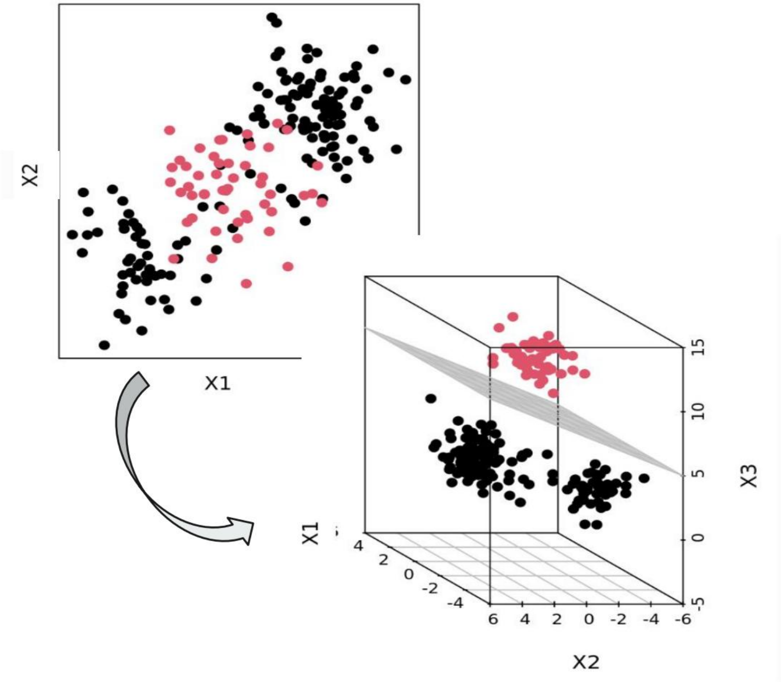
where C is a method tuning parameter.

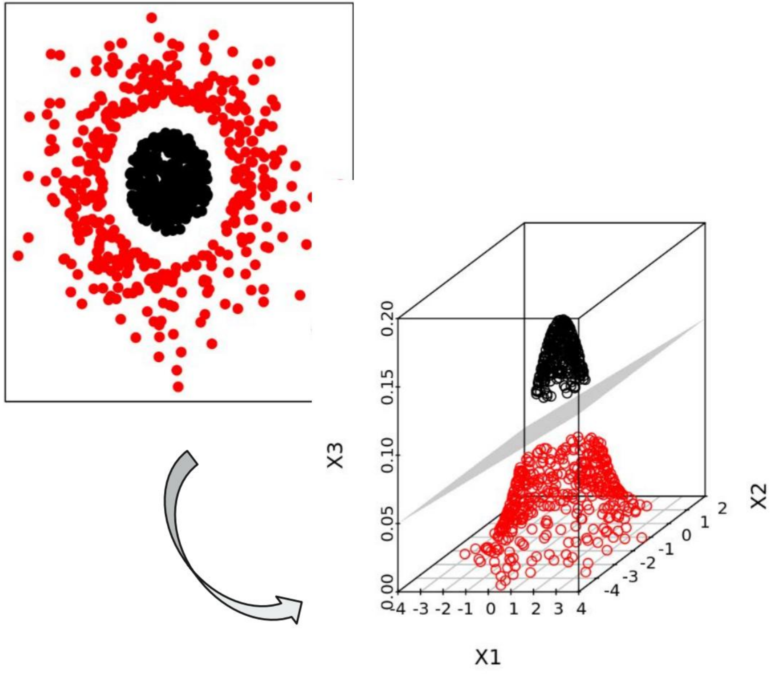
Classification with nonlinear decision boundaries

In the example, we use $x^2_{eh,}$ but we could use another degree or another function.

The idea is to obtain a space where there is linear separability, which implies nonlinear separability in the original space of the examples of dimension p.

Classification with nonlinear decision boundaries





Support Vector Machine

This proposal is a generalization of the Classification with limits of non-linear decisions and the way to generalize this idea is by introducing the concept of **Kernel.**

Support Vector Machine

The maximum margin classifier only depends on the vectors, ÿk such that:

support, then, ÿ, ÿ1 , . . .

Given an observation xÿ, if we want to know which class it belongs to,

we calculate f (xÿ) as:

f (xÿ) = b0 + ÿki=1 ÿi ÿxÿ, xi ÿ

where the xi are the support vectors.

Support Vector Machine

We can write af(x) as:

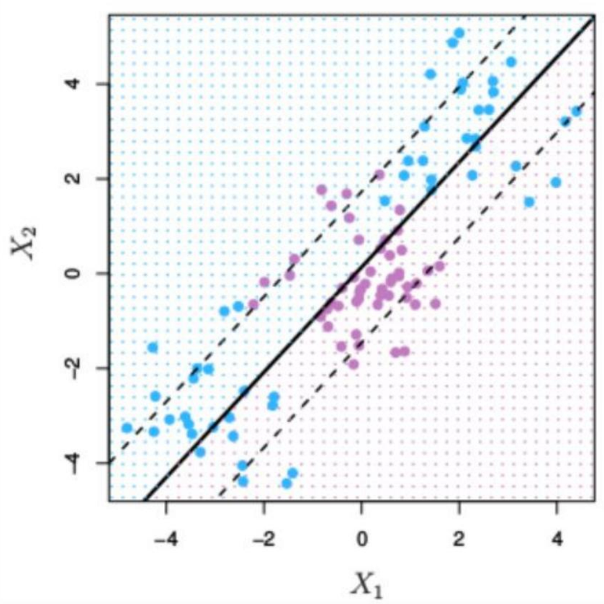
$$f(x) = b_0 + \sum_{i=1}^{k} \alpha_i K(x, x_i)$$

where K(x, xi) = ÿx, xi ÿ and **K** is called **the Kernel.**

Support Vector Machine

Linear core

$$K(\mathbf{x}, \mathbf{x'}) = \mathbf{x} \cdot \mathbf{x'}$$



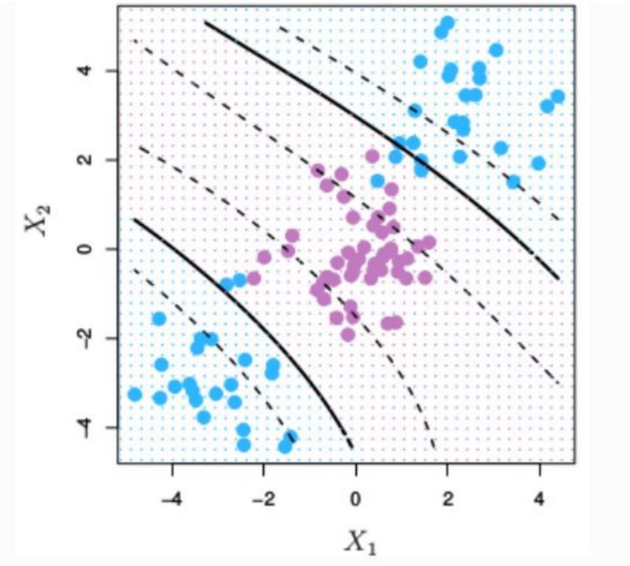
Support Vector Machine

Polynomial kernel

$$K(x', x_i) = (1 + \sum_{j=1}^{p} x_{ij} x'_j)^d$$

where d is the degree of the polynomial.

As d increases there will be more flexibility to find a linear separation in the new space of the examples (the expanded space).



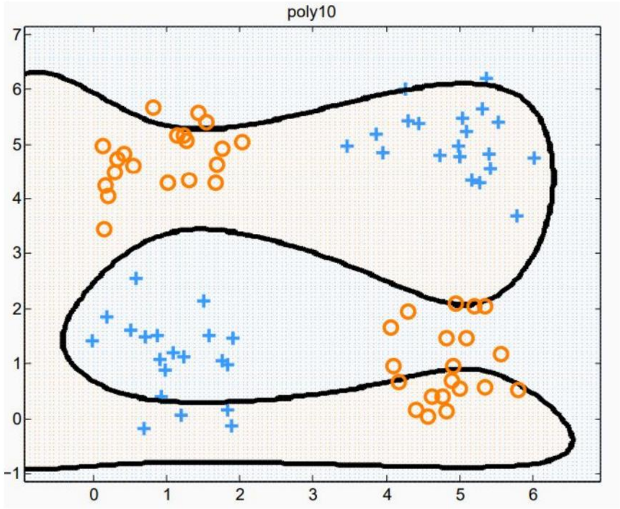
Support Vector Machine

Polynomial kernel

$$K(x', x_i) = (1 + \sum_{j=1}^{p} x_{ij} x'_j)^d$$

where d is the degree of the polynomial.

As d increases there will be more flexibility to find a linear separation in the new space of the examples (the expanded space).

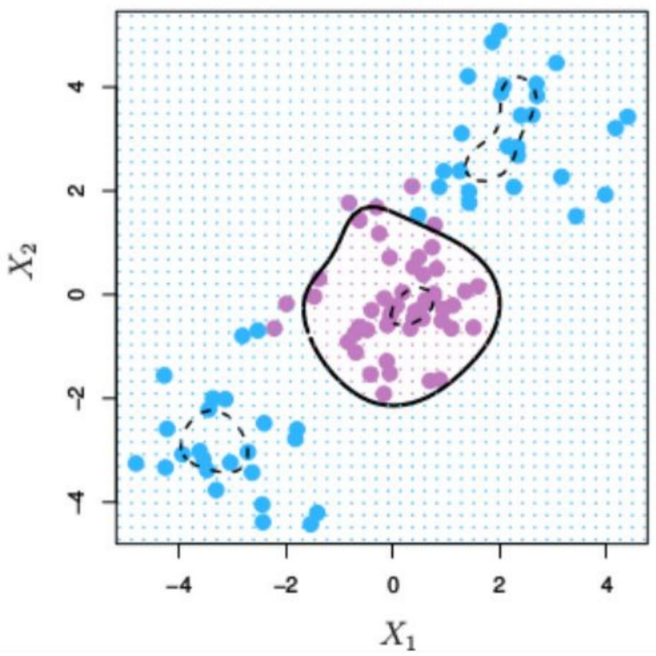


Support Vector Machine

Radial core

$$K(x', x_i) = e^{-\gamma \sum_{j=1}^{p} (x_{ij} - x'_j)^2}$$

where ÿ is a positive constant.

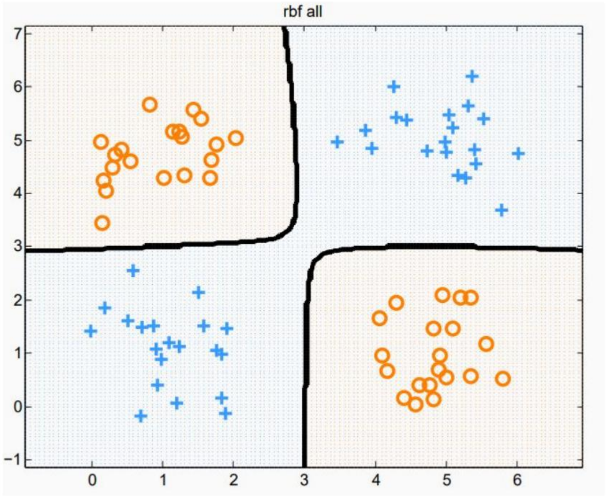


Support Vector Machine

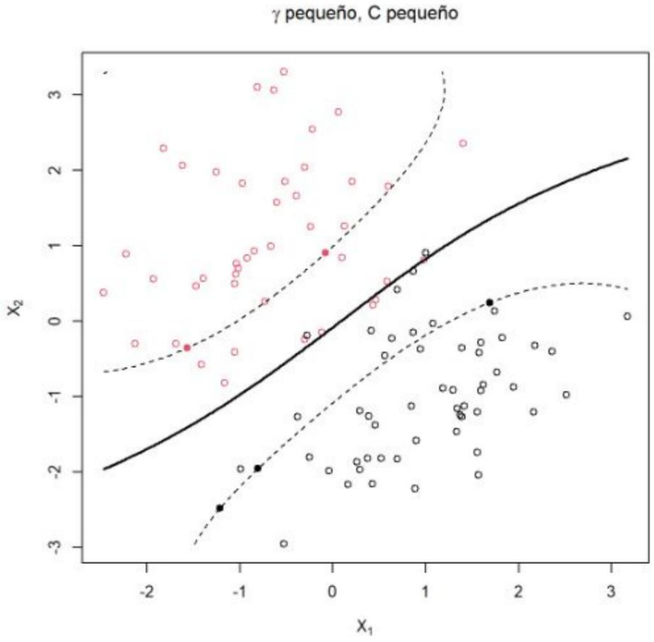
Radial core

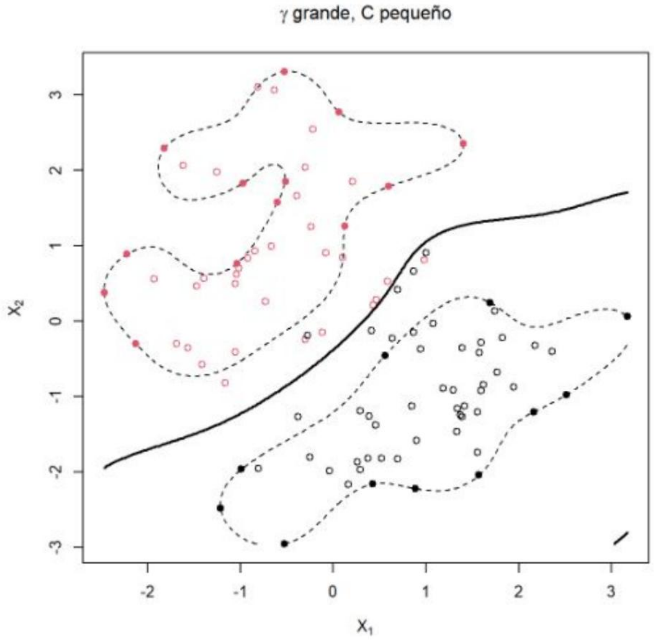
$$K(x', x_i) = e^{-\gamma \sum_{j=1}^{p} (x_{ij} - x'_j)^2}$$

where ÿ is a positive constant.



Support Vector Machine





Support Vector Machine

Even if we change the Core, the formulation for the problem does not change.

The calculation of f(x) will continue to be:

$$f(x) = b_0 + \sum_{i=1}^{n} \alpha_i K(x, x_i)$$

Support vector machine multiclass

What happens when our training set has more than 2 classes?

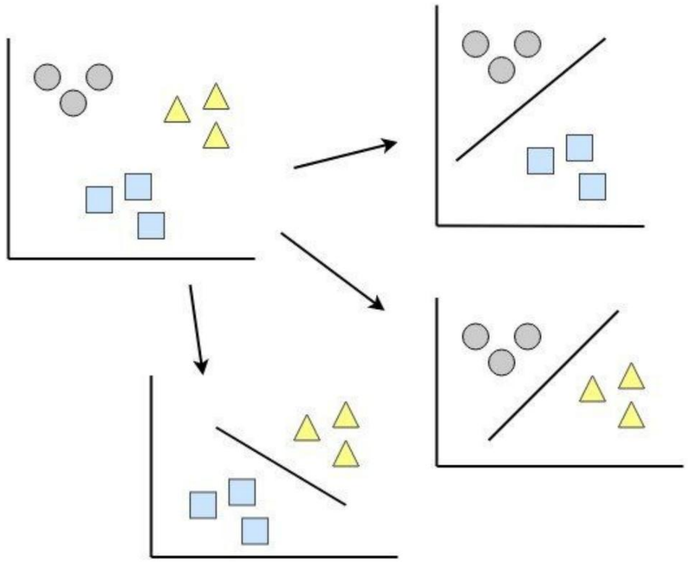
There are two approaches:

- One against one
- One against the rest

Support vector machine multiclass

One against one

An SVM is constructed for each distinct pair of classes (one class will be assigned +1 and the other -1). The rest of the examples from the remaining classes will be ignored.



When a new observation is presented for each constructed SMV, a response is obtained and the class with the most votes is chosen.

SVM - Support multiclass vector machine

one against all

An SVM is constructed for each class (the class examples will be assigned ÿa +1 and the rest of the training set examples will be assigned ÿa -1).

When a new observation of each SMV constructed is presented, a

answer and the resulting class will be the one that corresponds to the SVM whose f (xÿ) is greater.