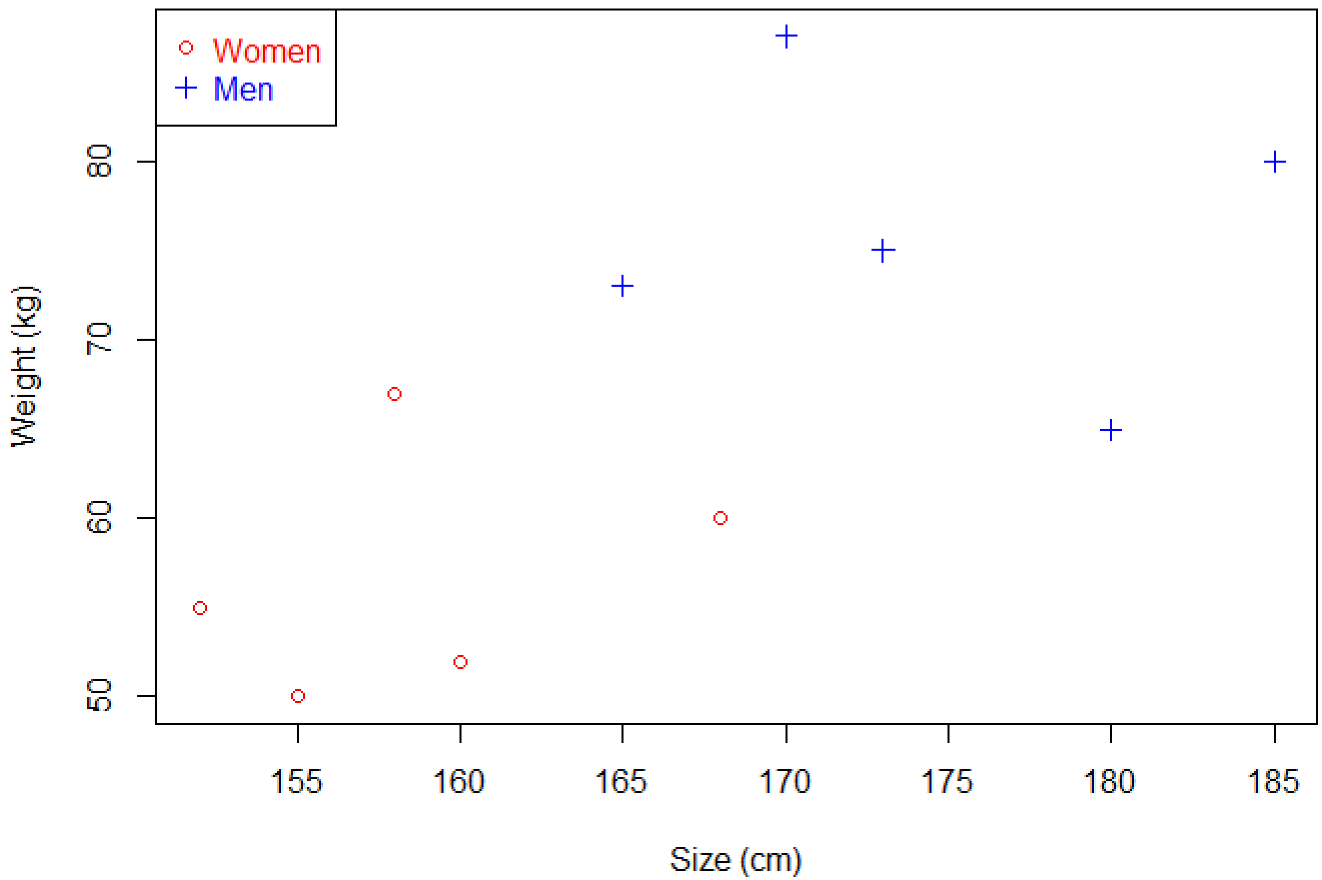
Support Vector Machine (SVM)

Much of this material is adapted from:

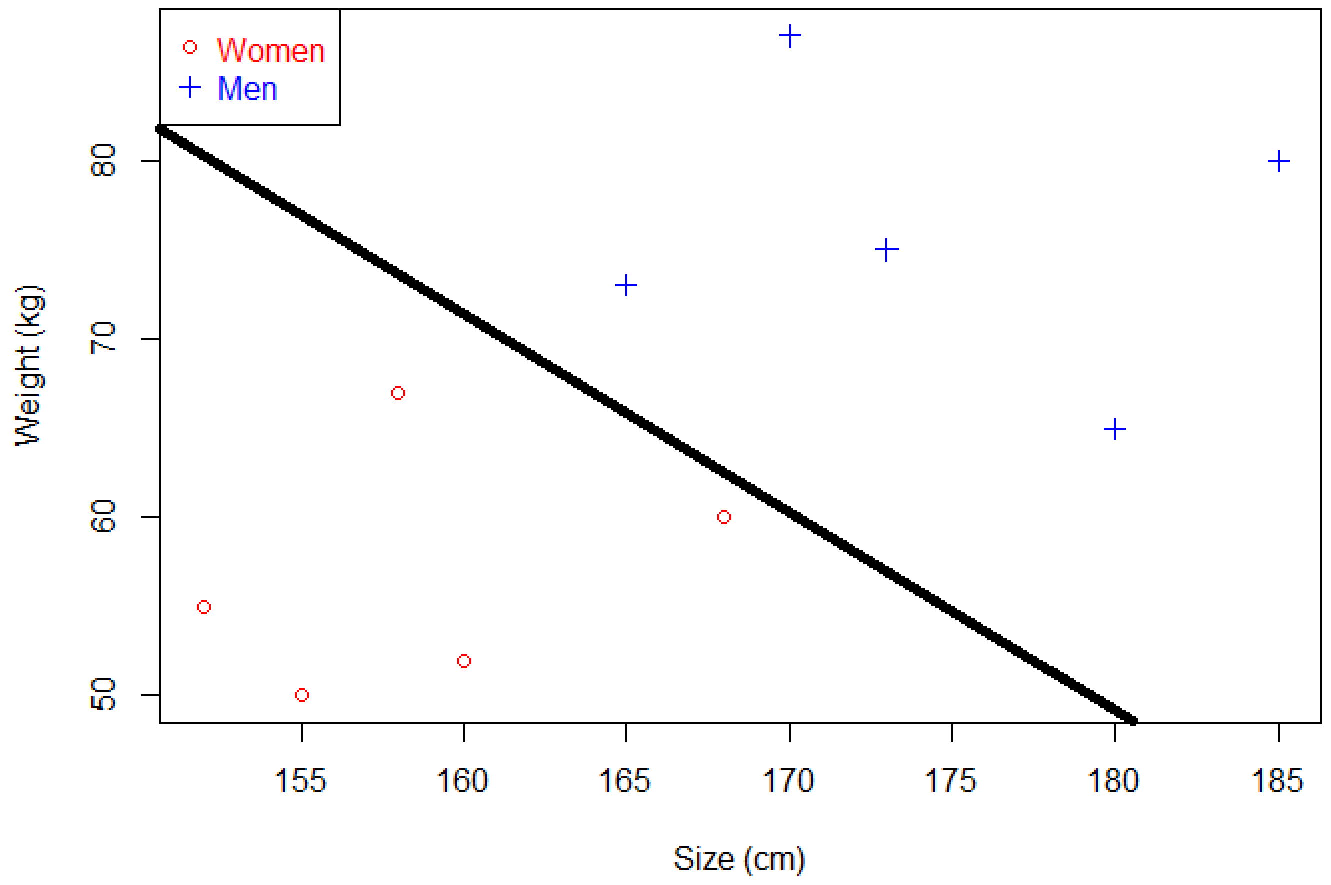
- https://www.svm-tutorial.com/2017/02/svms-overview-support-vector-machines/
- https://www.youtube.com/watch?v=9_DJ4KvyYoo

Many of the images were taken from the internet



Is it possible to separate the data? yes

How? We could trace a line.


Women
Men
Weight (kg)
50
60
70
80
155
160
165
170
175
180
185
Size (cm)

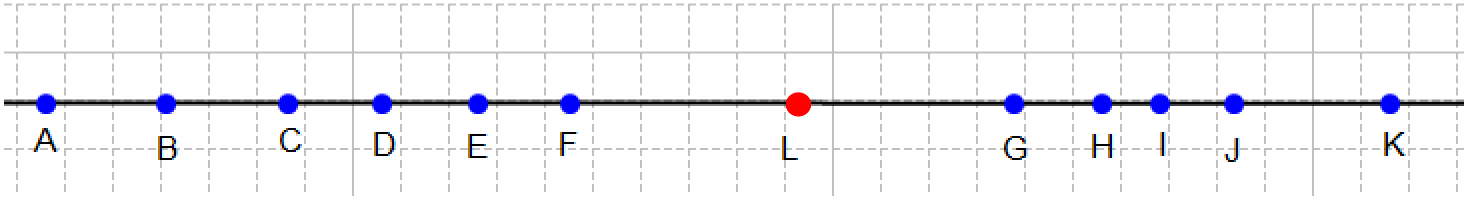
- All the data points representing men will be above the line, and all the data points representing women will be below the line.

- Such a line is called a **separating hyperplane.**

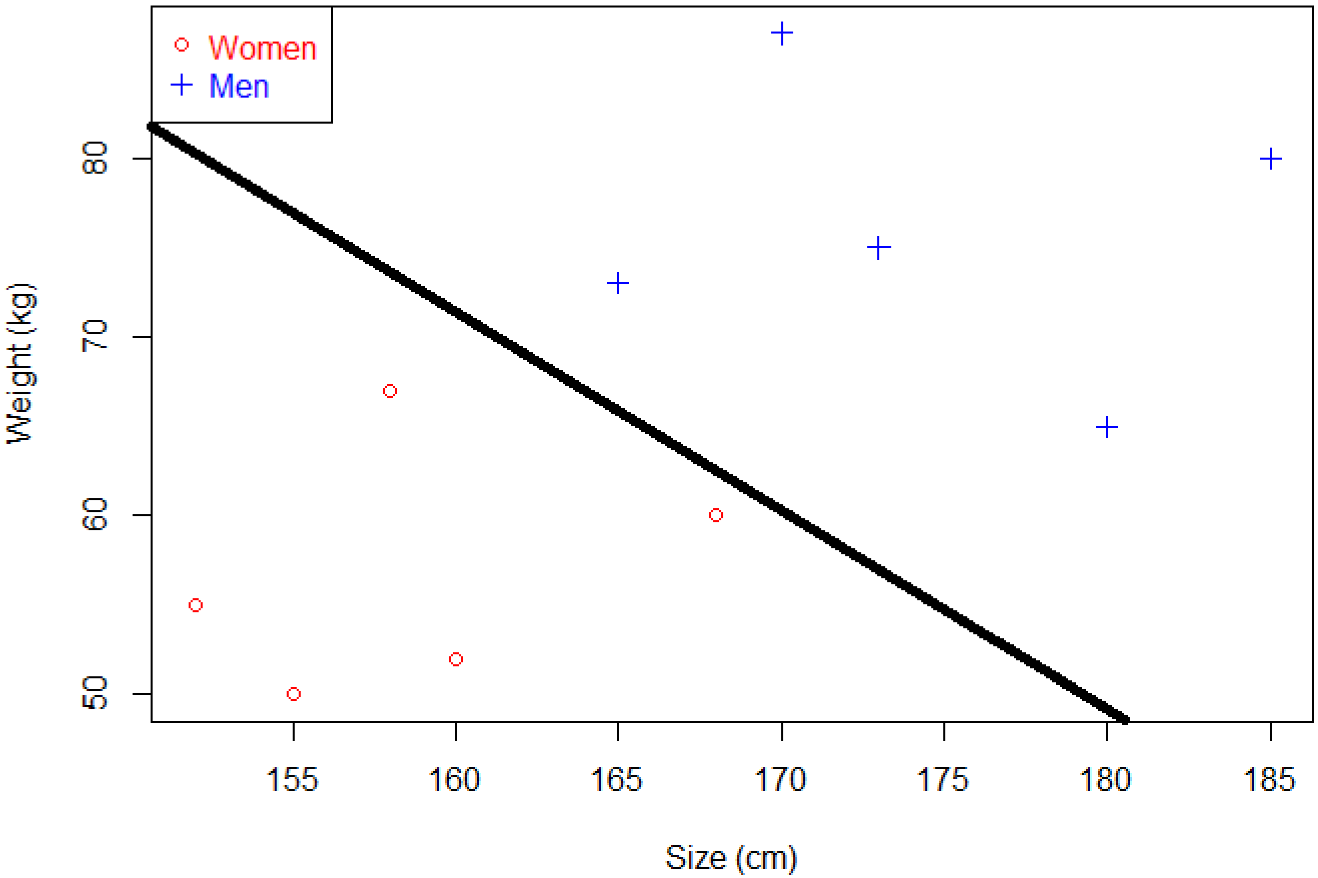
If it is just a line, why do we call it an hyperplane ?

- An hyperplane is a generalization of a plane.
- in one dimension, an hyperplane is called a point
- in two dimensions, it is a line
- in three dimensions, it is a plane
- in more dimensions you can call it an hyperplane

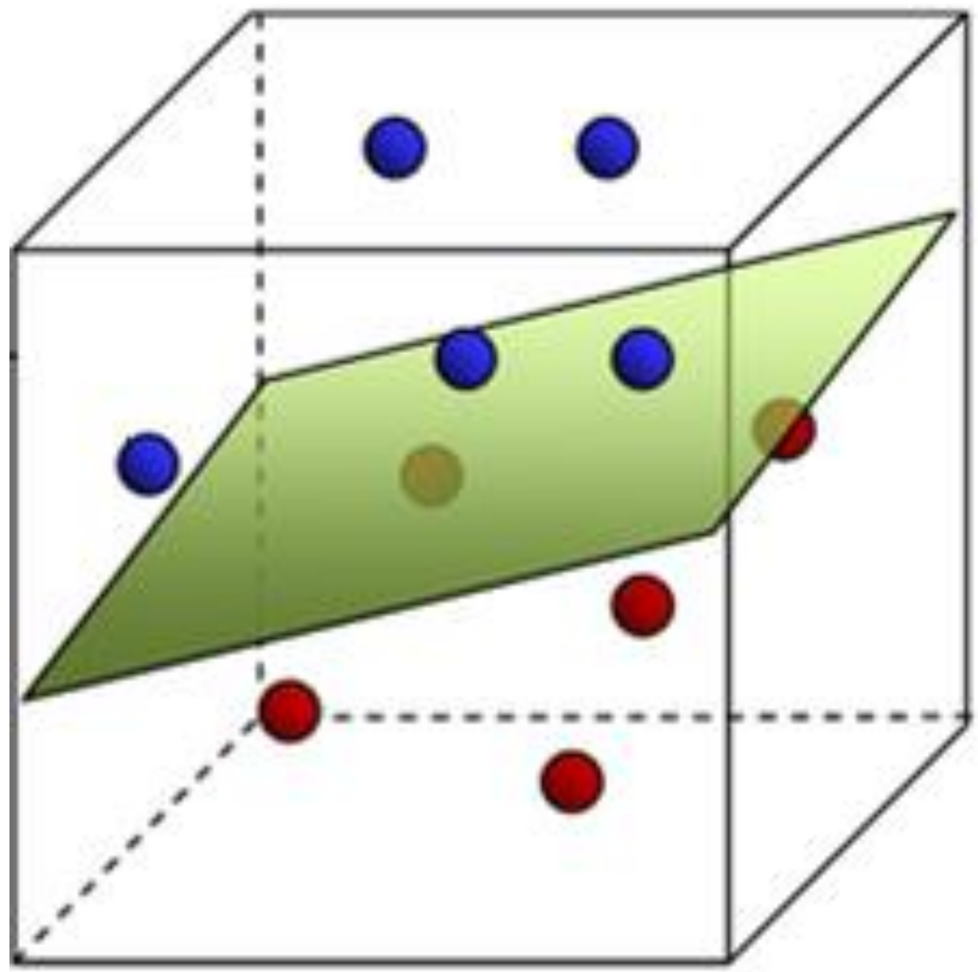
R^1


A
B
C
D
E
F
L
G
H
I
J
K

R^2

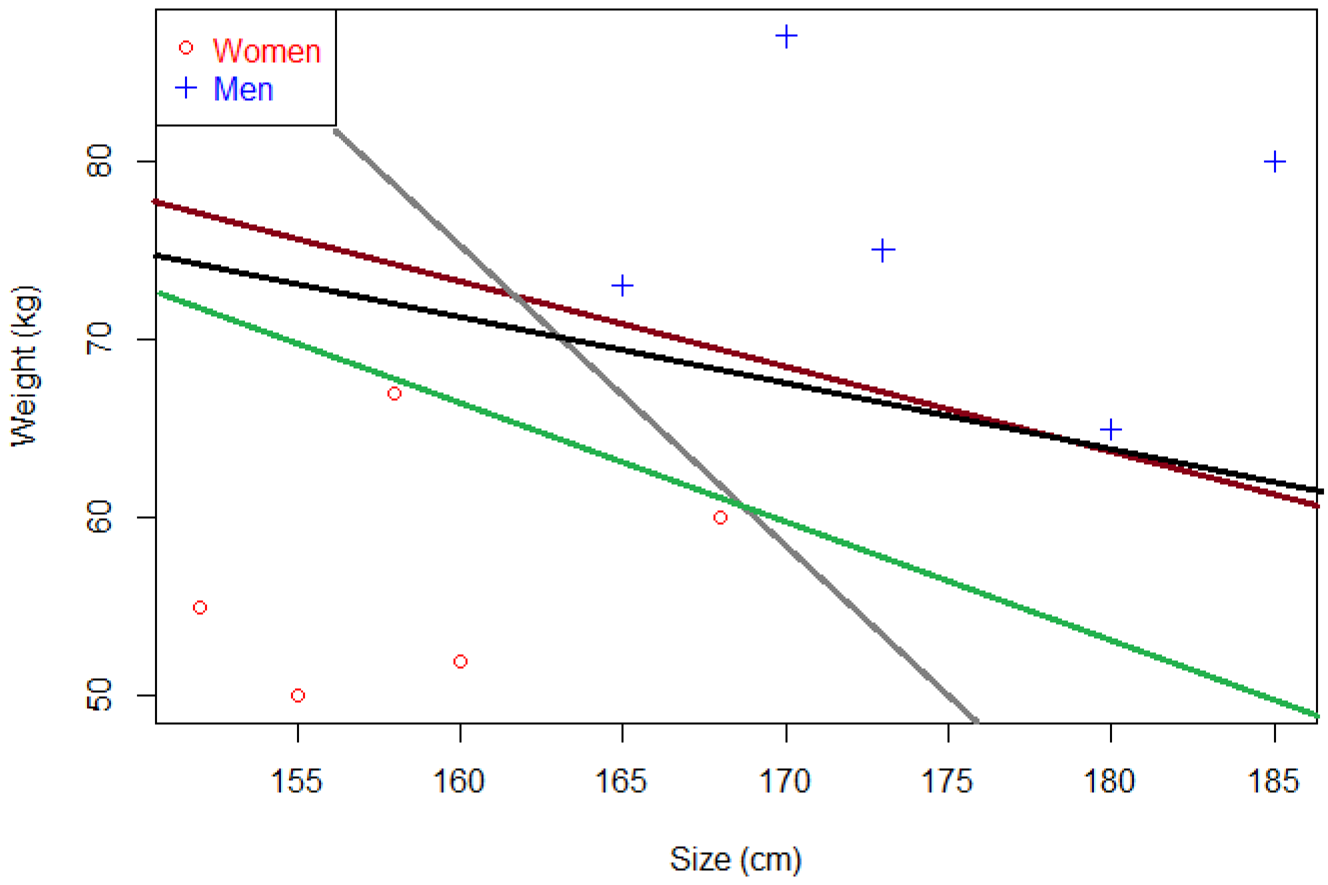

Women
Men
Weight (kg)
50
60
70
80
155
160
165
170
175
180
185
Size (cm)

R^3

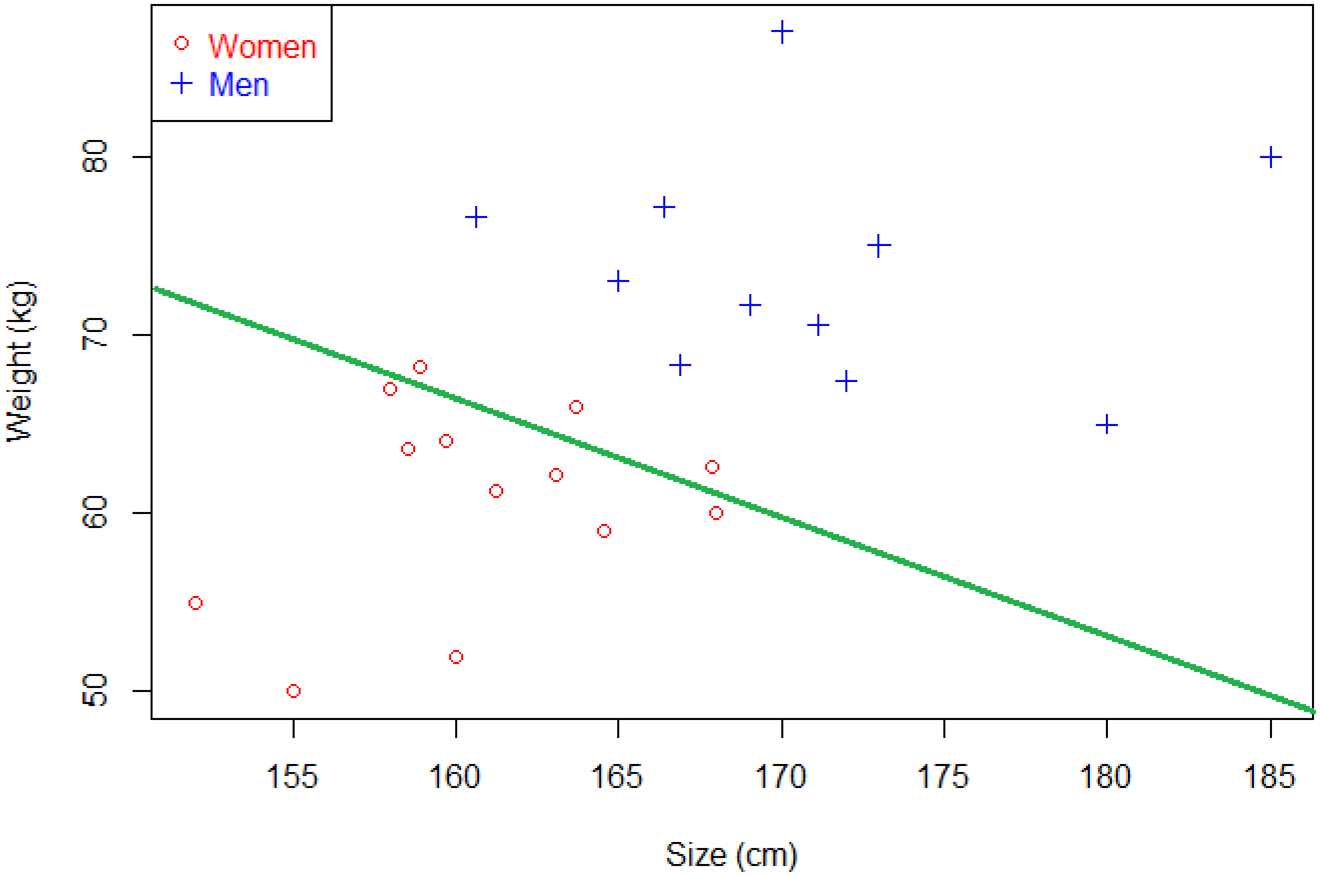


Source: Norman Di Palo, https://www.quora.com/How-do-you-explain-support-vector-machines-to-not-so-good-at-math-people

There can be a lot of separating hyperplanes. Which one is the best?



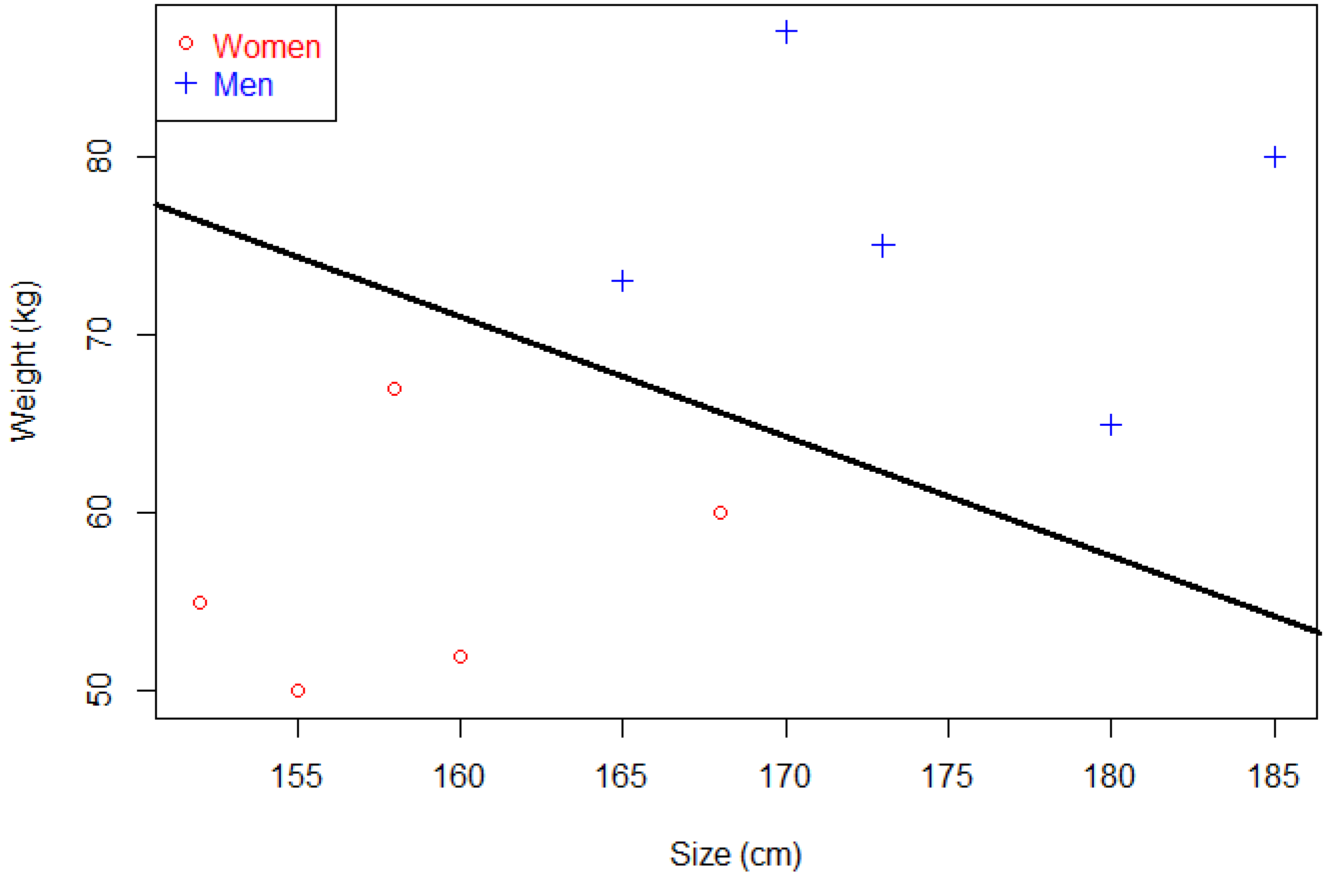
Suppose we select the green hyperplane and use it to classify data.



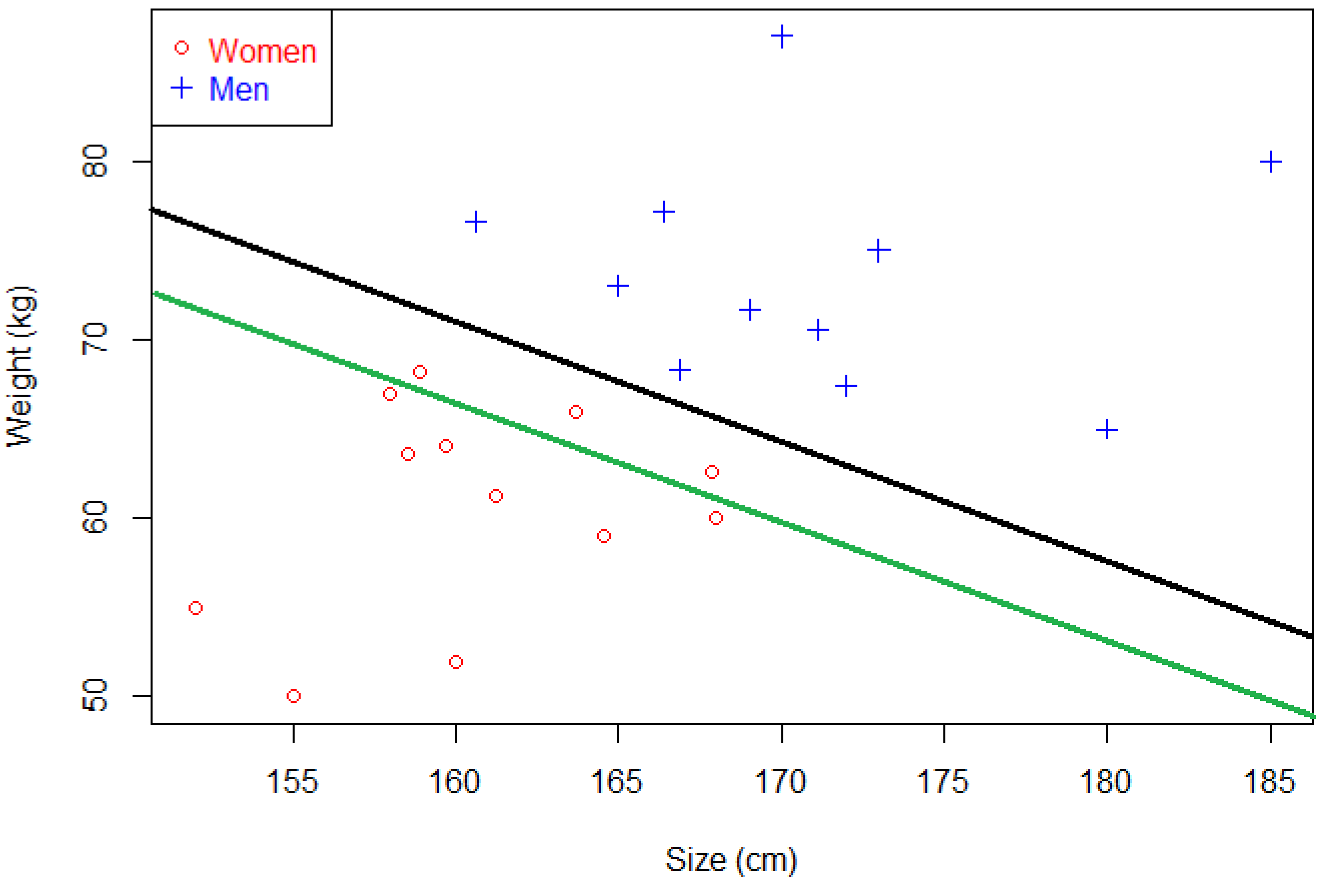
- It makes some mistakes → wrongly classify 3 women.
- Intuitively, we can see that:

  if we select an hyperplane which is close to the data points of one class, then it might not generalize well.

- So we will try to select an hyperplane **as far as possible from data points from each class**



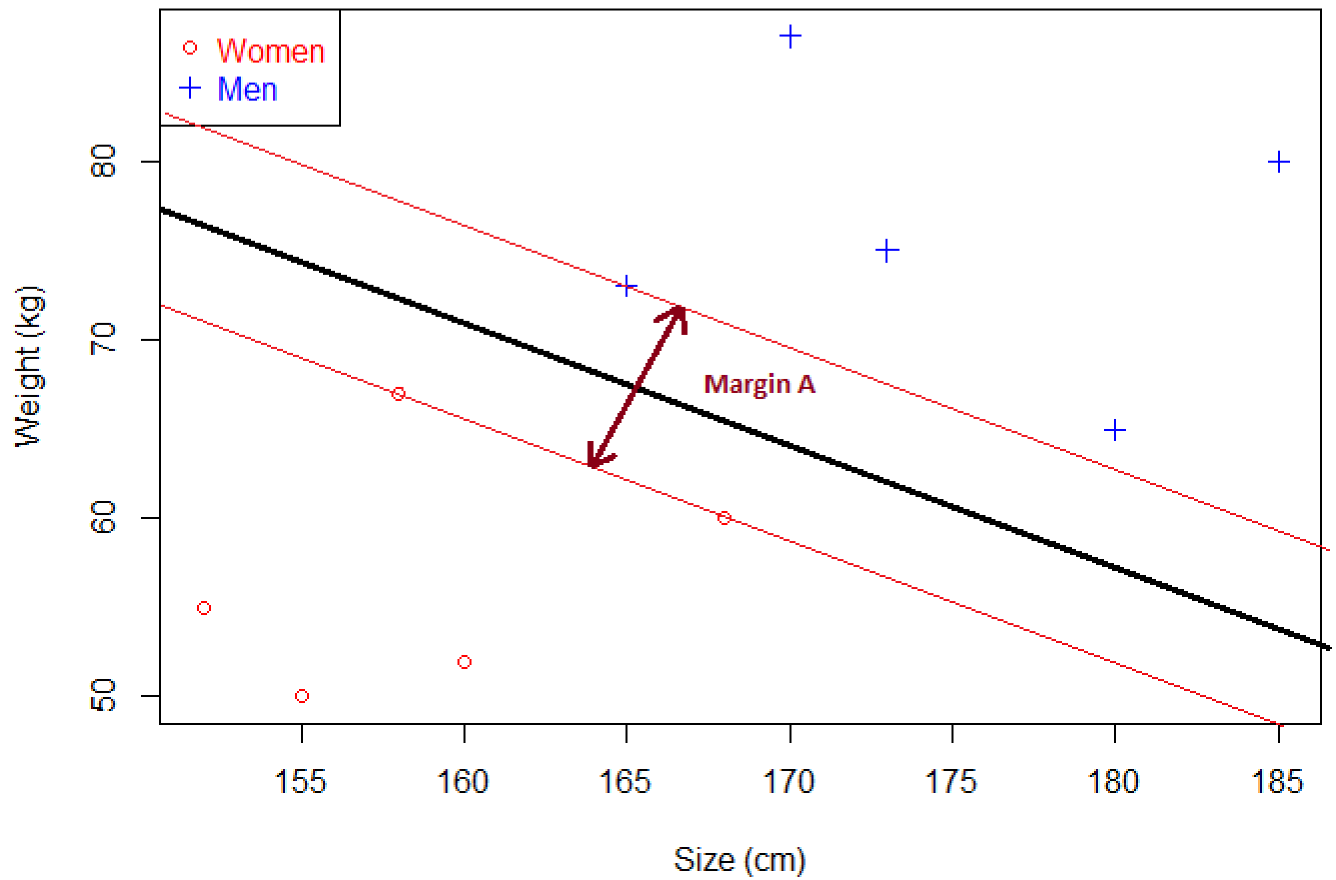
This black hyperplane is better than the green one.



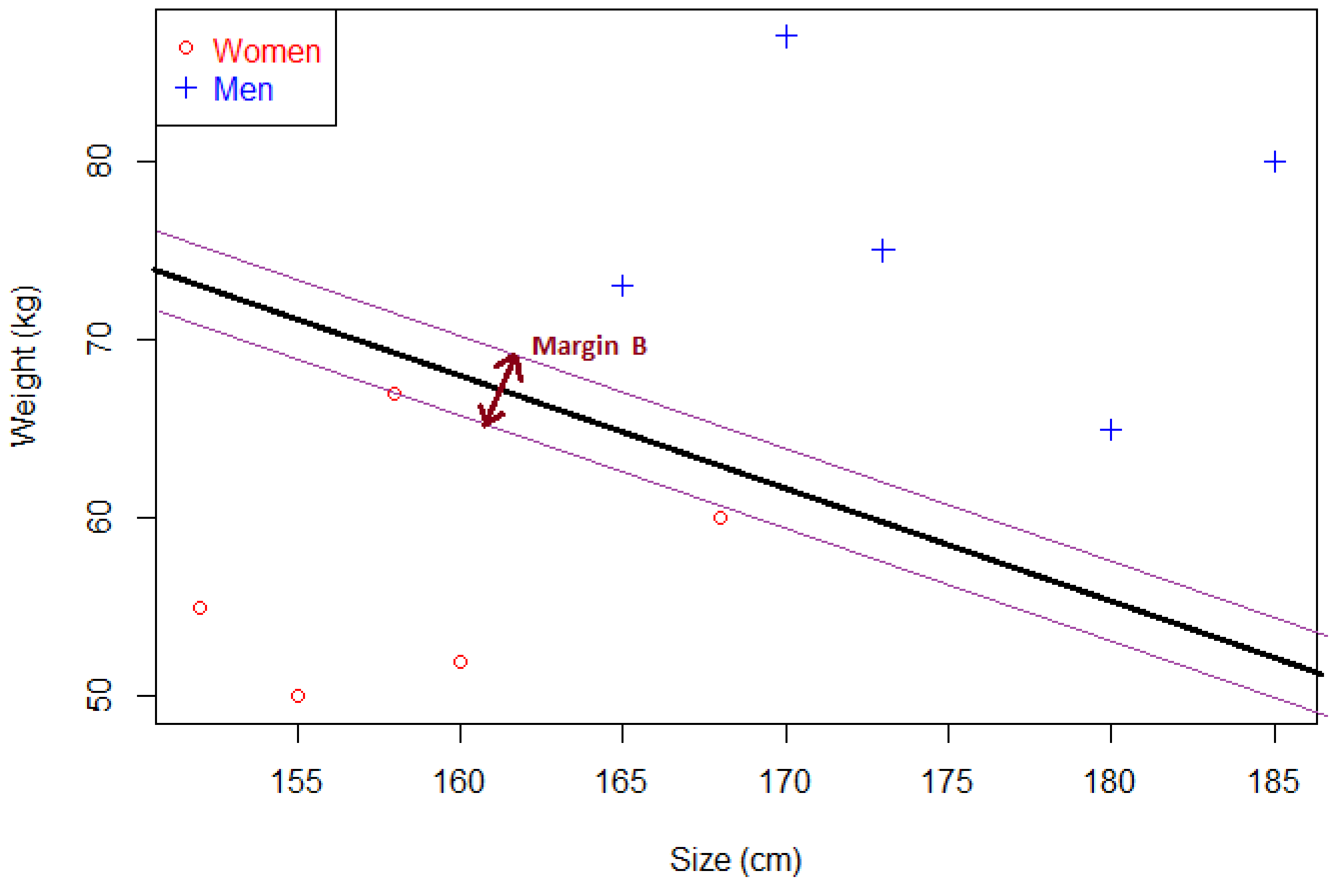
Women
Men
Weight (kg)
50
60
70
80
155
160
165
170
175
180
185
Size (cm)

- That's why the objective of a SVM is to **find the optimal separating hyperplane,** because:
- it correctly classifies the training data
- it is the one which will generalize better with unseen data (test data)

- Given a particular hyperplane, we can compute the distance between the hyperplane and the closest data point



Margin: double distance between the hyperplane and the closest data point



Margin B is smaller than Margin A

- If an hyperplane is very close to a data point, its margin will be small.
- The further an hyperplane is from a data point, the larger its margin will be.
- **the optimal hyperplane will be the one with the biggest margin.**
- the objective of the SVM **is to find the optimal separating hyperplane which maximizes the margin of the training data.**
- How do we calculate this margin?

An hyperplane

- Line equation: $y = ax + b$

- Hyperplane equation: $\mathbf{w}^T \mathbf{x} = 0$

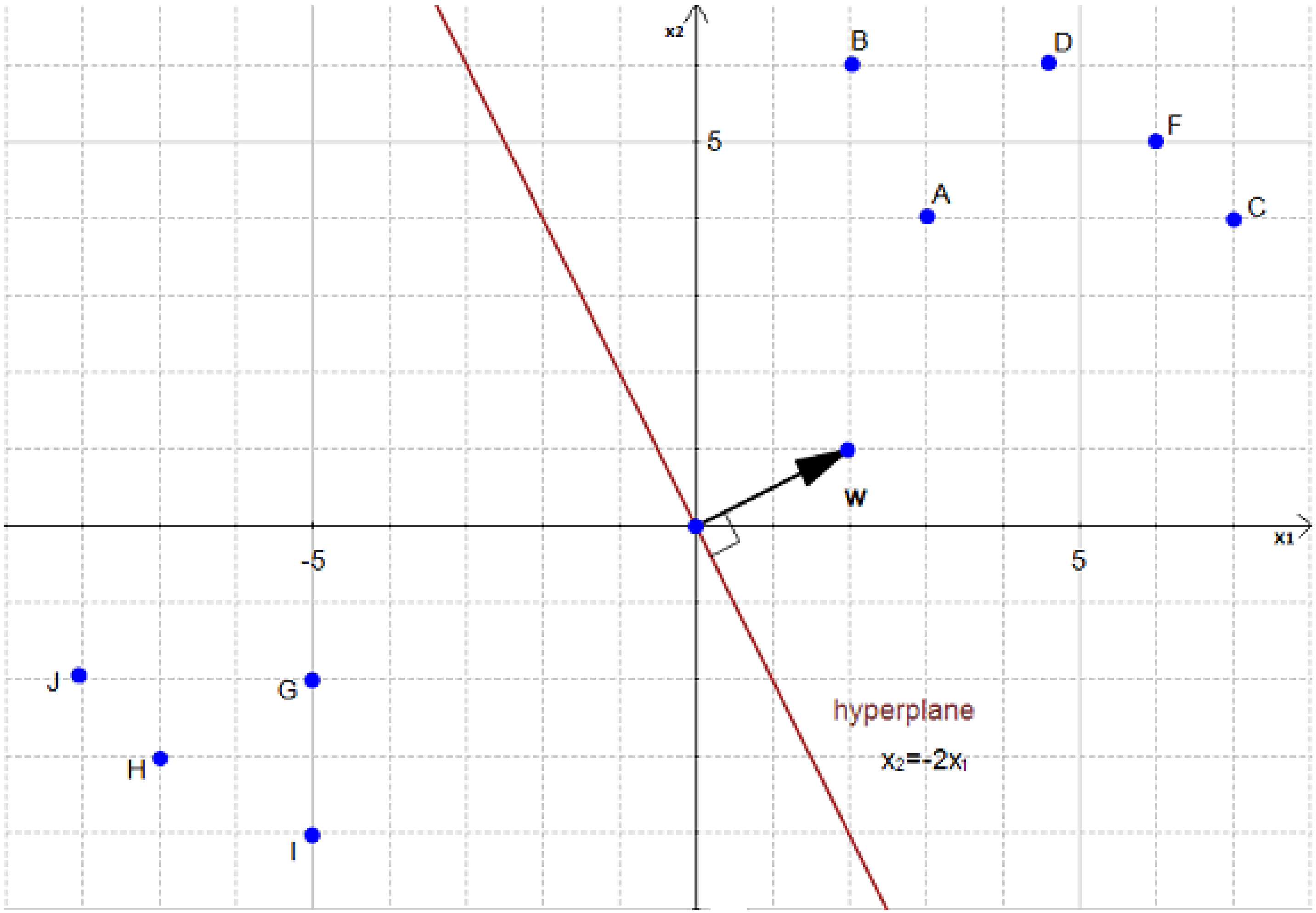
$$
\begin{aligned}
& y = ax + b \\
\Longleftrightarrow \quad & y - ax - b = 0
\end{aligned}
$$

Given two vectors $\mathbf{w} \begin{pmatrix} -b \\ -a \\ 1 \end{pmatrix}$ and $\mathbf{x} \begin{pmatrix} 1 \\ x \\ y \end{pmatrix}$

$$\mathbf{w}^T\mathbf{x} = -b \times (1) + (-a) \times x + 1 \times y$$

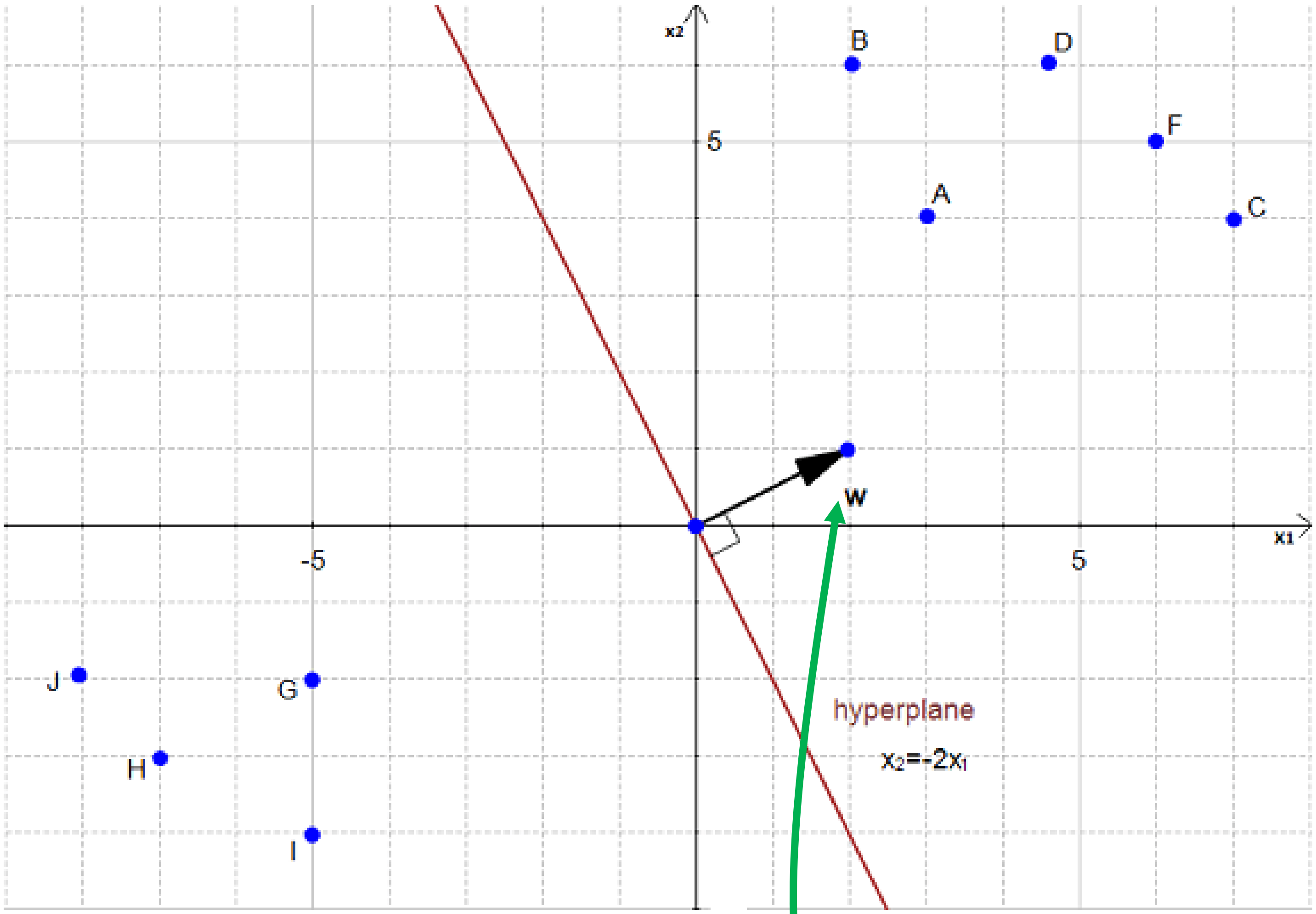
$$\mathbf{w}^T\mathbf{x} = y - ax - b$$

The two equations are just different ways of expressing the same thing.



The equation of the hyperplane is : $x_2 = -2x_1$

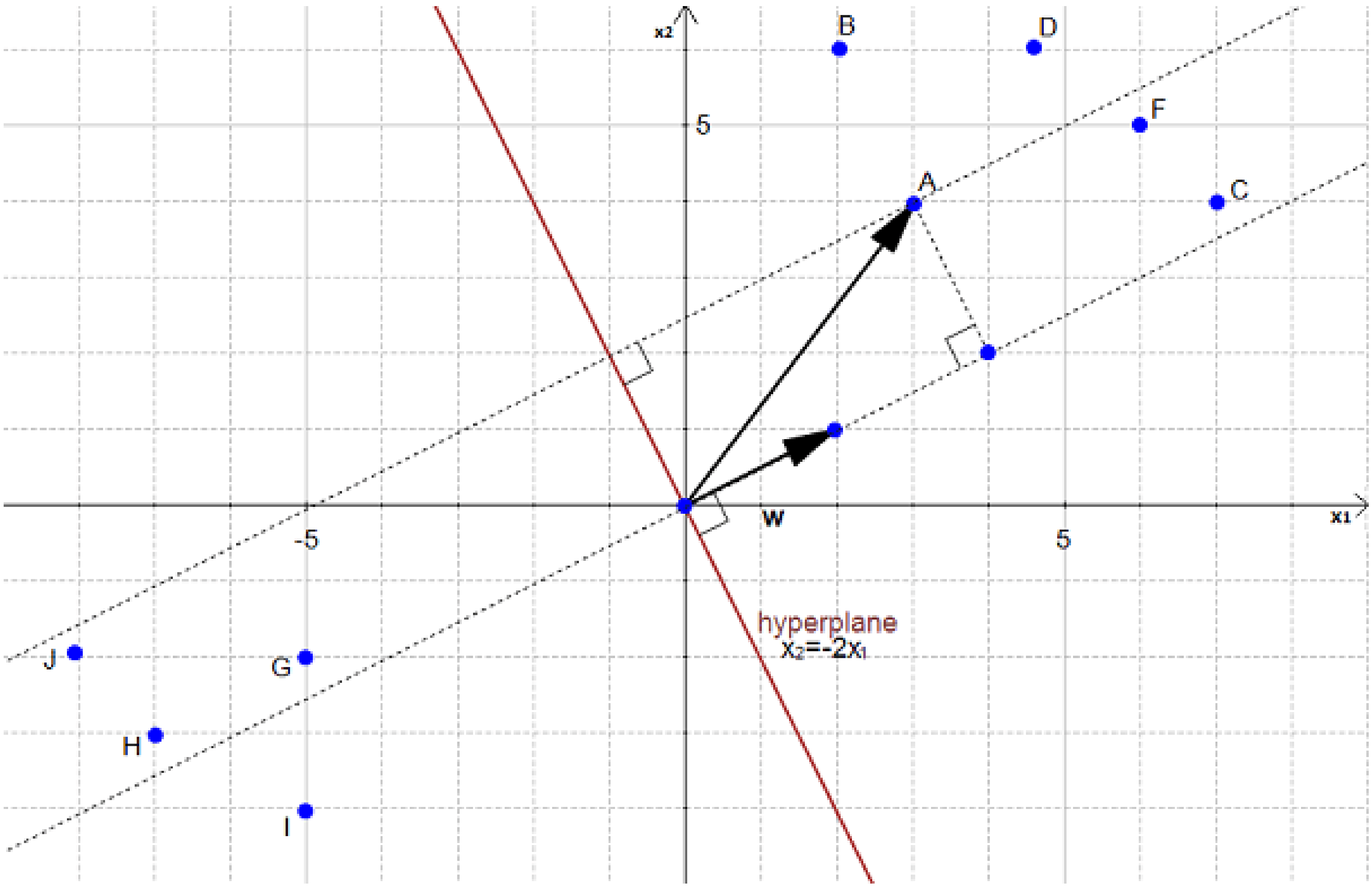
which is equivalent to: $\mathbf{w}^T \mathbf{x} = 0$ with $\mathbf{w} \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ and $\mathbf{x} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$



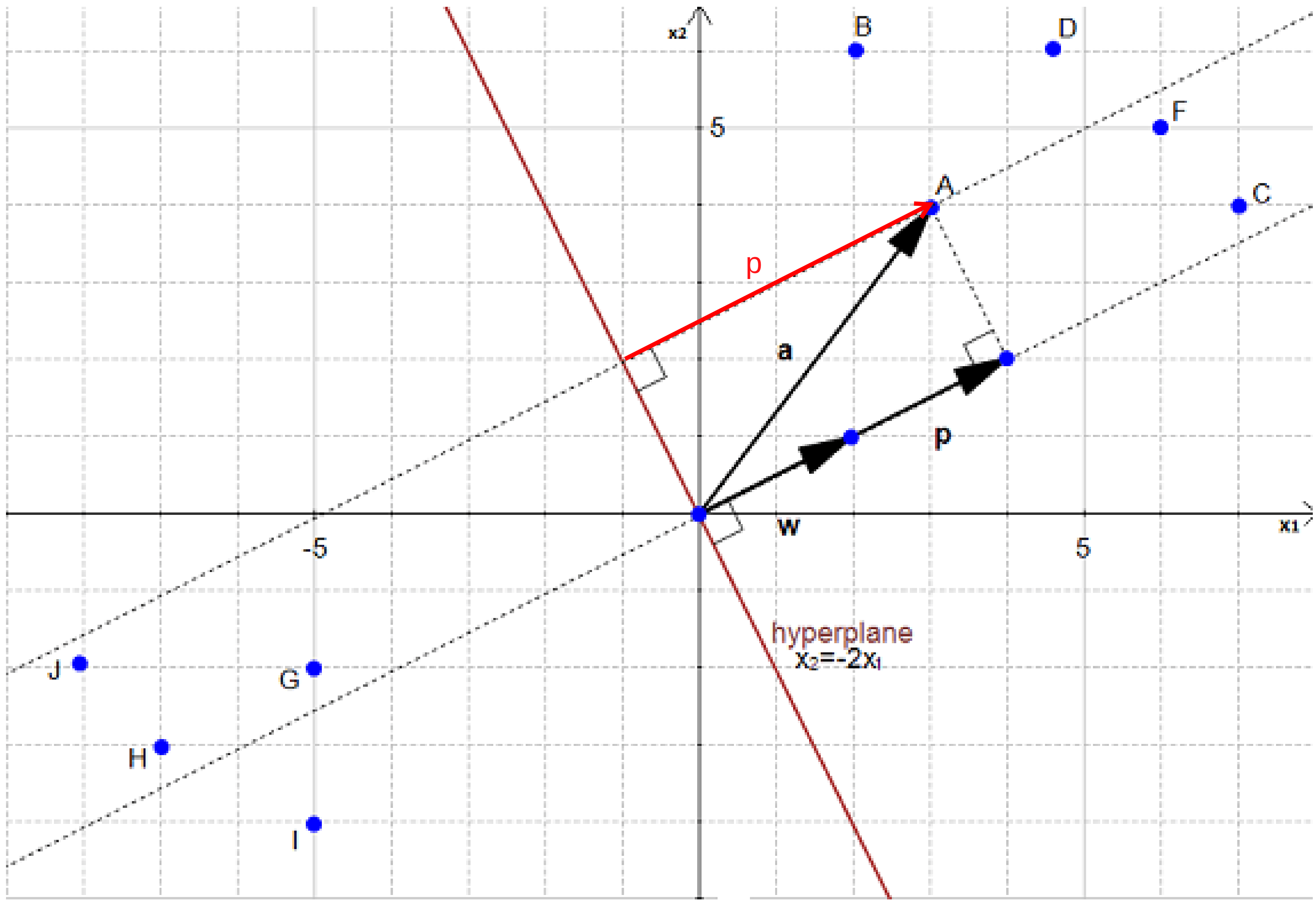
The equation of the hyperplane is : $x_2 = -2x_1$

which is equivalent to: $\mathbf{w}^T\mathbf{x} = 0$ with $\mathbf{w}\begin{pmatrix} 2 \\ 1 \end{pmatrix}$ and $\mathbf{x}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

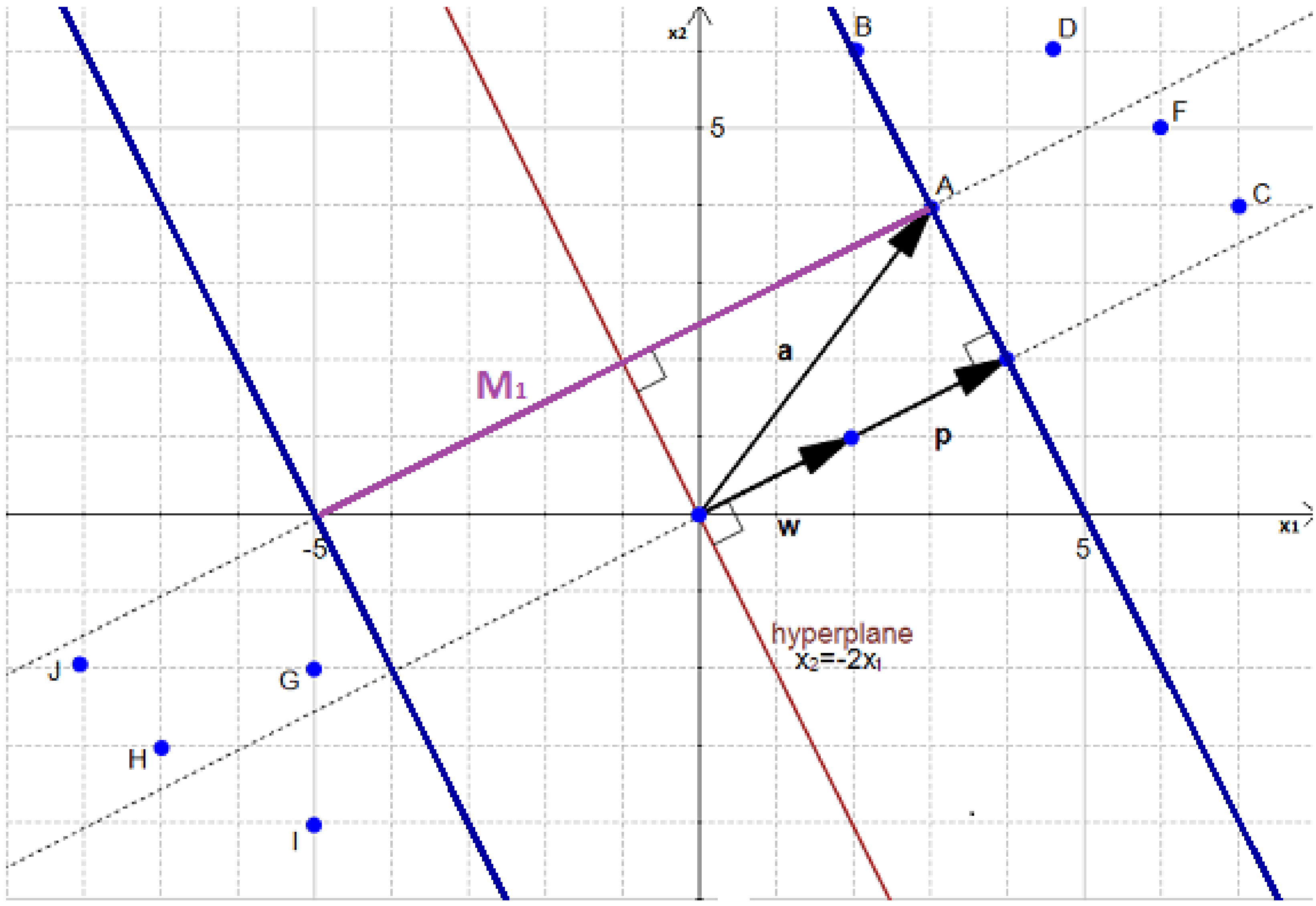
- We would like to compute the distance between the point ***A(3,4)*** and the hyperplane.
- This is the distance between ***A*** and its projection onto the hyperplane.



x2
B
D
F
5
A
C
w
-5
5
x1
hyperplane
x_2=-2x_1
J
G
H
I



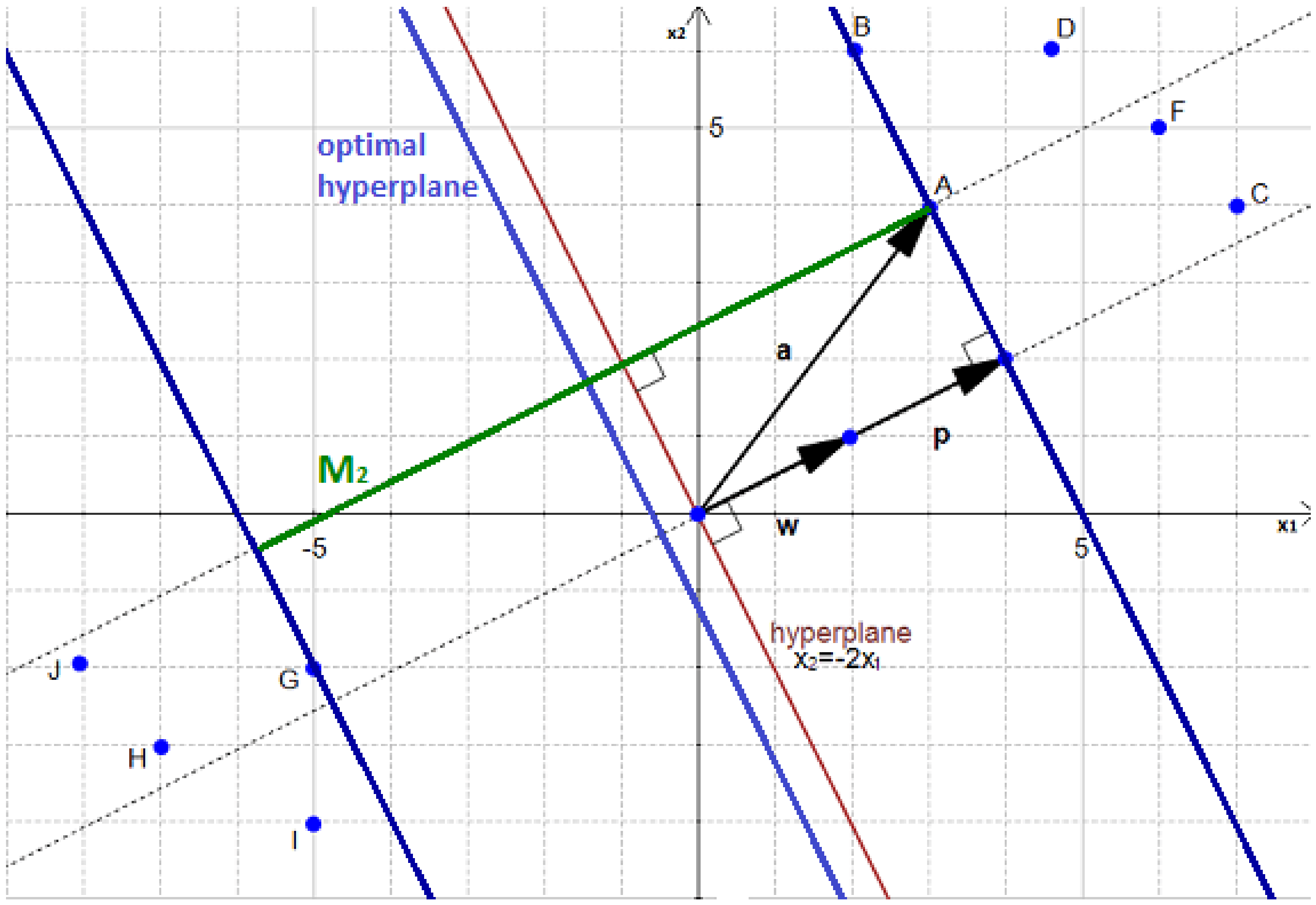
The distance between the point A and the hyperplane : $\|p\|$



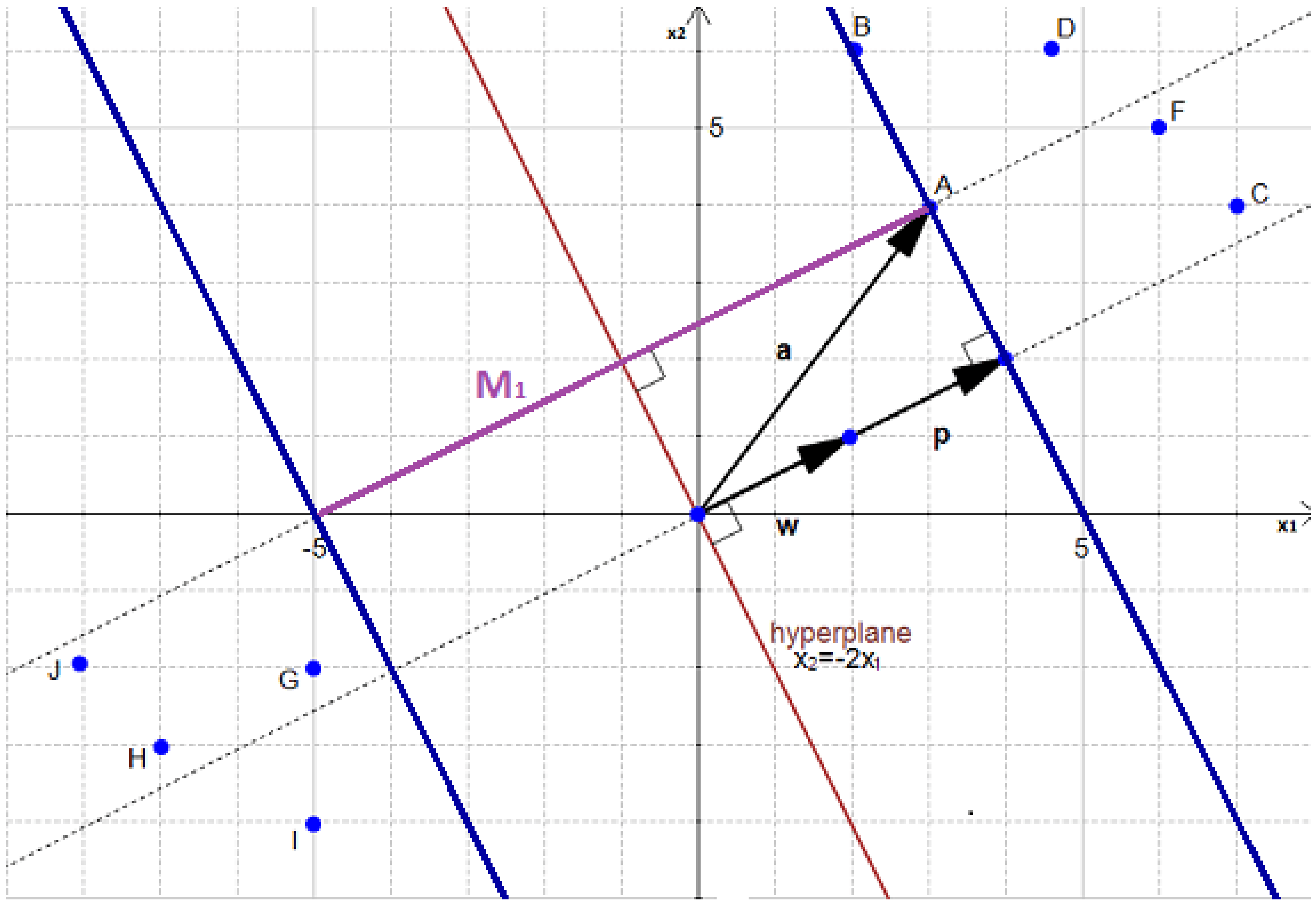
The margin: 2 ||p||

There is no single data on the margin area

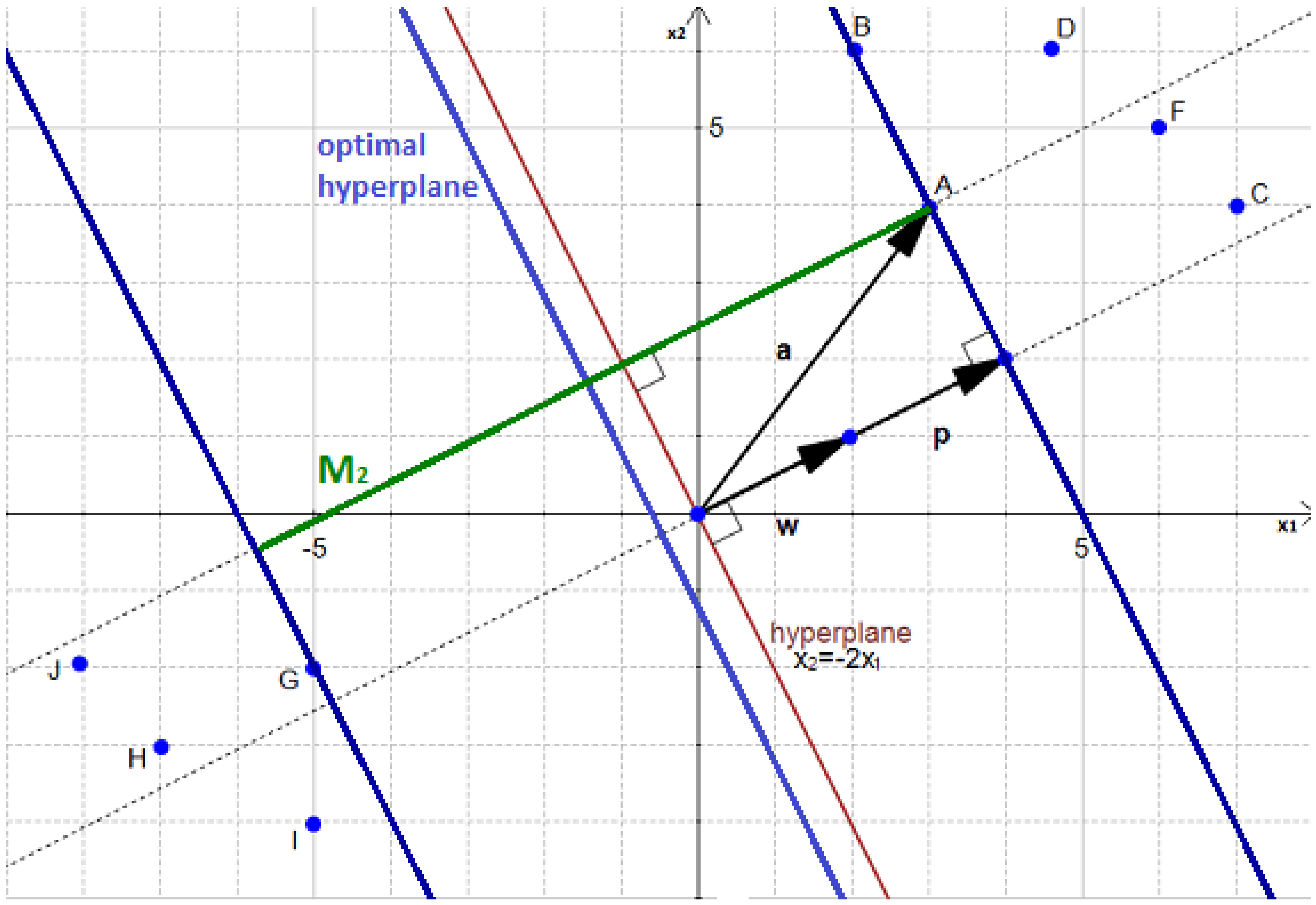
- However, even if it is quite a good at separating the data it was not the optimal hyperplane
- The optimal hyperplane is the one which maximizes the margin of the training data
- The margin M_1, delimited by the two blue lines, is not the biggest margin separating perfectly the data.
- The biggest margin is the margin M_2



x2
x1
5
-5
5
B
D
F
C
A
optimal
hyperplane
a
p
M2
w
J
G
H
I
hyperplane
x2=-2x1



x2
x1
5
-5
5
B
D
F
A
C
J
G
H
I
a
p
w
M1
hyperplane
$x_2=-2x_1$



x2
x1
5
-5
5
B
D
F
C
A
optimal
hyperplane
a
p
w
M2
J
G
H
I
hyperplane
x2=-2x1

- If we have an hyperplane, we can compute its margin with respect to some data point.
- If we have a margin delimited by two hyperplanes (the dark blue lines in *Figure above*), I can find a third hyperplane passing right in the middle of the margin.
- **Finding the biggest margin, is the same thing as finding the optimal hyperplane.**

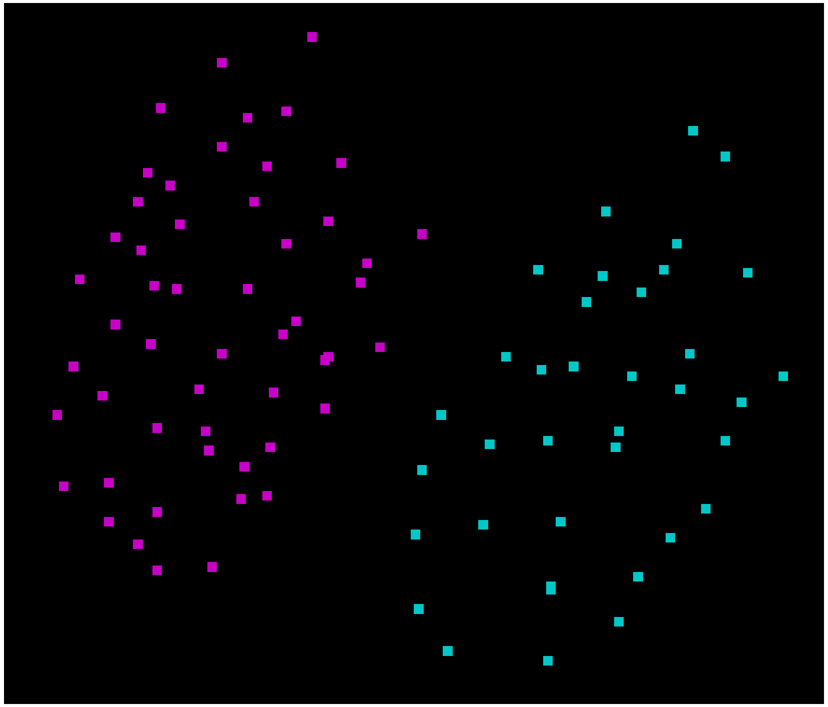
How can we find the biggest margin?

1. We have a dataset
2. Select two hyperplanes which separate the data with no points between them
3. Maximize their distance (the margin)

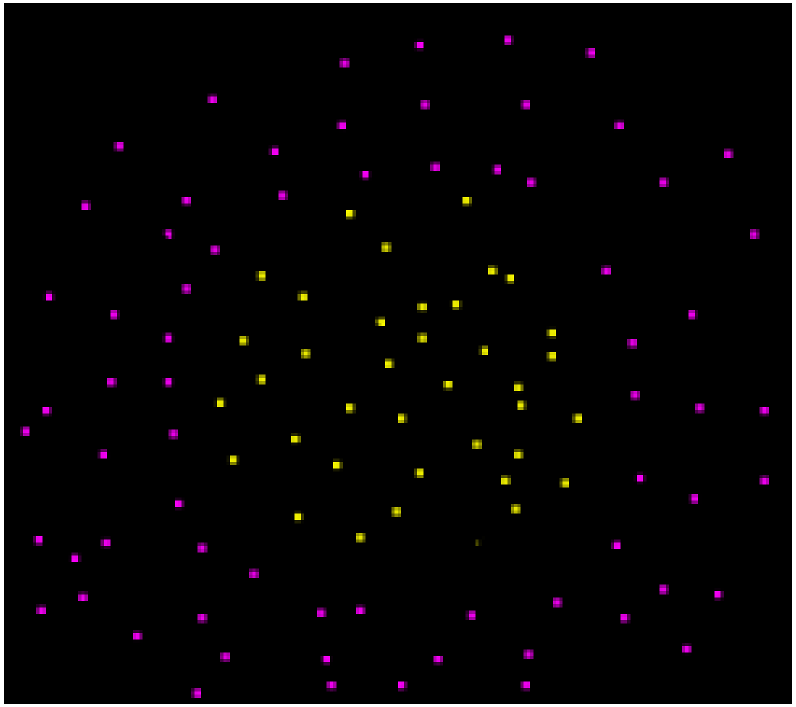
- Dataset *D* is the set of *n* couples of element $(\mathbf{x}_i, y_i)$
- y_i can only have two possible values -1 or +1

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p,\, y_i \in \{-1, 1\}\}_{i=1}^{n}$$

- We can only select two hyperplanes separating the data with no points between them that if our data **is** linearly separable



Linearly separable data



Non linearly separable data

- If our data is non linearly separable, then we use kernel function to transform our data from original space into other space (dimension). One of this visualization can be seen at https://www.youtube.com/watch?v=3liCbRZPrZA and https://www.youtube.com/watch?v=ffF8UnbheLk

- let's assume that our dataset D is linearly separable
- $\mathbf{w} \cdot \mathbf{x} + b = 0$

Given two 3-dimensional vectors $\mathbf{w}(b, -a, 1)$ and $\mathbf{x}(1, x, y)$

$$\mathbf{w} \cdot \mathbf{x} = b \times (1) + (-a) \times x + 1 \times y$$

$$\mathbf{w} \cdot \mathbf{x} = y - ax + b \tag{1}$$

Given two 2-dimensional vectors $\mathbf{w}'(-a, 1)$ and $\mathbf{x}'(x, y)$

$$\mathbf{w}' \cdot \mathbf{x}' = (-a) \times x + 1 \times y$$

$$\mathbf{w}' \cdot \mathbf{x}' = y - ax \tag{2}$$

Now if we add b on both side of the equation (2) we got :

$$\mathbf{w}' \cdot \mathbf{x}' + b = y - ax + b$$

$$\mathbf{w}' \cdot \mathbf{x}' + b = \mathbf{w} \cdot \mathbf{x} \tag{3}$$

Given a hyperplane H_0 separating the dataset and satisfying:

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

We can select two others hyperplanes H_1 and H_2 which also separate the data and have the following equations :

$$\mathbf{w} \cdot \mathbf{x} + b = \delta$$

and

$$\mathbf{w} \cdot \mathbf{x} + b = -\delta$$

so that H_0 is equidistant from H_1 and H_2.

However, here the variable δ is not necessary. So we can set $\delta = 1$ to simplify the problem.

$$\mathbf{w} \cdot \mathbf{x} + b = 1$$

and

$$\mathbf{w} \cdot \mathbf{x} + b = -1$$

Now we want to be sure that they have no points between them.

We won't select *any* hyperplane, we will only select those who meet the two following **constraints**:

For each vector $\mathbf{x_i}$ either :

$$\mathbf{w}\cdot\mathbf{x_i}+b\geq 1 \text{ for } \mathbf{x_i} \text{ having the class } 1 \qquad (4)$$

or

$$\mathbf{w}\cdot\mathbf{x_i}+b\leq -1 \text{ for } \mathbf{x_i} \text{ having the class } -1 \qquad (5)$$

Understanding the constraints

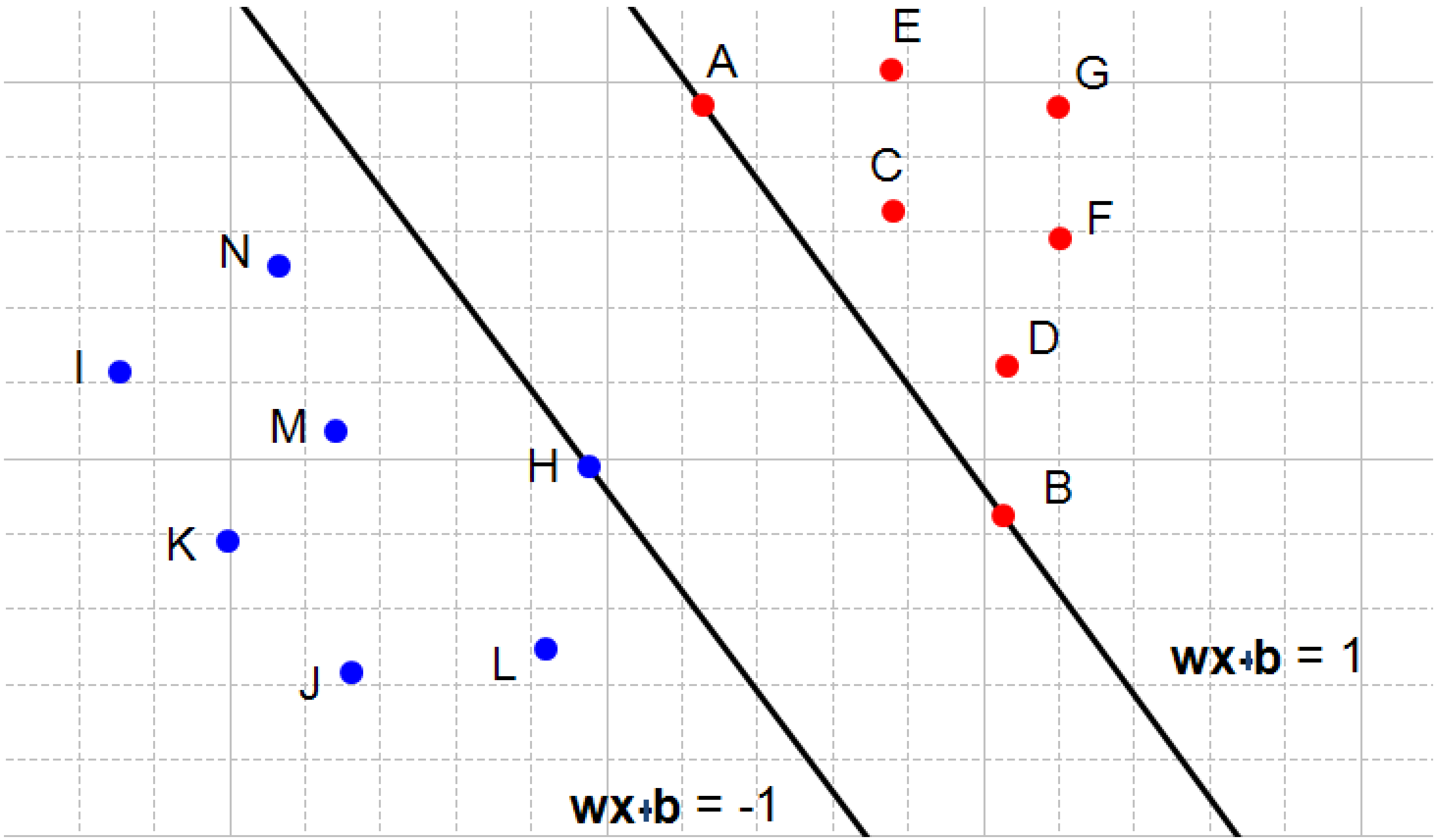
On the following figures, all red points have the class 1 and all blue points have the class -1.

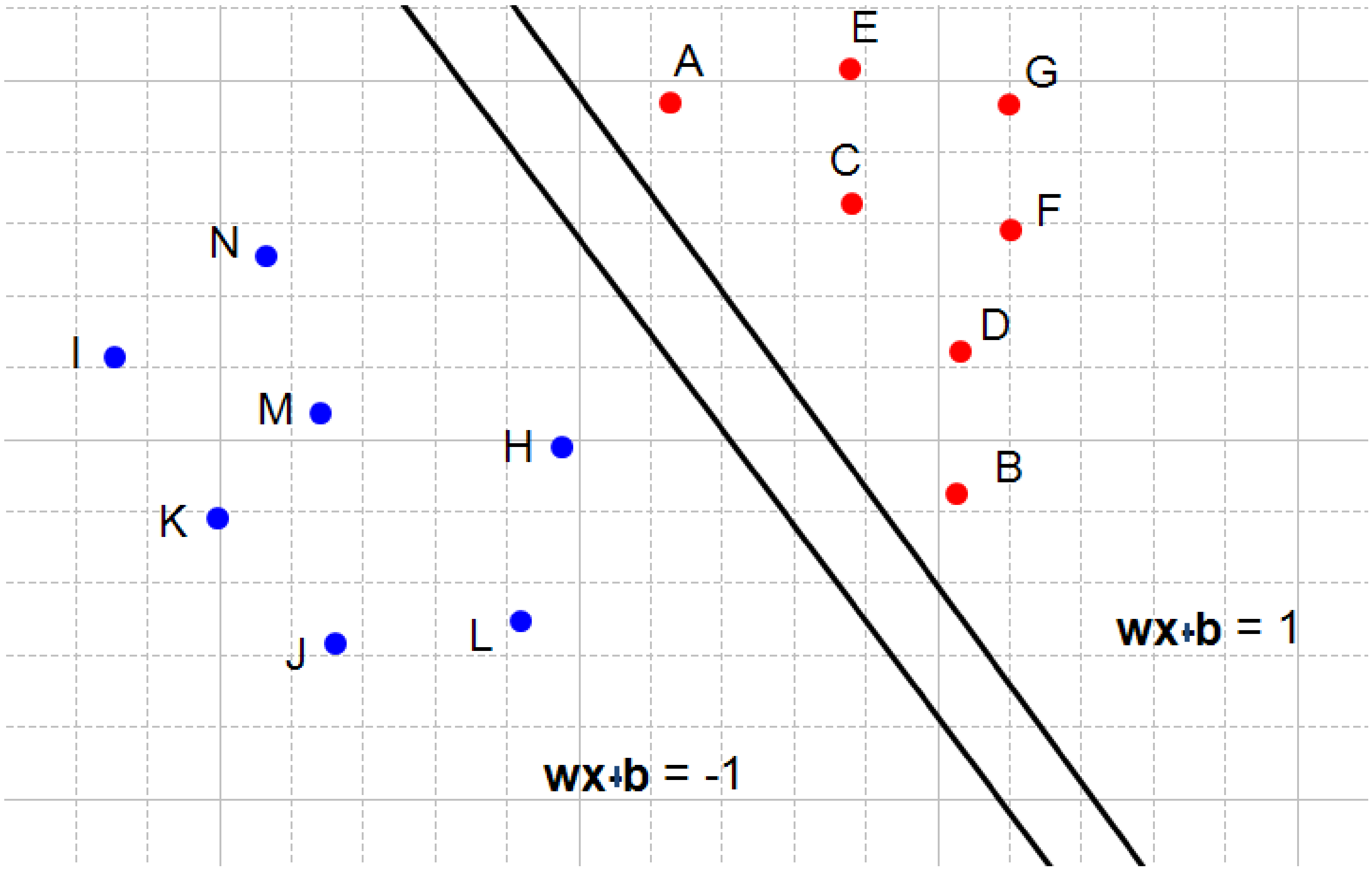
So let's look at *Figure 4* below and consider the point A. It is red so it has the class 1 and we need to verify it does not violate the constraint $\mathbf{w} \cdot \mathbf{x_i} + b \geq 1$

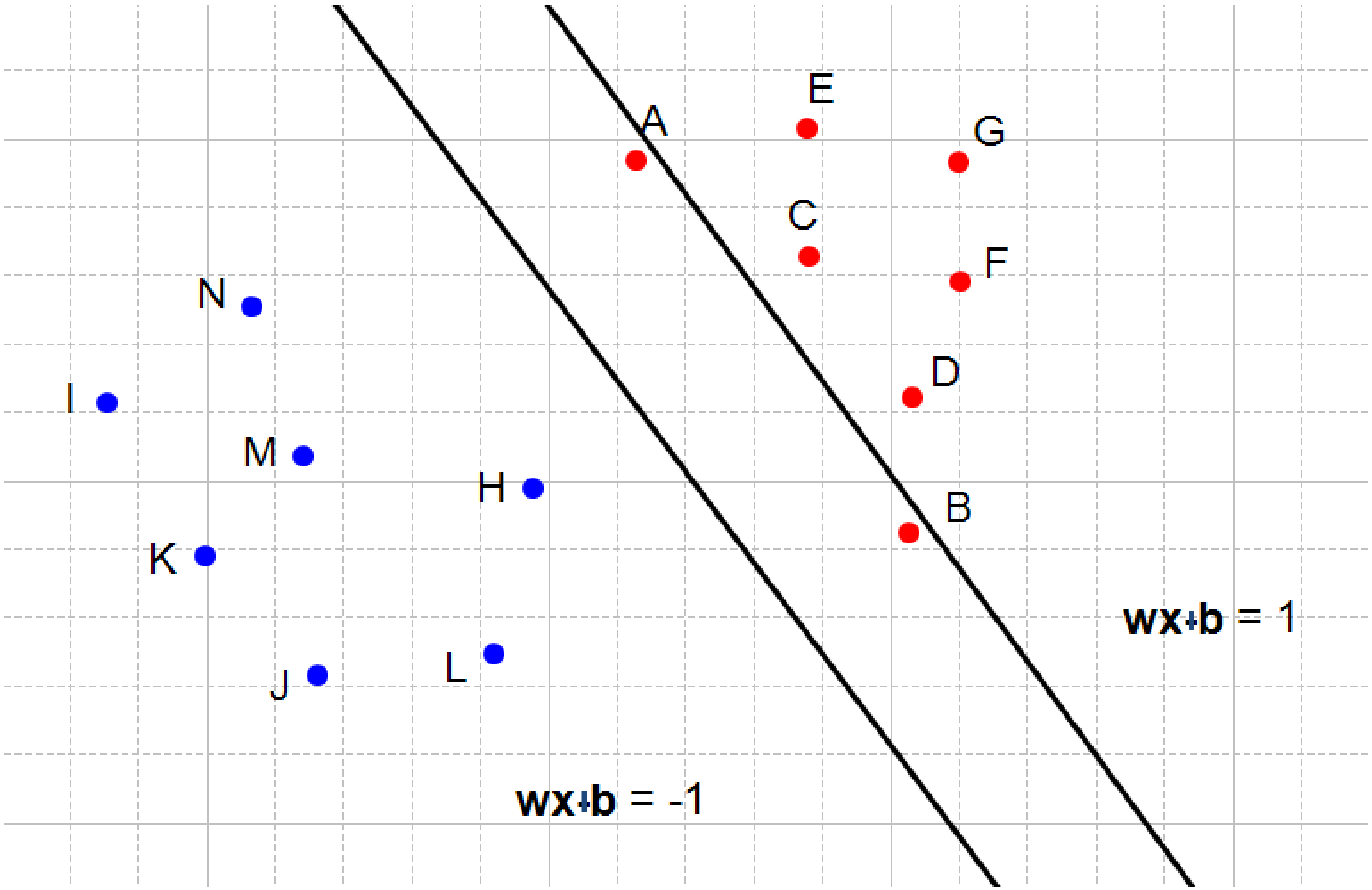
When $\mathbf{x_i} = A$ we see that the point is on the hyperplane so $\mathbf{w} \cdot \mathbf{x_i} + b = 1$ and the constraint is respected. The same applies for B.

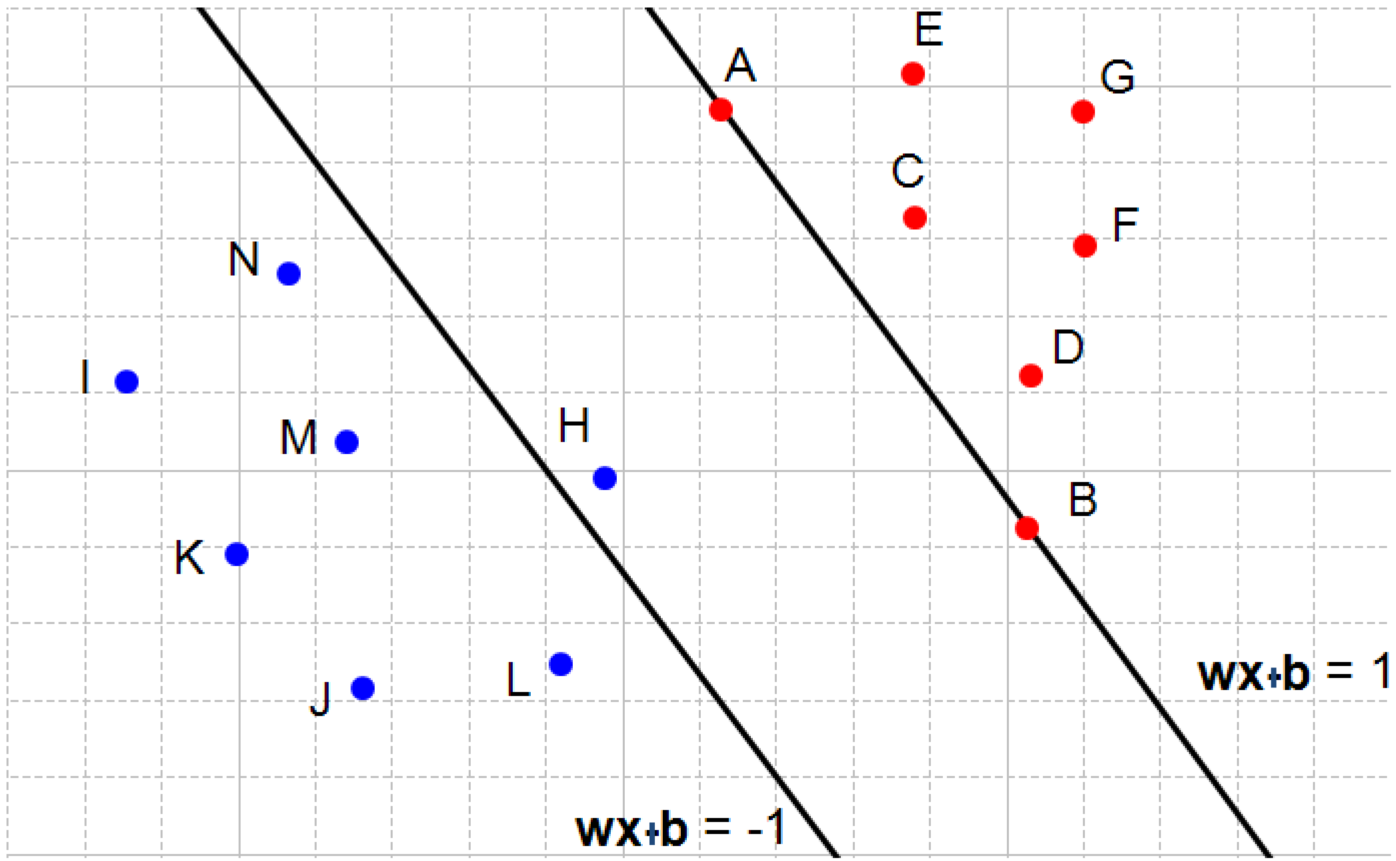
When $\mathbf{x_i} = C$ we see that the point is above the hyperplane so $\mathbf{w} \cdot \mathbf{x_i} + b > 1$ and the constraint is respected. The same applies for D, E, F and G.

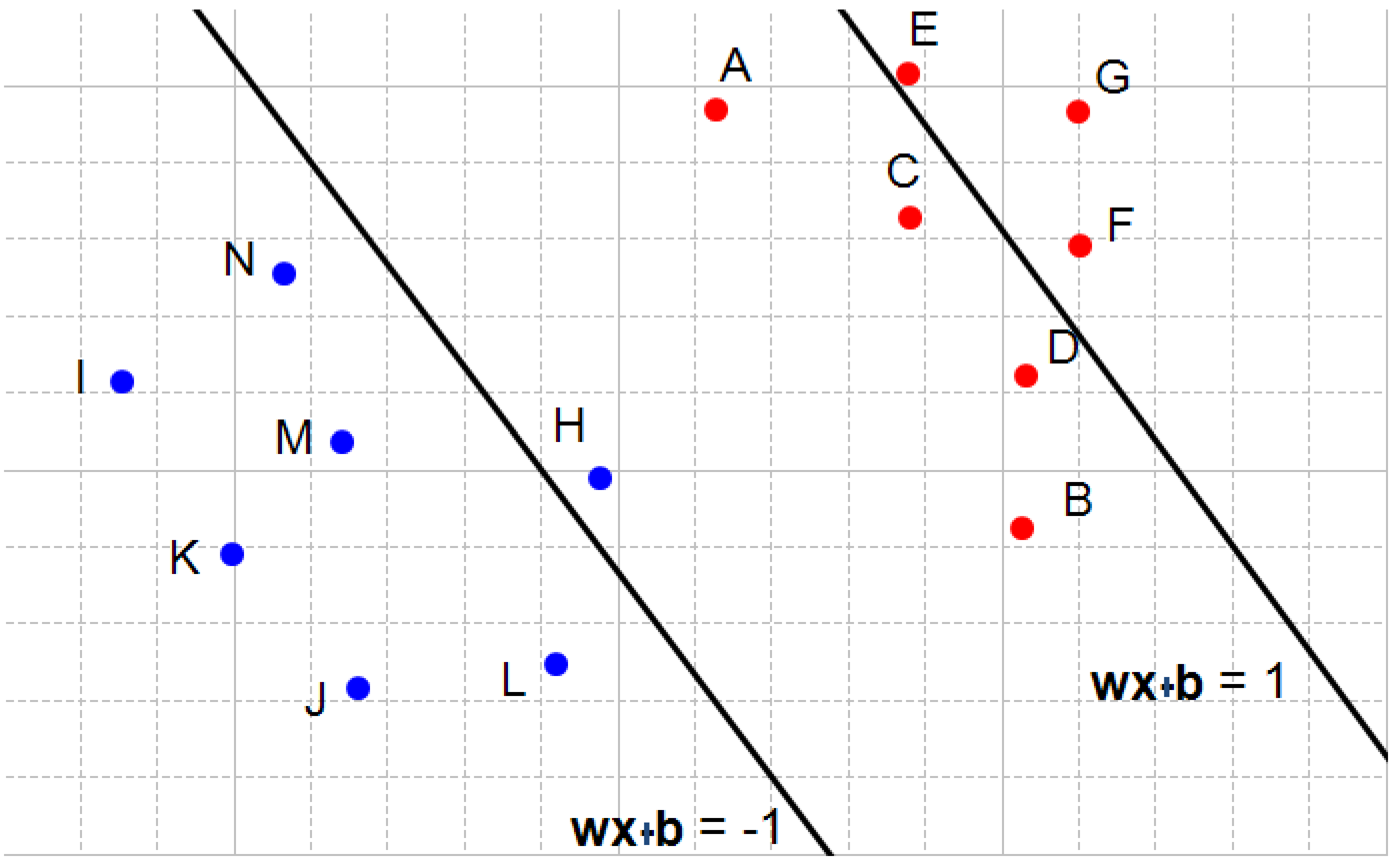
With an analogous reasoning you should find that the second constraint is respected for the class -1.


A
E
G
C
F
N
D
I
M
H
B
K
wx+b = 1
L
J
wx+b = -1


A
E
G
C
F
N
D
I
M
H
B
K
L
J
wx+b = 1
wx+b = -1


A
E
G
C
F
N
D
I
M
H
B
K
wx+b = 1
L
J
wx+b = -1


A
E
G
C
F
N
D
I
H
M
B
K
wx+b = 1
L
J
wx+b = -1



A
E
G
C
F
N
D
I
H
M
B
K
L
J
wx+b = 1
wx+b = -1

Combining both constraints

Equations **(4)** and **(5)** can be combined into a single constraint:

We start with equation **(5)**

$$\text{for } \mathbf{x_i} \text{ having the class } -1$$

$$\mathbf{w} \cdot \mathbf{x_i} + b \leq -1$$

And multiply both sides by y_i (which is always -1 in this equation)

$$y_i(\mathbf{w} \cdot \mathbf{x_i} + b) \geq y_i(-1)$$

Which means equation **(5)** can also be written:

$$y_i(\mathbf{w}\cdot\mathbf{x_i}+b)\geq 1 \text{ for } \mathbf{x_i} \text{ having the class } -1 \quad (6)$$

In equation **(4)**, as $y_i = 1$ it doesn't change the sign of the inequation.

$$y_i(\mathbf{w}\cdot\mathbf{x_i}+b)\geq 1 \text{ for } \mathbf{x_i} \text{ having the class } 1 \quad (7)$$

We combine equations **(6)** and **(7)** :

$$y_i(\mathbf{w}\cdot\mathbf{x_i}+b)\geq 1 \text{ for all } 1\leq i\leq n \quad (8)$$

We now have a unique constraint (equation **8**) instead of two (equations **4** and **5**) , but they are mathematically equivalent. So their effect is the same (there will be no points between the two hyperplanes).

Step 3: Maximize the distance between the two hyperplanes

a) What is the distance between our two hyperplanes ?

Before trying to maximize the distance between the two hyperplane, we will first ask ourselves: how do we compute it ?

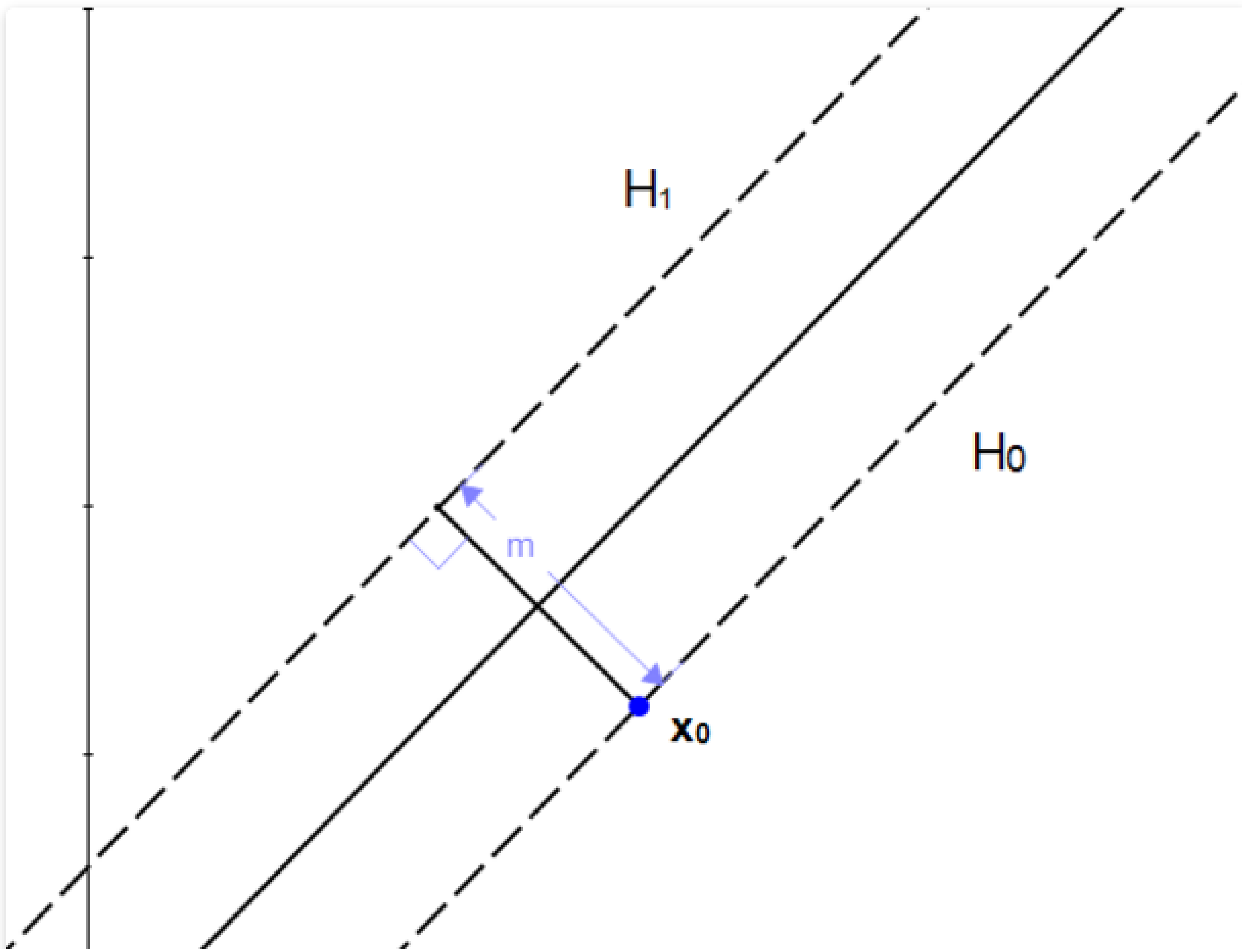
Let:

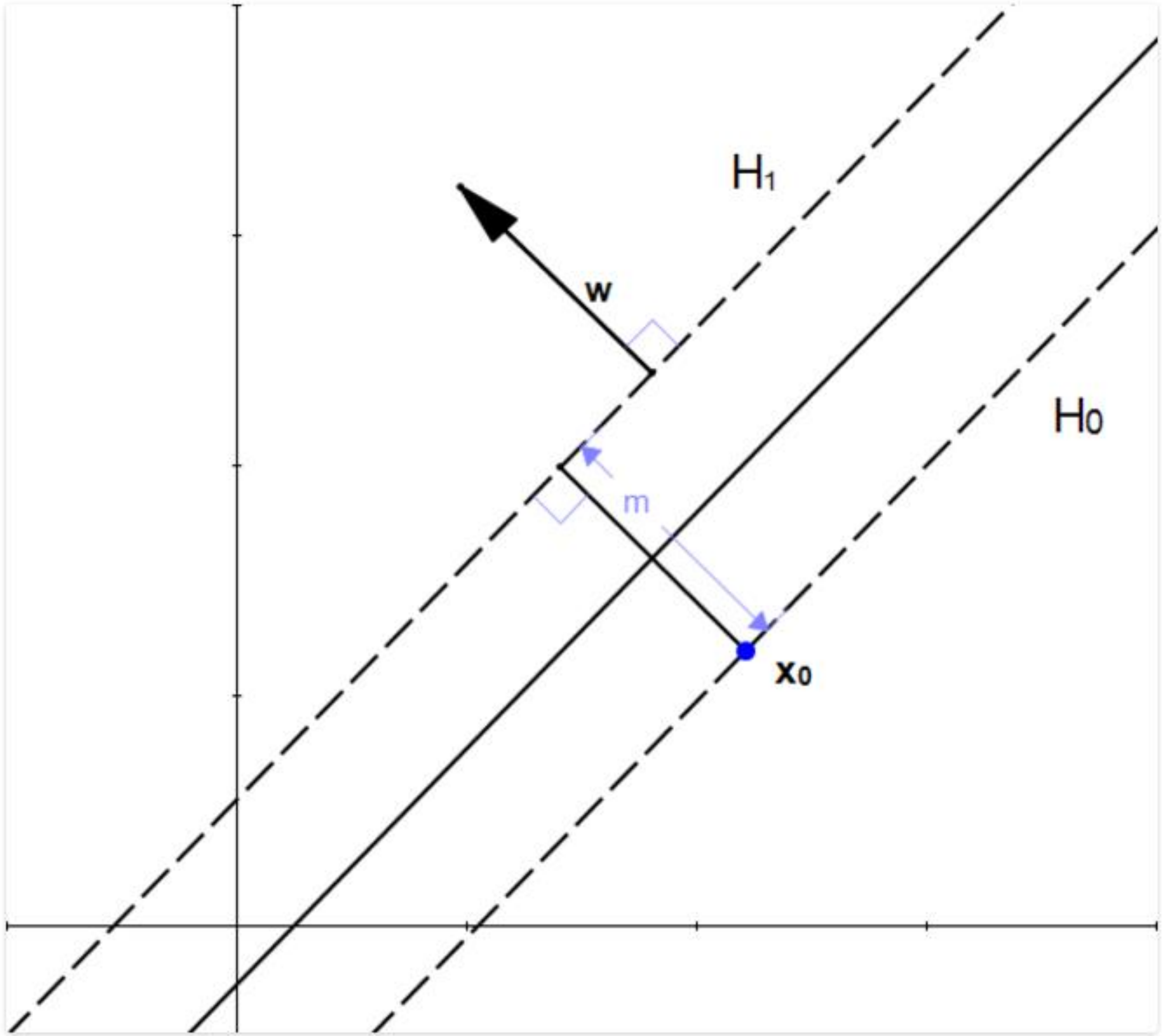
- $\mathcal{H}_0$ be the hyperplane having the equation $\mathbf{w} \cdot \mathbf{x} + b = -1$
- $\mathcal{H}_1$ be the hyperplane having the equation $\mathbf{w} \cdot \mathbf{x} + b = 1$
- $\mathbf{x}_0$ be a point in the hyperplane $\mathcal{H}_0$.

We will call m the perpendicular distance from $\mathbf{x}_0$ to the hyperplane $\mathcal{H}_1$. By definition, m is what we are used to call **the margin**.

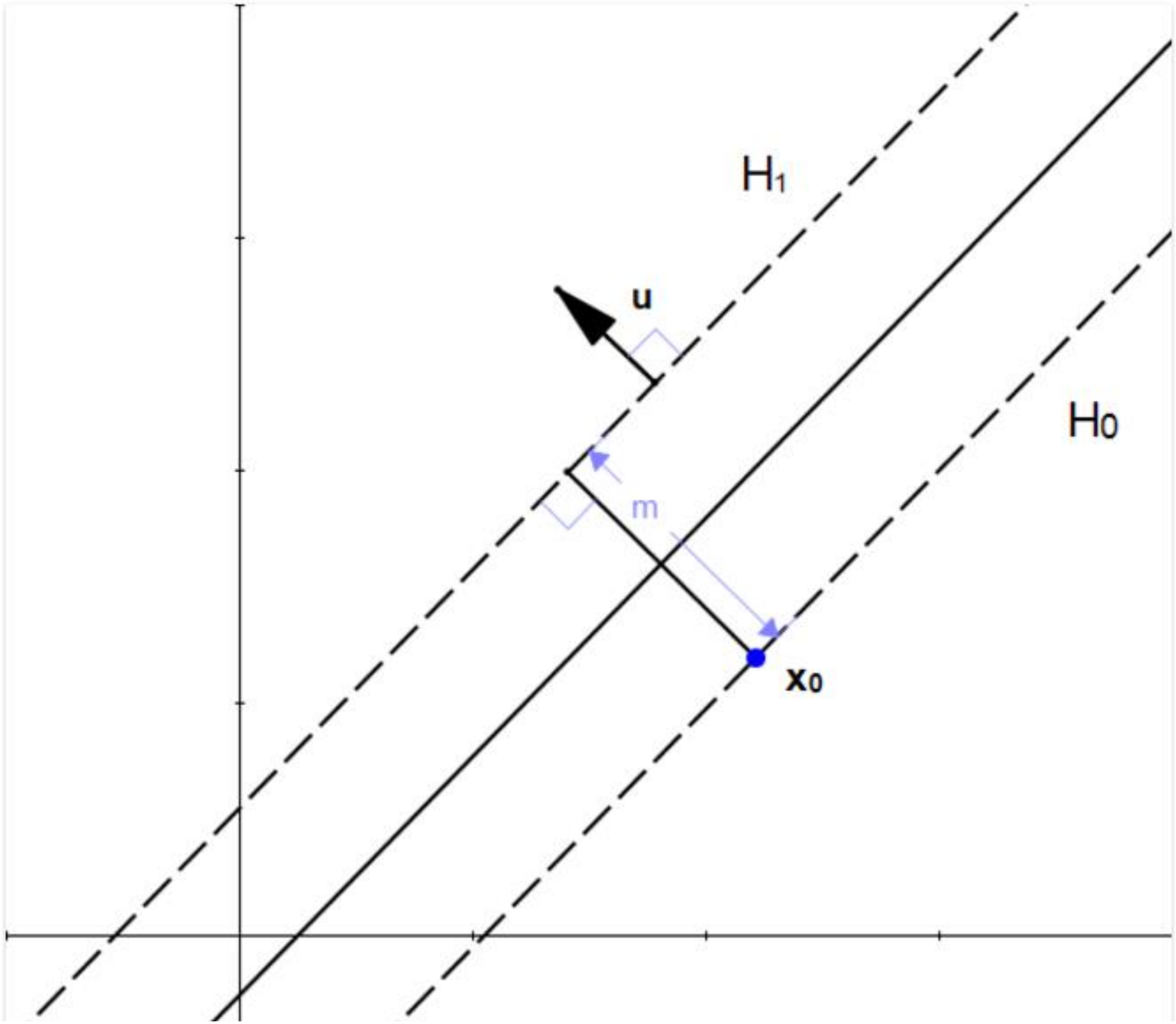
As $\mathbf{x}_0$ is in $\mathcal{H}_0$, m is the distance between hyperplanes $\mathcal{H}_0$ and $\mathcal{H}_1$.

We will now try to find the value of m.





H1
w
H0
m
x0



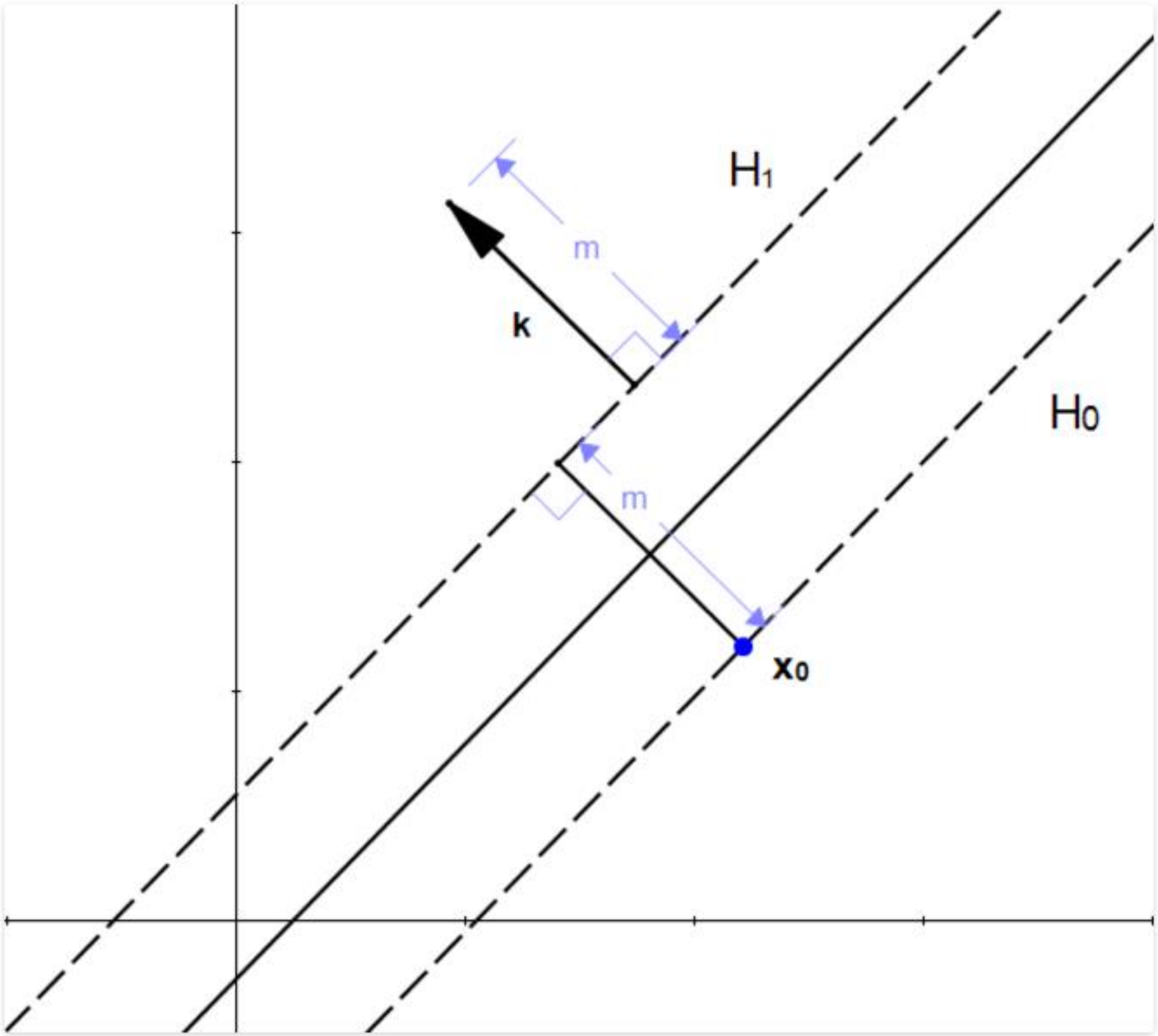
Let's define $\mathbf{u} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$ the unit vector of $\mathbf{w}$. As it is a unit vector $\|\mathbf{u}\| = 1$ and it has the same direction as $\mathbf{w}$ so it is also perpendicular to the hyperplane.

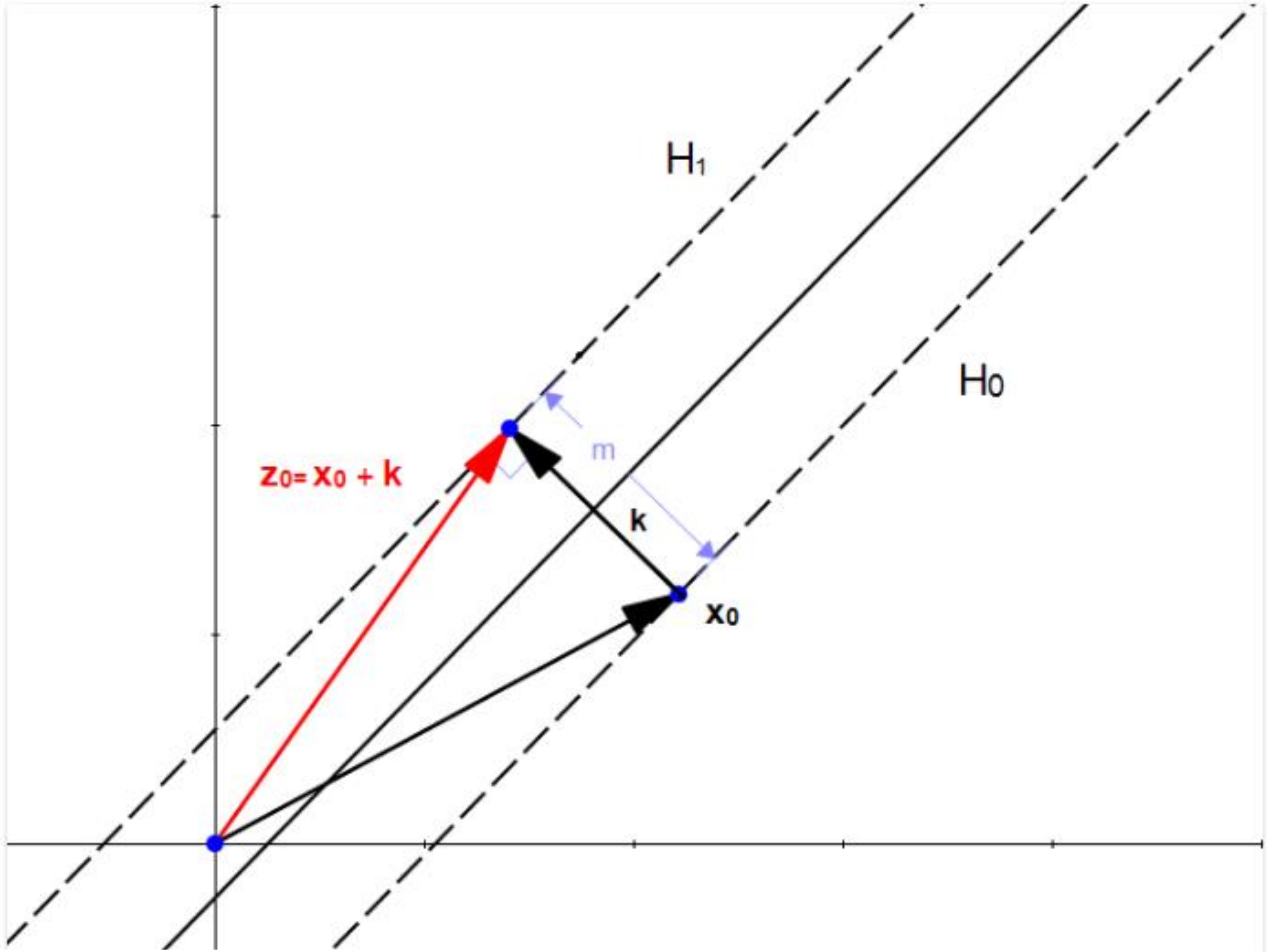
If we multiply $\mathbf{u}$ by m we get the vector $\mathbf{k} = m\mathbf{u}$ and :

1. $\|\mathbf{k}\| = m$
2. $\mathbf{k}$ is perpendicular to $\mathcal{H}_1$ (because it has the same direction as $\mathbf{u}$)

From these properties we can see that $\mathbf{k}$ is the vector we were looking for.

$$\mathbf{k} = m\mathbf{u} = m\frac{\mathbf{w}}{\|\mathbf{w}\|} \tag{9}$$


H1
m
k
H0
m
x0



If we start from the point $\mathbf{x}_0$ and add k we find that the point $\mathbf{z}_0 = \mathbf{x}_0 + \mathbf{k}$ is in the hyperplane $\mathcal{H}_1$

The fact that $\mathbf{z}_0$ is in $\mathcal{H}_1$ means that

$$\mathbf{w} \cdot \mathbf{z}_0 + b = 1 \tag{10}$$

We can replace $\mathbf{z}_0$ by $\mathbf{x}_0 + \mathbf{k}$ because that is how we constructed it.

$$\mathbf{w} \cdot (\mathbf{x}_0 + \mathbf{k}) + b = 1 \tag{11}$$

We can now replace $\mathbf{k}$ using equation (9)

$$\mathbf{w} \cdot (\mathbf{x}_0 + m\frac{\mathbf{w}}{\|\mathbf{w}\|}) + b = 1 \tag{12}$$

We now expand equation (12)

$$\mathbf{w} \cdot \mathbf{x}_0 + m\frac{\mathbf{w} \cdot \mathbf{w}}{\|\mathbf{w}\|} + b = 1 \tag{13}$$

The dot product of a vector with itself is the square of its norm so :

$$\mathbf{w} \cdot \mathbf{x}_0 + m\frac{\|\mathbf{w}\|^2}{\|\mathbf{w}\|} + b = 1 \tag{14}$$

$$\mathbf{w} \cdot \mathbf{x}_0 + m\|\mathbf{w}\| + b = 1 \tag{15}$$

$$\mathbf{w} \cdot \mathbf{x}_0 + b = 1 - m\|\mathbf{w}\| \tag{16}$$

As $\mathbf{x}_0$ is in $\mathcal{H}_0$ then $\mathbf{w} \cdot \mathbf{x}_0 + b = -1$

$$-1 = 1 - m\|\mathbf{w}\| \tag{17}$$

$$m\|\mathbf{w}\| = 2 \tag{18}$$

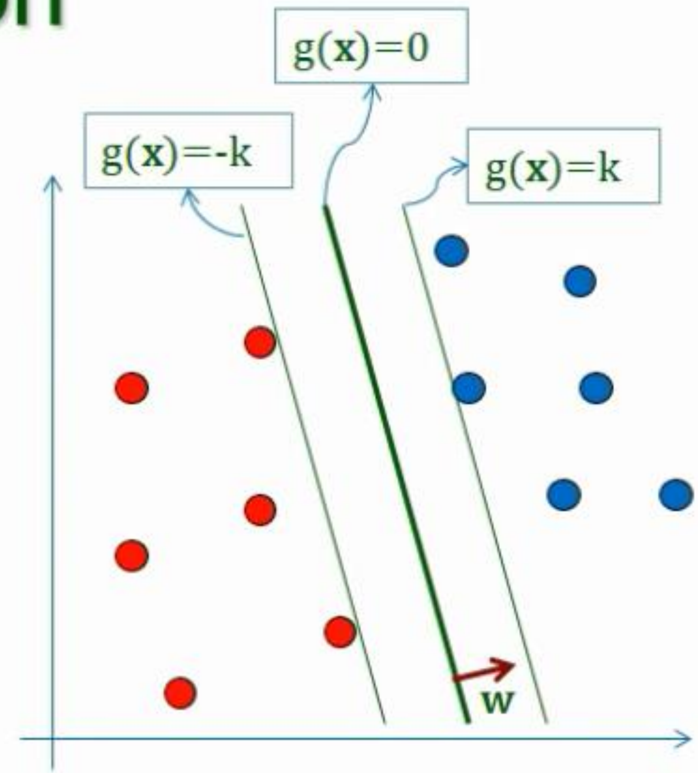
$$m = \frac{2}{\|\mathbf{w}\|} \tag{19}$$

This is it ! We found a way to compute m.

Margin ***m*** berbanding terbalik dengan norm **w**. Semakin kecil norm **w** maka semakin besar margin, semakin besar norm **w** maka semakin kecil margin). Dengan demikian, untuk mendapatkan margin terbesar (optimum) kita akan mencari **w** yang memiliki norm terkecil.

Formulation

- Let $g(x) = w^T x + b$.
- We want to maximize k such that:
  - $w^T x_i + b \geq k$ for $d_i = 1$
  - $w^T x_i + b \leq -k$ for $d_i = -1$



- Value of $g(x)$ dependents on $\|w\|$:
  1. Keep $\|w\|=1$, and maximize $g(x)$, or
  2. Let $g(x) \geq 1$, and minimize $\|w\|$.

- We use approach (2) and formulate the problem as:
  - Minimize: $\frac{1}{2} w^T w$
  - Subject to: $d_i(w^T x_i + b) \geq 1$, for $i=1..N$

The Optimization Problem

$$\text{Minimize} : \Phi(\mathbf{w}) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w}$$

$$\text{Subject to} : d_i(\mathbf{w}^T\mathbf{x_i} + b) - 1 \geq 0 \quad \forall i$$

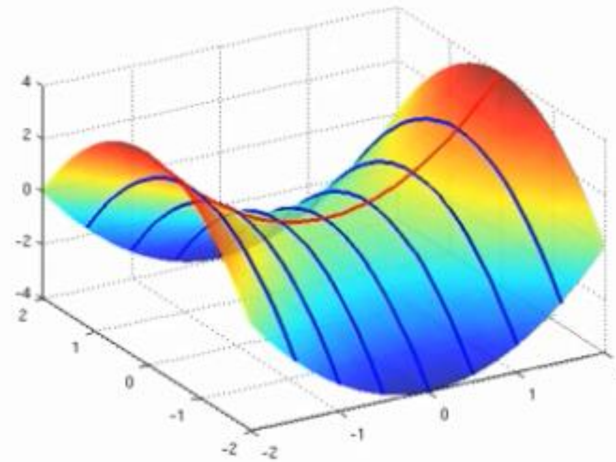
- Quadratic objective function with linear inequalities as constraints: QP Solver.
- Integrating the constraints into the Lagrangian form, we get:

$$\text{Minimize} : J(\mathbf{w}, b, \boldsymbol{\alpha}) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i d_i(\mathbf{w}^T\mathbf{x_i} + b) + \sum_{i=1}^{N} \alpha_i$$

$$\text{Subject to} : \alpha_i \geq 0 \quad \forall i$$

The Optimization Problem

$$\text{Minimize : } \Phi(\mathbf{w}) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w}$$
$$\text{Subject to : } d_i(\mathbf{w}^T\mathbf{x_i} + b) - 1 \geq 0 \quad \forall i$$



- Quadratic objective function with linear inequalities as constraints: QP Solver.
- Integrating the constraints into the Lagran form, we get:

$$\text{Minimize : } J(\mathbf{w}, b, \boldsymbol{\alpha}) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N}\alpha_i d_i(\mathbf{w}^T\mathbf{x_i} + b) + \sum_{i=1}^{N}\alpha_i$$
$$\text{Subject to : } \alpha_i \geq 0 \quad \forall i$$

- Minimize J with respect to w and b, and maximize with respect to α.

Converting to the Dual Form

$$\text{Objective}: J(\mathbf{w}, b, \boldsymbol{\alpha}) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i d_i (\mathbf{w}^T \mathbf{x_i} + b) + \sum_{i=1}^{N} \alpha_i$$

$$\text{At the optimum,} \quad 1: \frac{\partial J}{\partial \mathbf{w}} = 0 \quad \text{and} \quad 2: \frac{\partial J}{\partial b} = 0$$

Converting to the Dual Form

Objective : $J(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i d_i(\mathbf{w}^T\mathbf{x_i} + b) + \sum_{i=1}^{N} \alpha_i$

At the optimum, $\boxed{1: \ \frac{\partial J}{\partial \mathbf{w}} = 0}$ and $2: \ \frac{\partial J}{\partial b} = 0$

$$\boxed{1: \ \mathbf{w}_o = \sum_{i=1}^{N} \alpha_i d_i \mathbf{x_i}}$$

Converting to the Dual Form

Objective : $J(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i d_i (\mathbf{w}^T \mathbf{x_i} + b) + \sum_{i=1}^{N} \alpha_i$

At the optimum, $\boxed{1: \ \frac{\partial J}{\partial \mathbf{w}} = 0}$ and $\boxed{2: \ \frac{\partial J}{\partial b} = 0}$

$\boxed{1: \ \mathbf{w}_o = \sum_{i=1}^{N} \alpha_i d_i \mathbf{x_i}}$ $\boxed{2: \ \sum_{i=1}^{N} \alpha_i d_i = 0}$

Converting to the Dual Form

$$\text{Objective}: J(\mathbf{w}, b, \boldsymbol{\alpha}) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N} \alpha_i d_i (\mathbf{w}^T \mathbf{x_i} + b) + \sum_{i=1}^{N} \alpha_i$$

At the optimum, $\boxed{1: \ \frac{\partial J}{\partial \mathbf{w}} = 0}$ and $\boxed{2: \ \frac{\partial J}{\partial b} = 0}$

$$\boxed{1: \ \mathbf{w}_o = \sum_{i=1}^{N} \alpha_i d_i \mathbf{x_i}} \qquad \boxed{2: \ \sum_{i=1}^{N} \alpha_i d_i = 0}$$

KKT Conditions

$$\boxed{3: \alpha_i [d_i (\mathbf{w}_o{}^T \mathbf{x_i} + b_o) - 1] = 0}$$

Converting to the Dual Form

$$\text{Objective}:\ J(\mathbf{w},b,\boldsymbol{\alpha}) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N}\alpha_i d_i(\mathbf{w}^T\mathbf{x_i}+b) + \sum_{i=1}^{N}\alpha_i$$

$$\text{At the optimum,}\quad \boxed{1:\ \frac{\partial J}{\partial \mathbf{w}} = 0} \quad \text{and} \quad \boxed{2:\ \frac{\partial J}{\partial b} = 0}$$

KKT Conditions

$$\boxed{1:\ \mathbf{w}_o = \sum_{i=1}^{N}\alpha_i d_i \mathbf{x_i}} \qquad \boxed{2:\ \sum_{i=1}^{N}\alpha_i d_i = 0} \qquad \boxed{3:\alpha_i[d_i(\mathbf{w}_o{}^T\mathbf{x_i}+b_o)-1]=0}$$

$$\text{Obj}:\ J(\mathbf{w},b,\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i + \tfrac{1}{2}\mathbf{w}^T\mathbf{w} - \mathbf{w}^T\sum_{i=1}^{N}\alpha_i d_i\mathbf{x_i} - b\sum_{i=1}^{N}\alpha_i d_i$$

Converting to the Dual Form

$$\text{Objective : } J(\mathbf{w}, b, \boldsymbol{\alpha}) = \tfrac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{N}\alpha_i d_i(\mathbf{w}^T\mathbf{x_i} + b) + \sum_{i=1}^{N}\alpha_i$$

At the optimum, $\boxed{1:\ \frac{\partial J}{\partial \mathbf{w}} = 0}$ and $\boxed{2:\ \frac{\partial J}{\partial b} = 0}$

KKT Conditions

$$\boxed{1:\ \mathbf{w}_o = \sum_{i=1}^{N}\alpha_i d_i \mathbf{x_i}} \quad \boxed{2:\ \sum_{i=1}^{N}\alpha_i d_i = 0} \quad \boxed{3:\alpha_i[d_i(\mathbf{w}_o{}^T\mathbf{x_i} + b_o) - 1] = 0}$$

$$\text{Obj: } J(\mathbf{w}, b, \boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i + \tfrac{1}{2}\mathbf{w}^T\mathbf{w} - \mathbf{w}^T\sum_{i=1}^{N}\alpha_i d_i \mathbf{x_i} - b\sum_{i=1}^{N}\alpha_i d_i$$

$$\text{Using } 1,2 : \boxed{Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i - \tfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j \mathbf{x_i}^T\mathbf{x}_j}$$

Solving the Dual Form

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x_i}^T \mathbf{x}_j$$

$$\text{Subject to} \quad \alpha_i \geq 0 \quad \forall_i \quad \text{and} \quad \sum_{i=1}^{N} \alpha_i d_i = 0$$

- The only unknowns (variables) are α_is.
- The constraints are also on α_is only.
- Data vectors appear only as dot products
- Objective is convex, subject to linear constraints
- Can be solved using standard convex quadratic program solvers

Solving the Dual Form

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x_i}^T \mathbf{x}_j$$

$$\text{Subject to} \quad \alpha_i \geq 0 \quad \forall_{\mathrm{i}} \quad \text{and} \quad \sum_{\mathrm{i}=1}^{N} \alpha_{\mathrm{i}} d_i = 0$$

$$\longrightarrow \boxed{\text{QP Solver}} \longrightarrow \boxed{\alpha_i}$$

- The only unknowns (variables) are α_{i}s.
- The constraints are also on α_{i}s only.
- Data vectors appear only as dot products
- Objective is convex, subject to linear constraints
- Can be solved using standard convex quadratic program solvers

Solving the Dual Form

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x_i}^T \mathbf{x}_j$$

Subject to $\alpha_i \geq 0 \quad \forall_i$ and $\sum_{\mathrm{i}=1}^{N} \alpha_{\mathrm{i}} d_i = 0$

$\rightarrow$ QP Solver $\rightarrow$ α_i $\rightarrow$

$$\mathbf{w}_o = \sum_{i=1}^{N} \alpha_i d_i \mathbf{x_i}$$

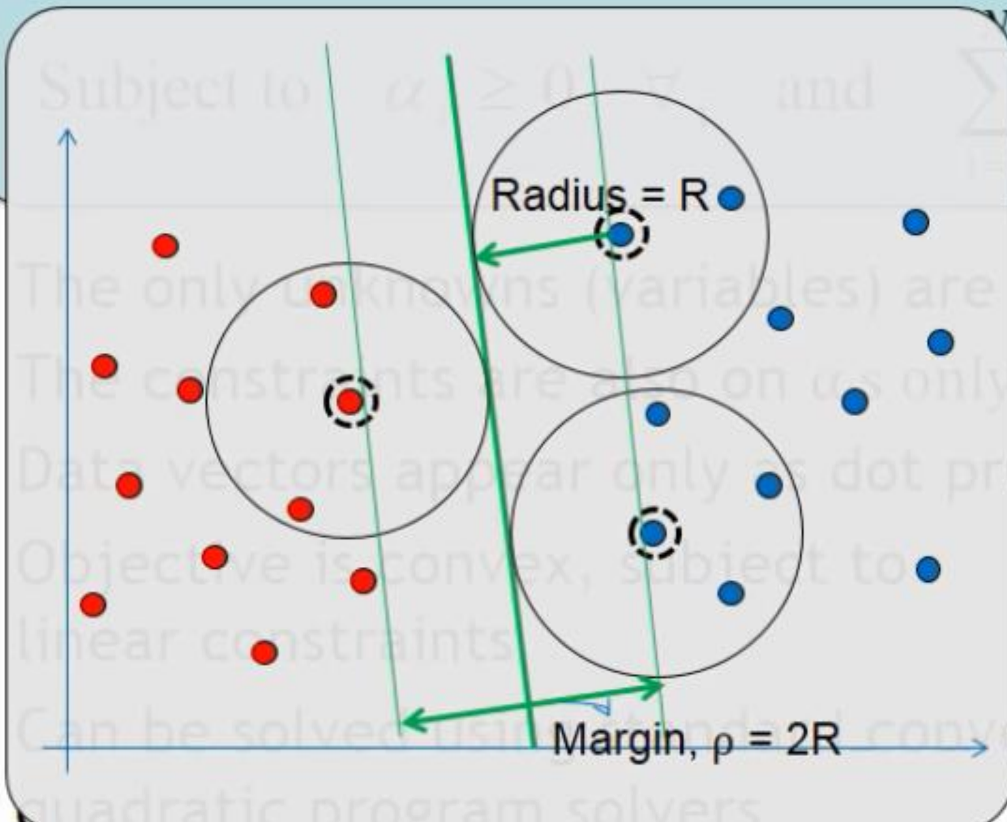
- The only unknowns (variables) are α_is.
- The constraints are also on α_is only.
- Data vectors appear only as dot products
- Objective is convex, subject to linear constraints
- Can be solved using standard convex quadratic program solvers

Solving the Dual Form

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x_i}^T \mathbf{x}_j$$

$$\text{Subject to} \quad \alpha_i \geq 0 \quad \forall i \quad \text{and} \quad \sum_{i=1}^{N} \alpha_i d_i = 0$$

QP Solver → α_i → $\mathbf{w}_o = \sum_{i=1}^{N} \alpha_i d_i \mathbf{x_i}$



- The only unknowns (variables) are α_is.
- The constraints are also on α_is only.
- Data vectors appear only as dot products
- Objective is convex, subject to linear constraints
- Can be solved using standard convex quadratic program solvers

Solving the Dual Form

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x_i}^T \mathbf{x}_j$$

$$\text{Subject to} \quad \alpha_i \geq 0 \quad \forall_{\text{i}} \quad \text{and} \quad \sum_{\text{i}=1}^{N} \alpha_{\text{i}} d_i = 0$$

QP Solver

α_i

$$\mathbf{w}_o = \sum_{i=1}^{N} \alpha_i d_i \mathbf{x_i}$$

KKT Conditions

$$\alpha_i [d_i (\mathbf{w}_o{}^T \mathbf{x_i} + b_o) - 1] = 0$$

- The only unknowns (variables) are α_is.
- The constraints are also on α_is only.
- Data vectors appear only as dot products
- Objective is convex, subject to linear constraints
- Can be solved using standard convex quadratic program solvers

Solving the Dual Form

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \tfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \mathbf{x_i}^T \mathbf{x}_j$$

$$\text{Subject to} \quad \alpha_i \geq 0 \quad \forall_i \quad \text{and} \quad \sum_{i=1}^{N} \alpha_i d_i = 0$$

QP Solver

α_i

$$\mathbf{w}_o = \sum_{i=1}^{N} \alpha_i d_i \mathbf{x_i}$$

- The only unknowns (variables) are α_is.
- The constraints are also on α_is only.
- Data vectors appear only as dot products
- Objective is convex, subject to linear constraints
- Can be solved using standard convex quadratic program solvers

KKT Conditions

$$\alpha_i[d_i(\mathbf{w}_o{}^T \mathbf{x_i} + b_o) - 1] = 0$$

$$b_o = 1 - \mathbf{w}_o{}^T \mathbf{x}_{s+}$$