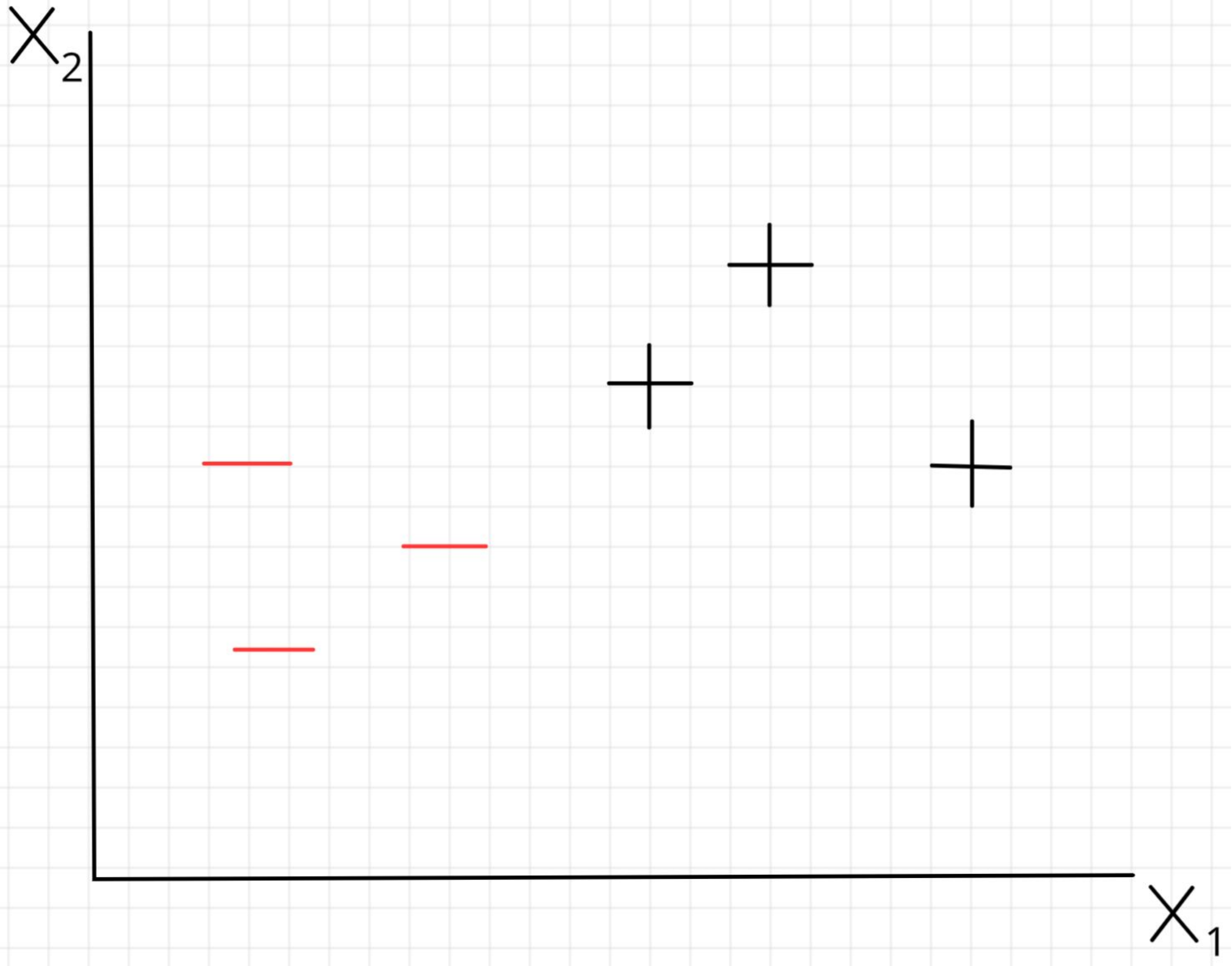
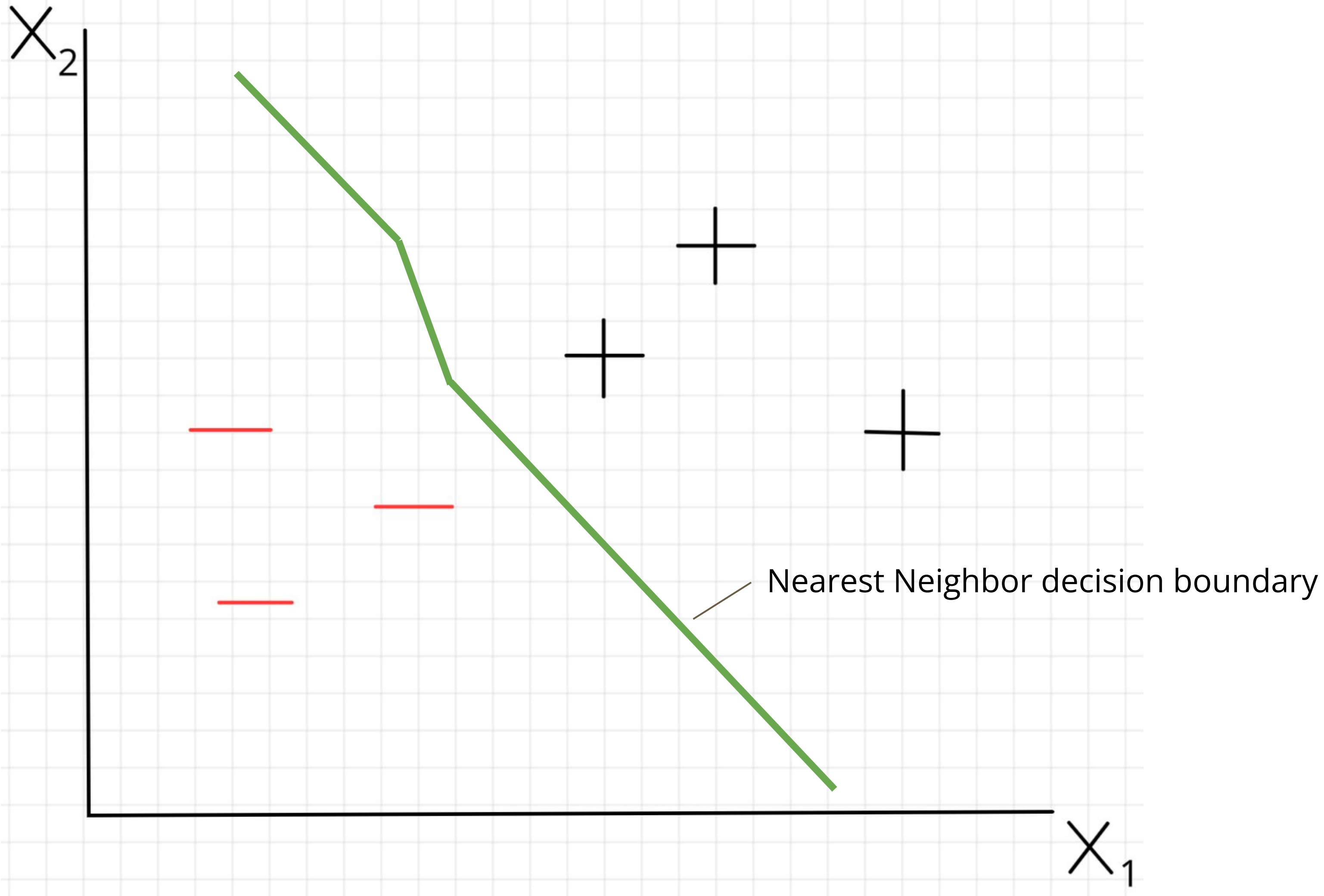
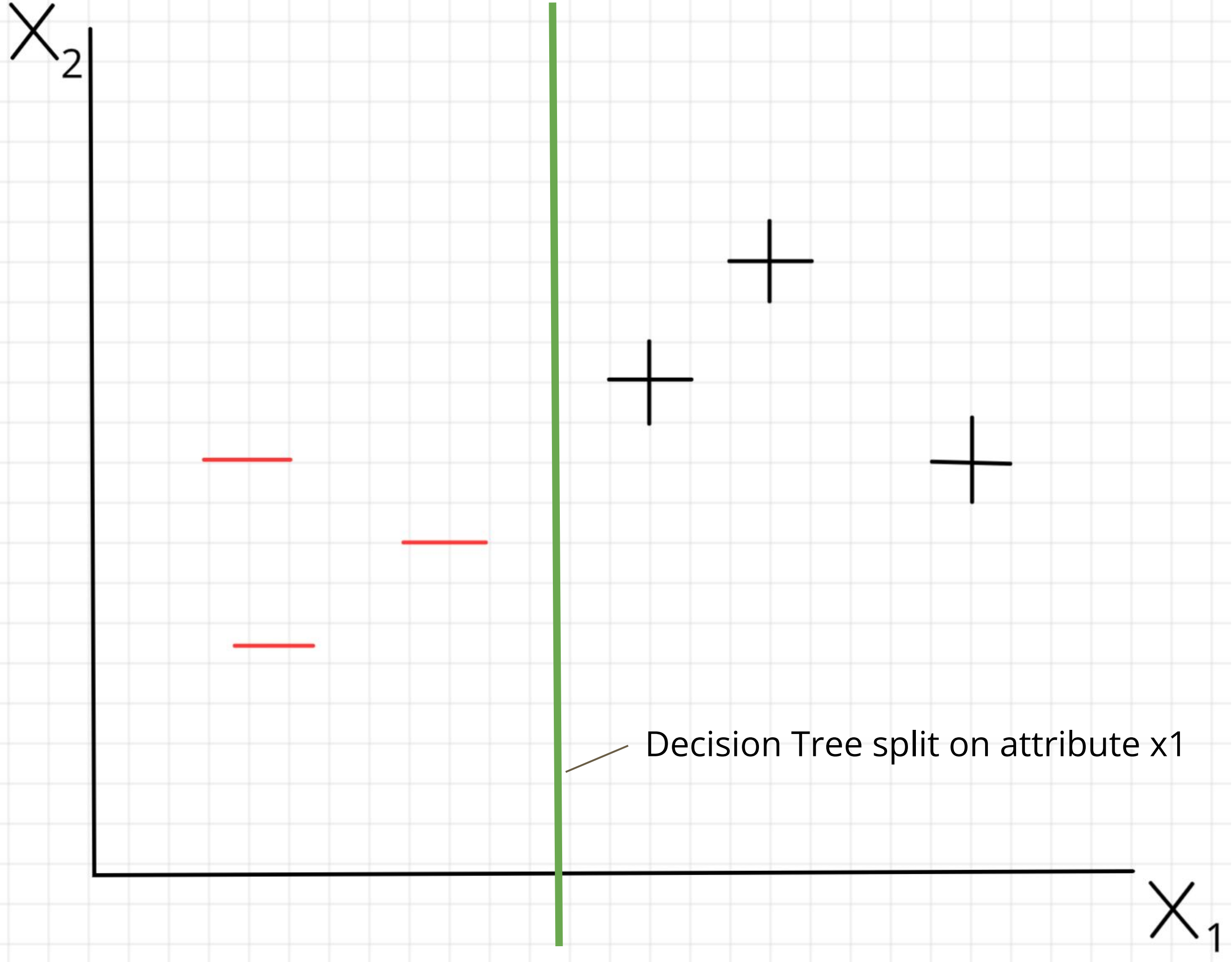
Support Vector Machines

Boston University CS 506 - Lance Galletti

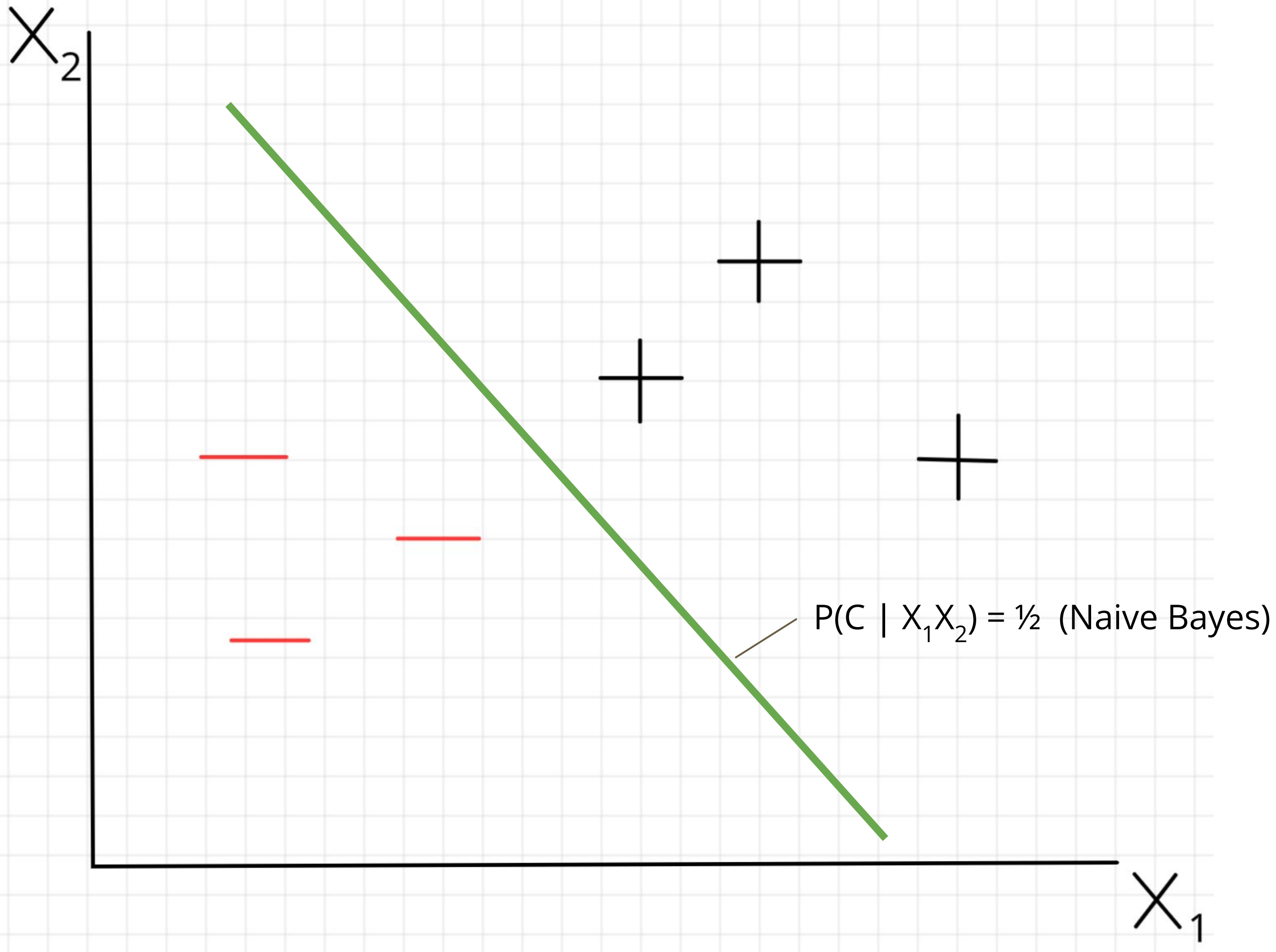


X_2
X_1


X2
X1
Nearest Neighbor decision boundary

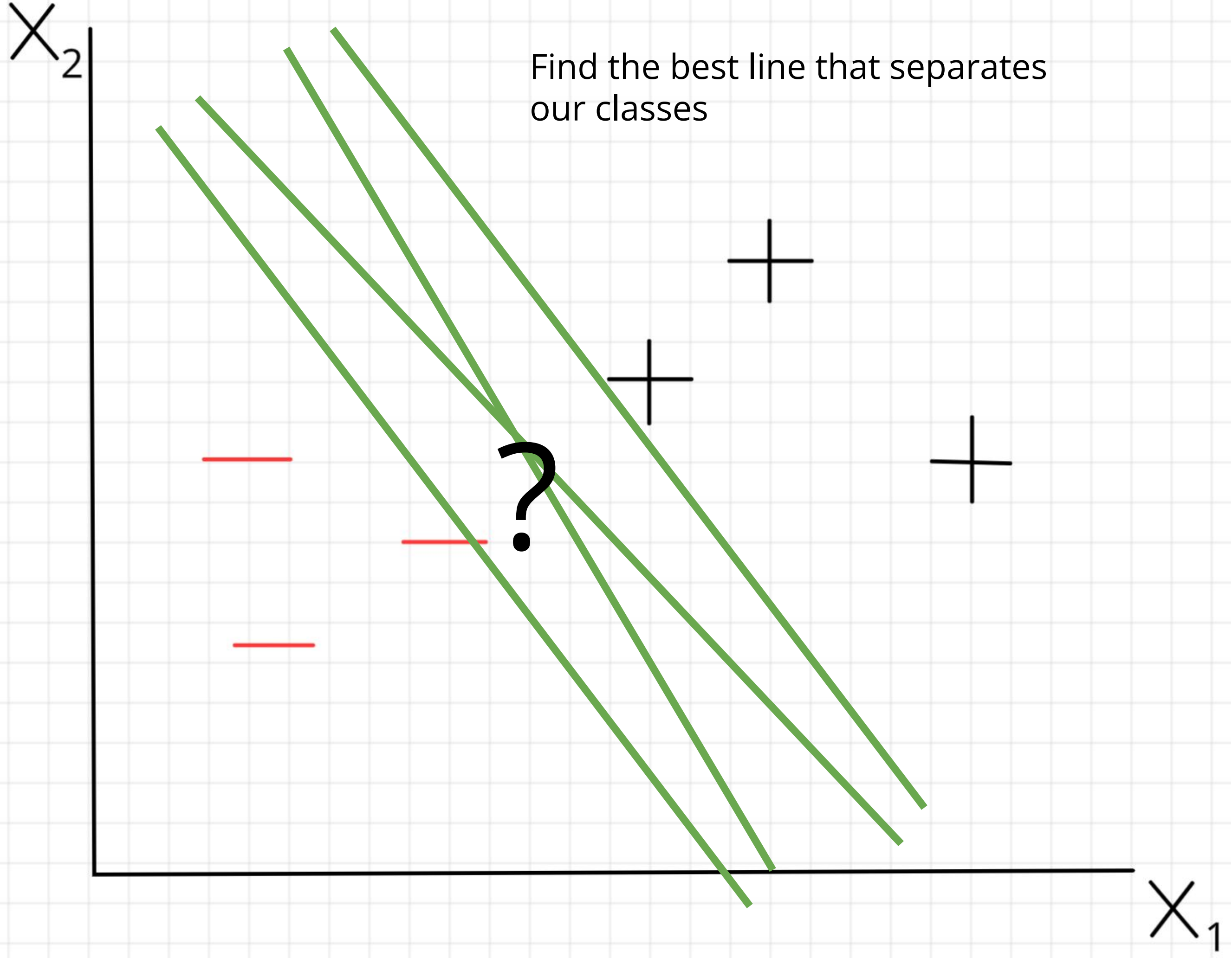


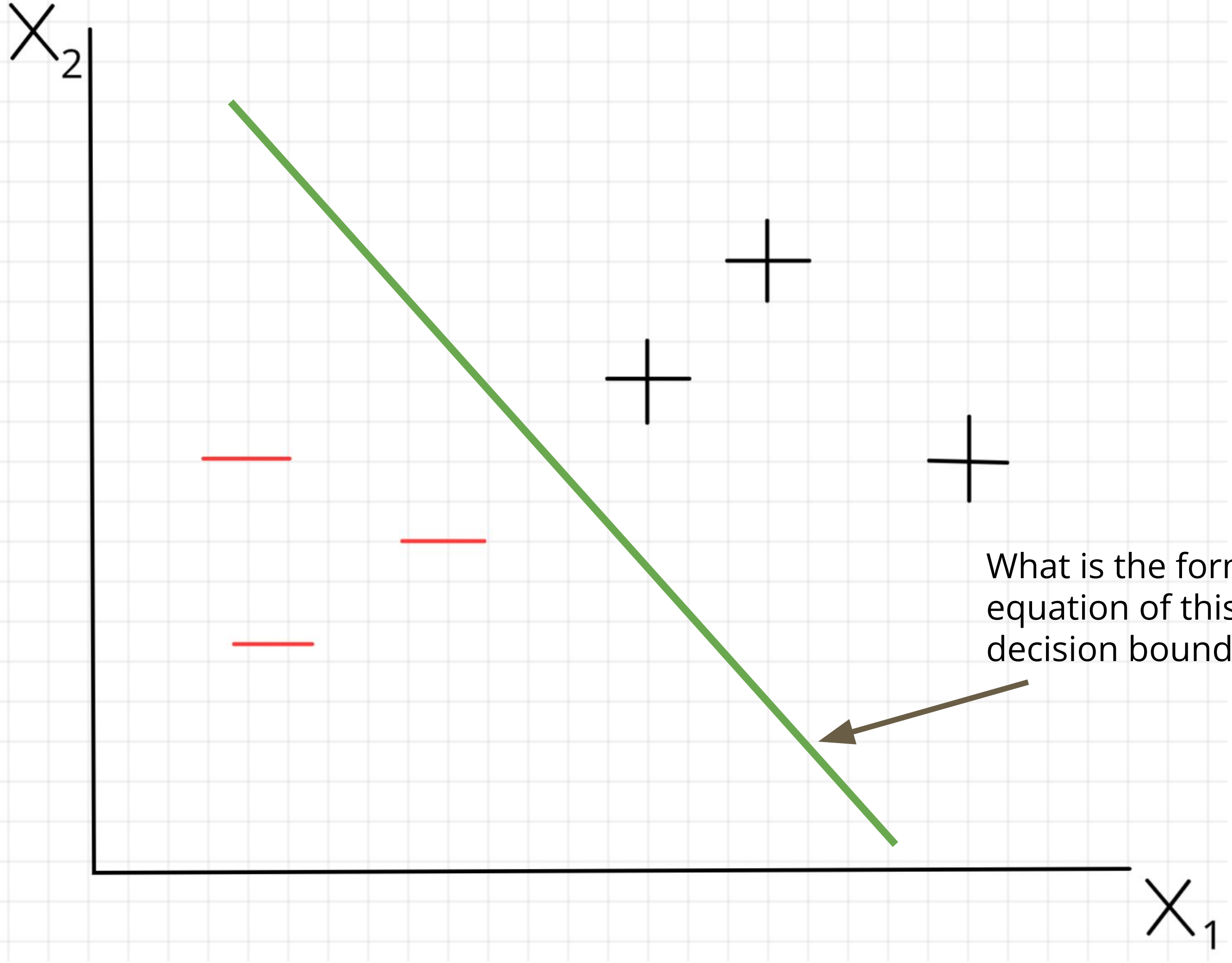
X_2
X_1
Decision Tree split on attribute x1



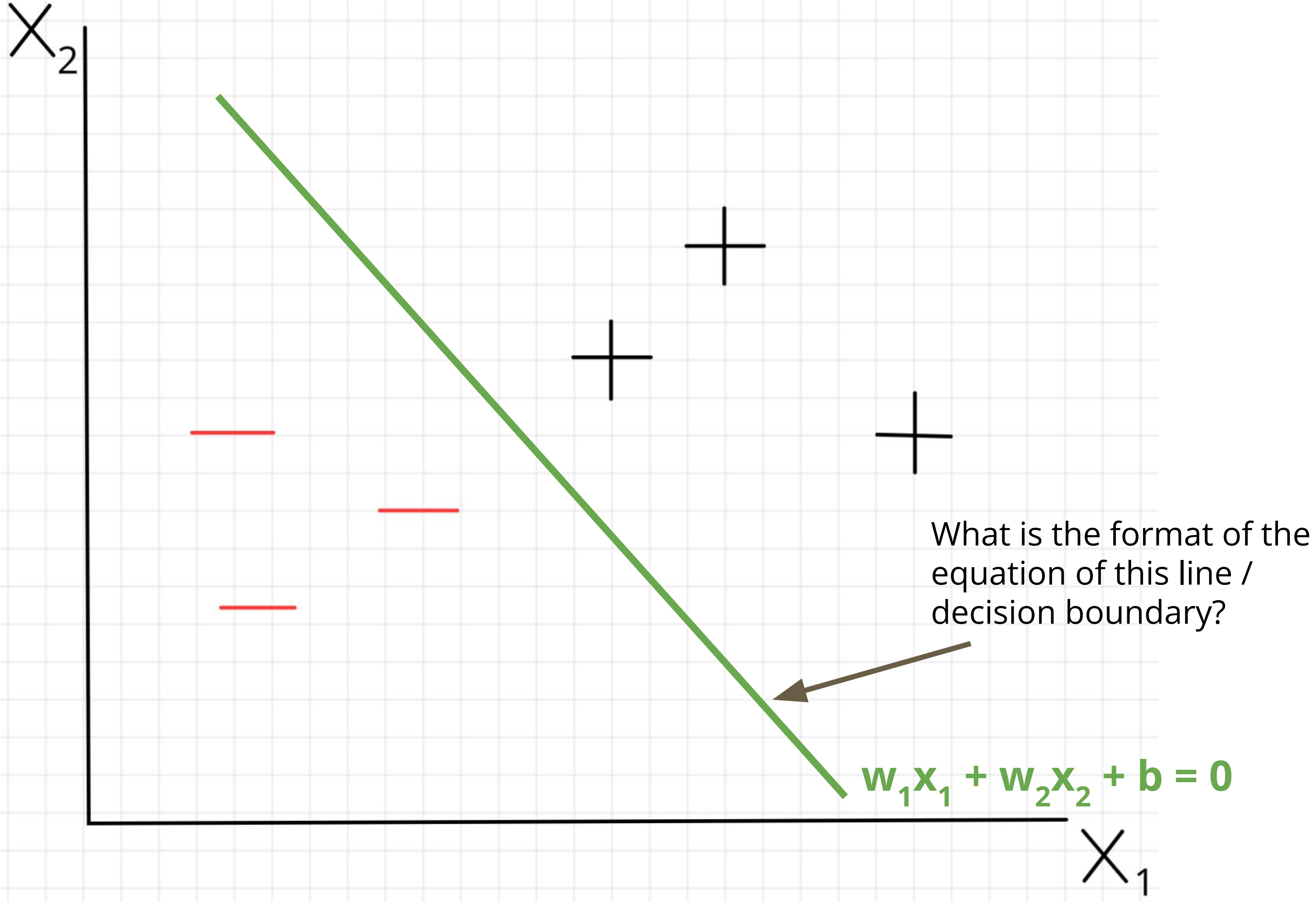
X_2
$P(C \mid X_1X_2) = ½$ (Naive Bayes)
X_1

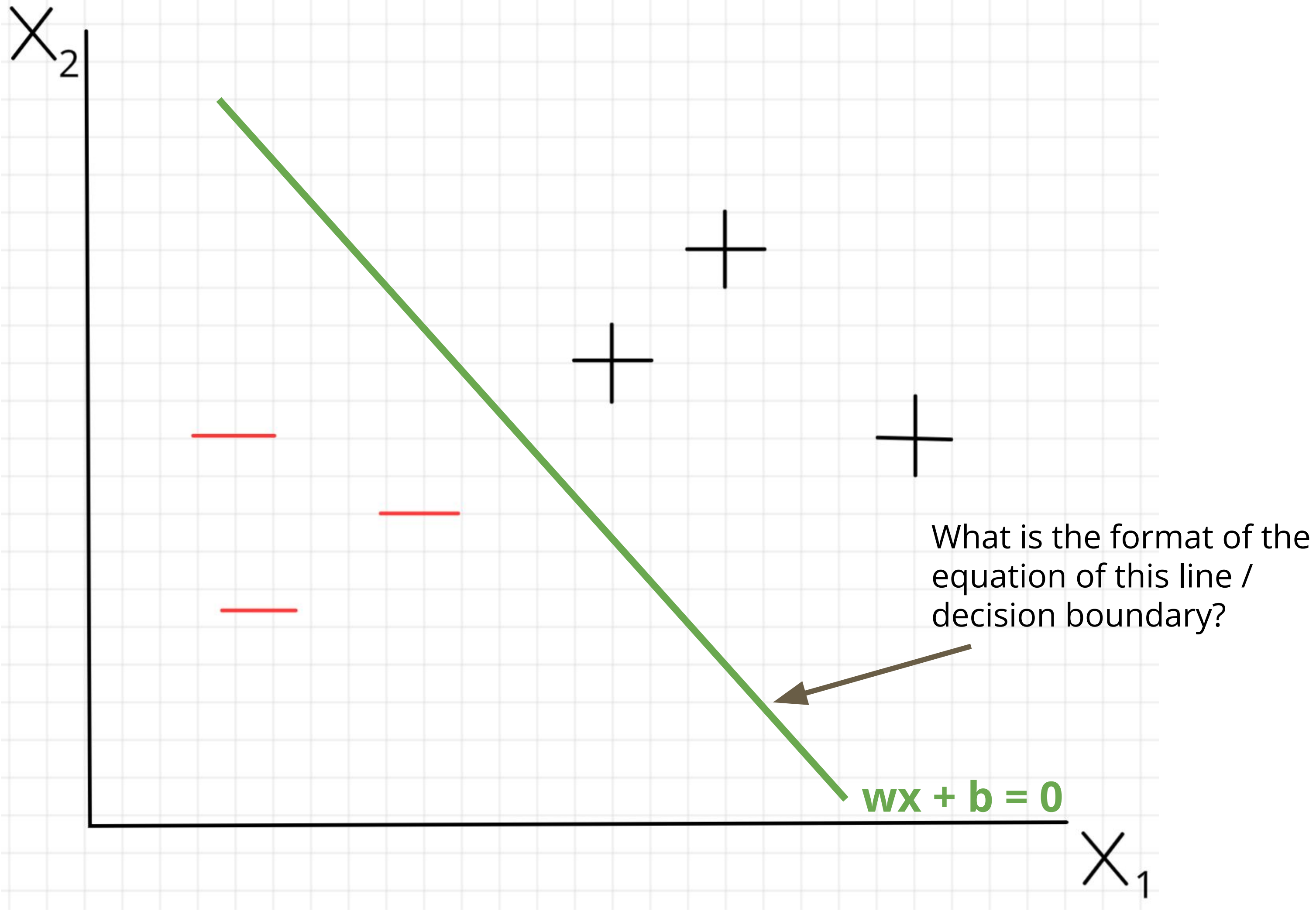
Find the best line that separates our classes

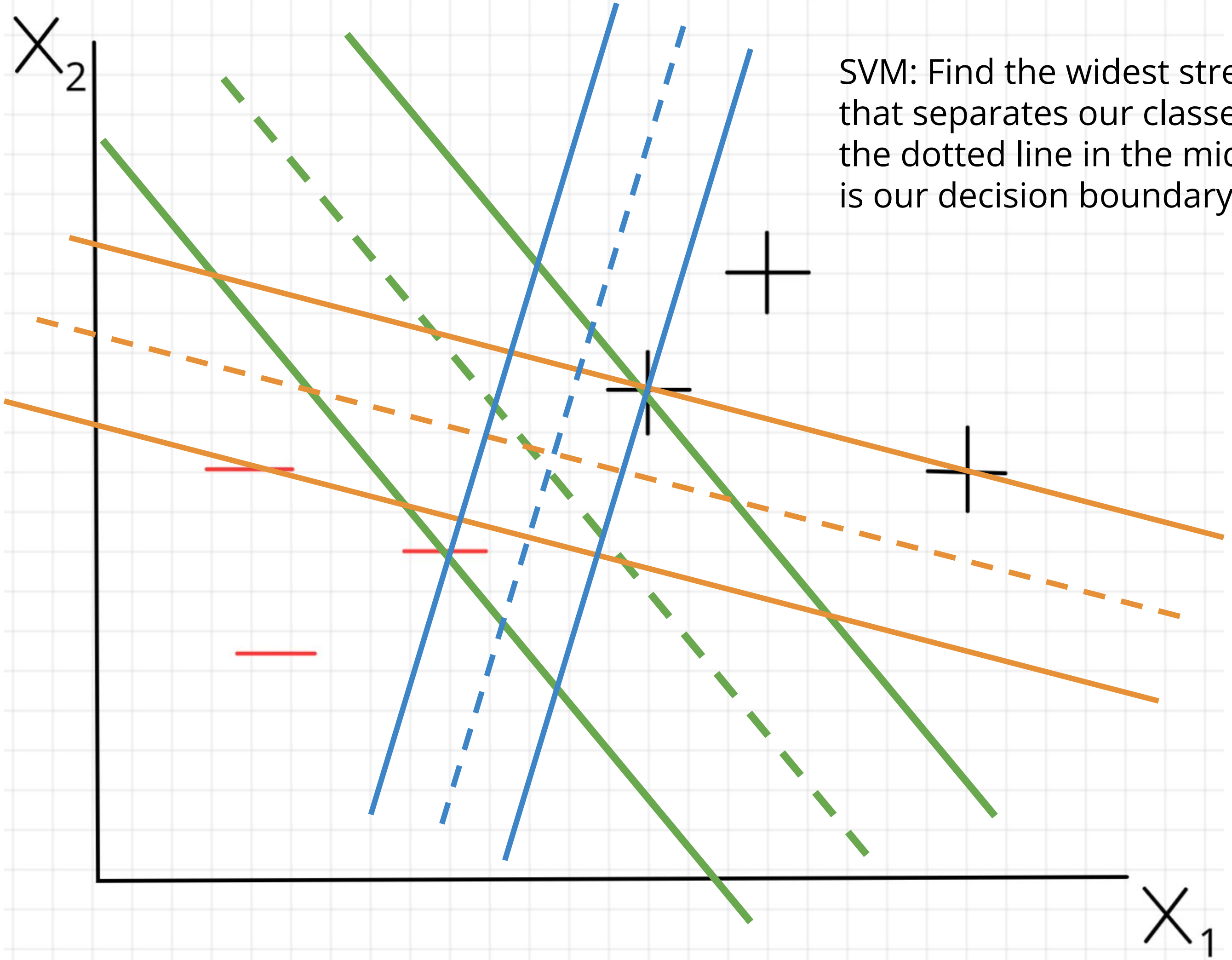




What is the format of the equation of this line / decision boundary?

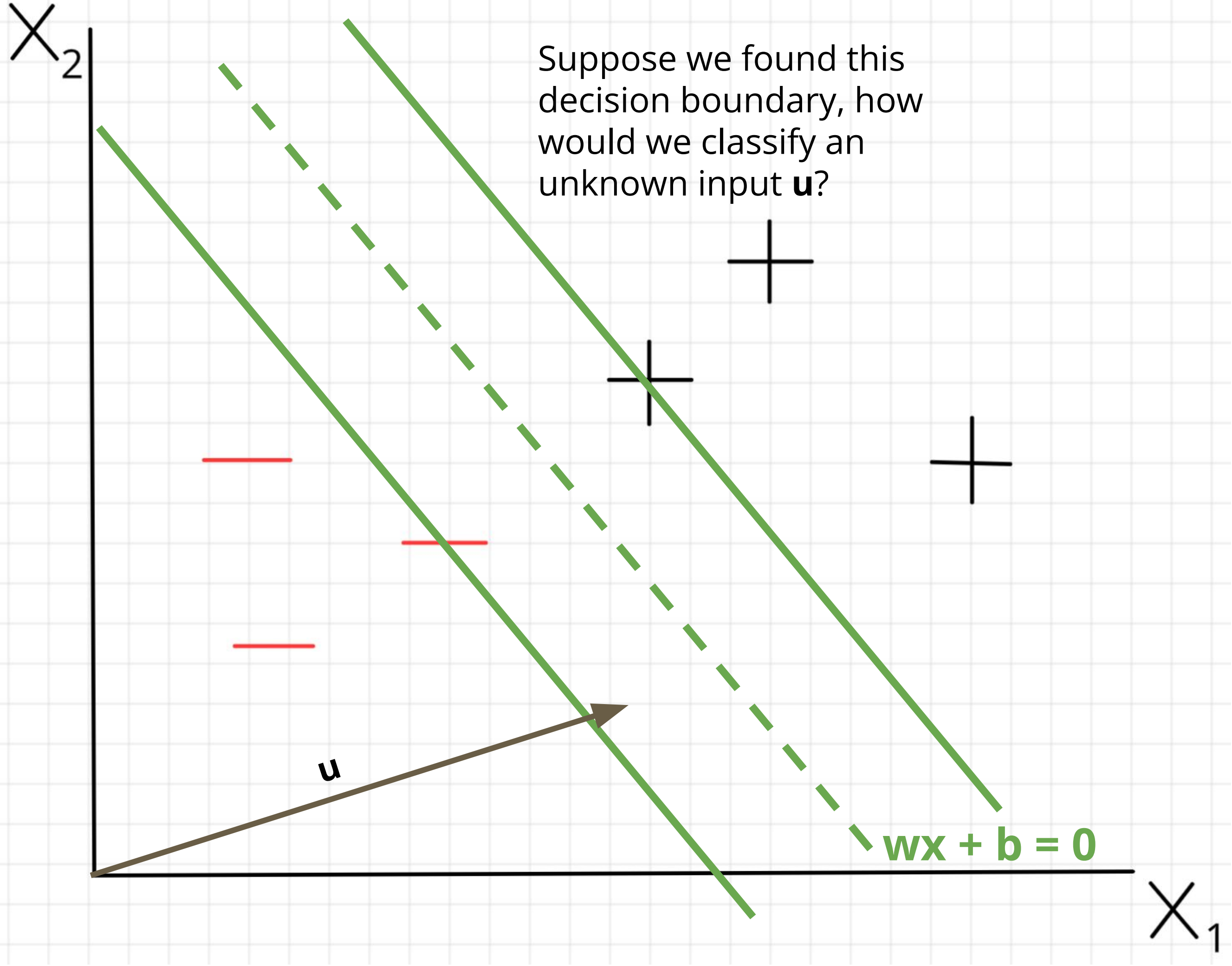

X_2
X_1
What is the format of the equation of this line / decision boundary?
$w_1x_1 + w_2x_2 + b = 0$


X_2
X_1
What is the format of the equation of this line / decision boundary?
wx + b = 0

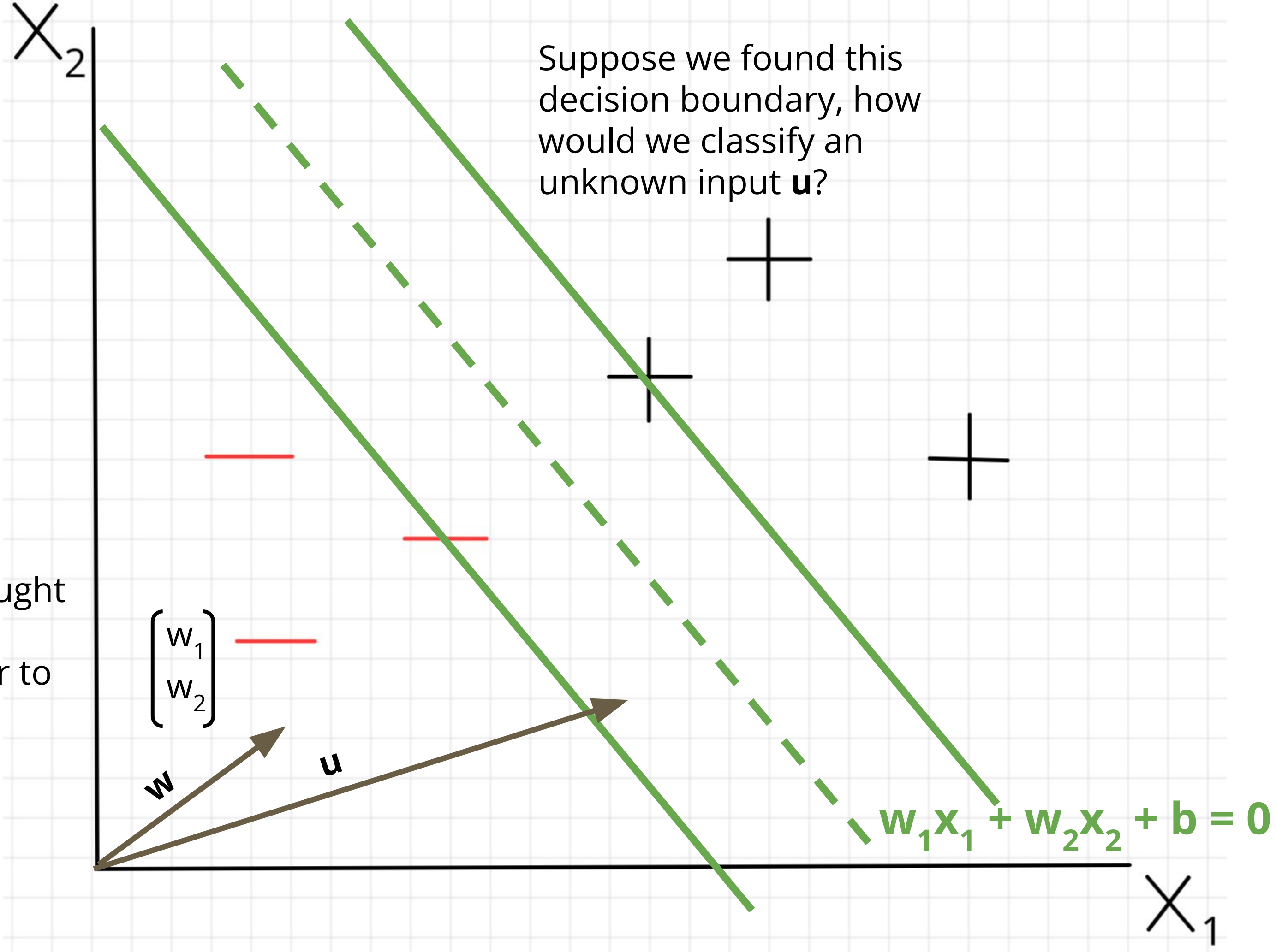


SVM: Find the widest street that separates our classes - the dotted line in the middle is our decision boundary

Suppose we found this decision boundary, how would we classify an unknown input **u**?



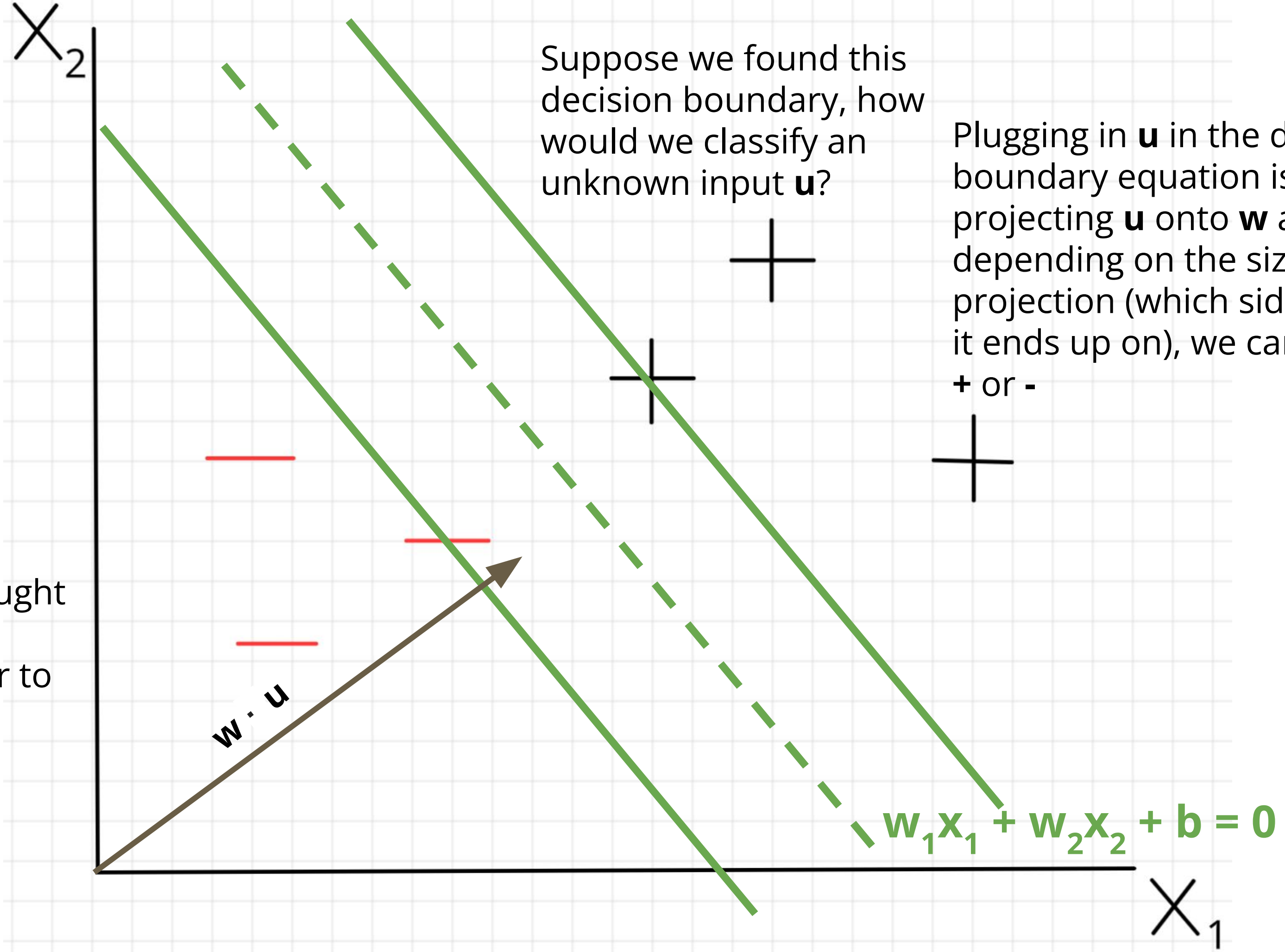
Suppose we found this decision boundary, how would we classify an unknown input **u**?



w can be thought of as a vector perpendicular to the decision boundary

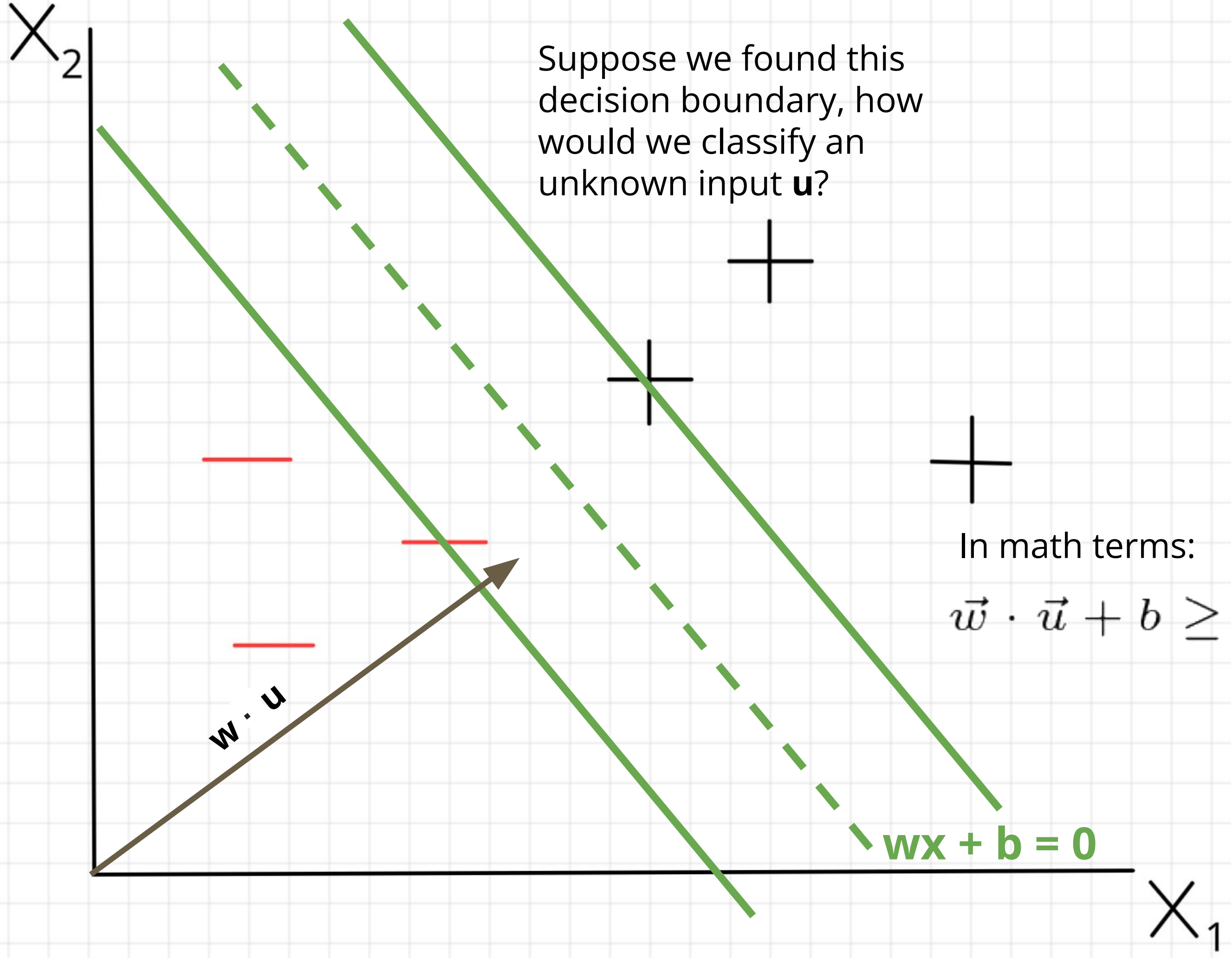
Suppose we found this decision boundary, how would we classify an unknown input **u**?

Plugging in **u** in the decision boundary equation is equivalent to projecting **u** onto **w** and depending on the size of that projection (which side of the street it ends up on), we can classify it as **+** or **-**



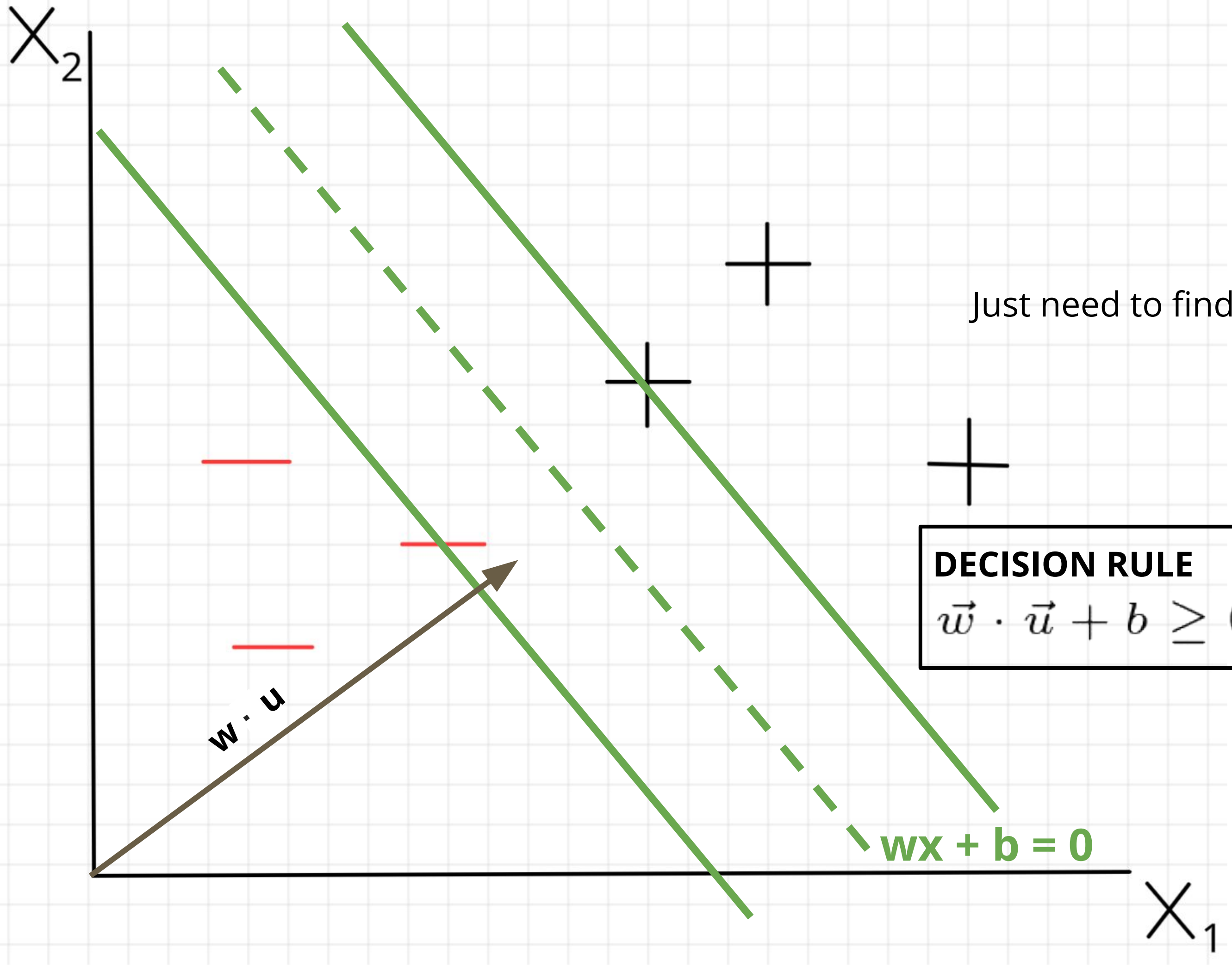
w can be thought of as a vector perpendicular to the decision boundary

Suppose we found this decision boundary, how would we classify an unknown input **u**?



In math terms:

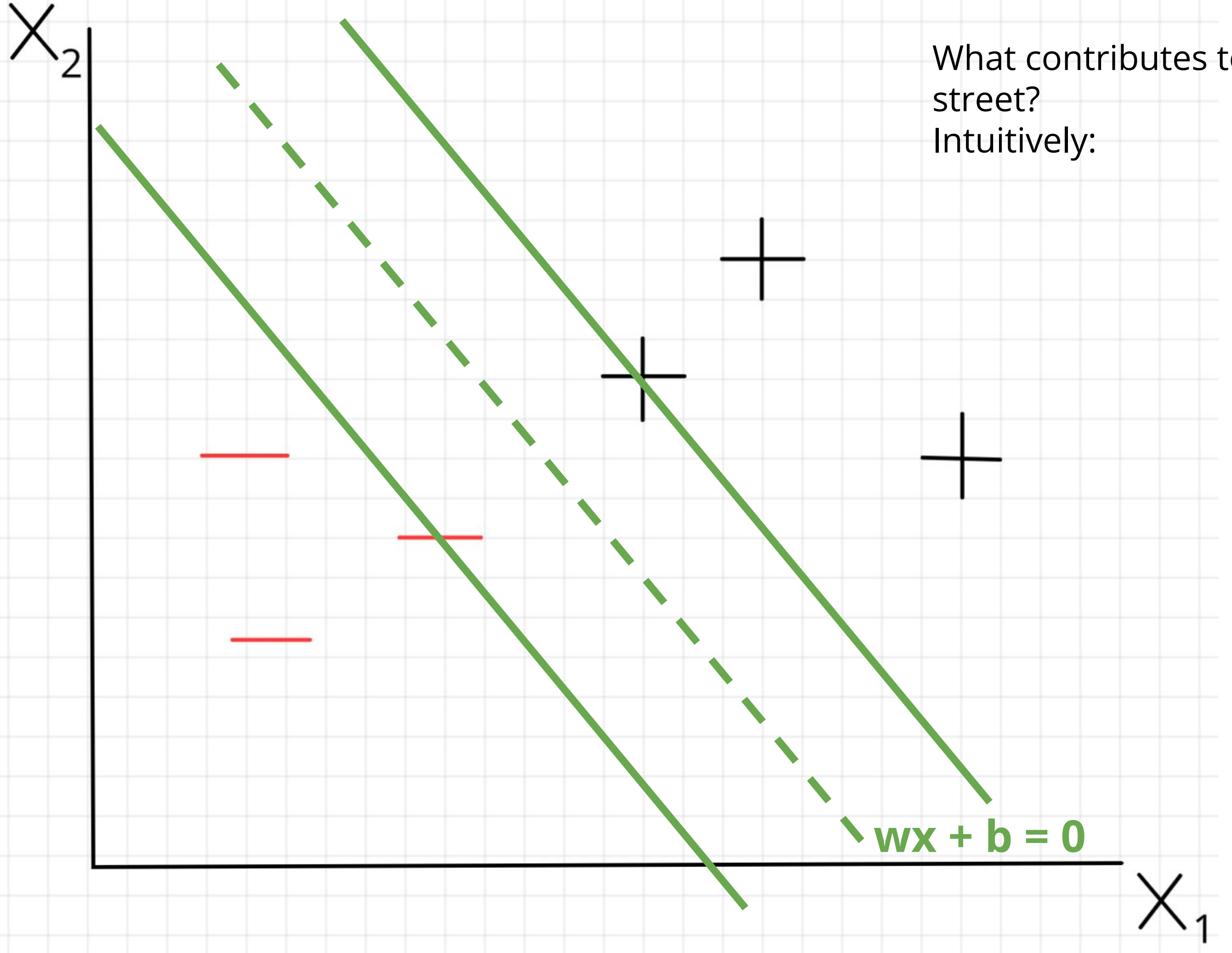
$$\vec{w} \cdot \vec{u} + b \geq 0 \quad \text{then } +$$



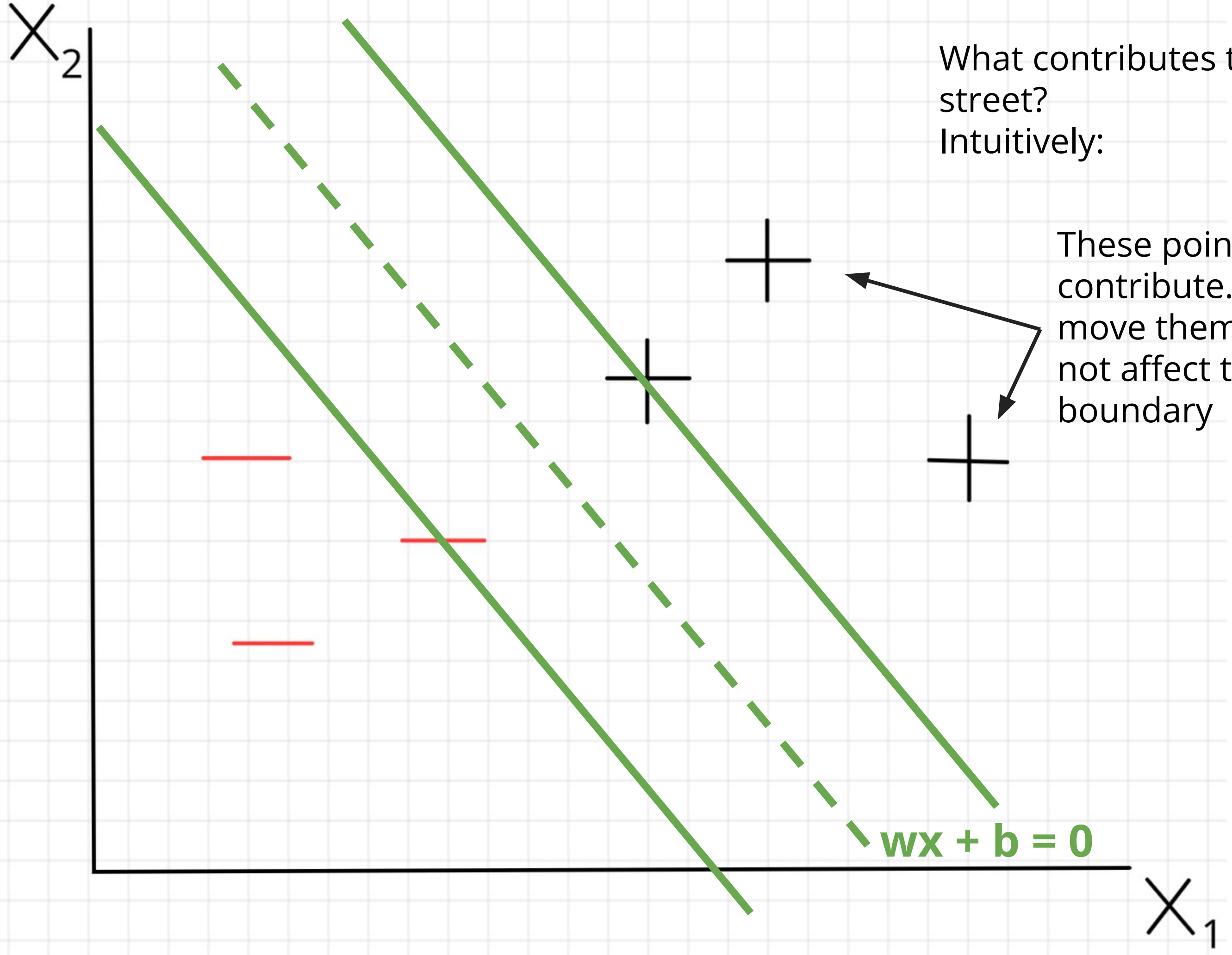
Just need to find **w** and **b**.

DECISION RULE

$$\vec{w} \cdot \vec{u} + b \geq 0 \quad \text{then } +$$



What contributes to the widest street?
Intuitively:

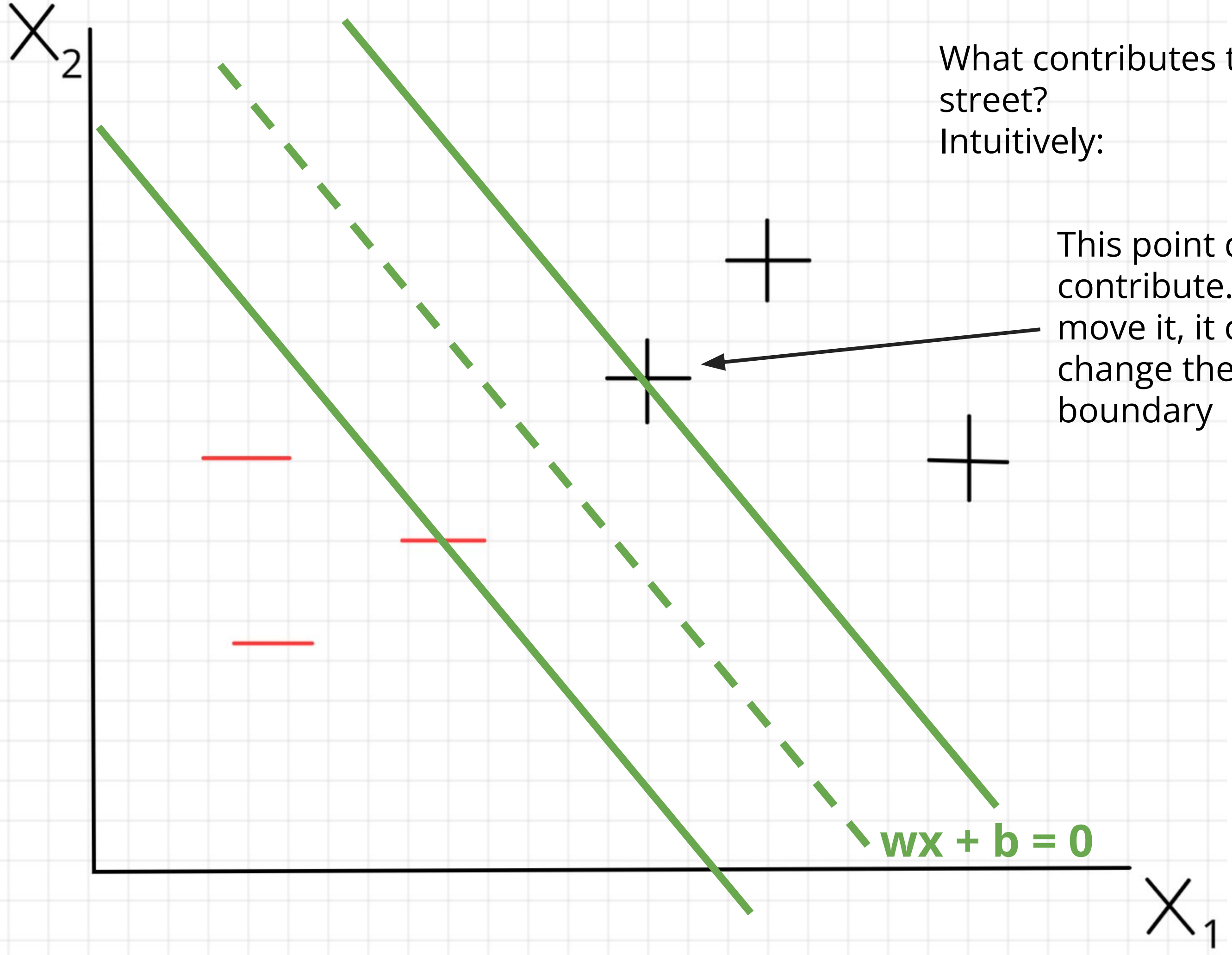


What contributes to the widest street?

Intuitively:

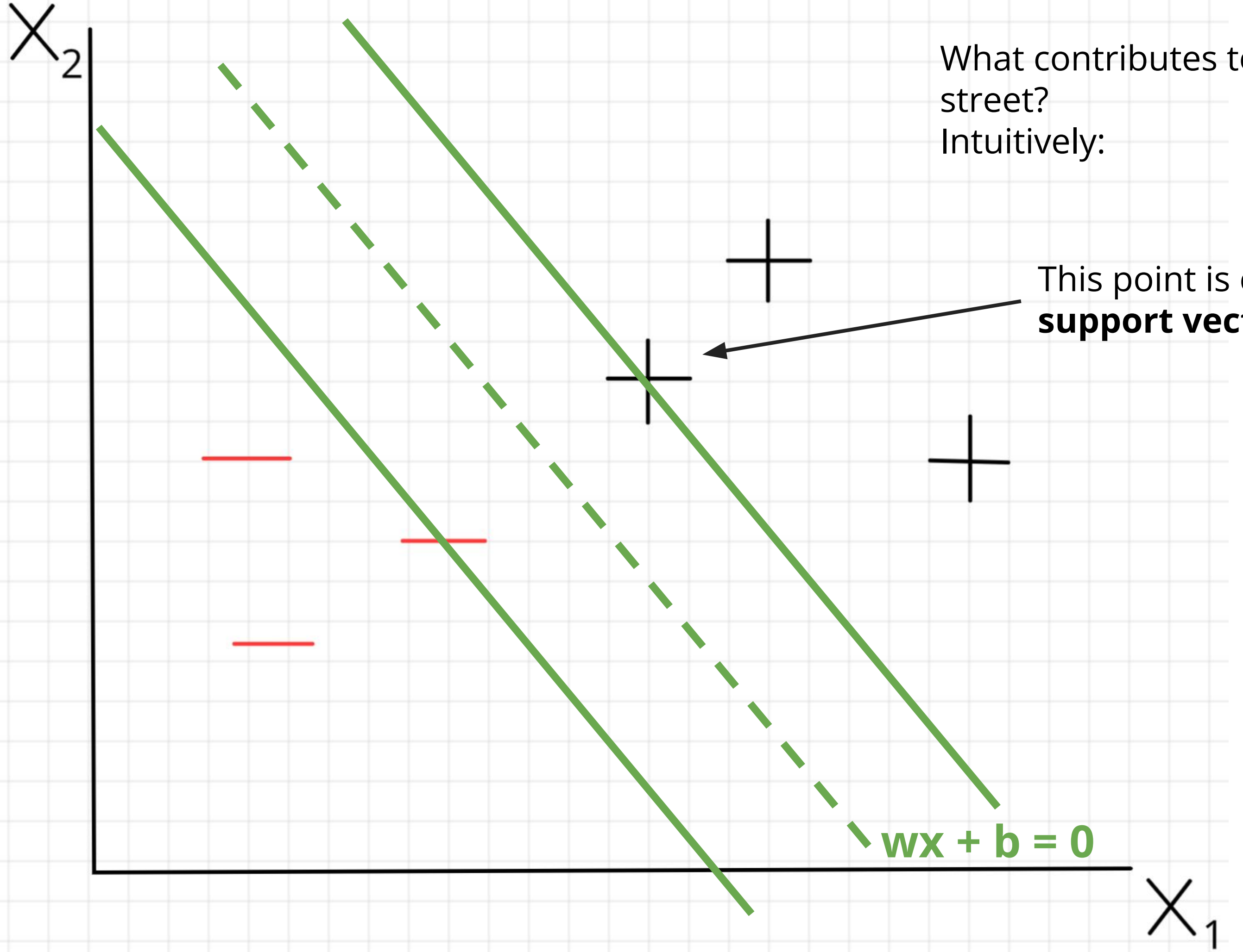
These points don’t contribute. If we were to move them they would not affect the decision boundary

What contributes to the widest street?
Intuitively:



This point does contribute. If we were to move it, it could totally change the decision boundary

What contributes to the widest street?
Intuitively:



Worksheet a) and b)

How to find the widest street

We want our samples to lie beyond the street. That is:

$$\vec{w} \cdot \vec{x}_+ + b \geq 1$$

$$\vec{w} \cdot \vec{x}_- + b \leq -1$$

Note: for an unknown **u**, we can have

$$-1 < \vec{w} \cdot \vec{u} + b < 1$$

How to find the widest street

Let's introduce a variable

$$y_i = \begin{cases} +1 & \text{if } x_i \text{ is a } + \text{sample} \\ -1 & \text{if } x_i \text{ is a } - \text{sample} \end{cases}$$

Note: this is effectively the class label of x_i

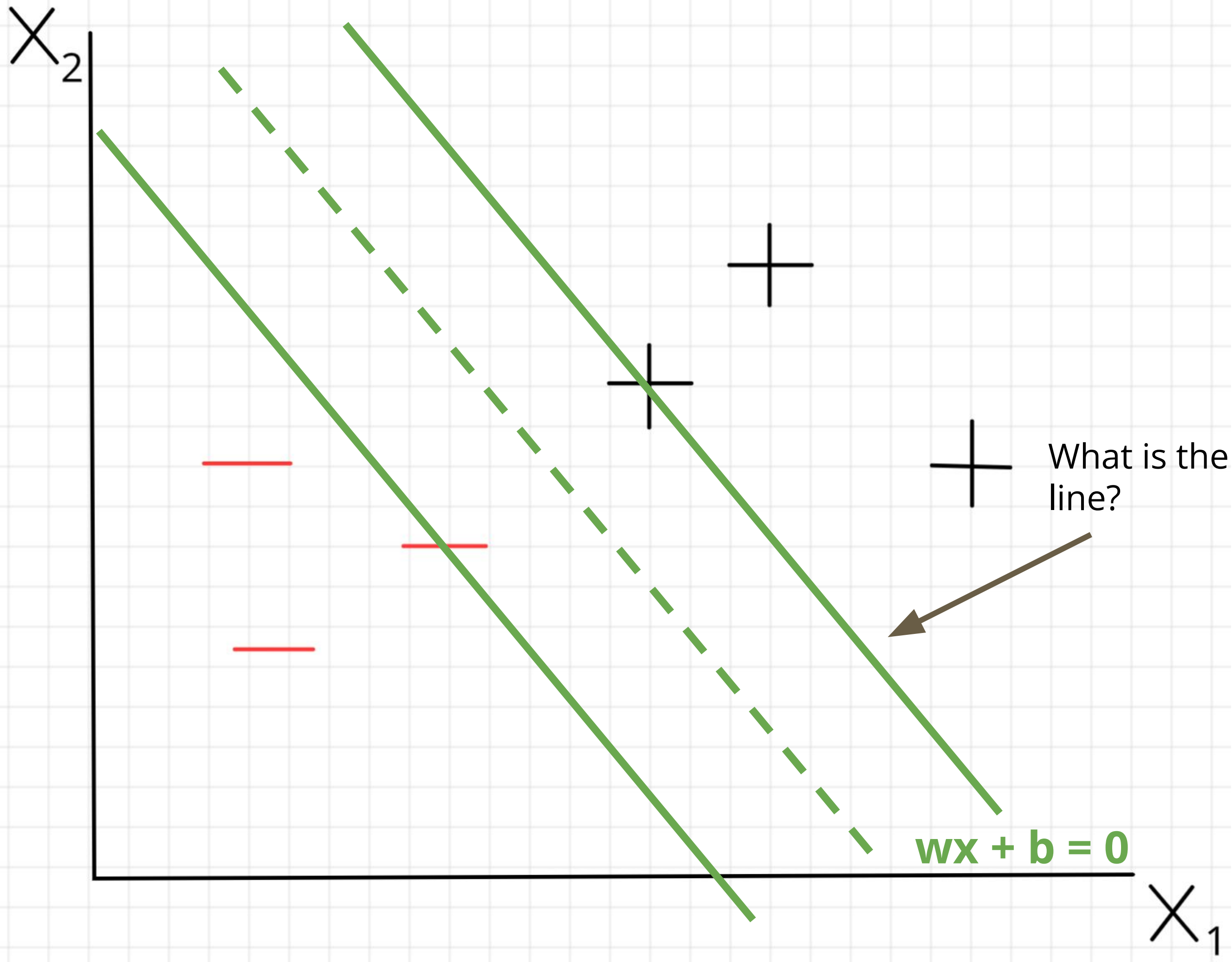
How to find the widest street

If we multiply our sample decision rules by this new variable:

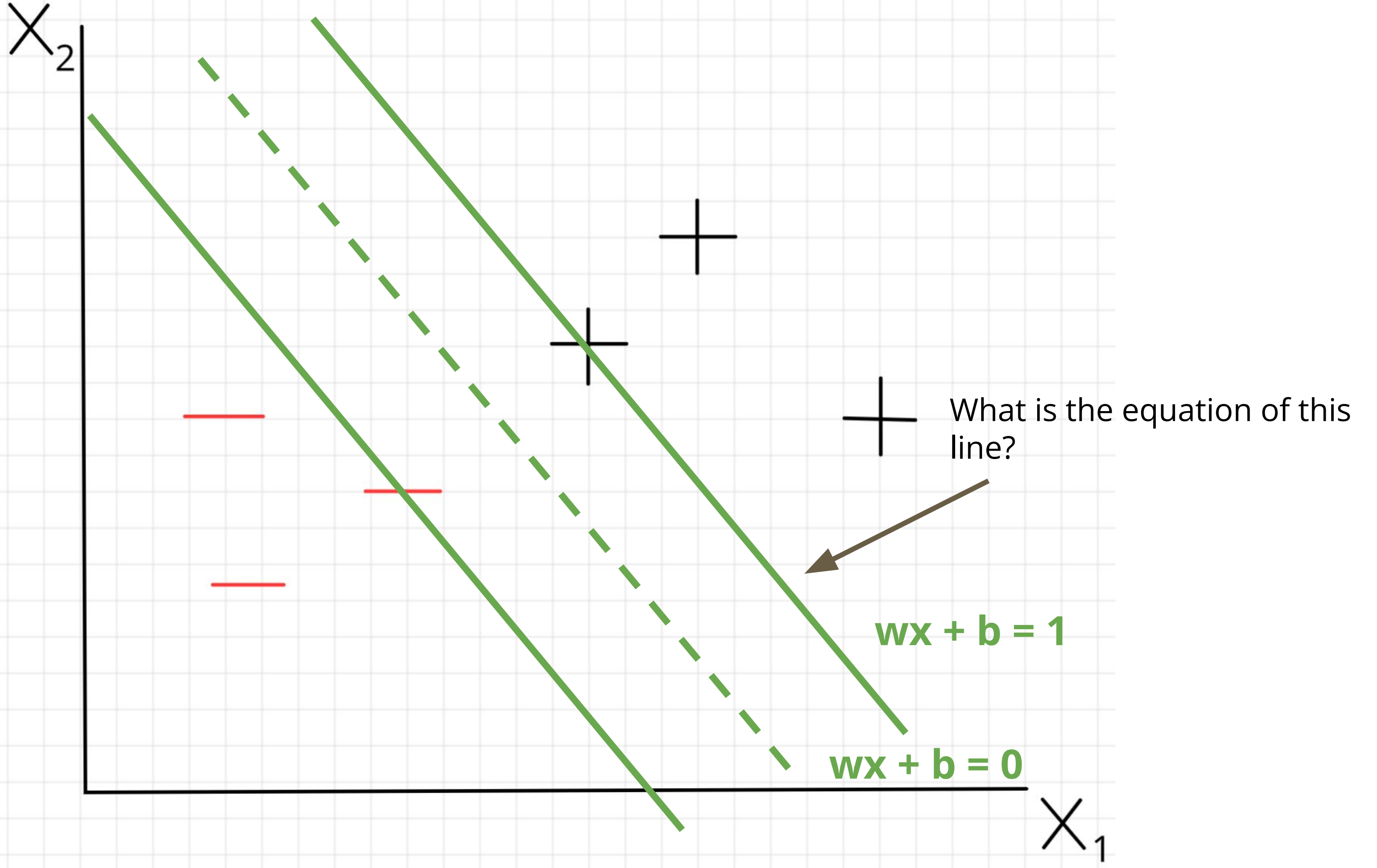
$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1$$

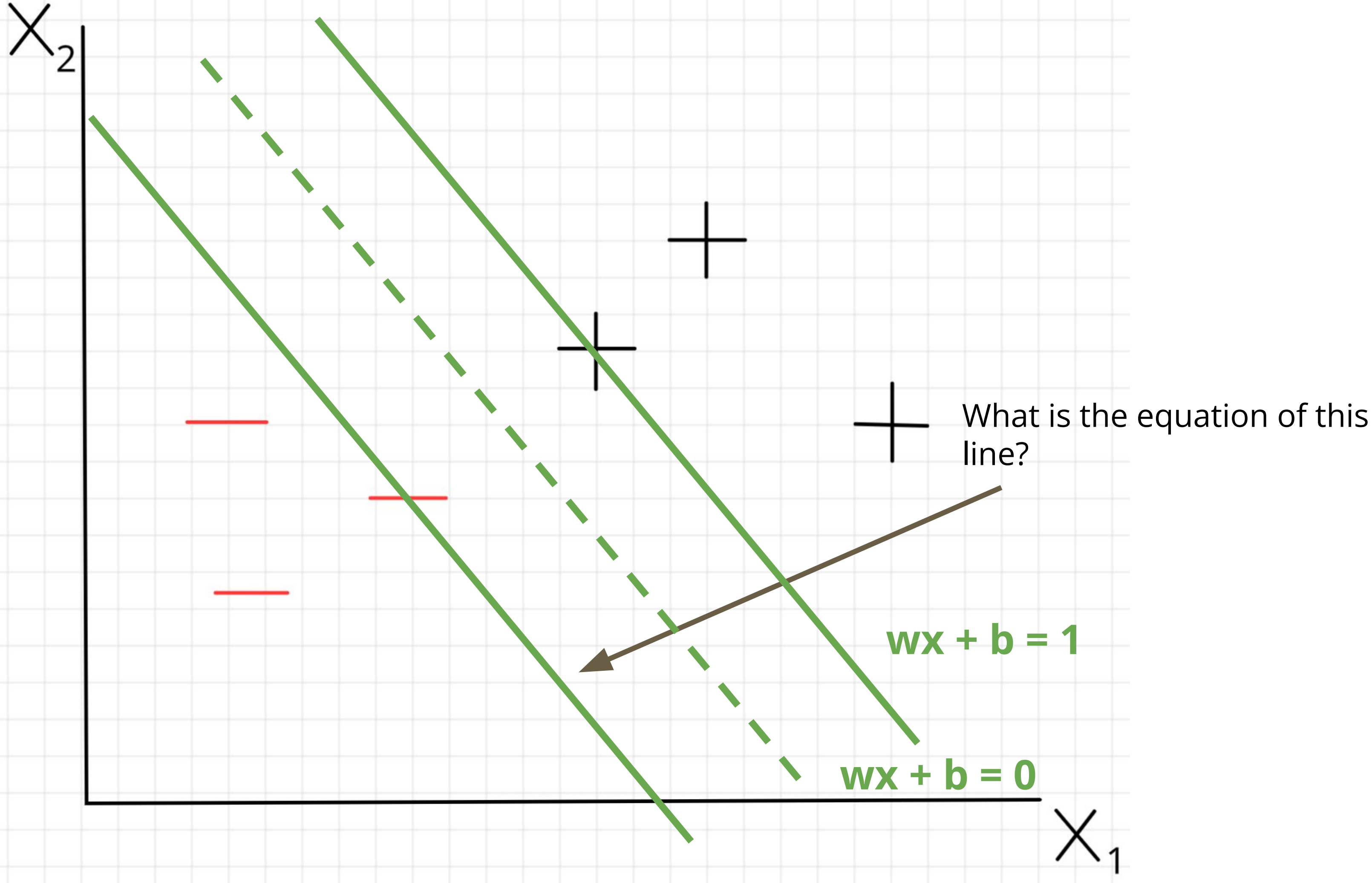
Meaning, for x_i on the decision boundary, we want:

$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 = 0$$

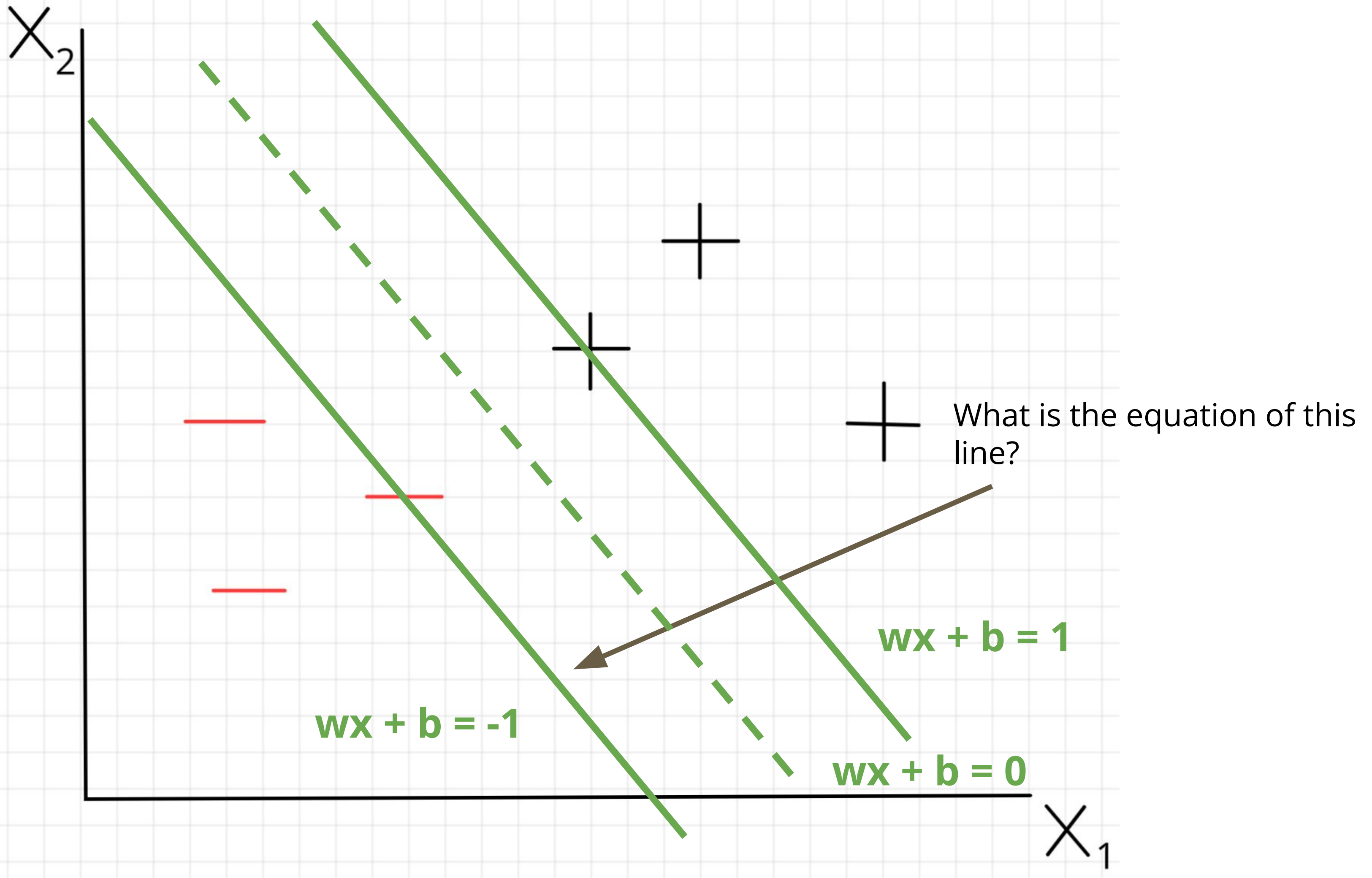


X_2
X_1
What is the equation of this line?
wx + b = 0

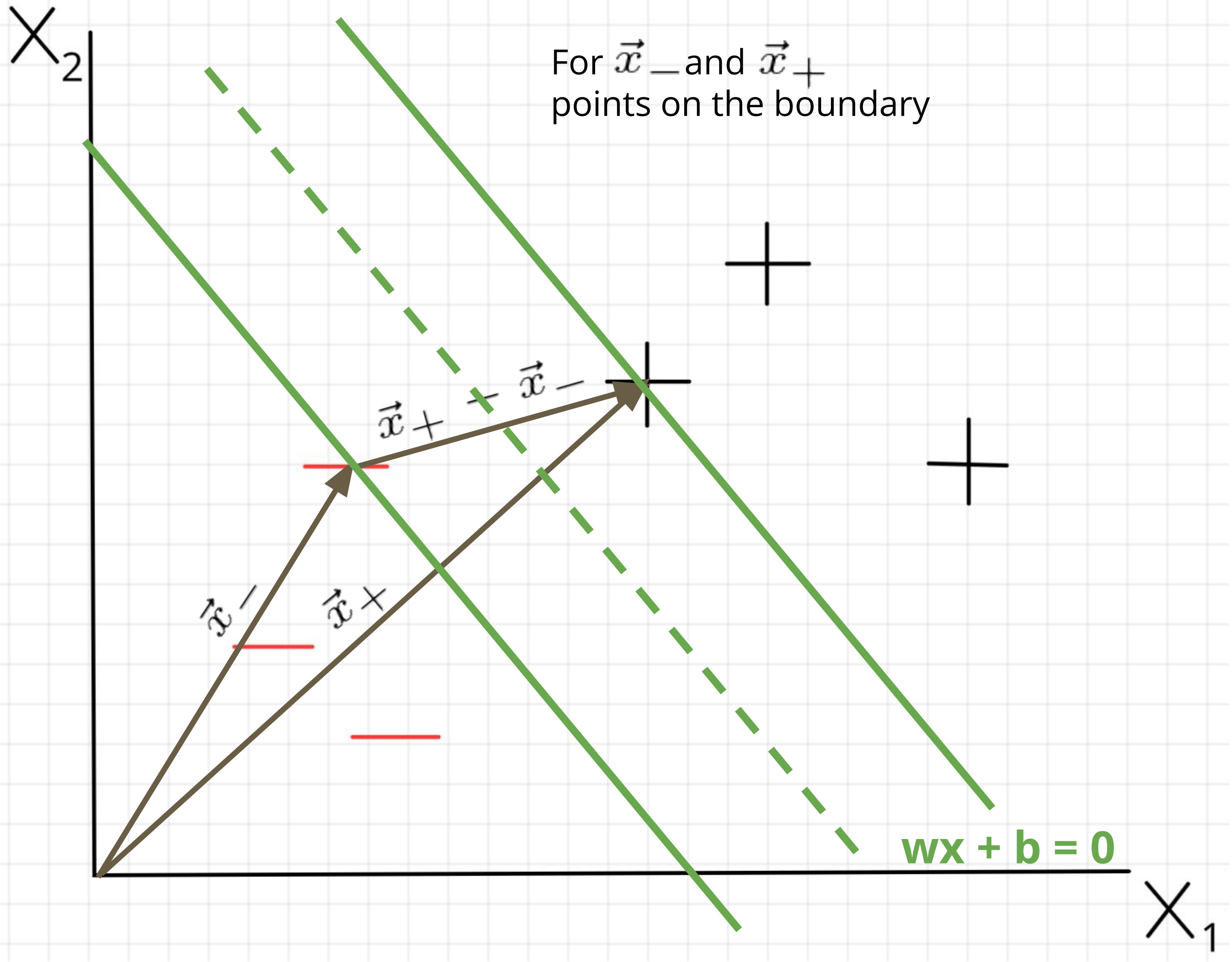

X_2
What is the equation of this line?
wx + b = 1
wx + b = 0
X_1



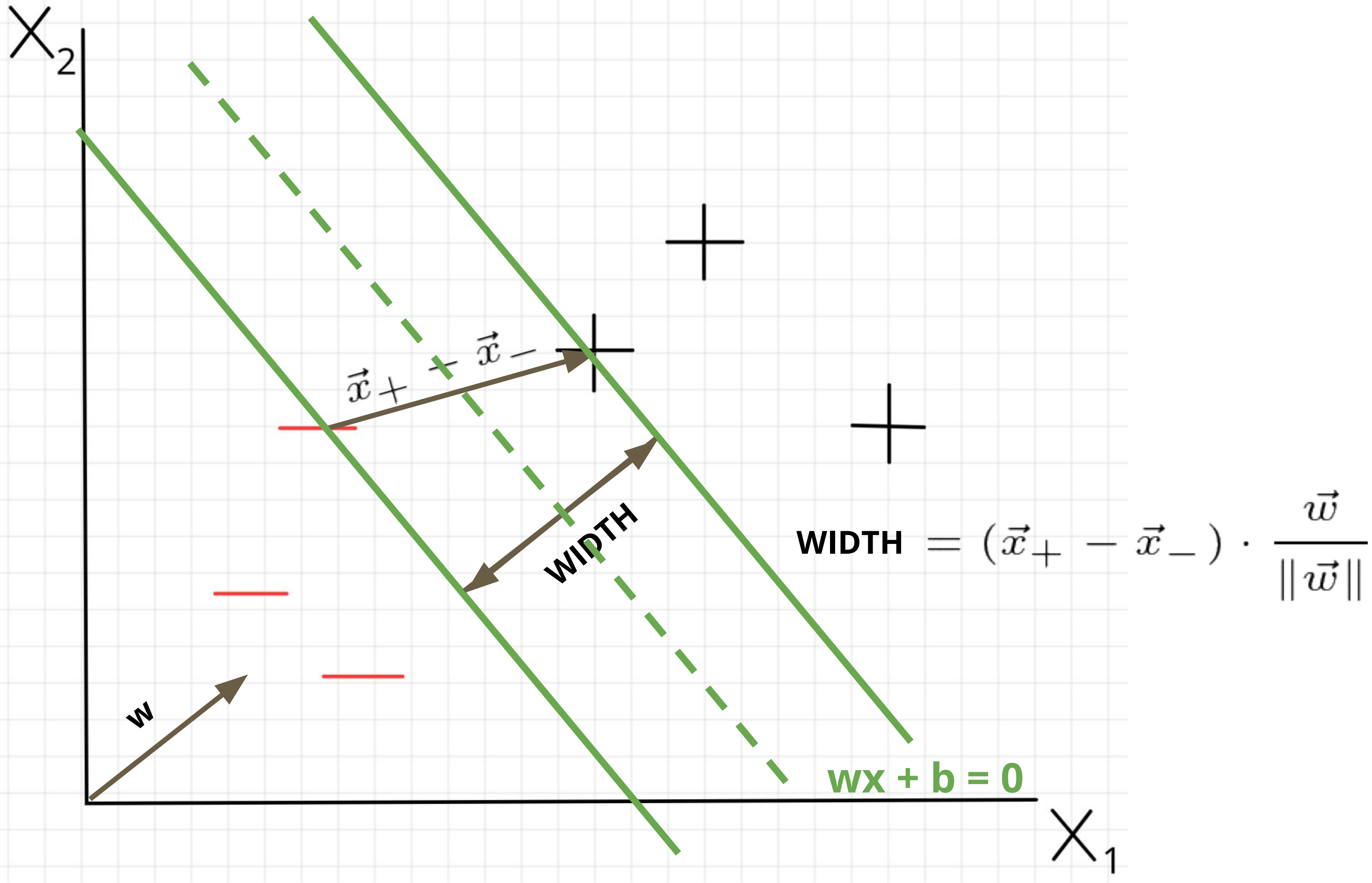
X_2
X_1
What is the equation of this line?
wx + b = 1
wx + b = 0



X2
X1
What is the equation of this line?
wx + b = 1
wx + b = -1
wx + b = 0



For $\vec{x}_-$ and $\vec{x}_+$
points on the boundary
X_2
X_1
$\vec{x}_+ - \vec{x}_-$
$\vec{x}_-$
$\vec{x}_+$
wx + b = 0



X_2
$\vec{x}_+ - \vec{x}_-$
WIDTH
w
wx + b = 0
X_1
WIDTH $= (\vec{x}_+ - \vec{x}_-) \cdot \frac{\vec{w}}{\|\vec{w}\|}$

How to find the widest street

We know that **WIDTH** $= (\vec{x}_+ - \vec{x}_-) \cdot \frac{\vec{w}}{||\vec{w}||}$ for $\vec{x}_-$ and $\vec{x}_+$ points on the boundary

And, since they are on the boundary, we know that

$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 = 0$$

Hence, **WIDTH = ?**

How to find the widest street

We know that **WIDTH** $= (\vec{x}_+ - \vec{x}_-) \cdot \dfrac{\vec{w}}{||\vec{w}||}$ for $\vec{x}_-$ and $\vec{x}_+$ points on the boundary

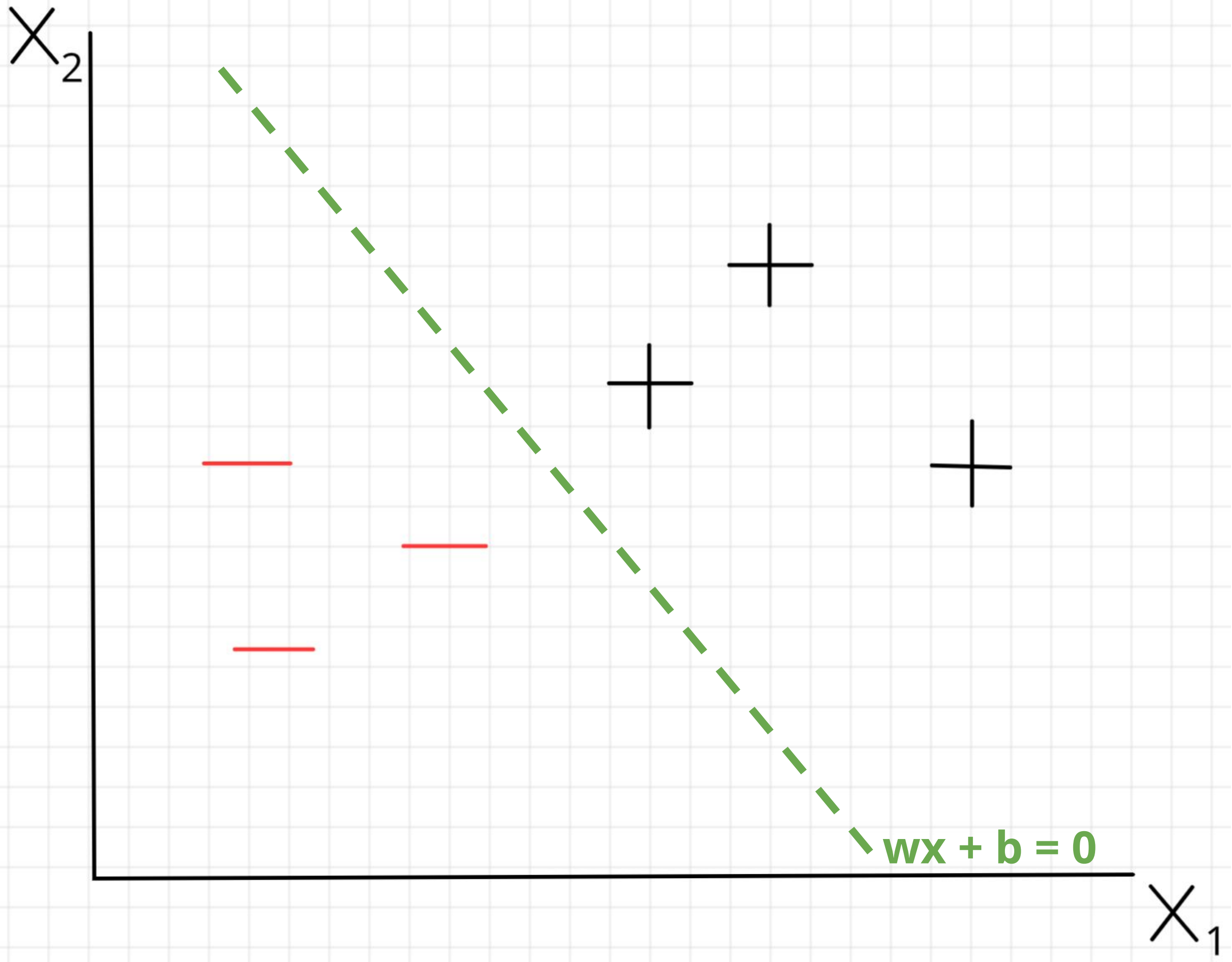
And, since they are on the boundary, we know that

$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 = 0$$

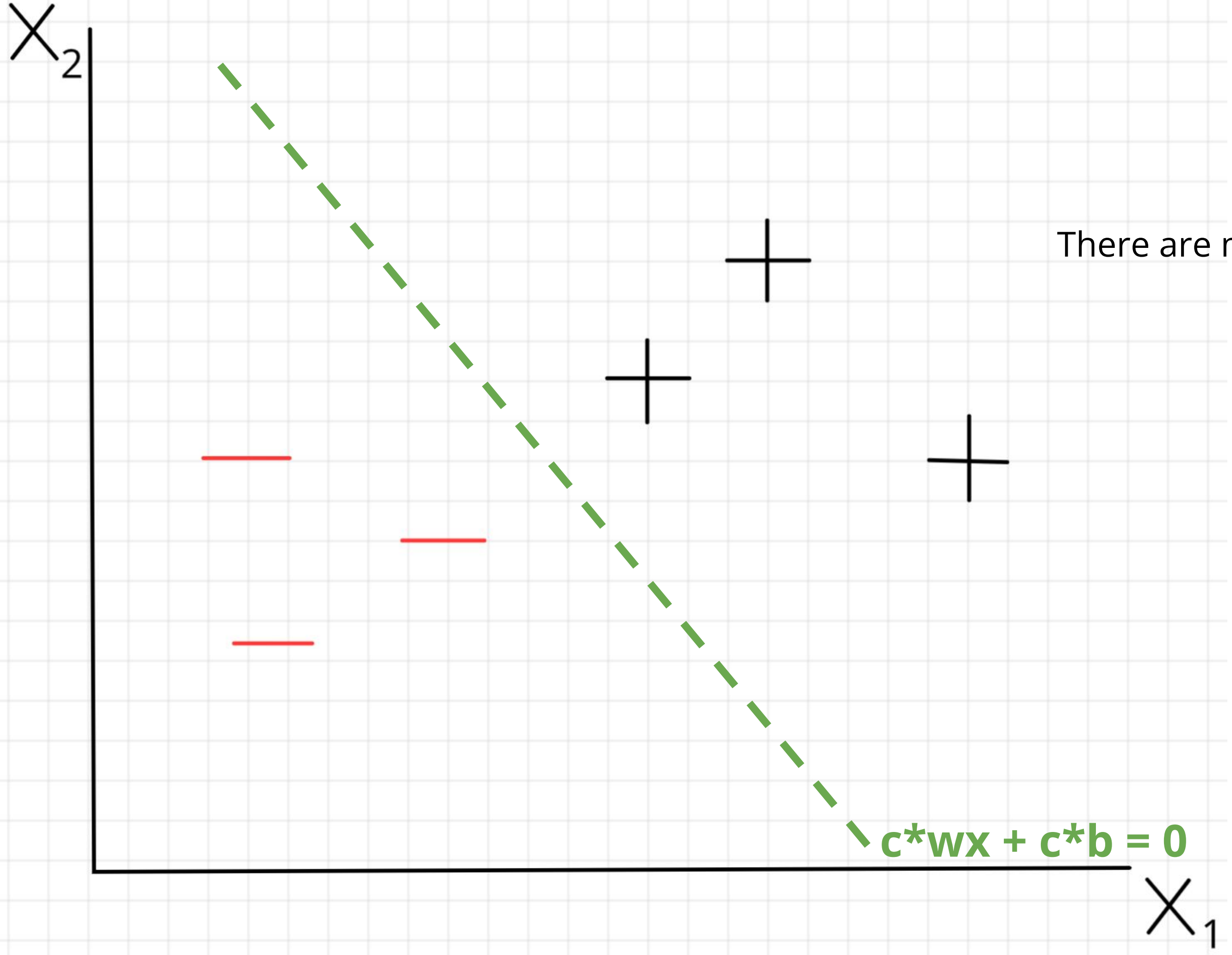
Hence, **WIDTH** $= \dfrac{2}{||\vec{w}||}$

What does that mean?

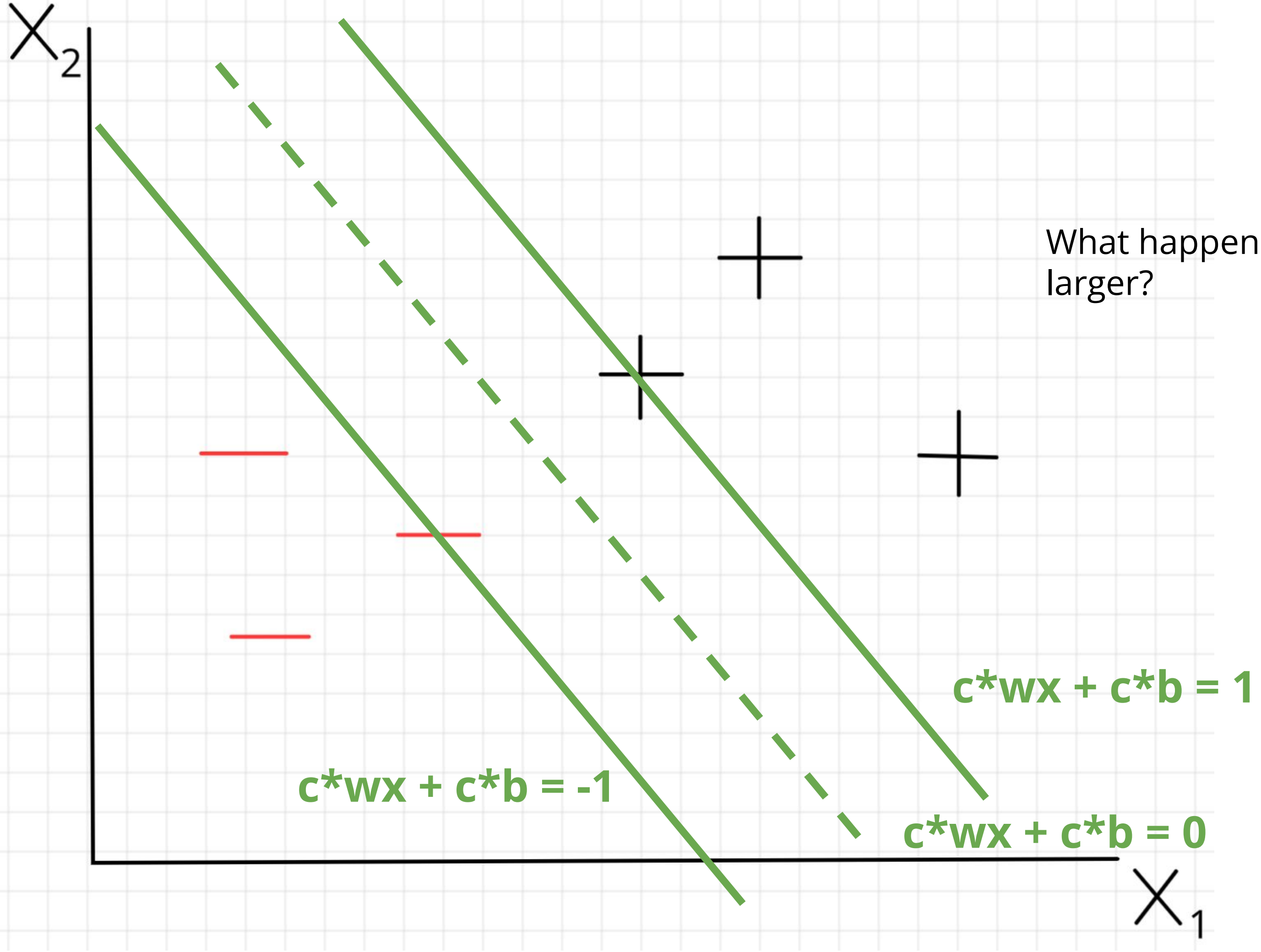
Size of **w** is inversely proportional to the width of the street.



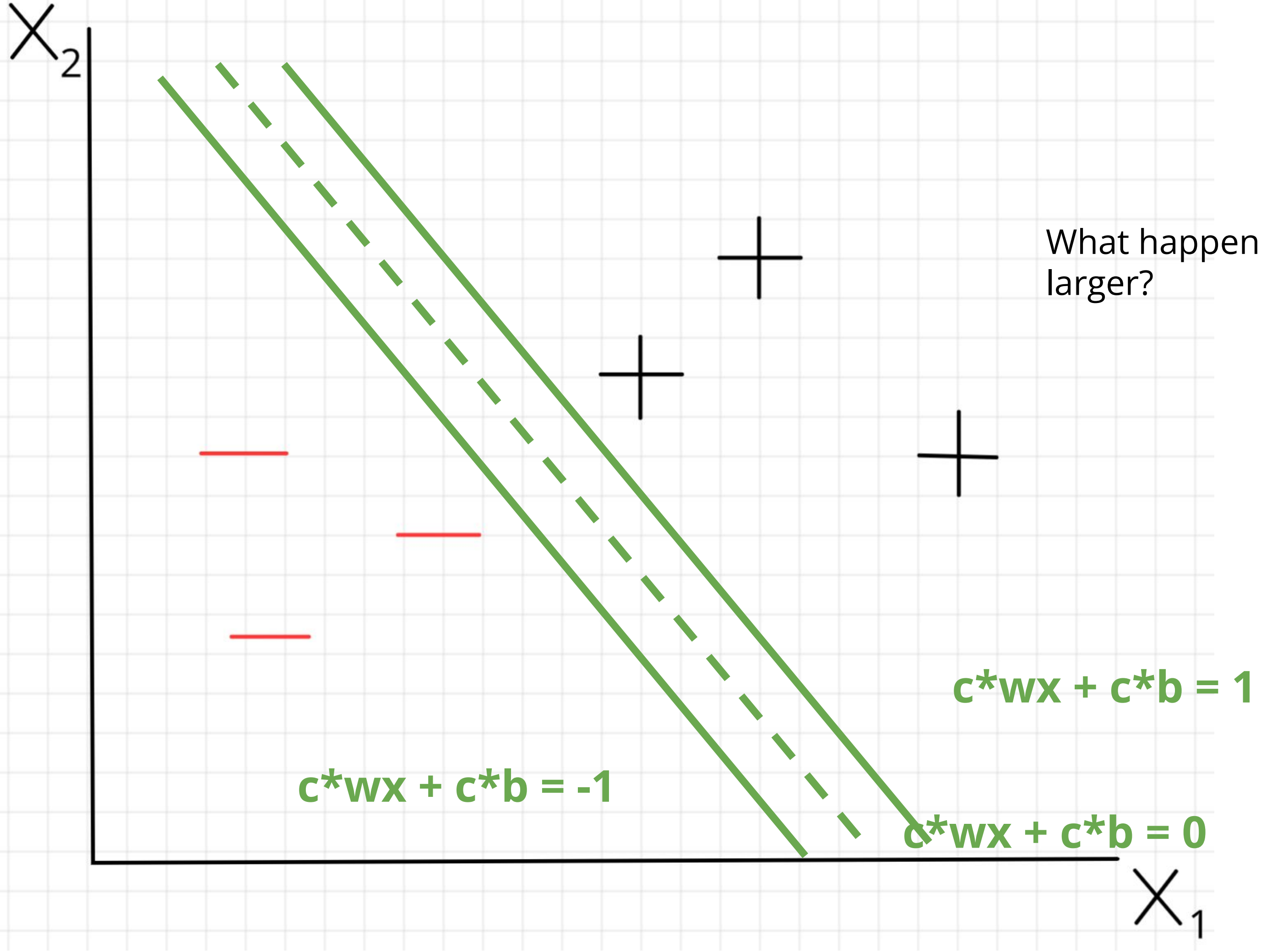
X_2
wx + b = 0
X_1



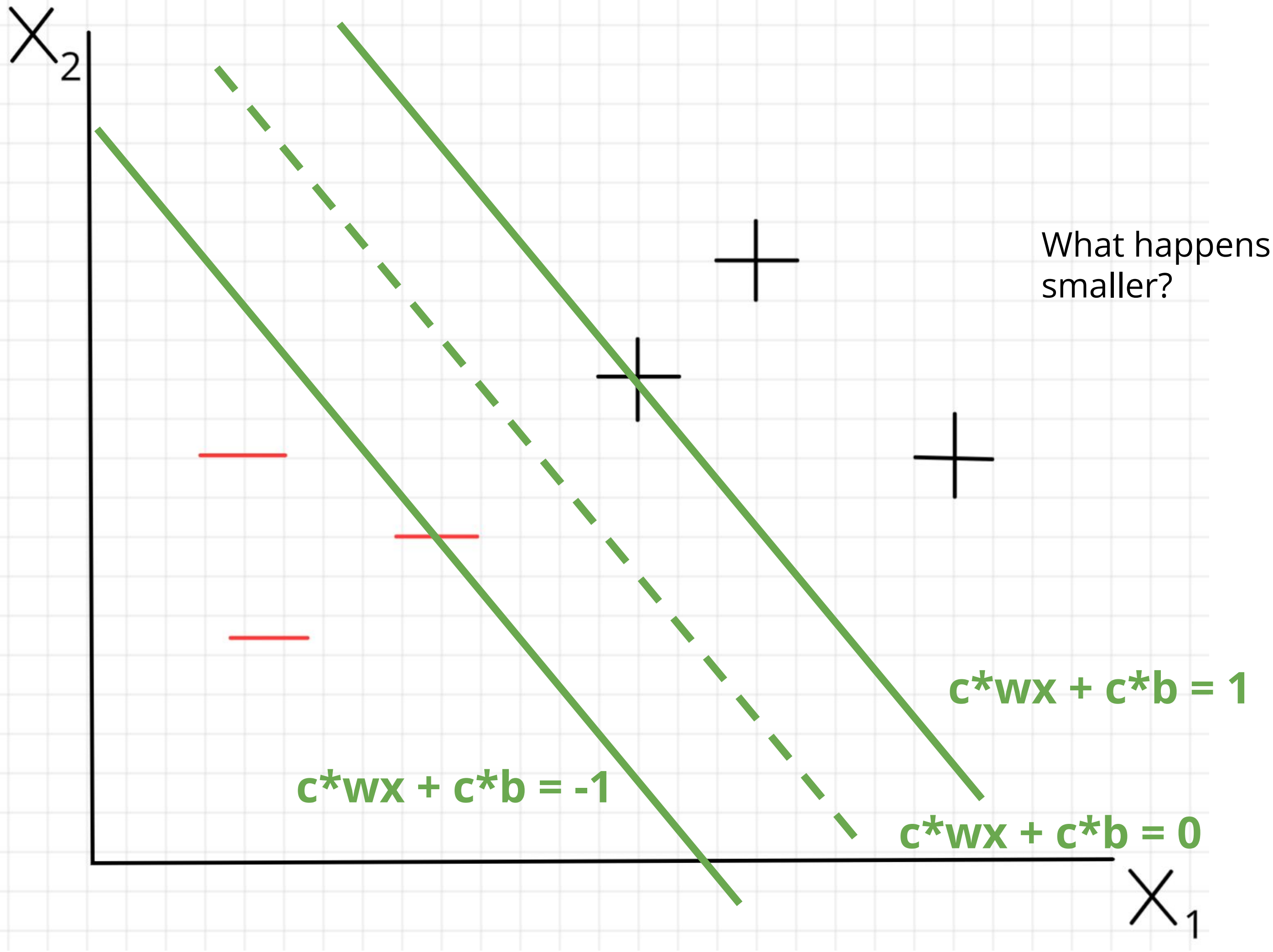
There are many **w**’s



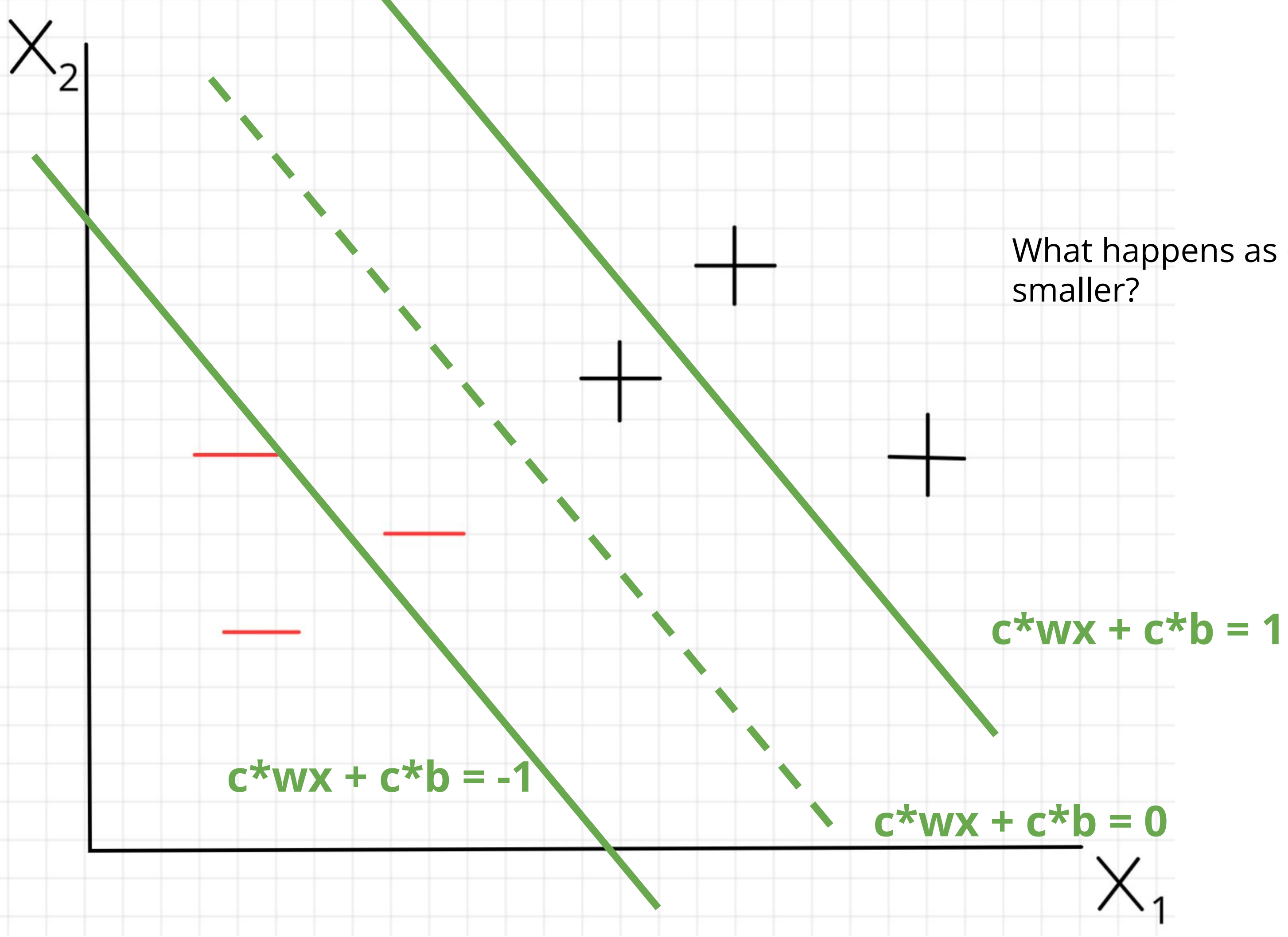
What happens as **c** gets larger?



What happens as **c** gets larger?



What happens as **c** gets smaller?



What happens as **c** gets smaller?

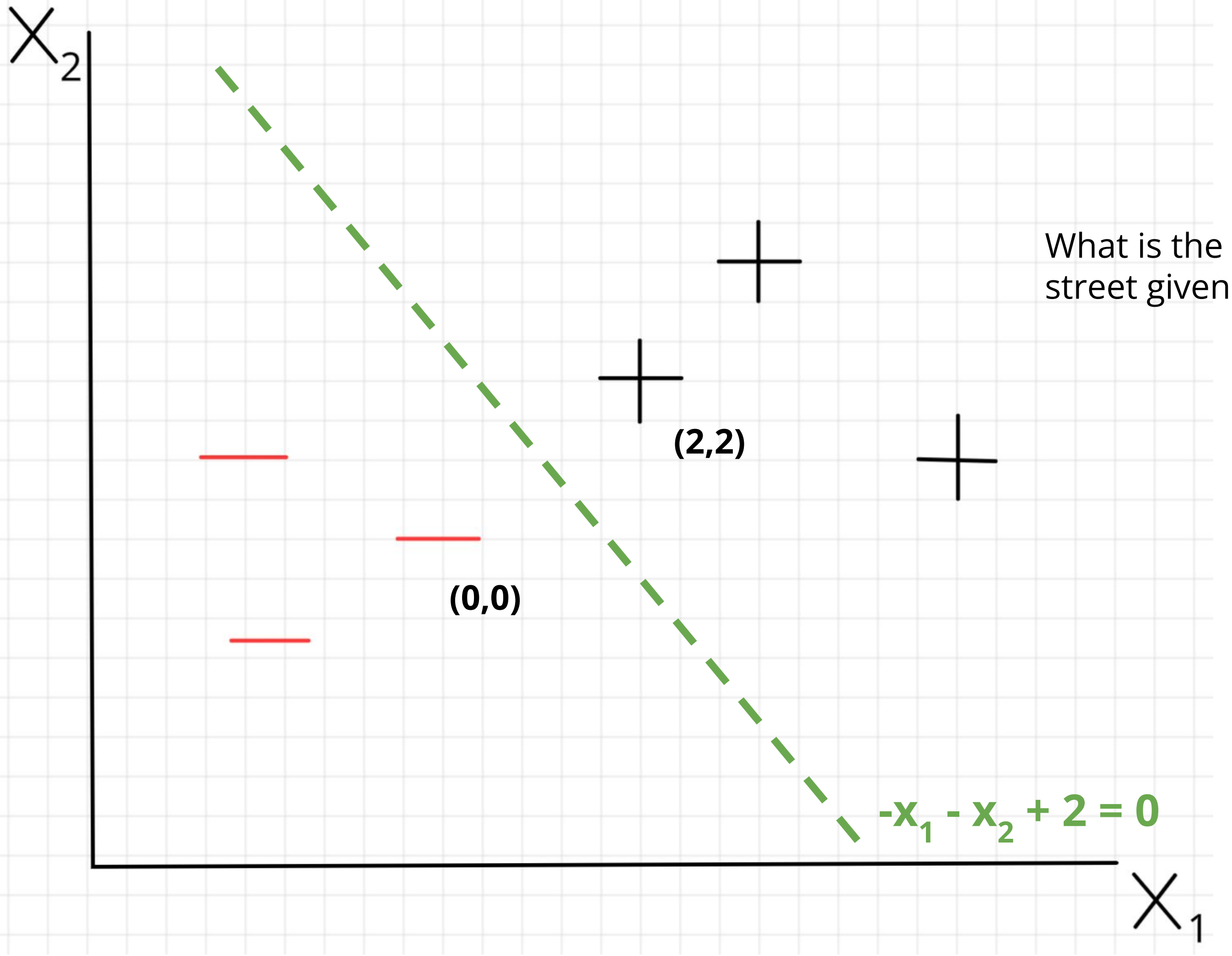
How to find the widest street

Goal is to maximize the width

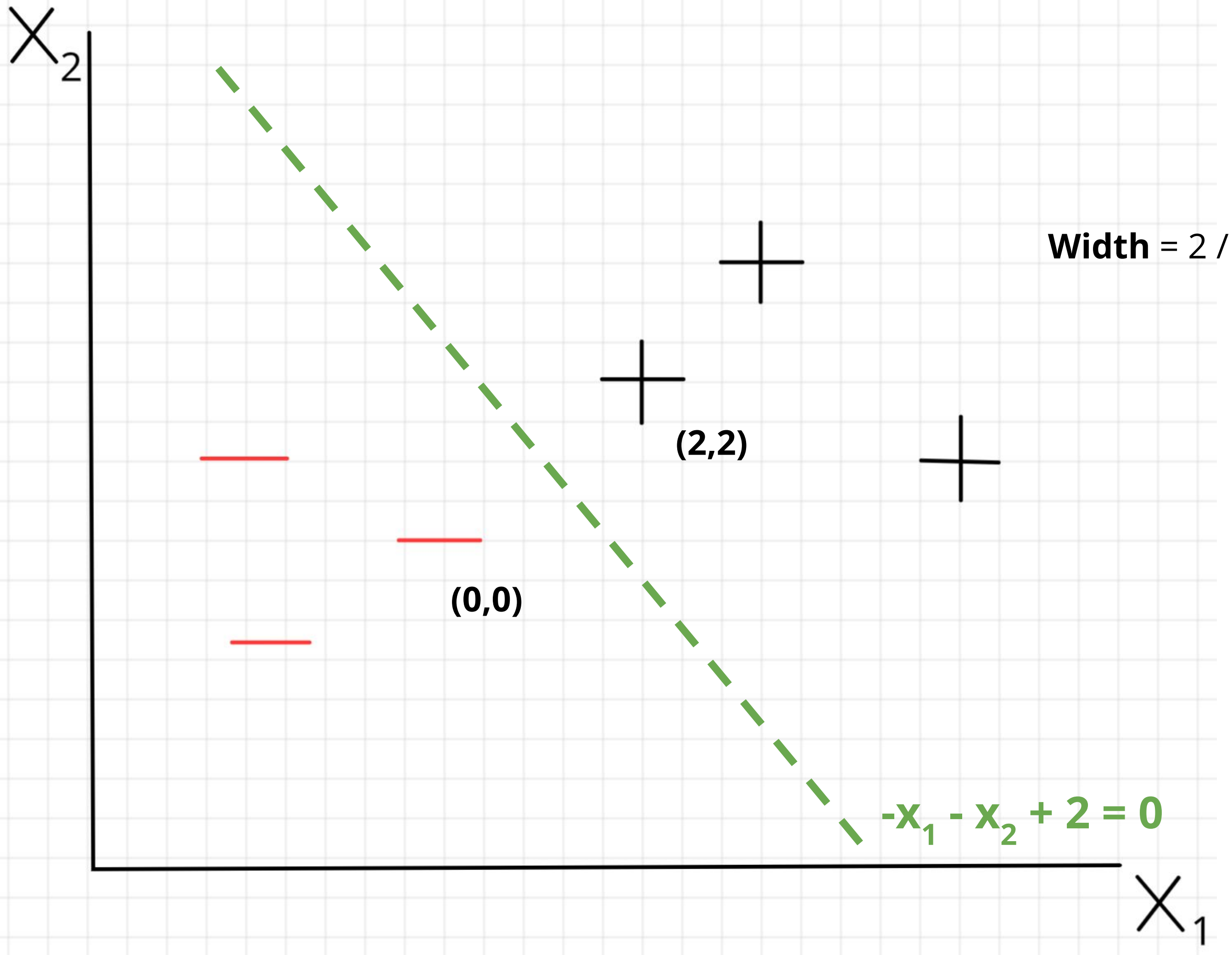
$$\max\left(\frac{2}{||\vec{w}||}\right)$$

Subject to:

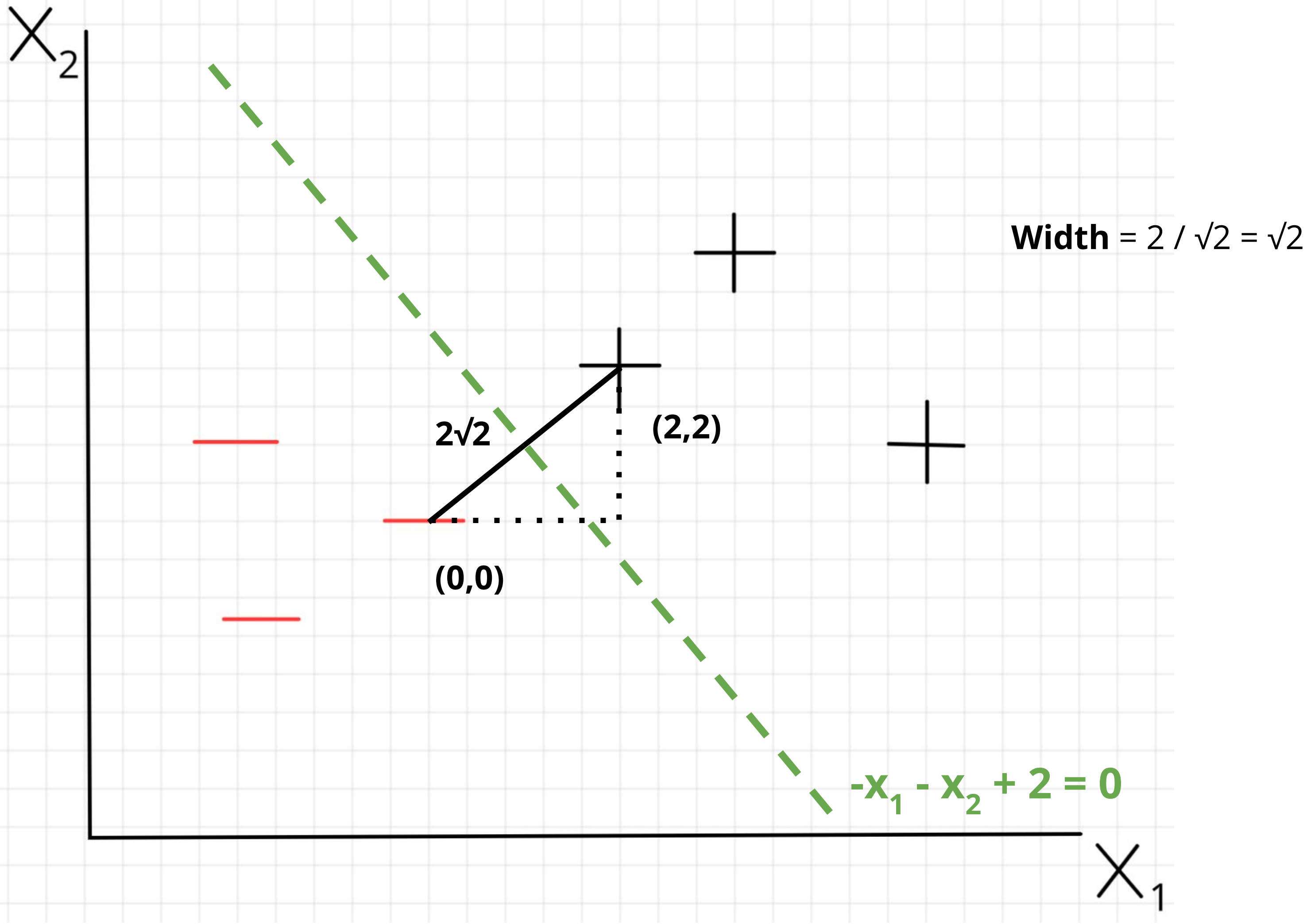
$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 = 0$$



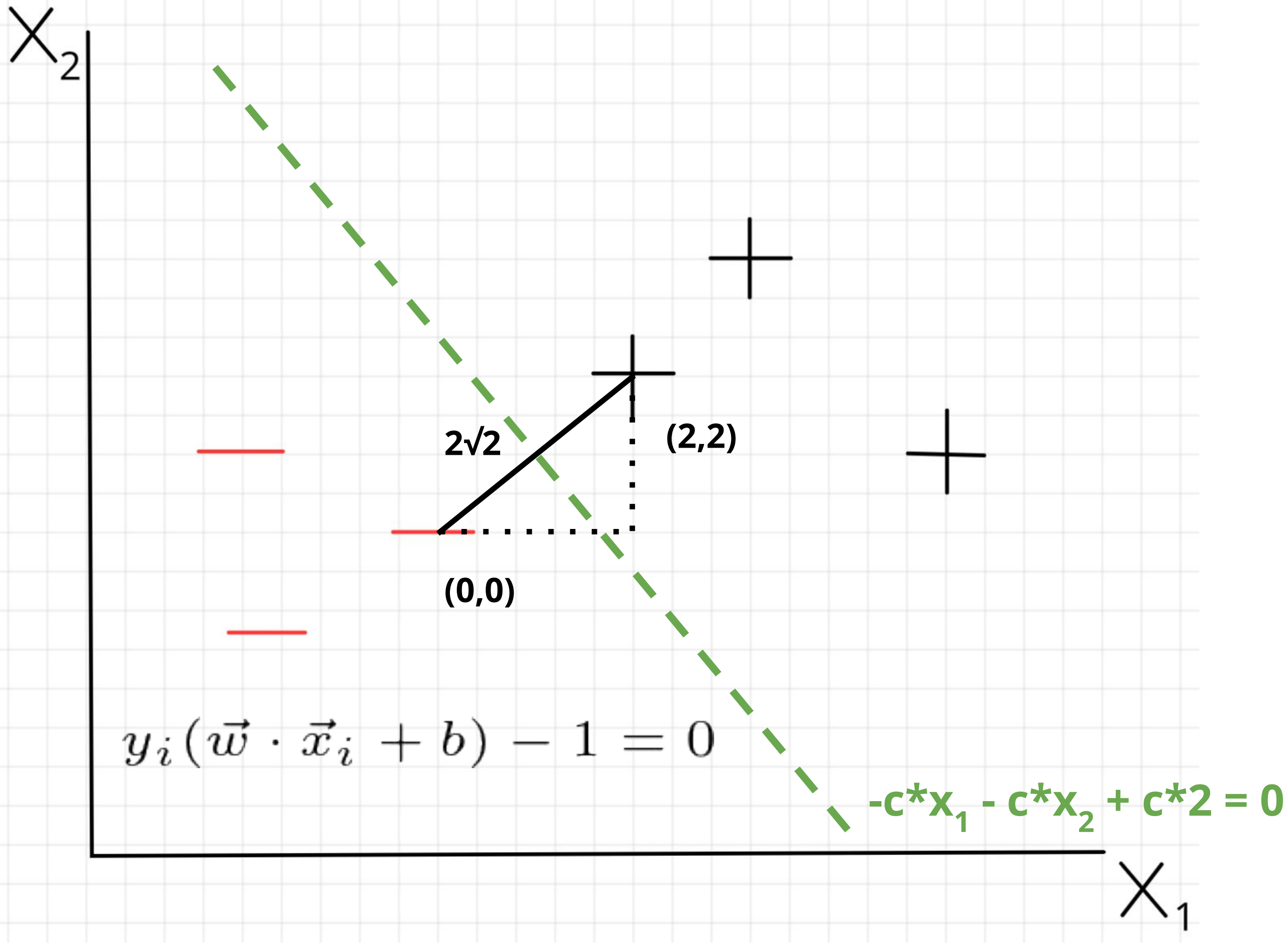
What is the width of the street given by **w**?



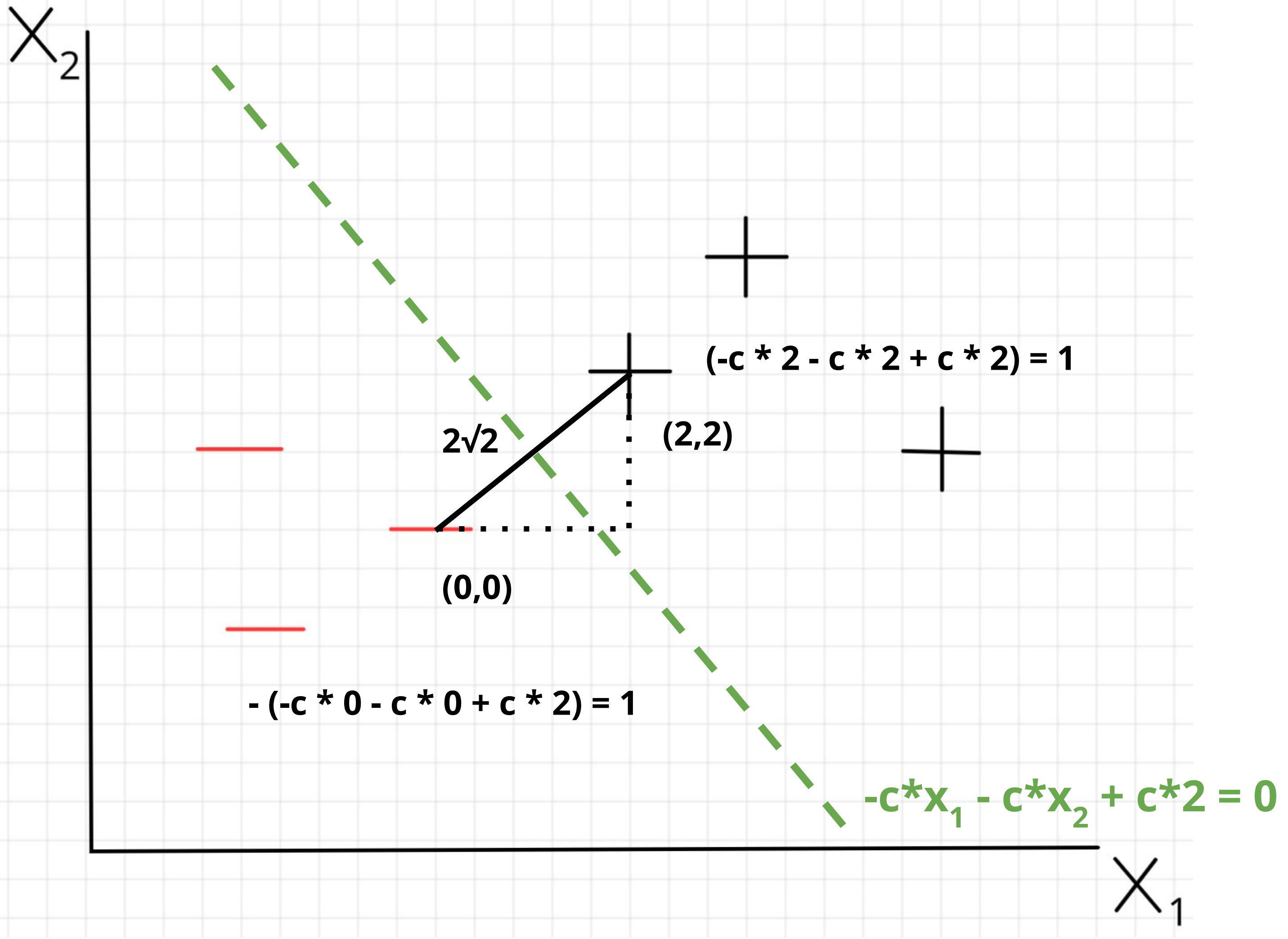
Width = $2 / \sqrt{2} = \sqrt{2}$



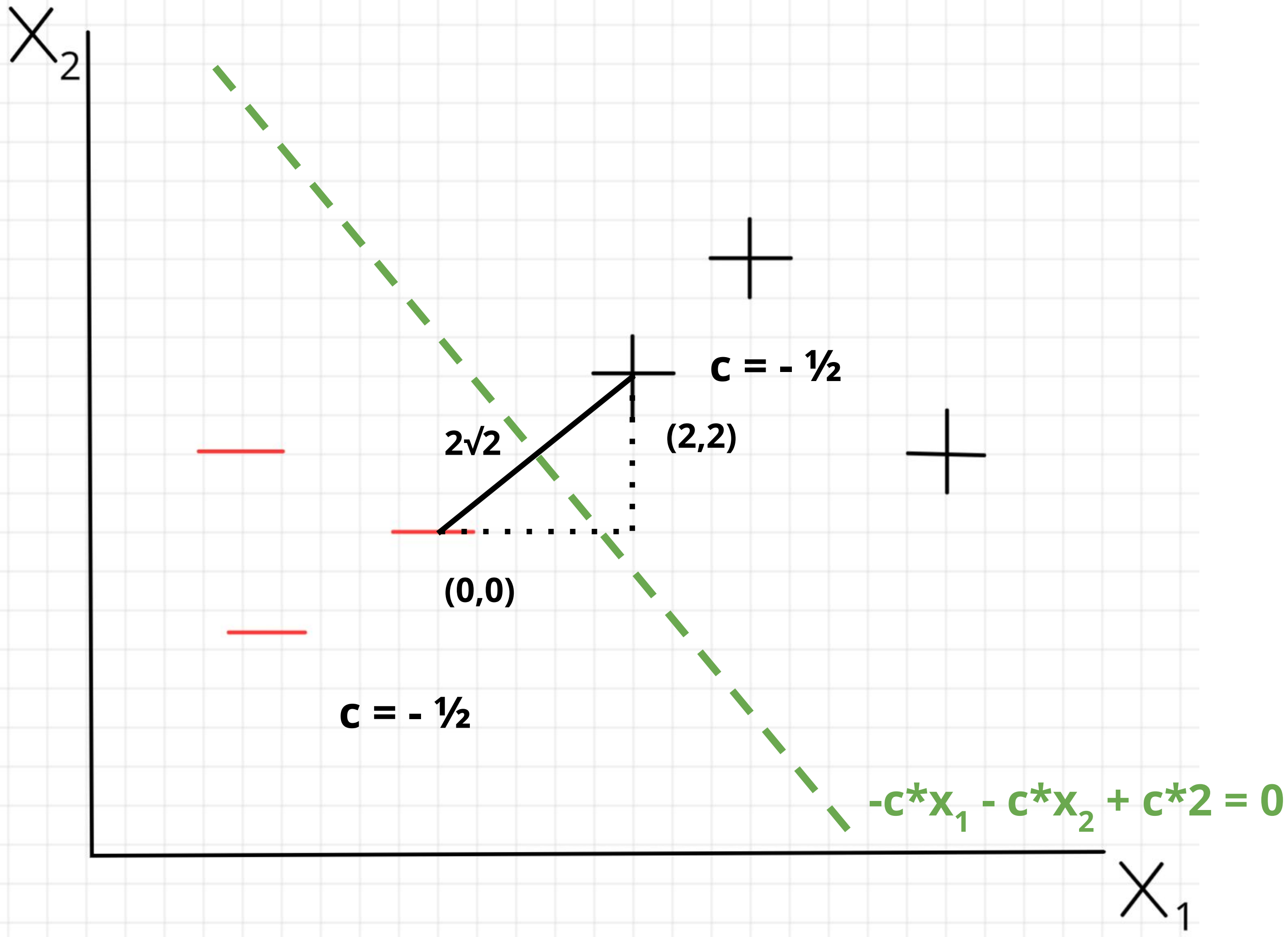
X_2
Width = 2 / √2 = √2
2√2
(2,2)
(0,0)
$-x_1 - x_2 + 2 = 0$
X_1

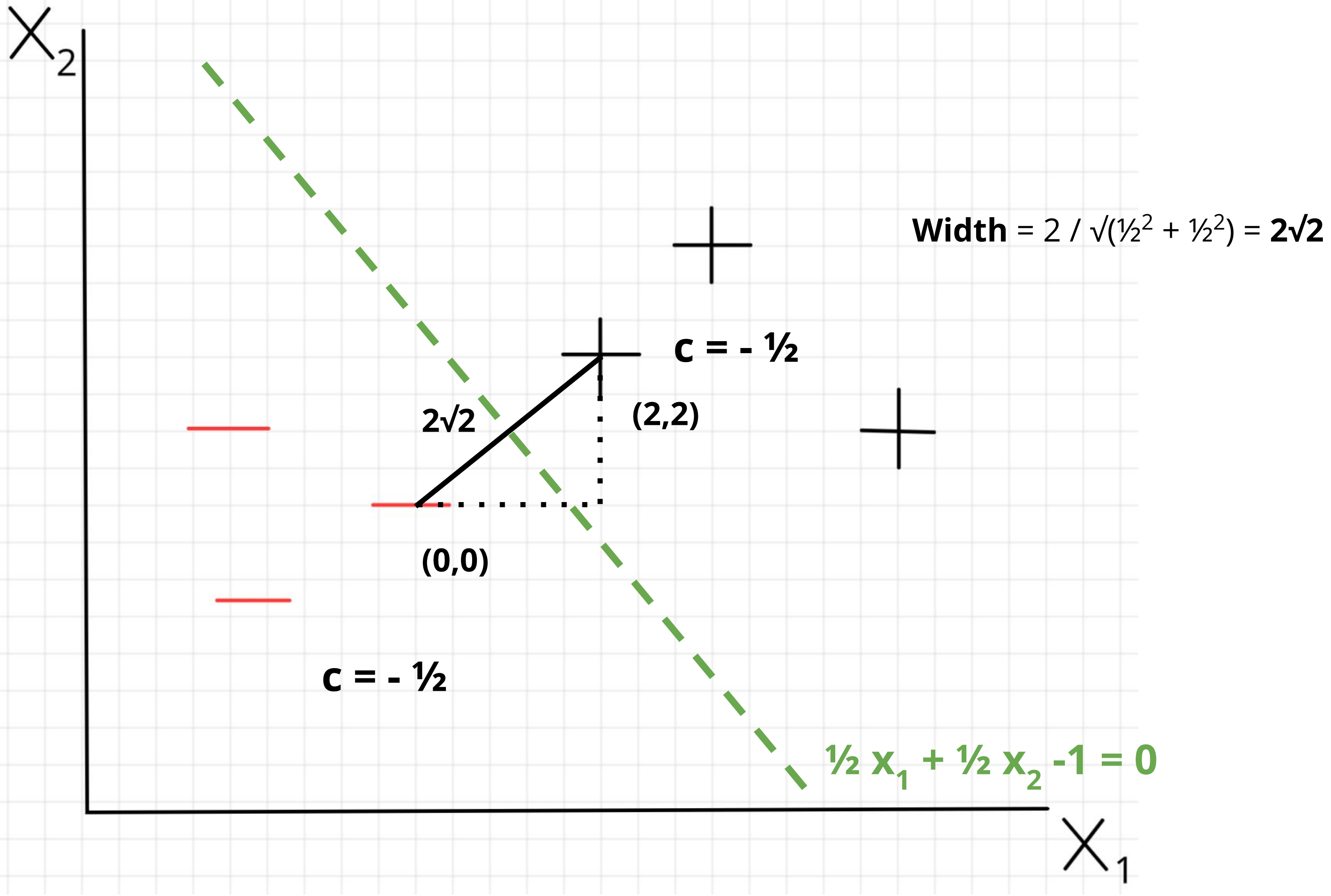


X_2
$2\sqrt{2}$
(2,2)
(0,0)
$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 = 0$
$-c*x_1 - c*x_2 + c*2 = 0$
X_1



X_2
$(-c * 2 - c * 2 + c * 2) = 1$
$2\sqrt{2}$
$(2,2)$
$(0,0)$
$-(-c * 0 - c * 0 + c * 2) = 1$
$-c*x_1 - c*x_2 + c*2 = 0$
X_1


X_2
c = - ½
2√2
(2,2)
(0,0)
c = - ½
$-c*x_1 - c*x_2 + c*2 = 0$
X_1



X_2
X_1
Width = 2 / √(½² + ½²) = **2√2**
c = - ½
(2,2)
2√2
(0,0)
c = - ½
½ x₁ + ½ x₂ -1 = 0

Worksheet c)

How to find the widest street

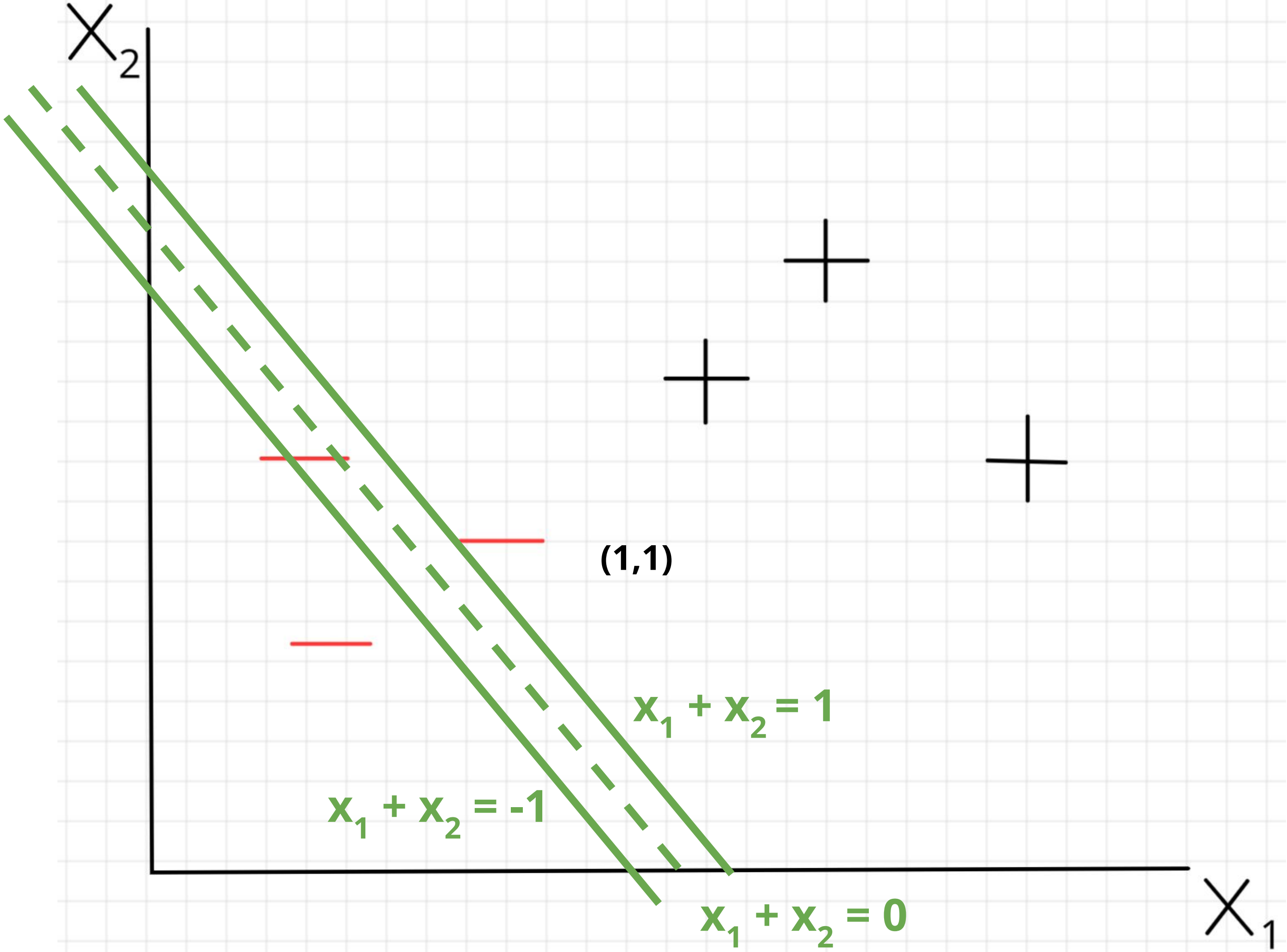
Goal is to maximize the width

$$\max\left(\frac{2}{\|\vec{w}\|}\right) = \min(\|\vec{w}\|)$$

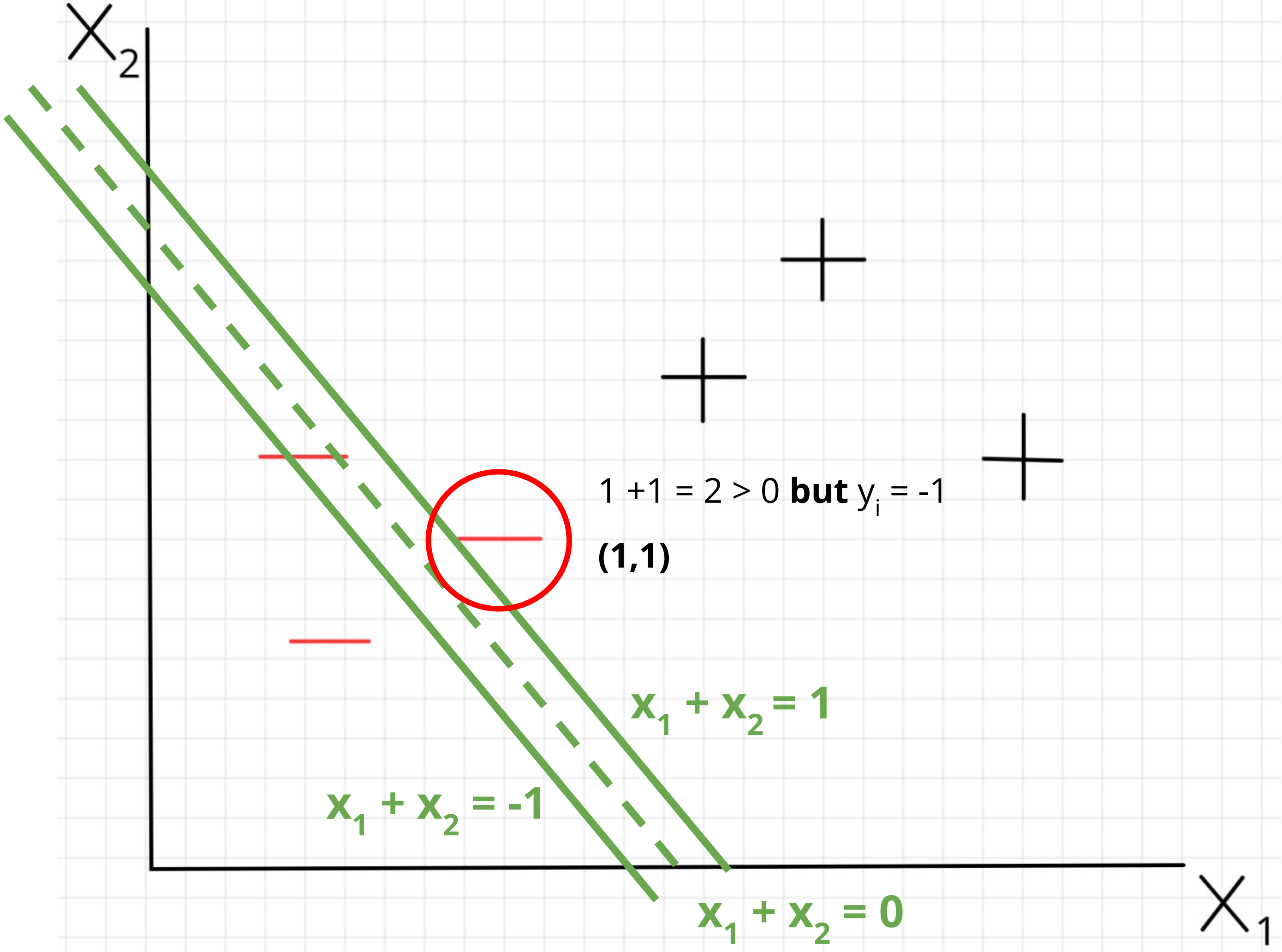
$$= \min\left(\frac{1}{2}\|\vec{w}\|^2\right)$$

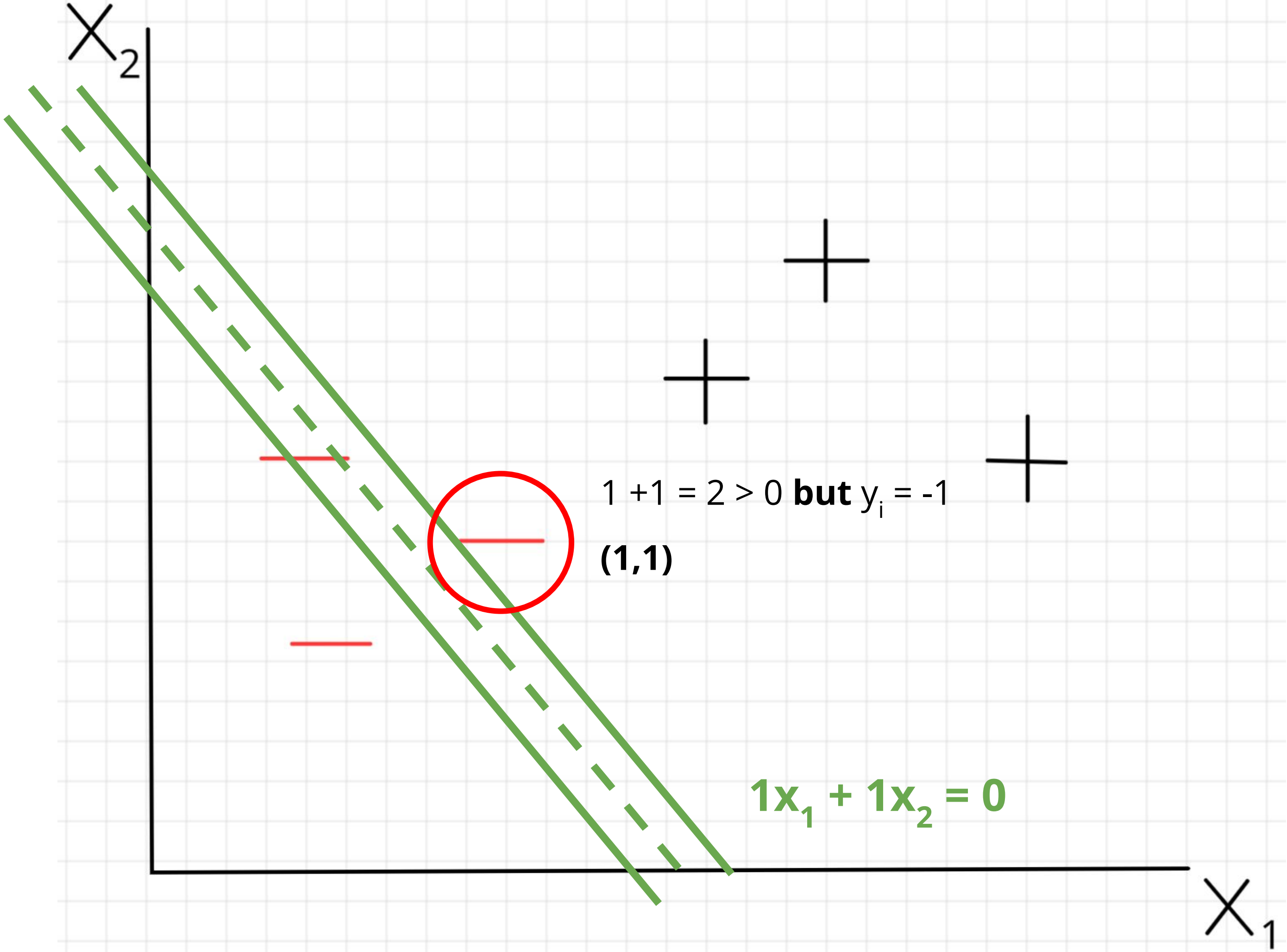
Subject to:

$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 = 0$$

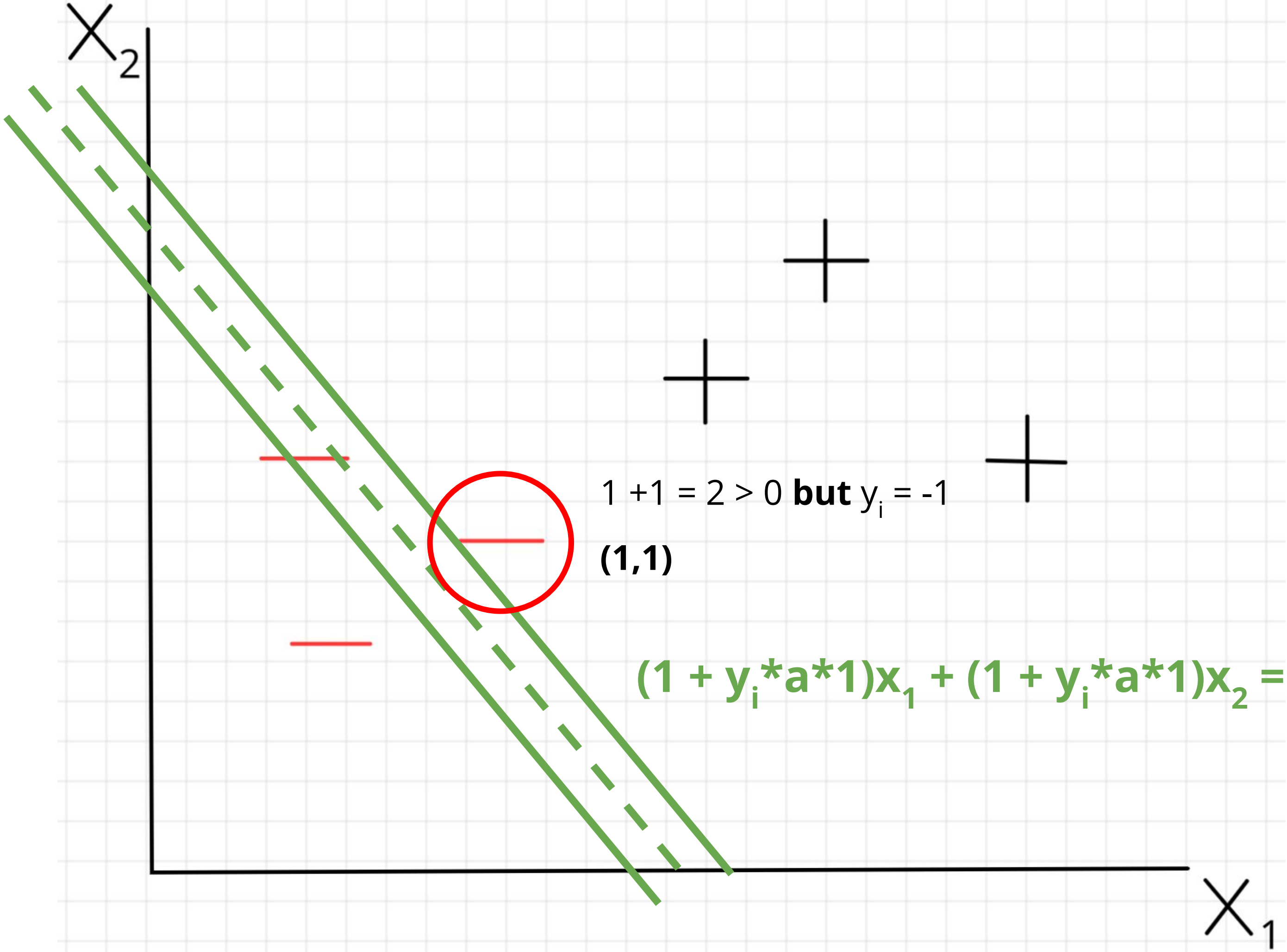


X_2
(1,1)
$x_1 + x_2 = 1$
$x_1 + x_2 = -1$
$x_1 + x_2 = 0$
X_1

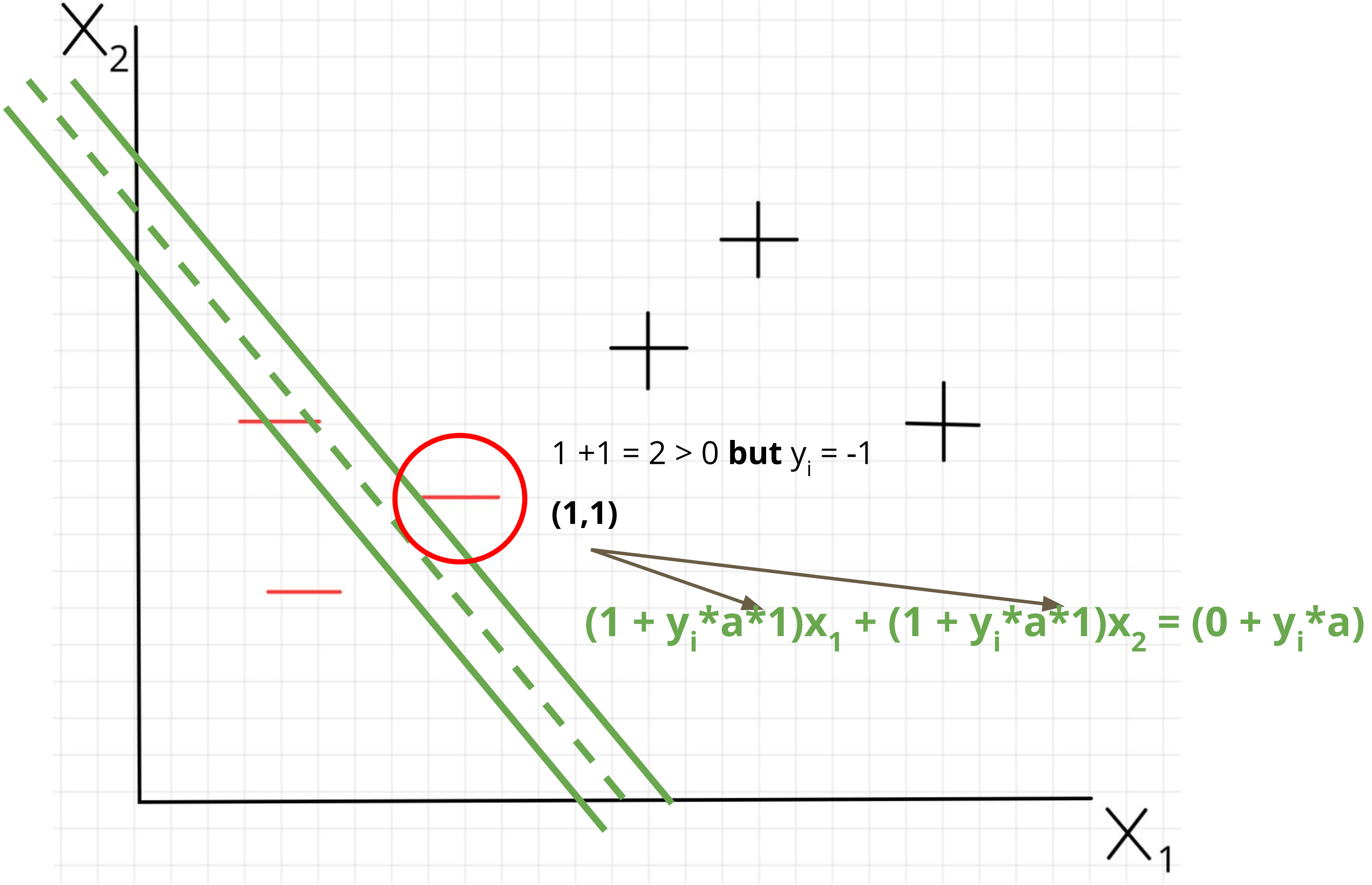

X_2
1 +1 = 2 > 0 **but** y_i = -1
(1,1)
$x_1 + x_2 = 1$
$x_1 + x_2 = -1$
$x_1 + x_2 = 0$
X_1



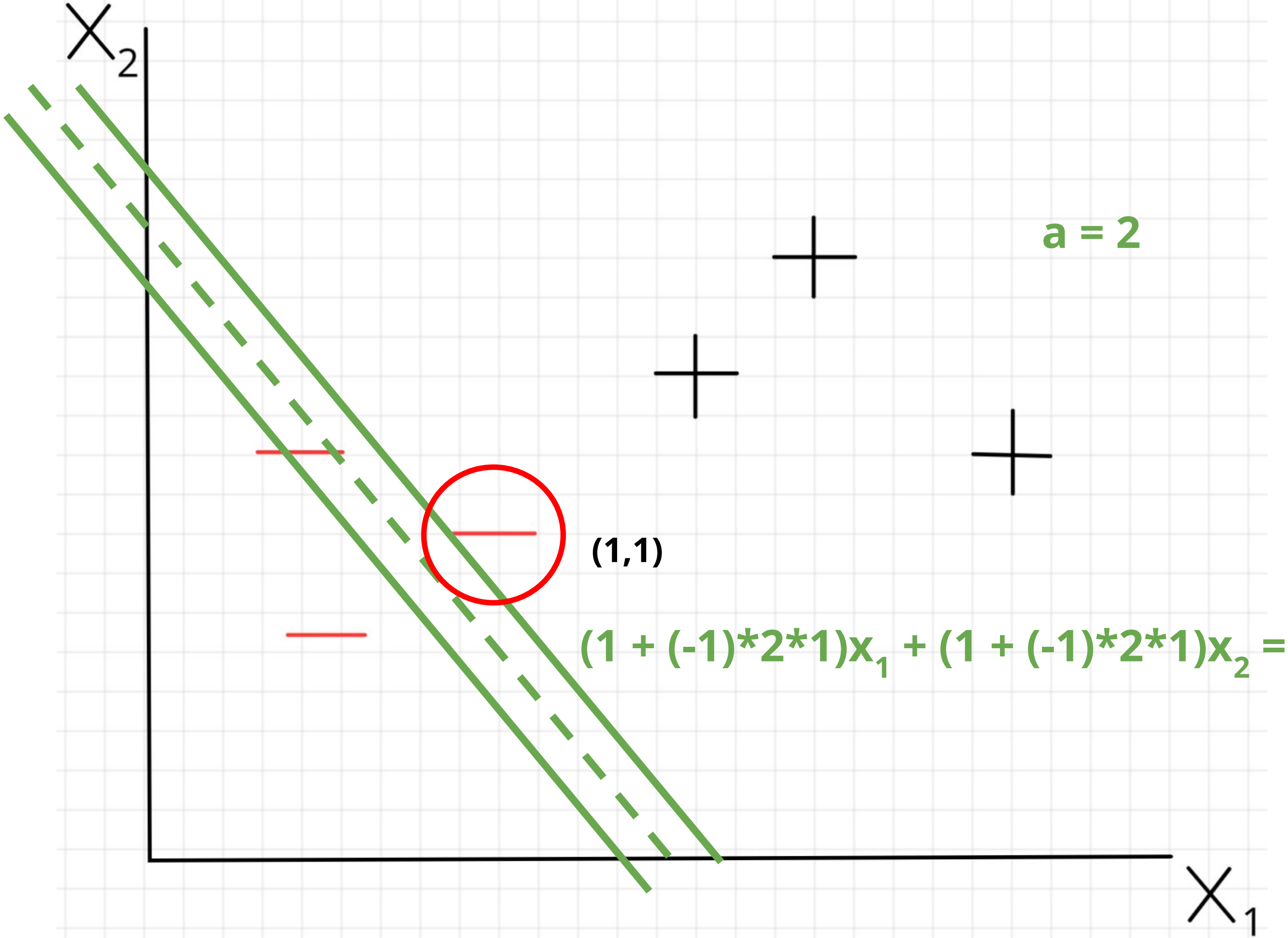
X_2
$1 + 1 = 2 > 0$ **but** $y_i = -1$
(1,1)
$1x_1 + 1x_2 = 0$
X_1



$$(1 + y_i*a*1)x_1 + (1 + y_i*a*1)x_2 = (0 + y_i*a)$$

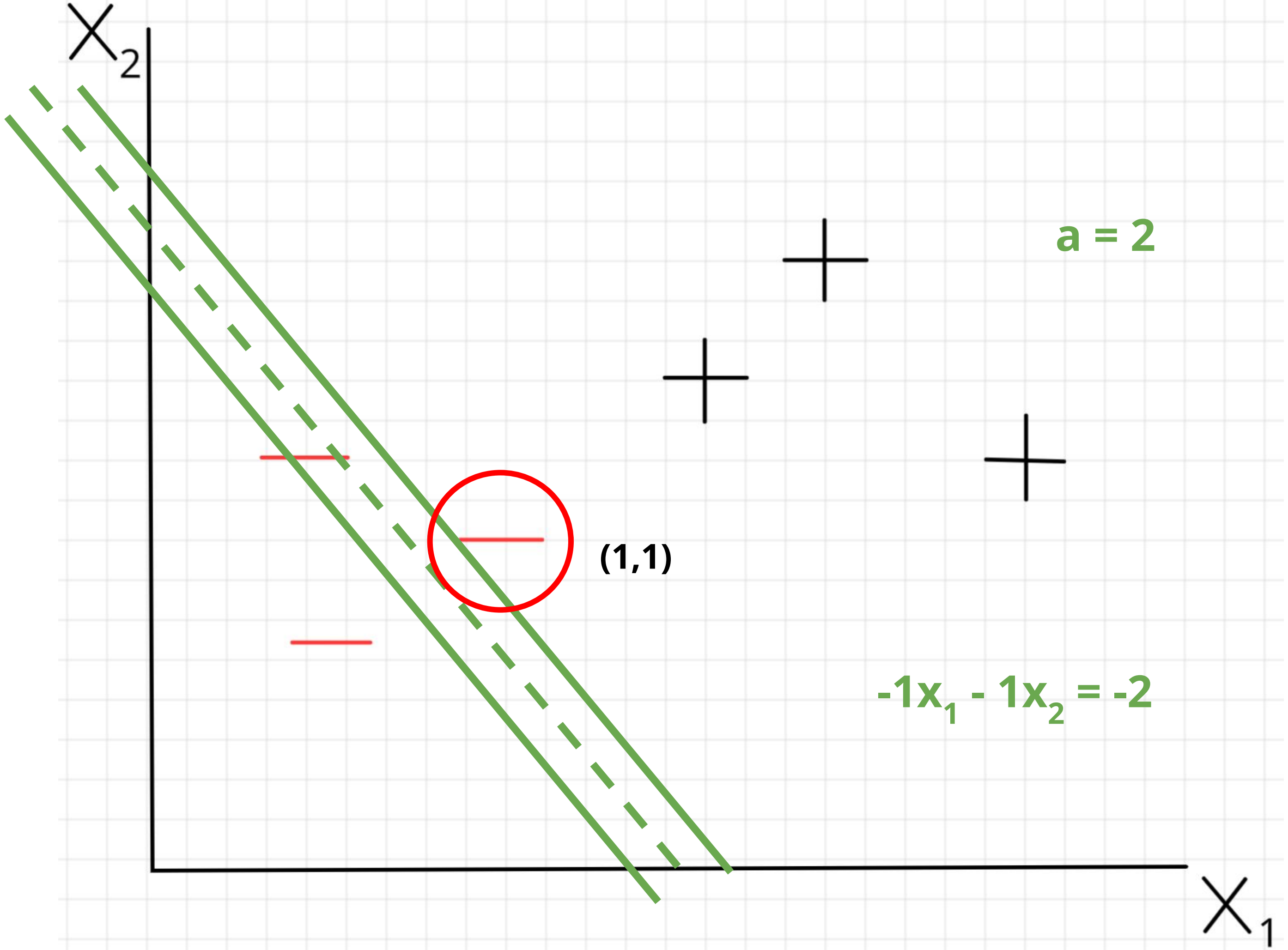


X_2
1 +1 = 2 > 0 **but** y_i = -1
(1,1)
$(1 + y_i*a*1)x_1 + (1 + y_i*a*1)x_2 = (0 + y_i*a)$
X_1

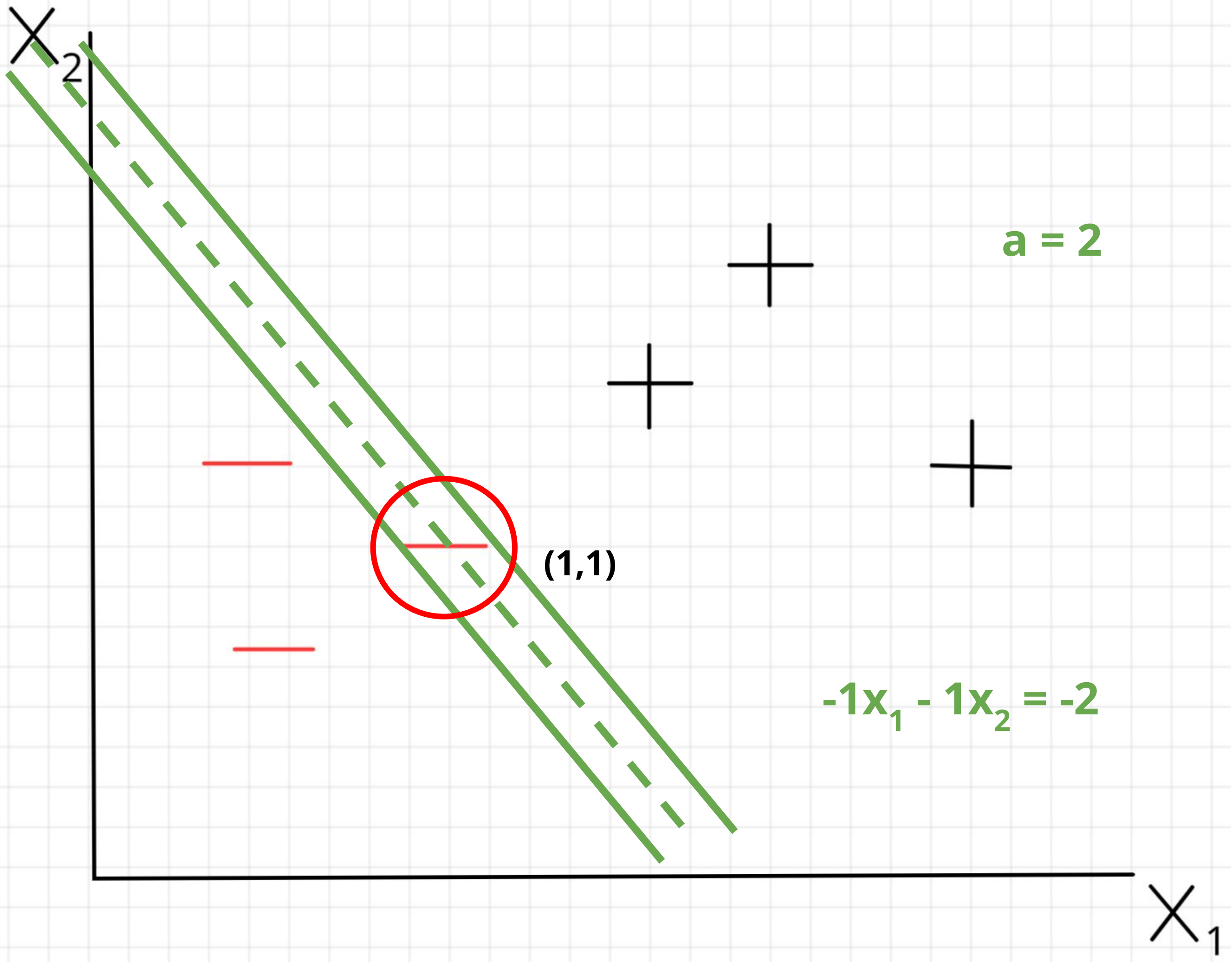


$a = 2$

$$(1 + (-1)*2*1)x_1 + (1 + (-1)*2*1)x_2 = (0 + (-1)*2)$$

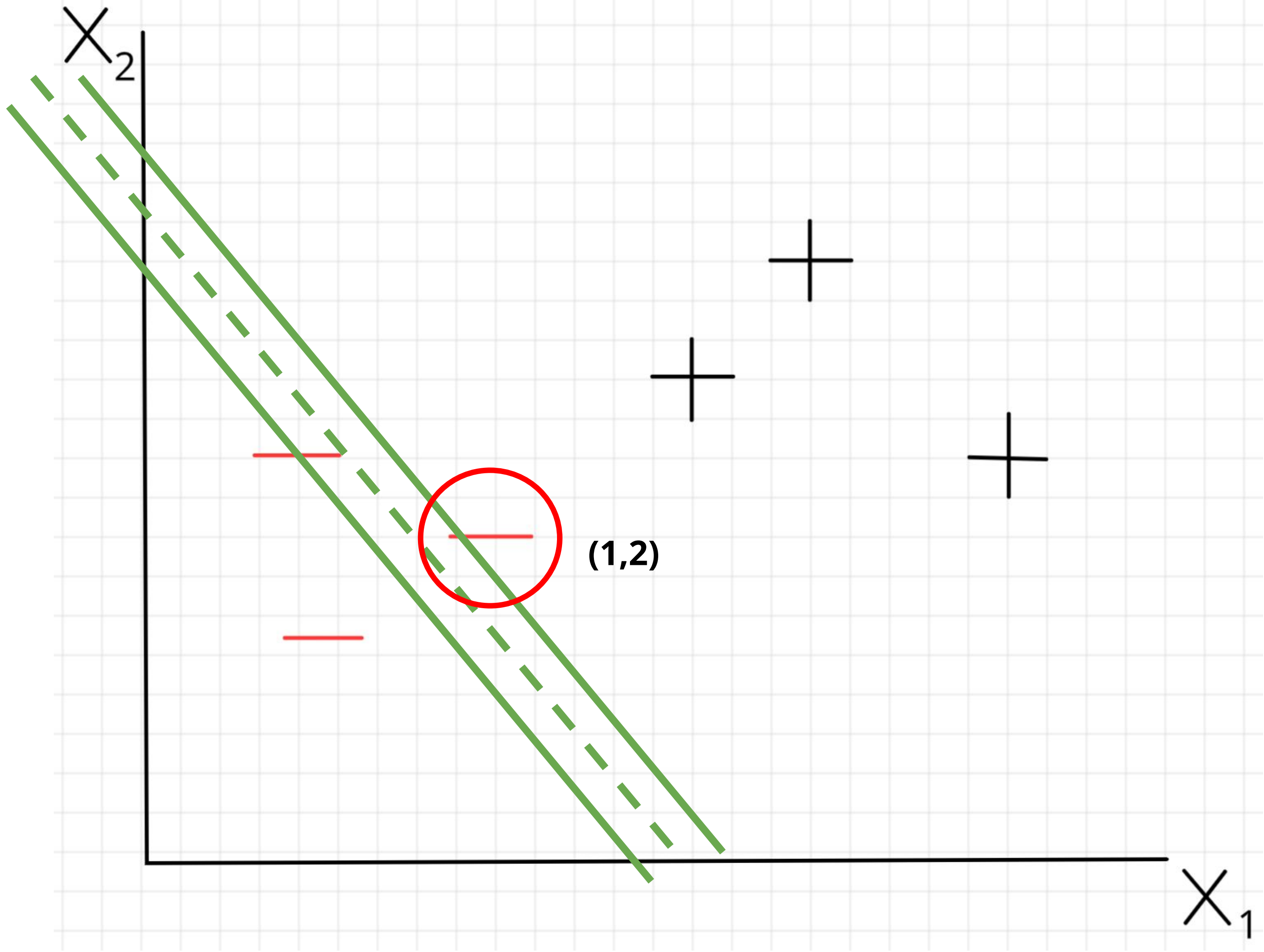


X_2
a = 2
(1,1)
$-1x_1 - 1x_2 = -2$
X_1

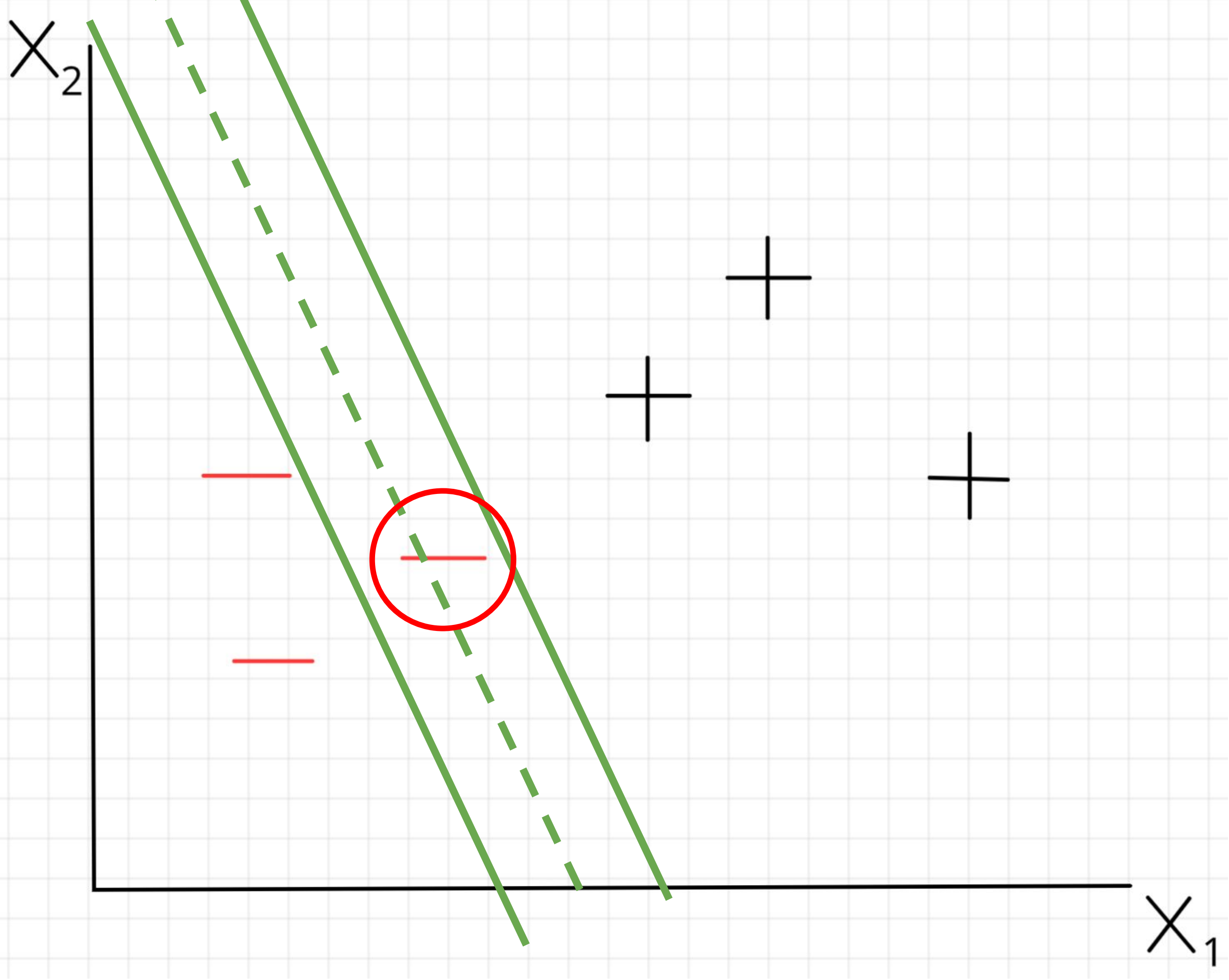


X_2
a = 2
(1,1)
$-1x_1 - 1x_2 = -2$
X_1

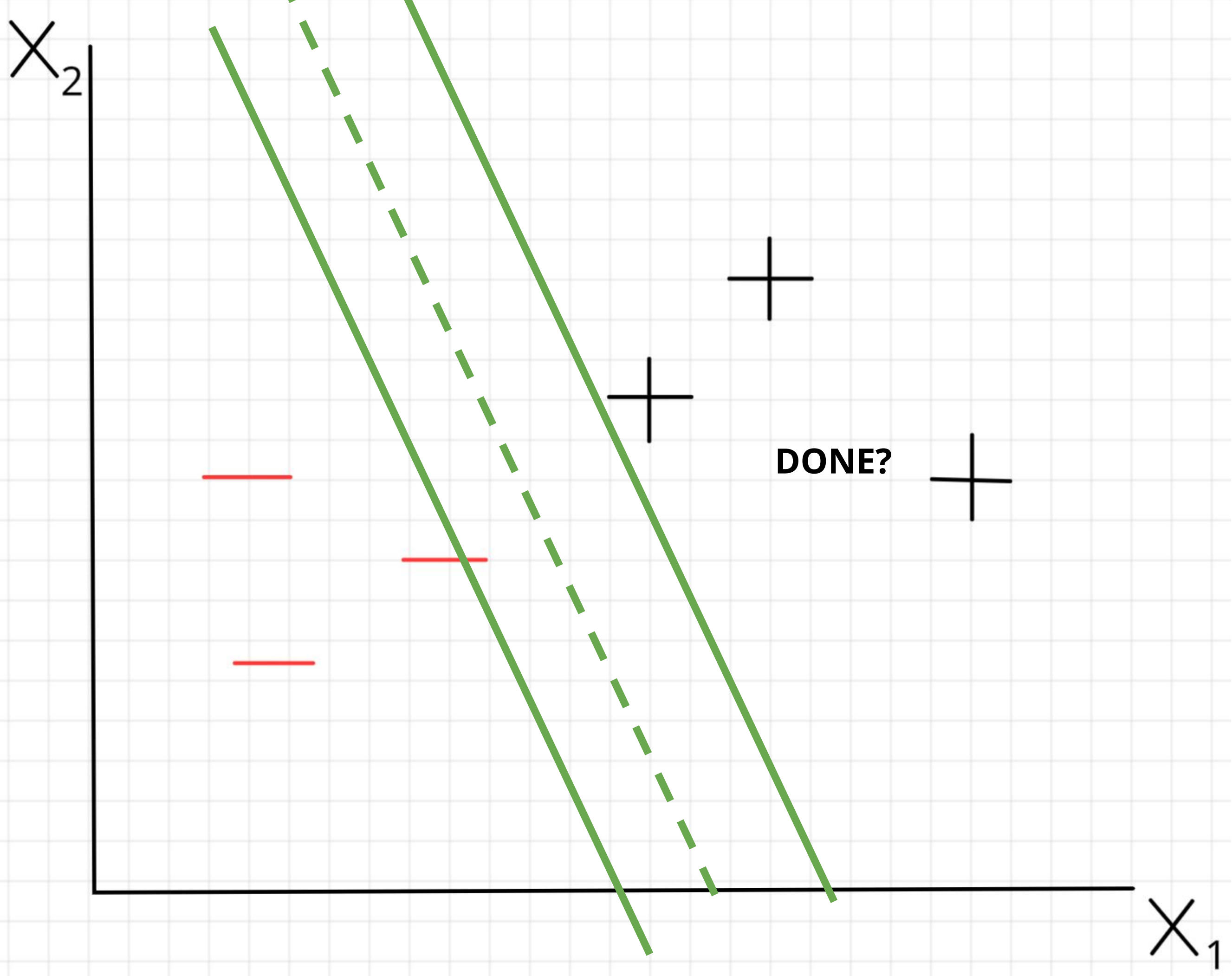
Now we know how to move the street in the right direction - but...



X_2
(1,2)
X_1

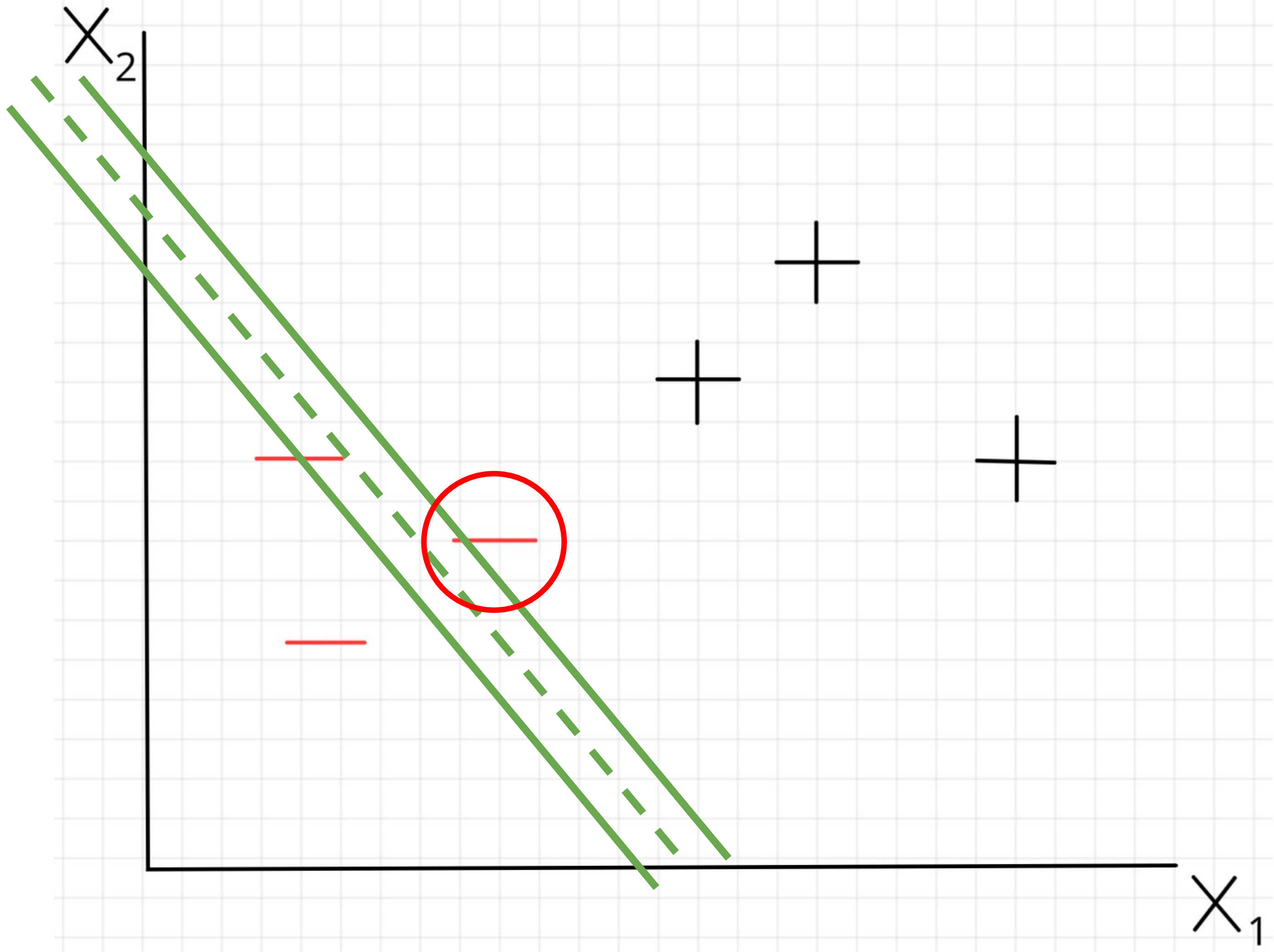


X_2
X_1

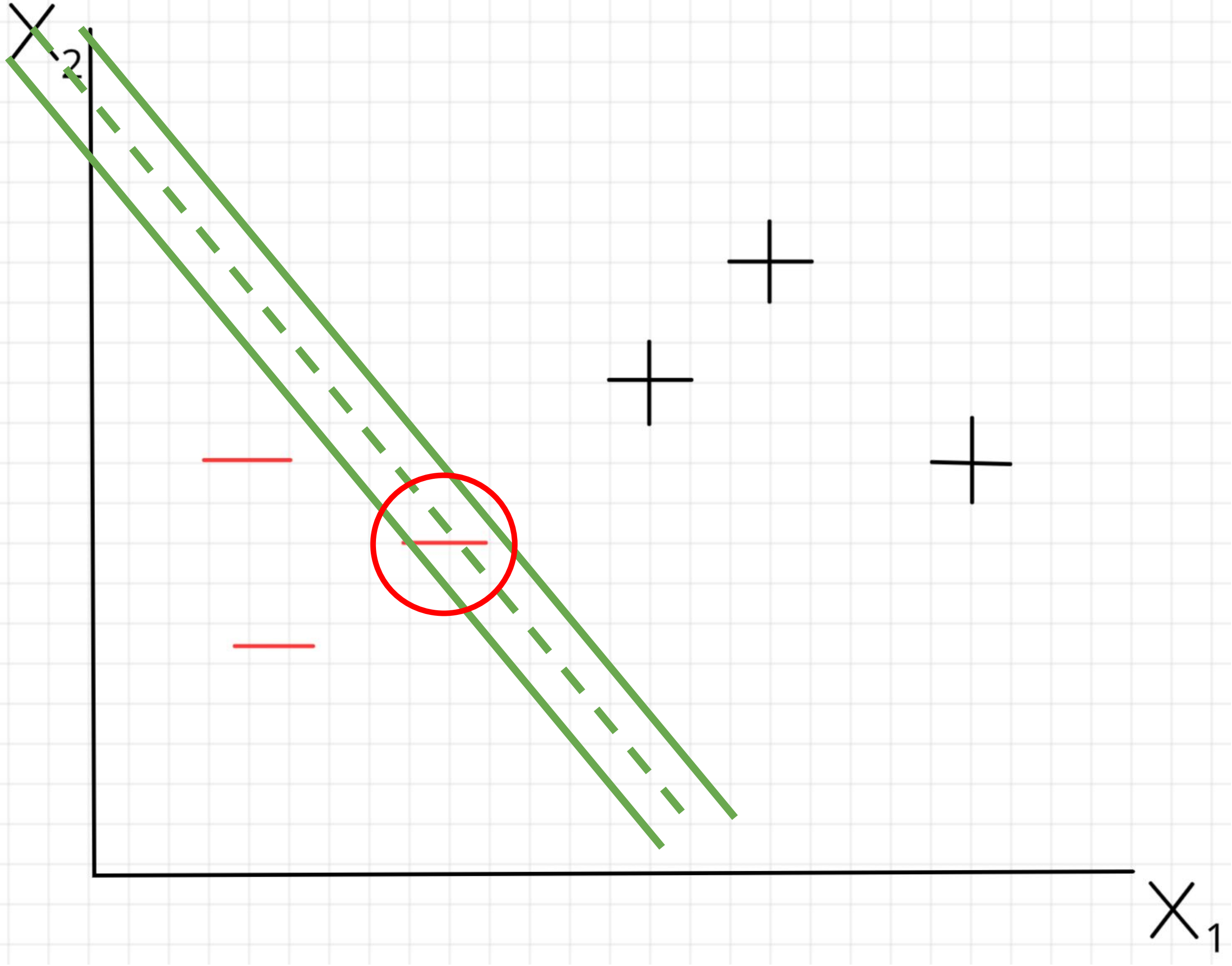

X2
DONE?
X1

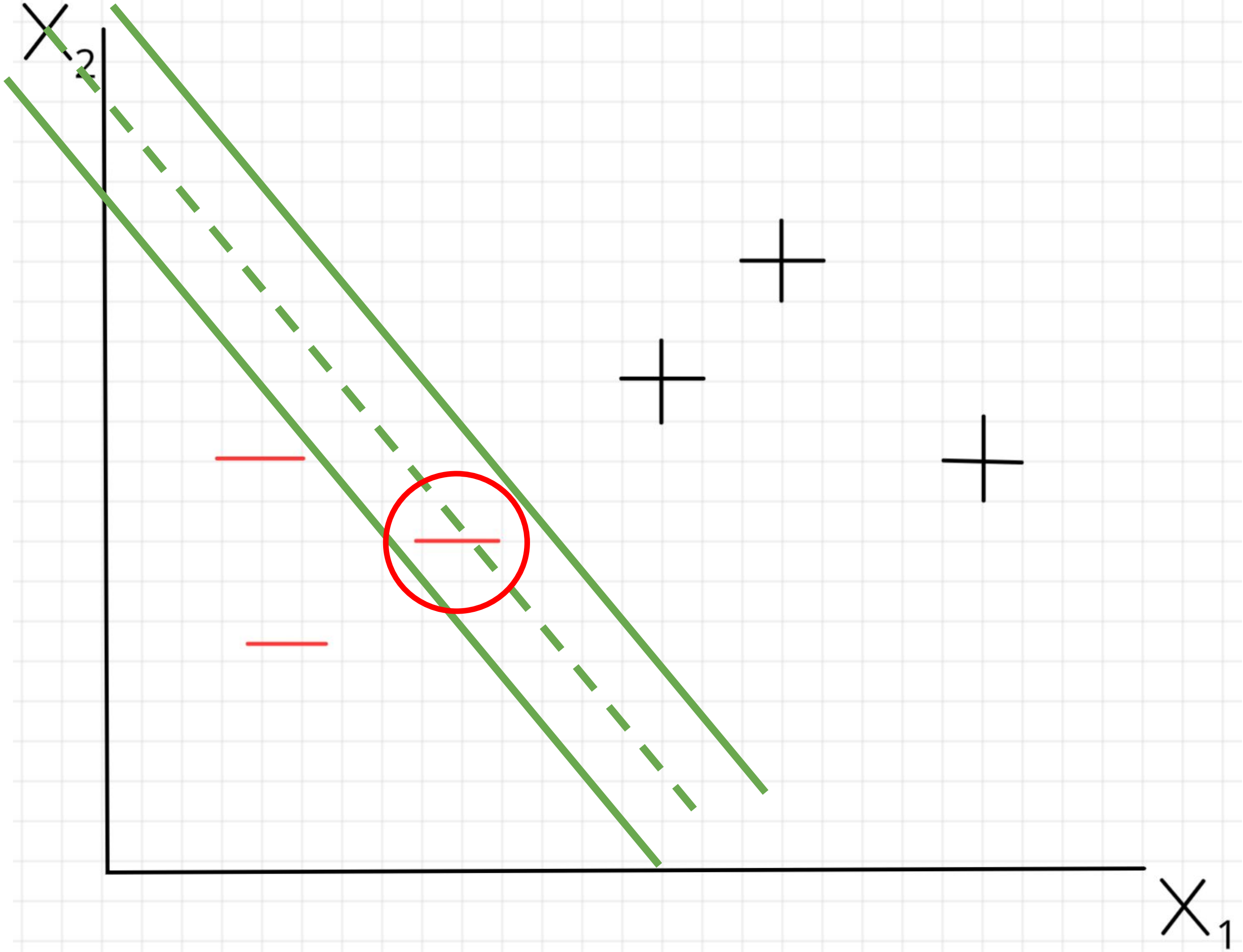
Full Algorithm (Perceptron Algorithm)

- Start with random line $w_1x_1 + w_2x_2 + b = 0$
- Define:
    - A total number of iterations (ex: 100)
    - A learning rate **a** (not too big not too small)
    - An expanding rate **c** (< 1 but not too close to 1)
- Repeat **number of iterations** times:
    - Pick a point (x_i, y_i) from the dataset
    - If correctly classified: do nothing
    - If incorrectly classified:
        - Adjust w_1 by adding $(y_i * a * x_1)$, w_2 by adding $(y_i * a * x_2)$, and b by adding $(y_i * a)$
    - Expand the width by **c** (multiply the new line by **c**)

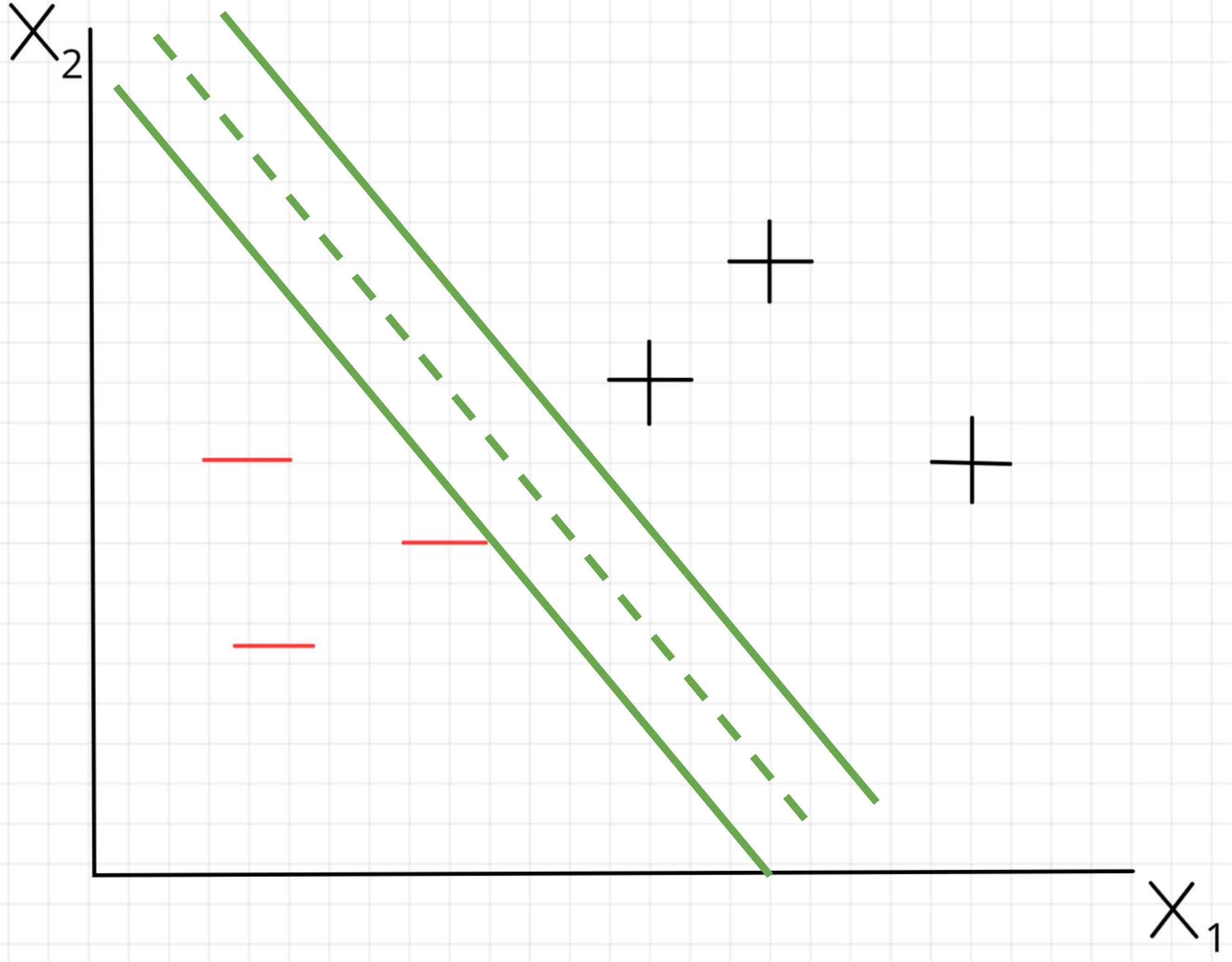


X_2
X_1

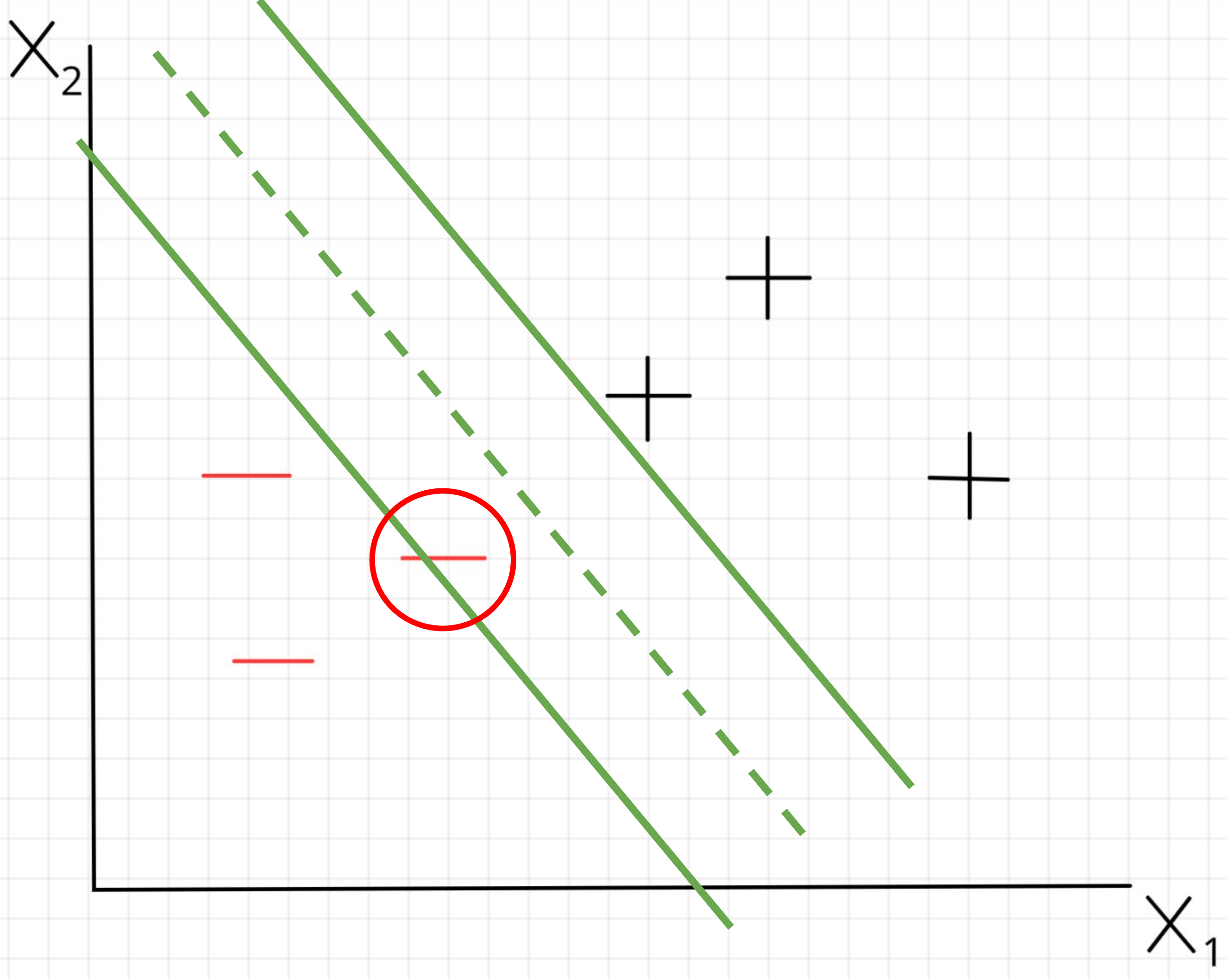

X_2
X_1



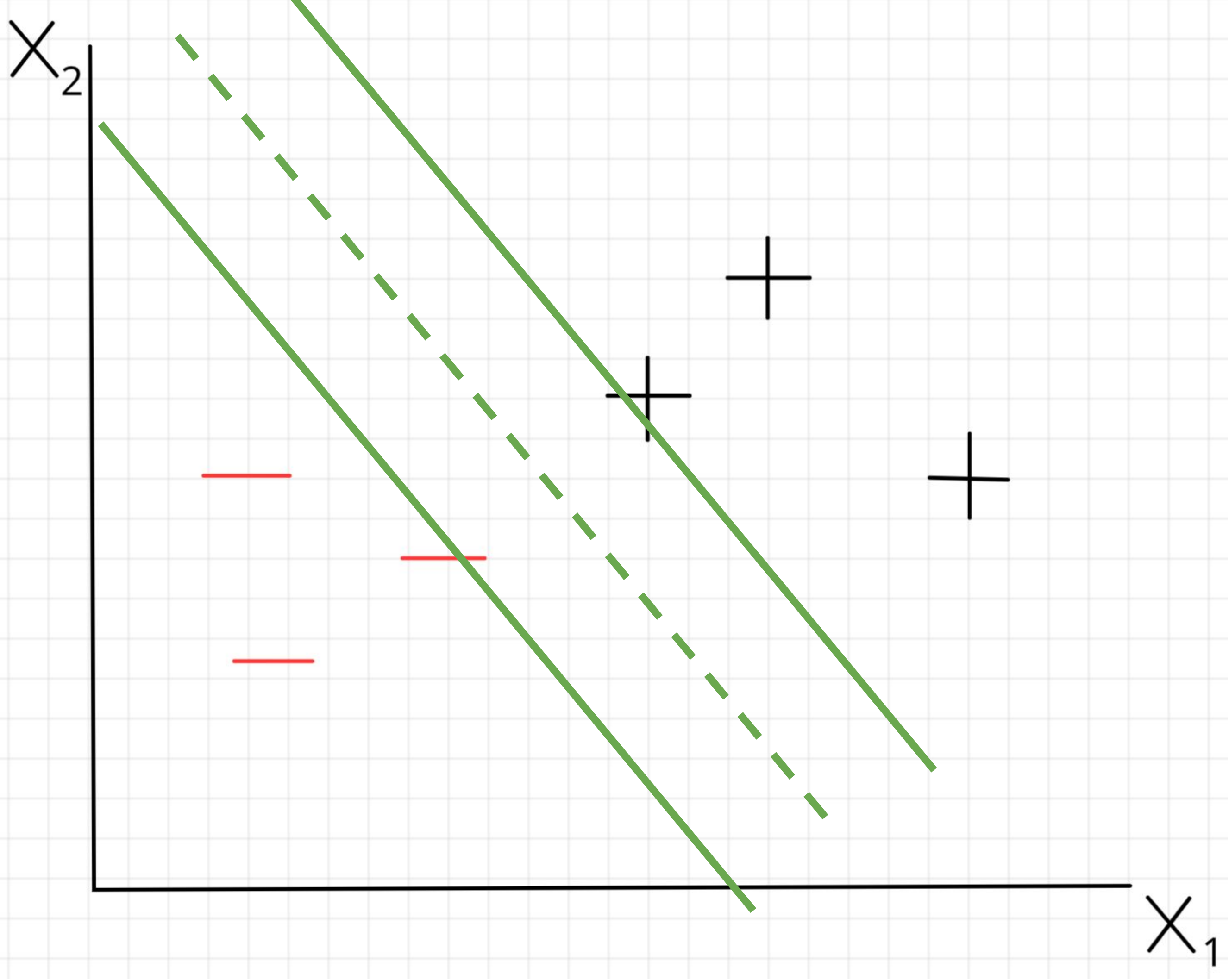
X_2
X_1



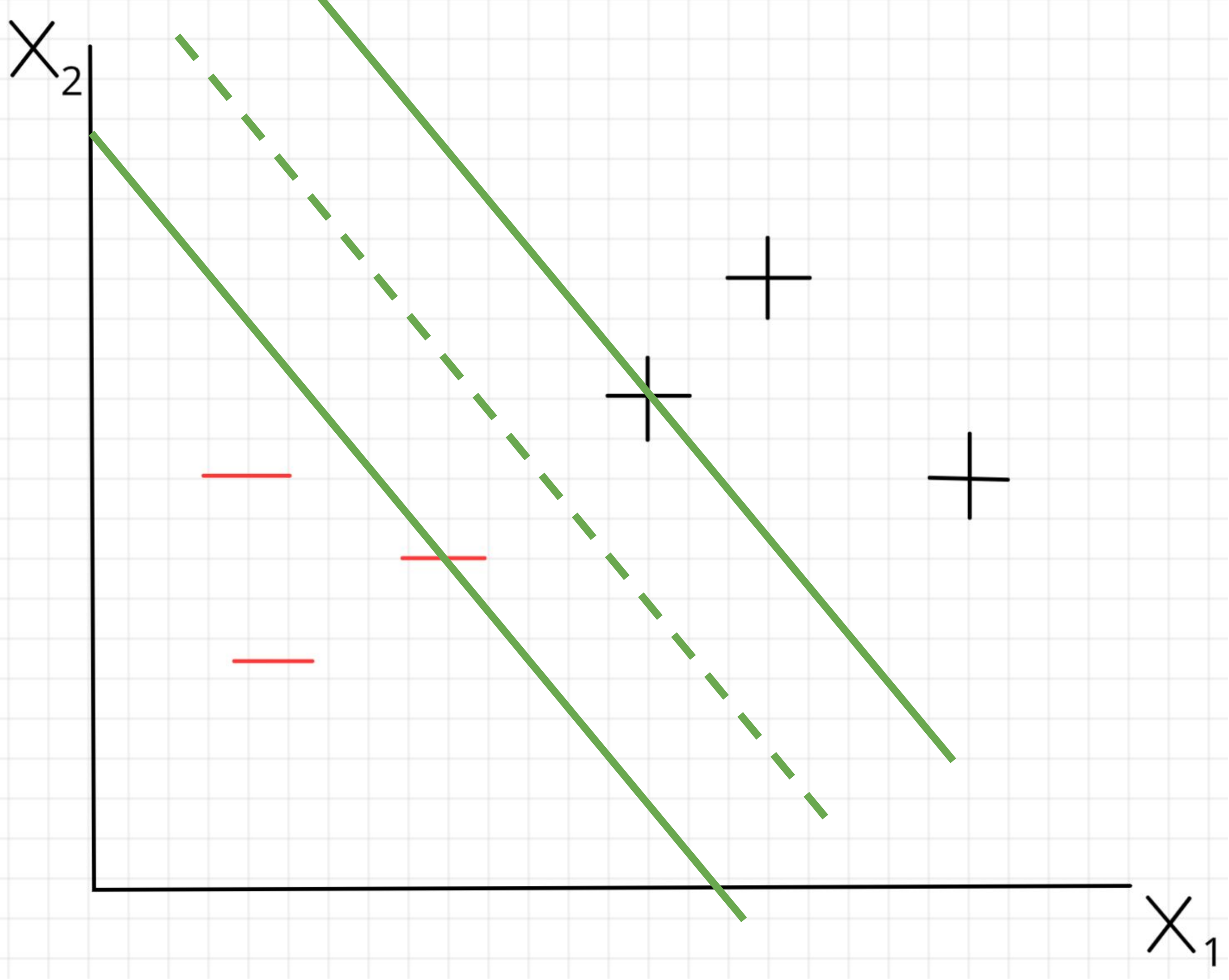
X_2
X_1



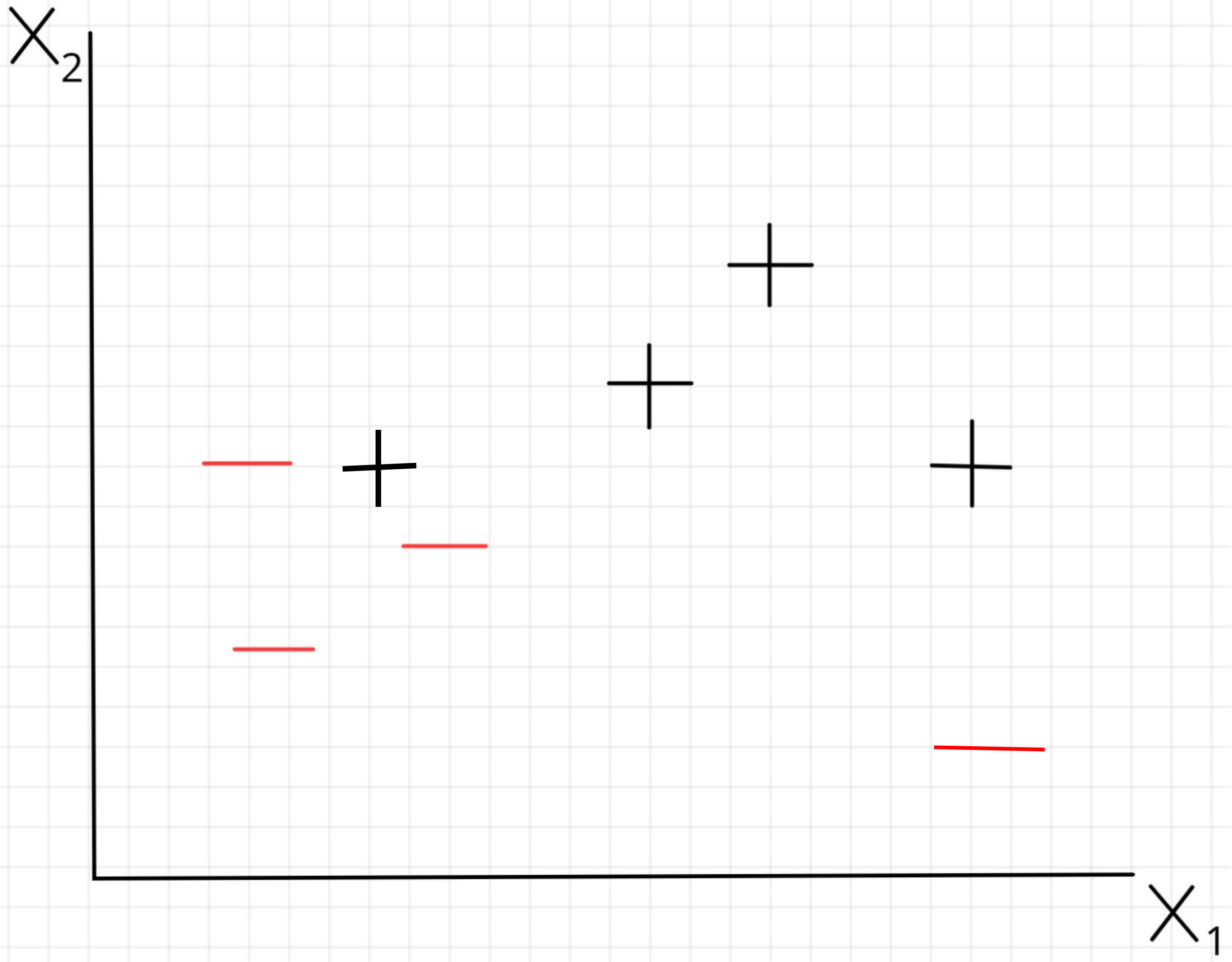
X_2
X_1

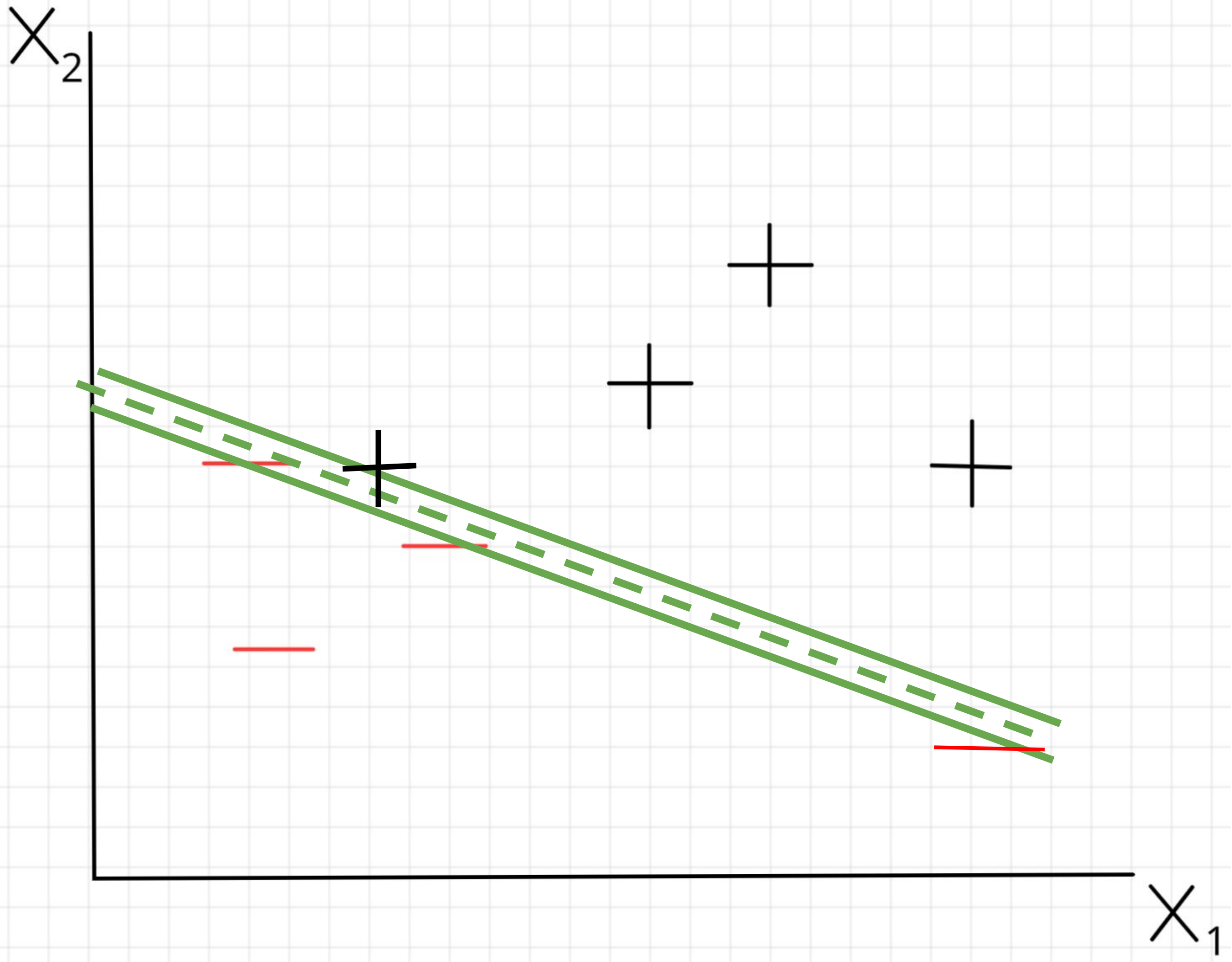


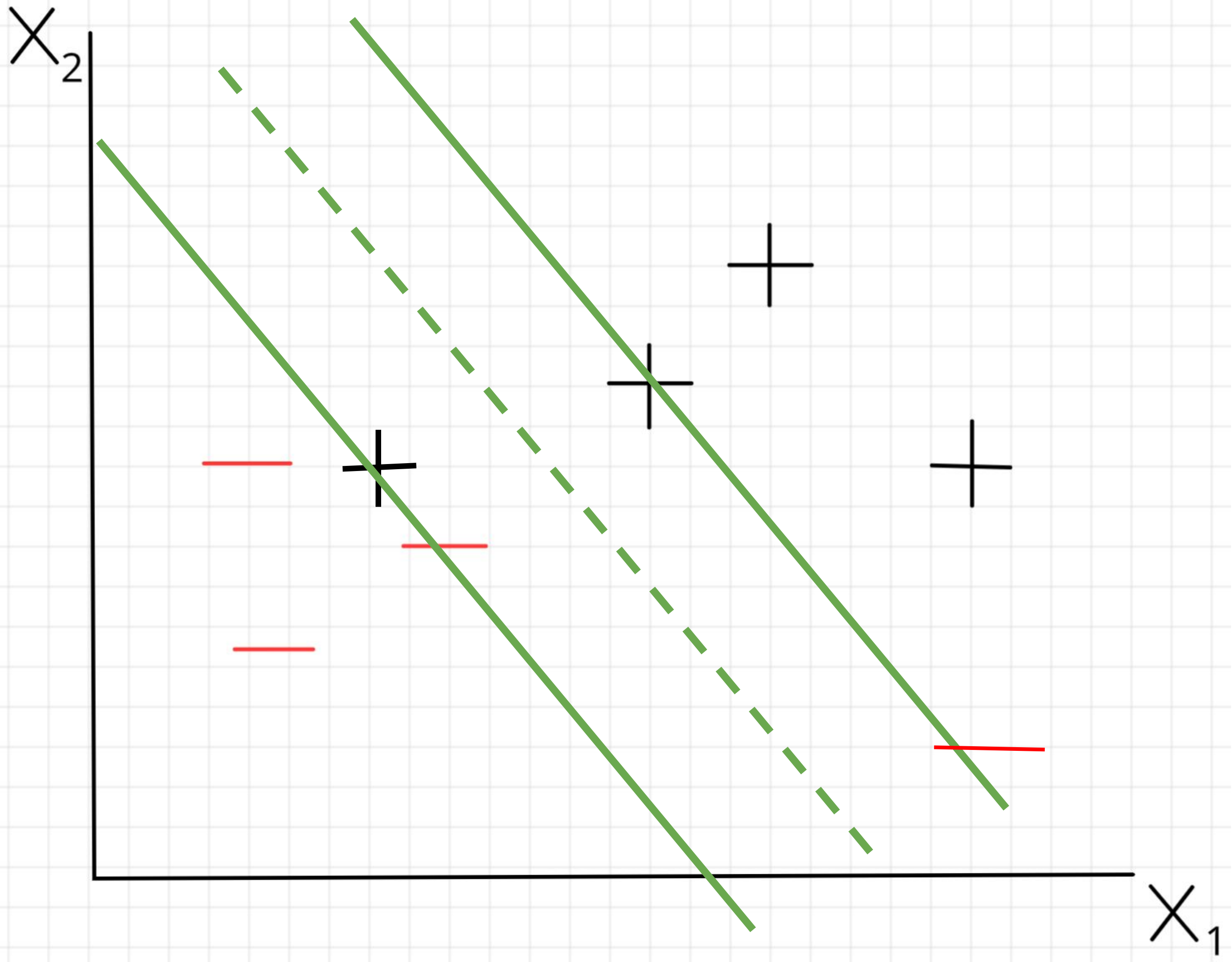
X_2
X_1


X_2
X_1

Trade-off between width and error

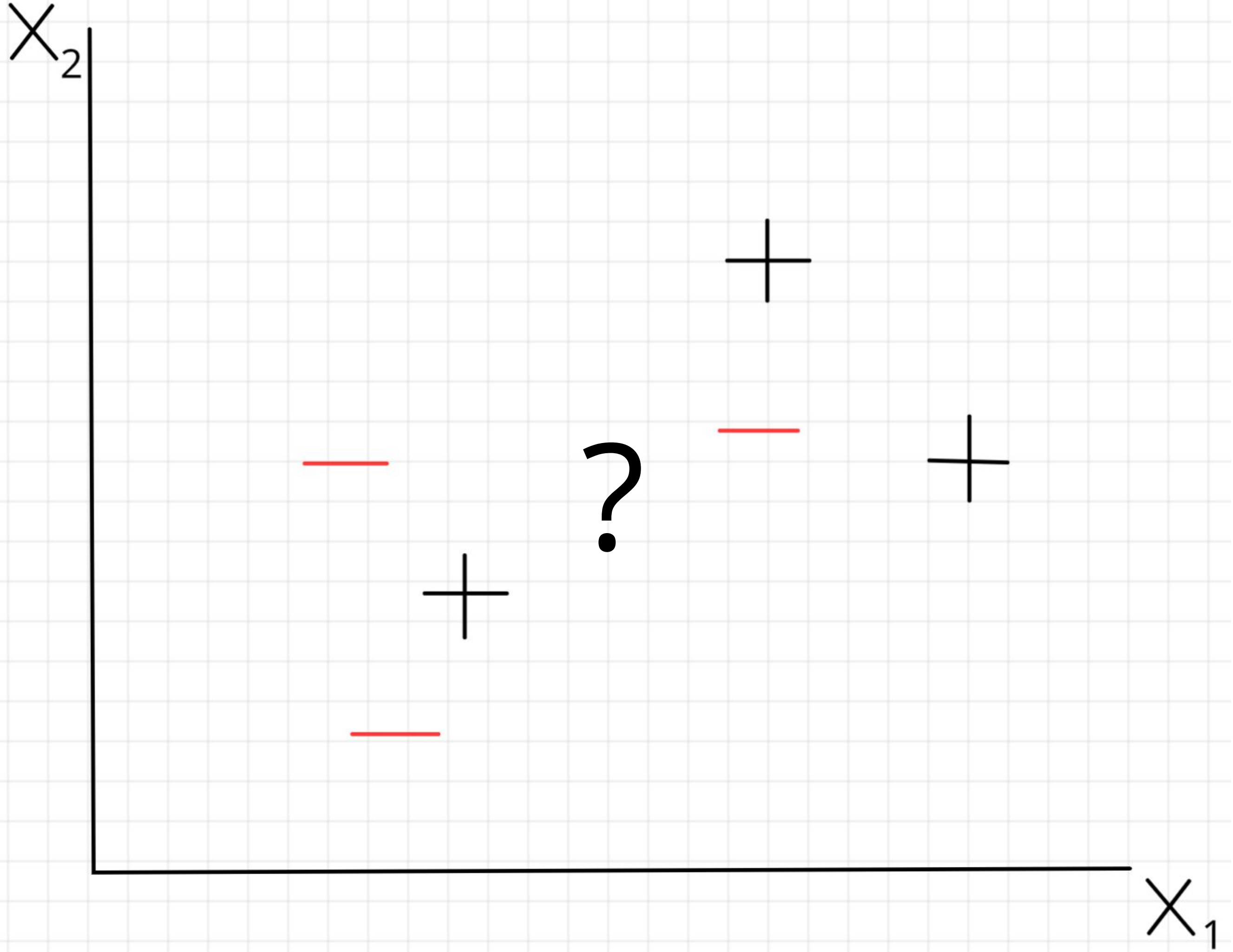

X_2
X_1


X_2
X_1



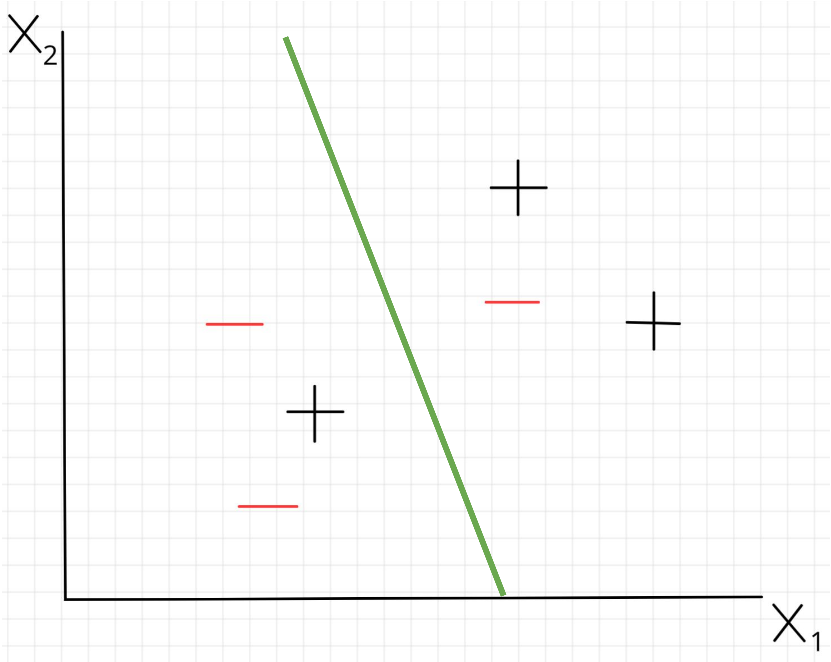
X_2
X_1

What if there is no line?

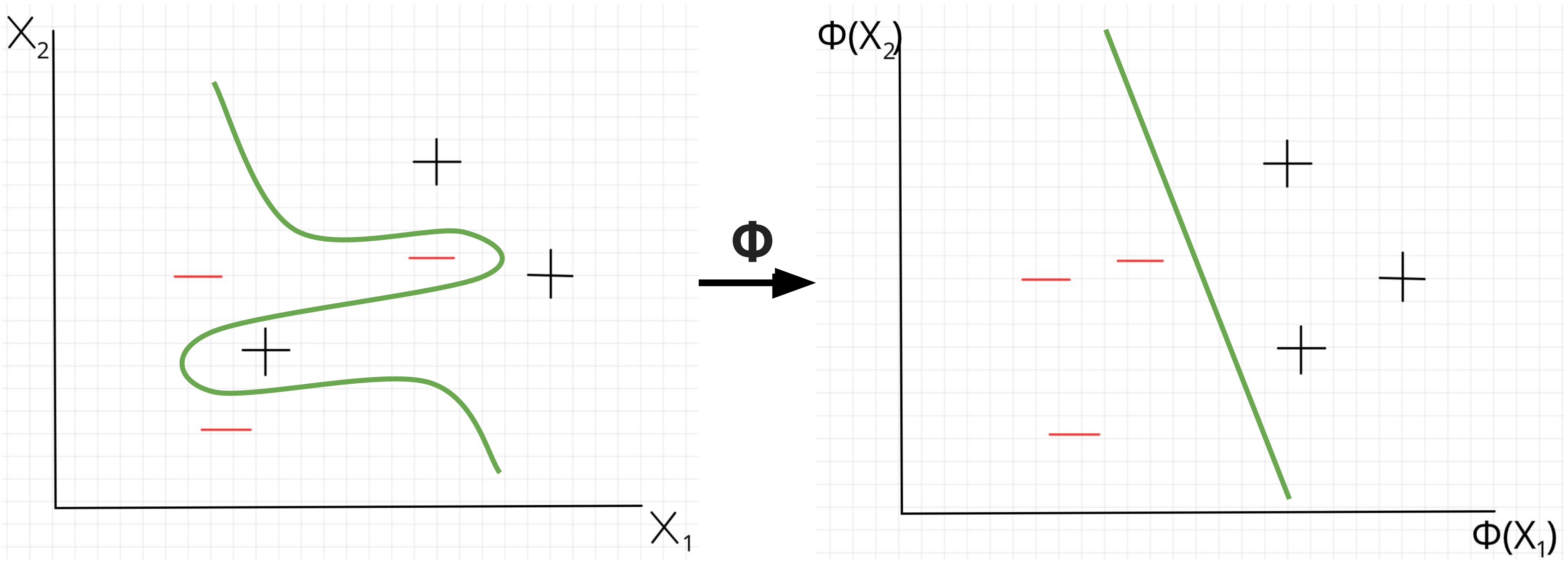

X2
X1
?

Option 1: Soft Margins

Can allow for some points in the dataset to be misclassified.



Option 2: Change perspective



But how to find Φ?

How to find the widest street

Goal is to maximize the width

$$\max\left(\frac{2}{||\vec{w}||}\right) = \min(||\vec{w}||)$$

$$= \min\left(\frac{1}{2}||\vec{w}||^2\right)$$

Subject to:

$$y_i(\vec{w} \cdot \vec{x}_i + b) - 1 = 0$$

How to find the widest street

Can use Lagrange multipliers to form a single expression to find the extremum of

$$L = \frac{1}{2}\|\vec{w}\|^2 - \sum_i \alpha_i \left[y_i(\vec{x}_i \cdot \vec{w} + b) - 1 \right]$$

where α_i is 0 for x_i not on the boundary.

Now we can take derivatives to find the extremum of **L**.

How to find the widest street

$$\frac{\partial L}{\partial \vec{w}} = \vec{w} - \sum_i \alpha_i y_i \vec{x}_i = 0$$

$$\implies \vec{w} = \sum_i \alpha_i y_i \vec{x}_i$$

Means **w** is a linear sum of vectors in our sample/training set!

$$\frac{\partial L}{\partial b} = -\sum_i \alpha_i y_i = 0$$

$$\implies \sum_i \alpha_i y_i = 0$$

How to find the widest street

Let's plug these values back into L to see what happens to L at its extremum

$$L = \frac{1}{2}\left(\sum_i \alpha_i y_i \vec{x}_i\right) \cdot \left(\sum_i \alpha_i y_i \vec{x}_i\right) - \left(\sum_i \alpha_i y_i \vec{x}_i\right) \cdot \left(\sum_i \alpha_i y_i \vec{x}_i\right) - \cancel{\sum_i \alpha_i y_i b} + \sum_i \alpha_i$$

Simplifying, we get:

$$L = \sum_i \alpha_i - \frac{1}{2}\left(\sum_i \alpha_i y_i \vec{x}_i\right) \cdot \left(\sum_i \alpha_i y_i \vec{x}_i\right)$$

$$= \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j$$

Quadratic Programming

Solving lagrange multipliers in general requires numerical optimization methods called quadratic programming solvers.

But how to find Φ?

Turns out we don't need to find or define a transformation Φ!

Looking back at L, since **it depends only on the dot product of our input**, we only need to define the dot product in our transformed space.

$$L := \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j$$

But how to find Φ?

Turns out we don't need to find or define a transformation Φ!

Looking back at L, since **it depends only on the dot product of our input**, we only need to define the dot product in our transformed space.

i.e. we only need to define

$$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$$

Called a Kernel function. This is often referred to as the "kernel trick".

$$L := \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(\vec{x}_i, \vec{x}_j)$$

Example Kernel Functions

Polynomial Kernel

$$K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + 1)^n$$

Radial Basis Function Kernel

$$K(\vec{x}_i, \vec{x}_j) = e^{\frac{\|\vec{x}_i - \vec{x}_j\|}{\sigma}}$$

Kernel Function (intuition)

- The inner product of a space describes how close / similar points are
- Kernel Functions allow for specifying the closeness / similarity of points in a hypothetical transformed space
- The hope is that with that new notion of closeness, points in the dataset are linearly separable.

Worksheet