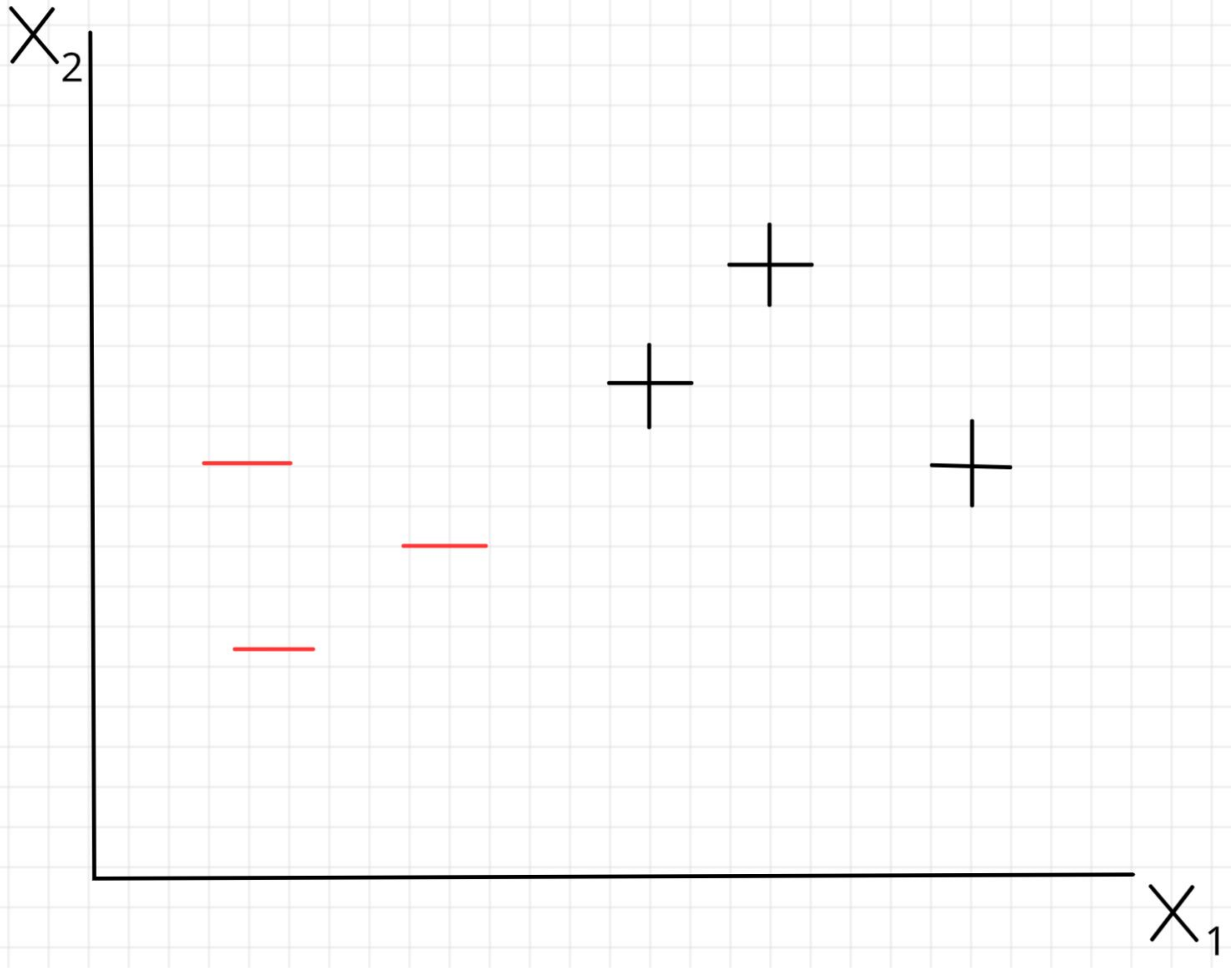
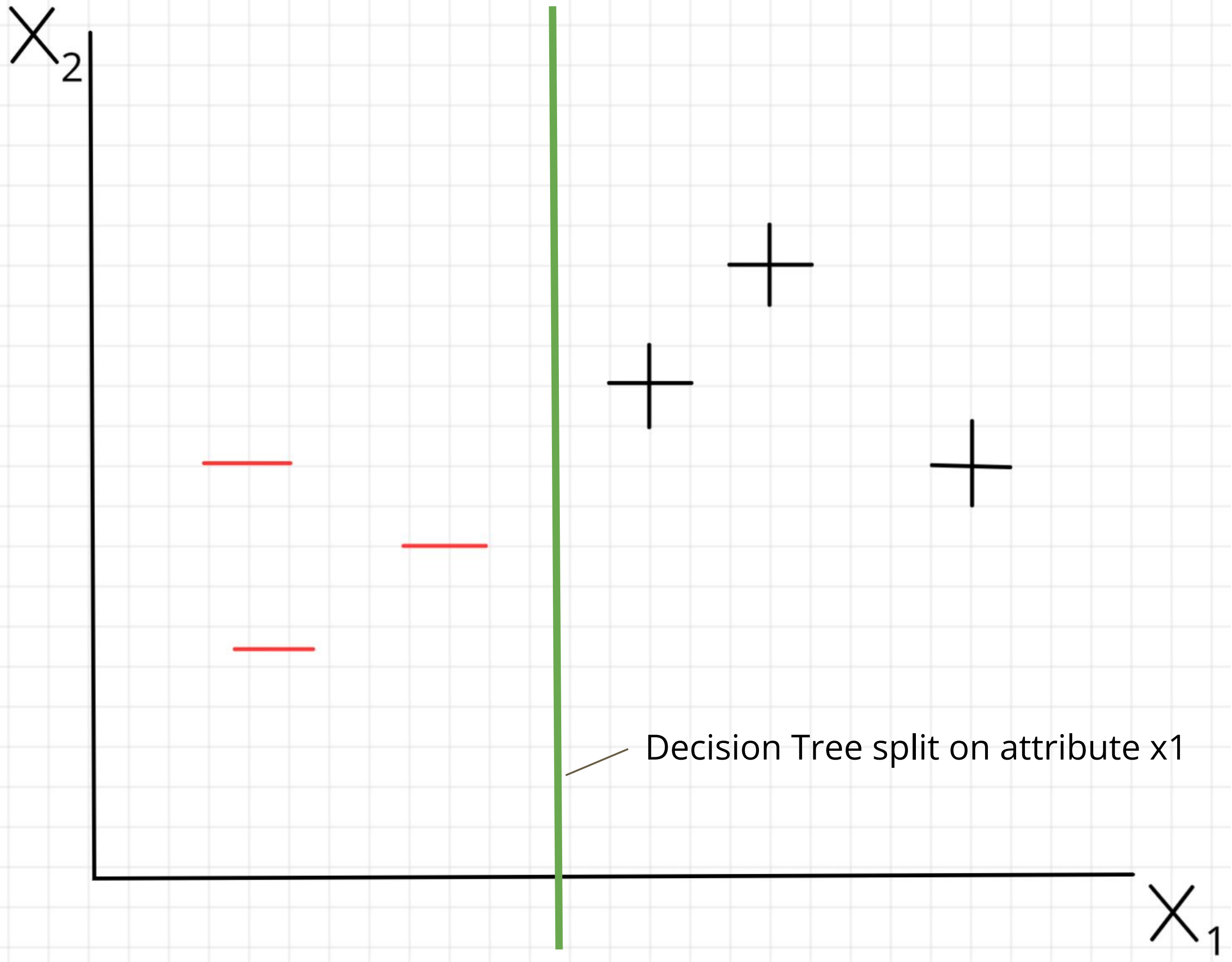
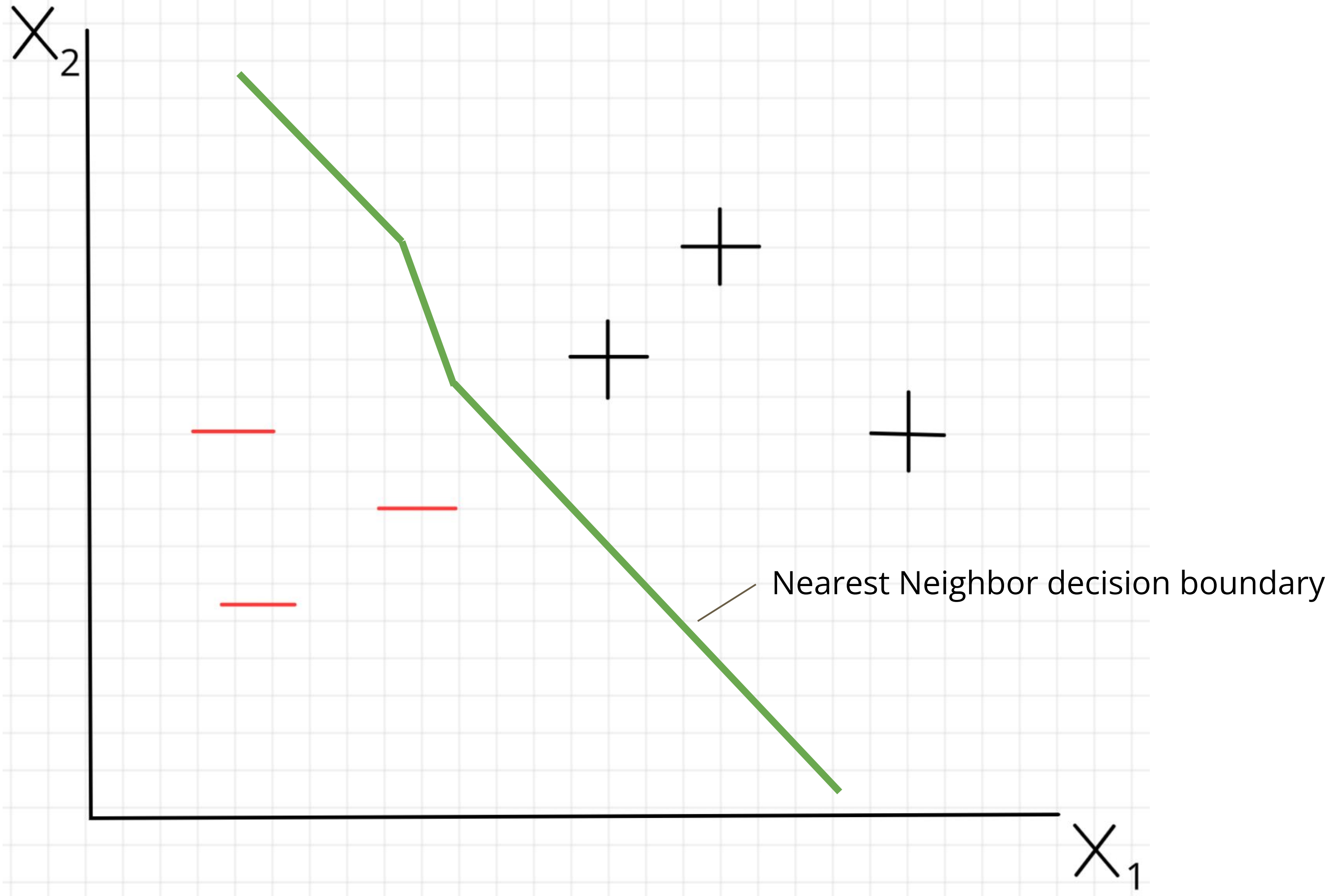
Support Vector Machines

Boston University CS 506 - Lance Galletti


X_2
X_1

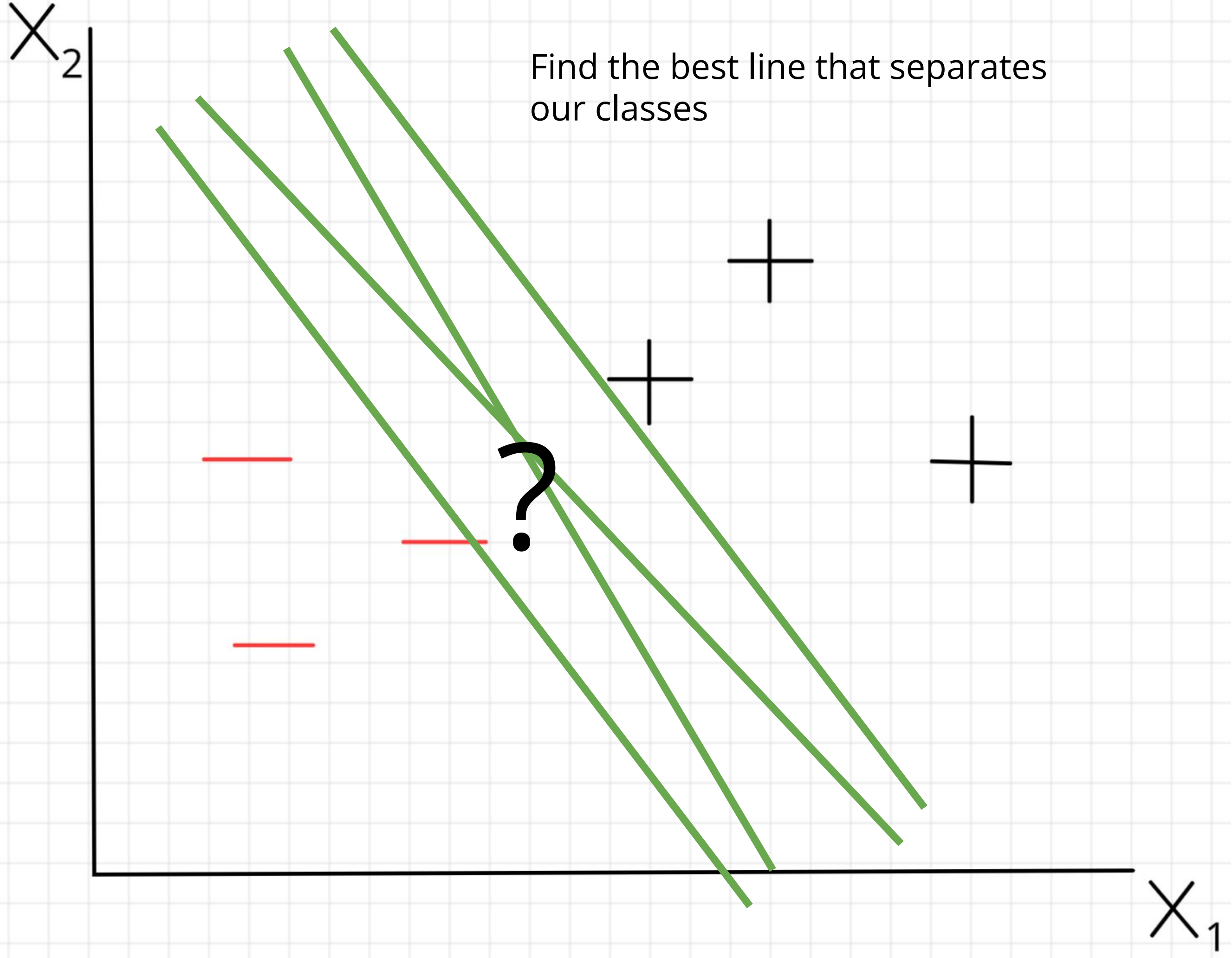


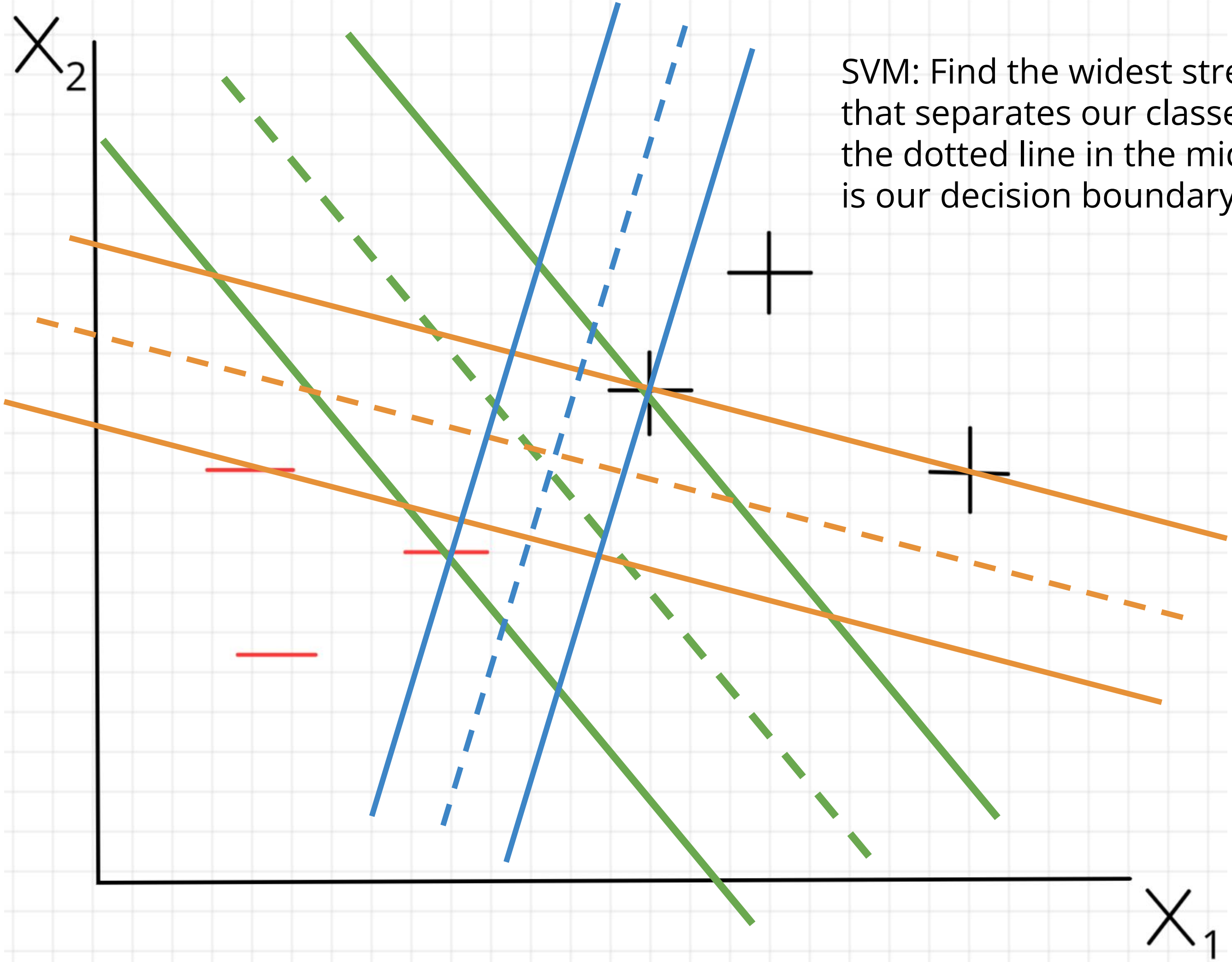
X2
X1
Decision Tree split on attribute x1



X_2
Nearest Neighbor decision boundary
X_1

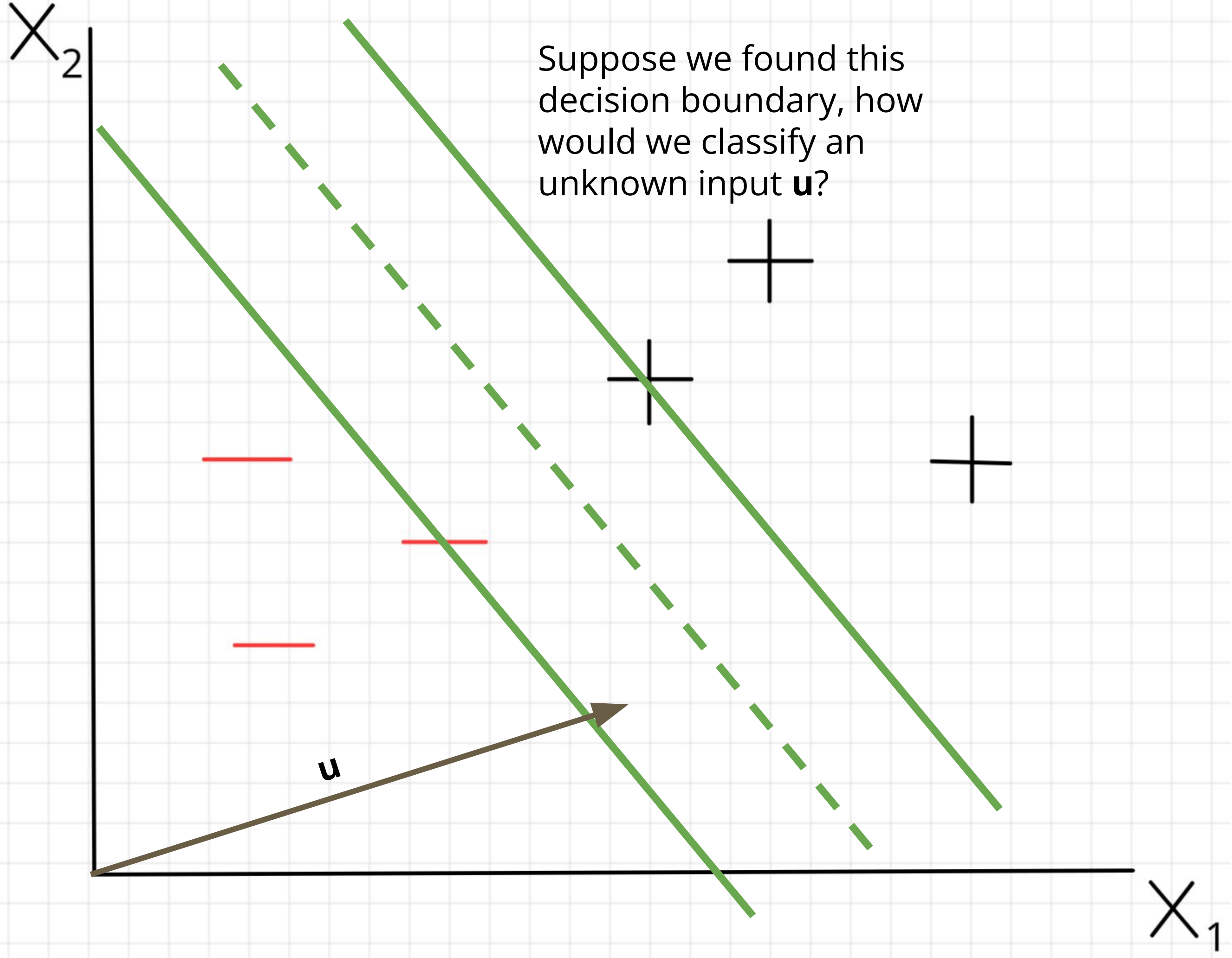
Find the best line that separates our classes



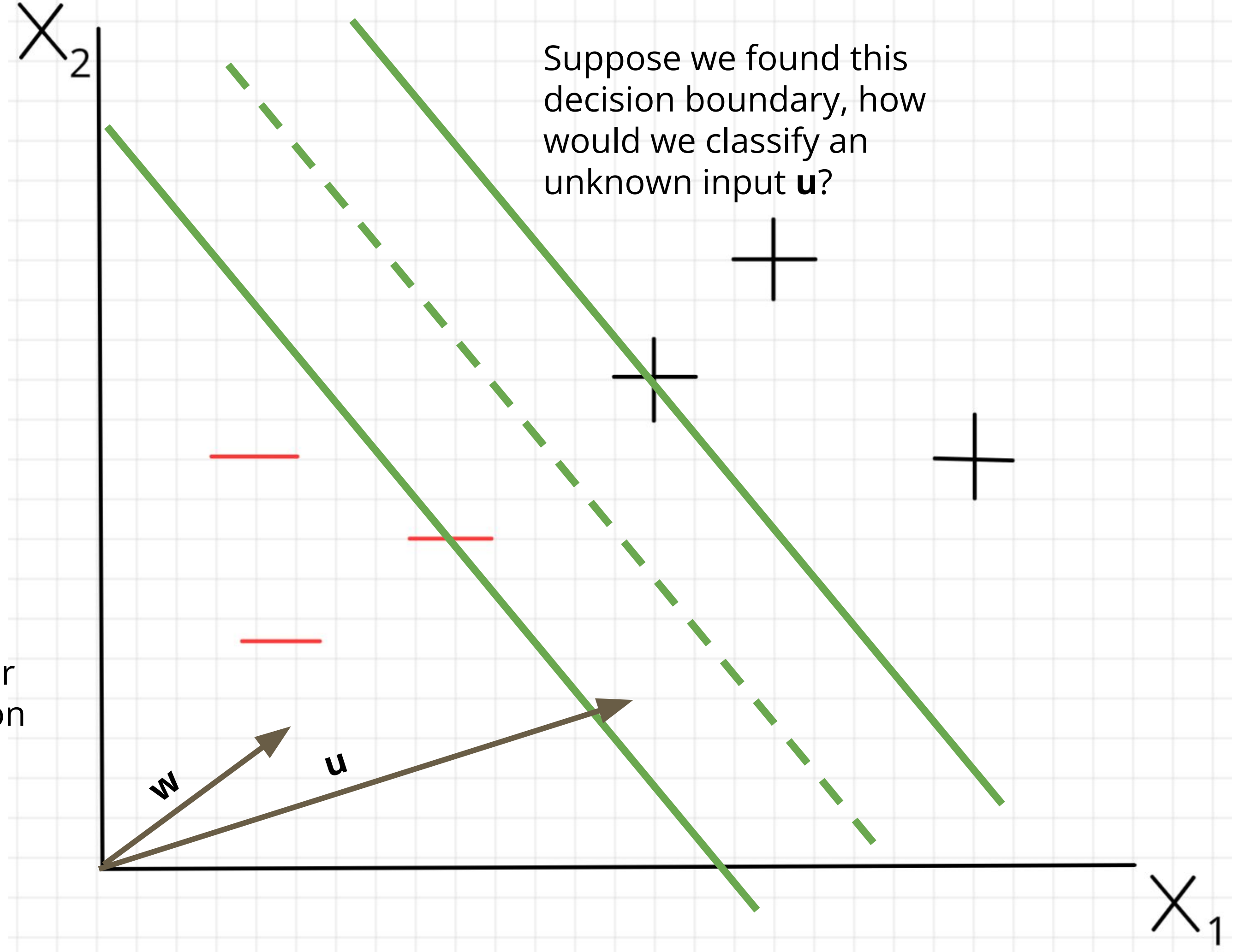


SVM: Find the widest street that separates our classes - the dotted line in the middle is our decision boundary

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

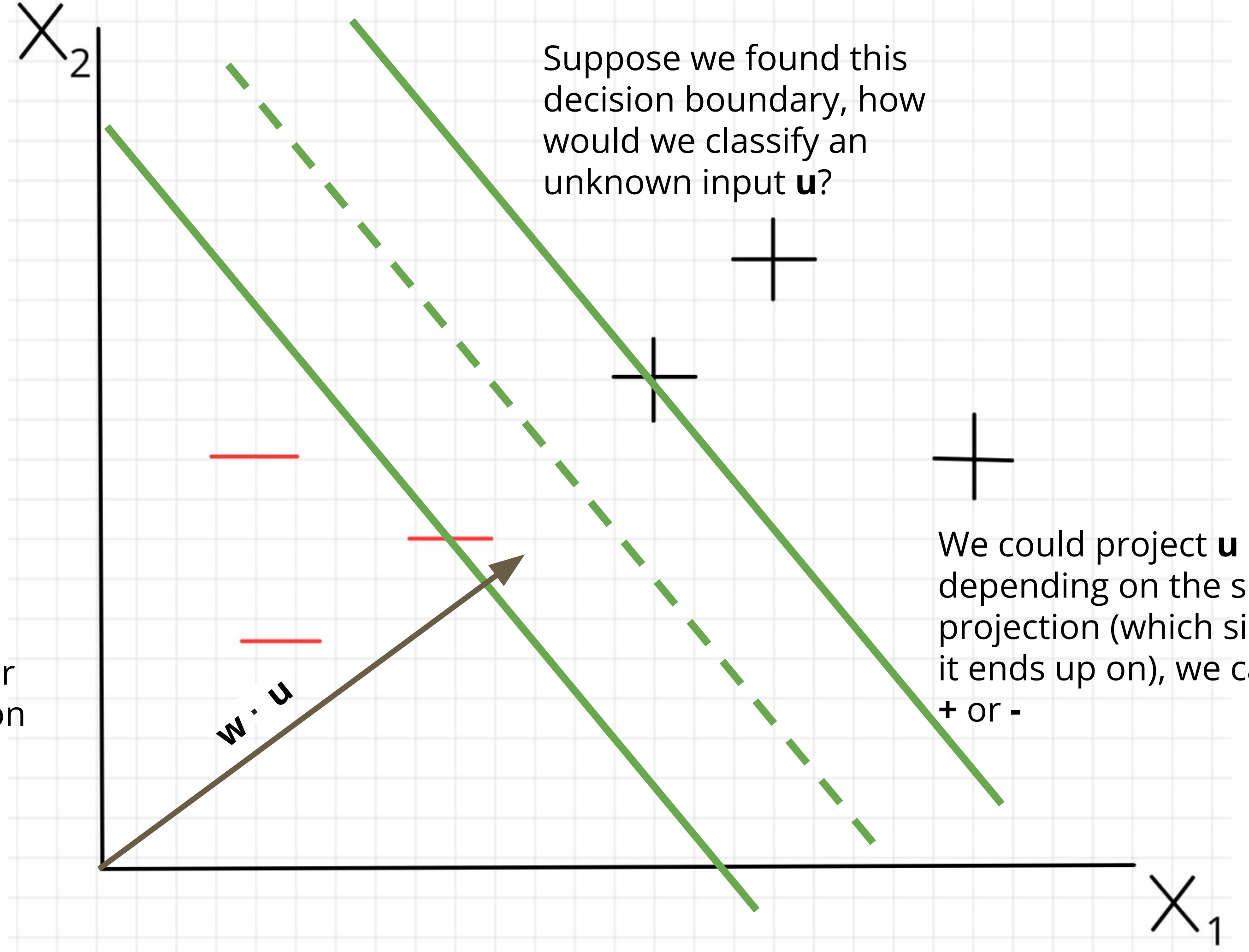


Suppose we found this decision boundary, how would we classify an unknown input **u**?



Let **w** be a vector perpendicular to the decision boundary

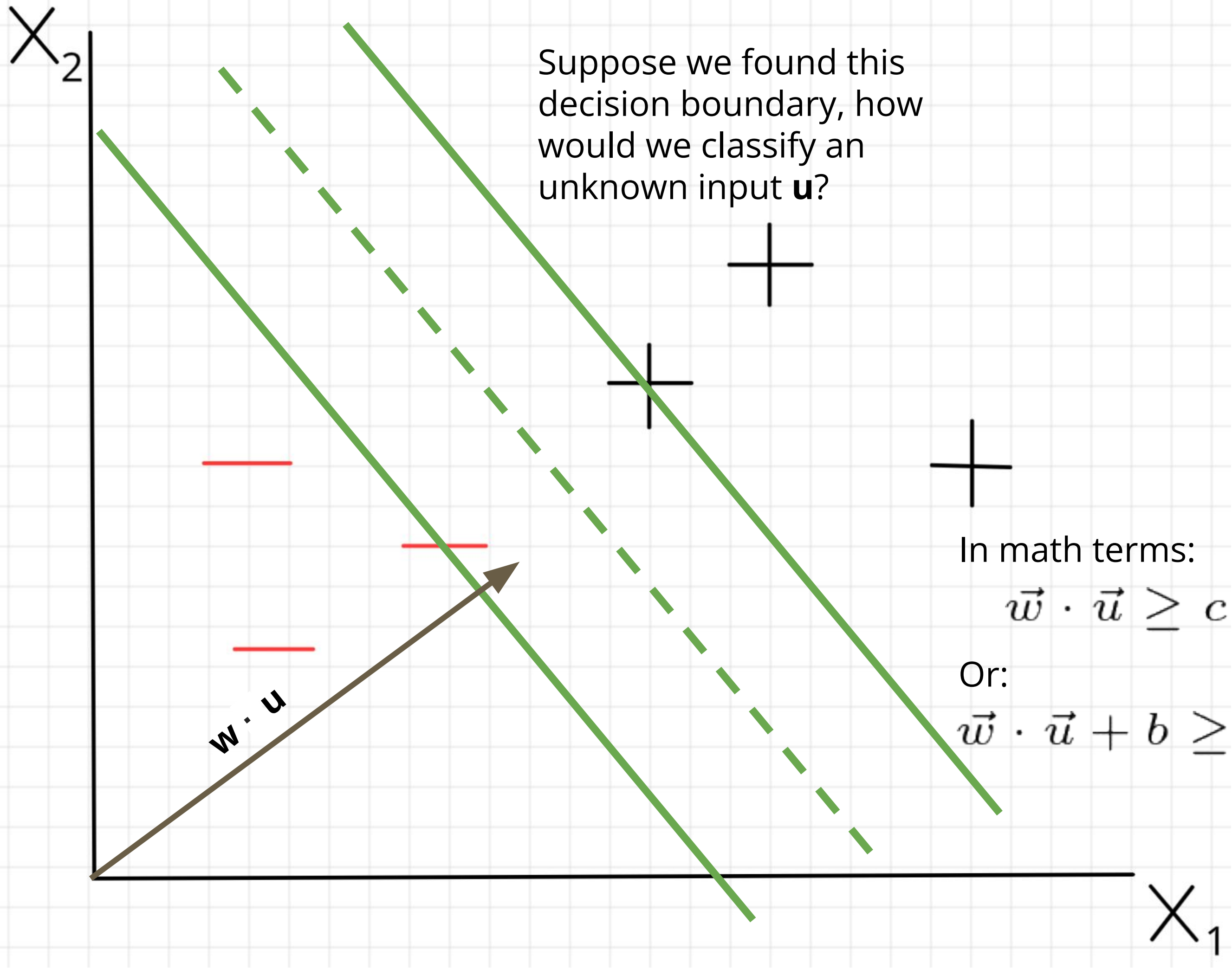
Suppose we found this decision boundary, how would we classify an unknown input **u**?



We could project **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 **-**

Let **w** be a vector perpendicular to the decision boundary

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

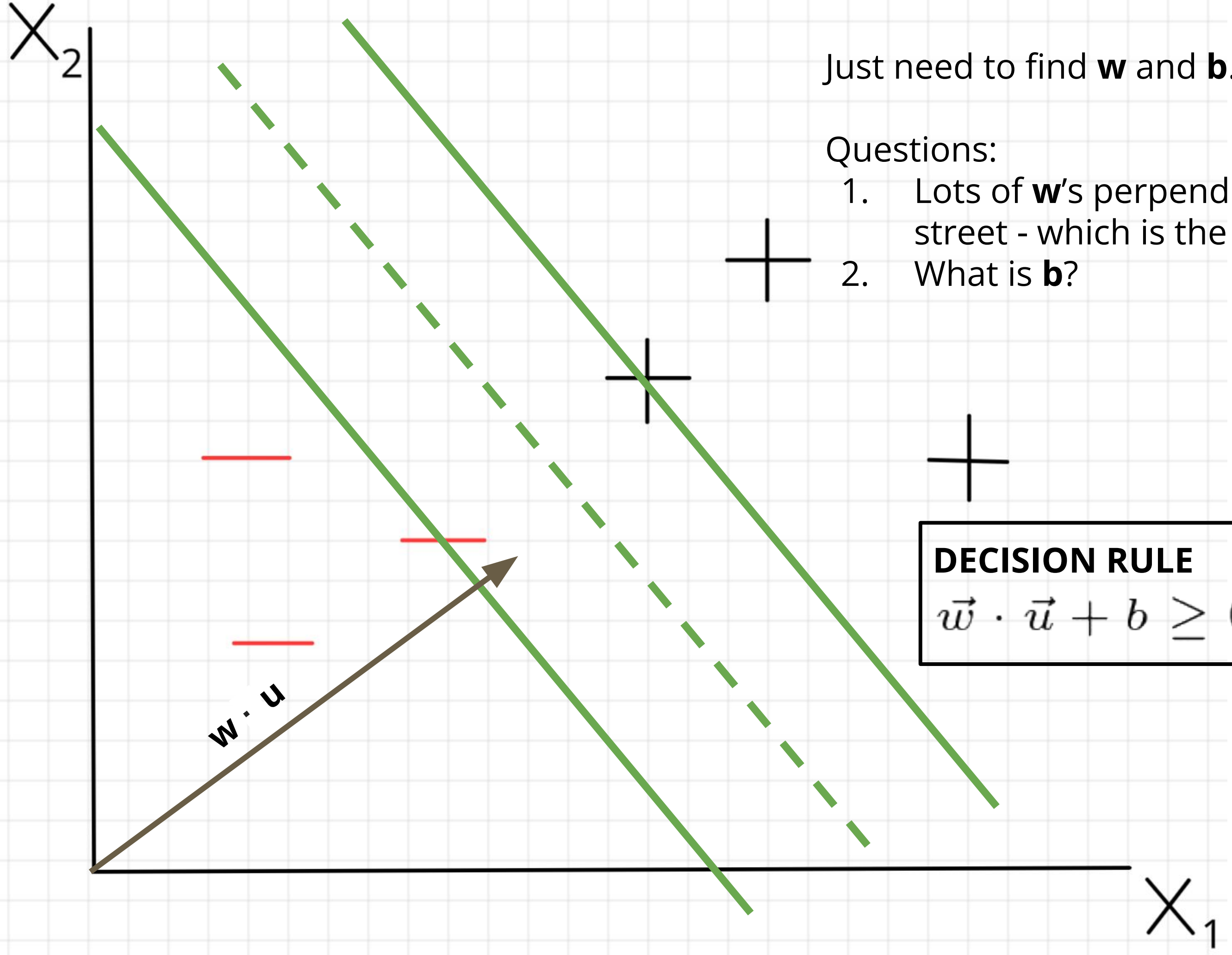


In math terms:

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

Or:

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



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

Questions:

1. Lots of **w**'s perpendicular to the street - which is the best one?
2. What is **b**?

DECISION RULE

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

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

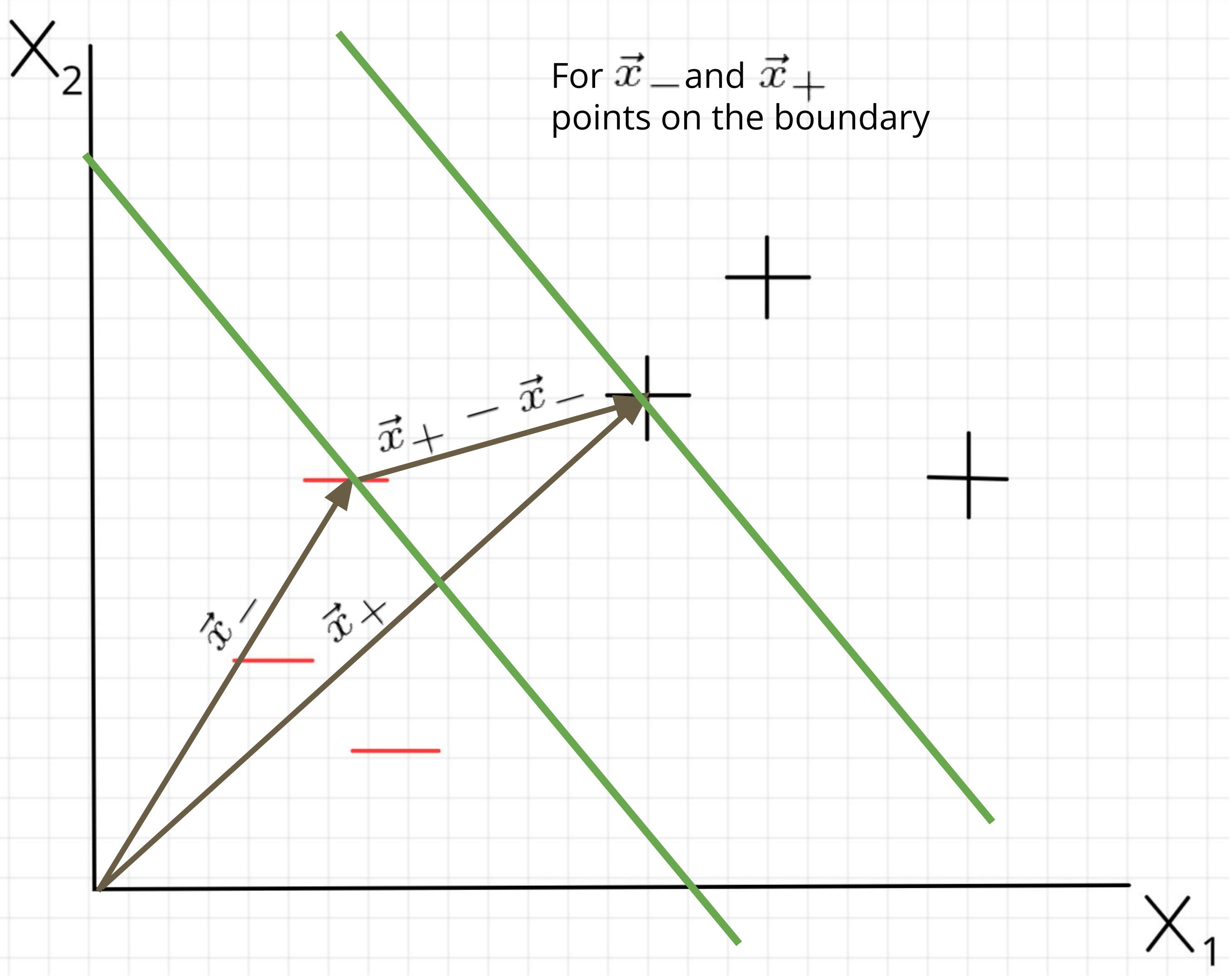
If we multiply our sample decision rules by this new variable:

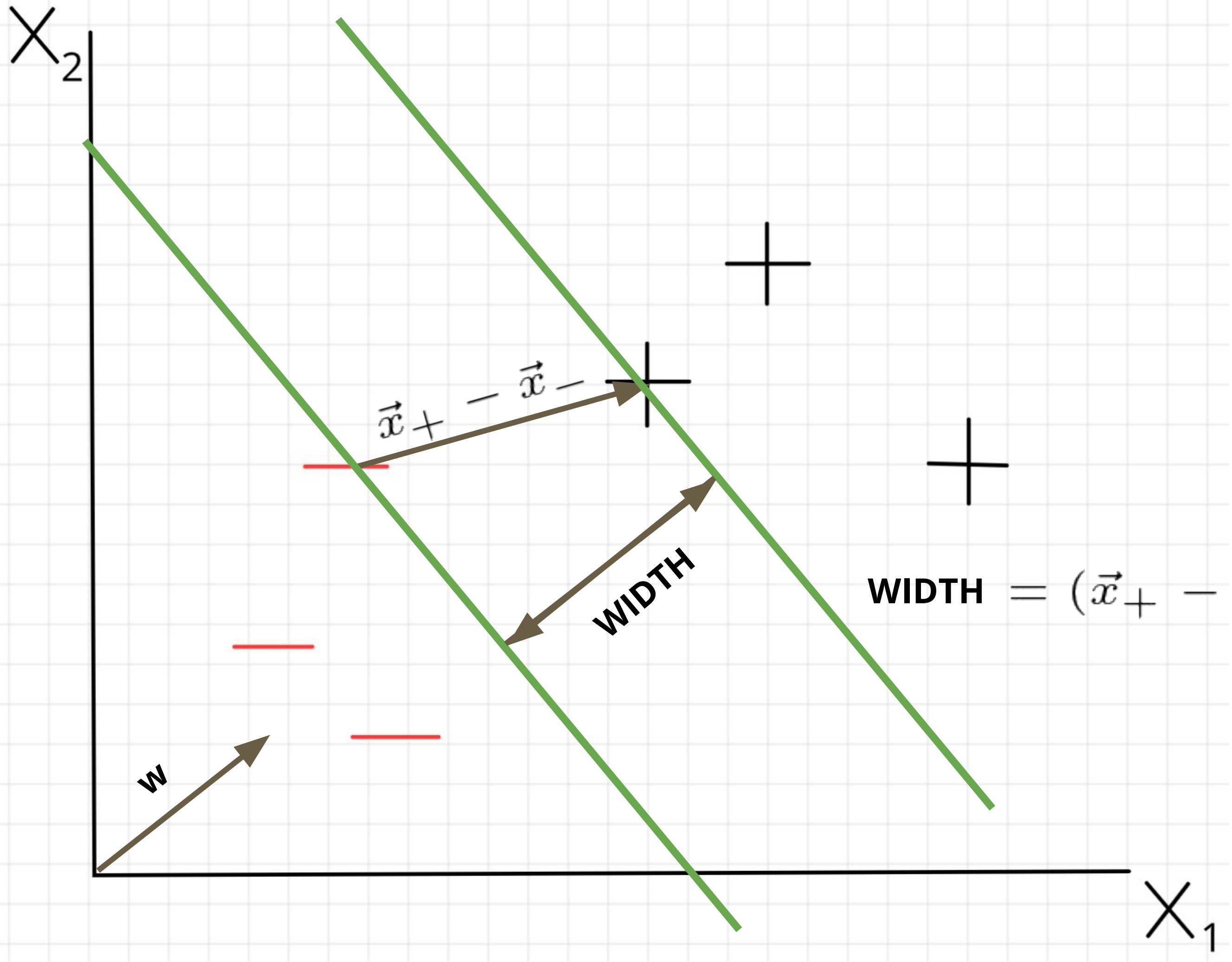
$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1$$

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

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

For $\vec{x}_-$ and $\vec{x}_+$
points on the boundary





$$\text{WIDTH} = (\vec{x}_+ - \vec{x}_-) \cdot \frac{\vec{w}}{\|\vec{w}\|}$$

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** $= \frac{2}{||\vec{w}||}$

(as an exercise, try to show this)

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

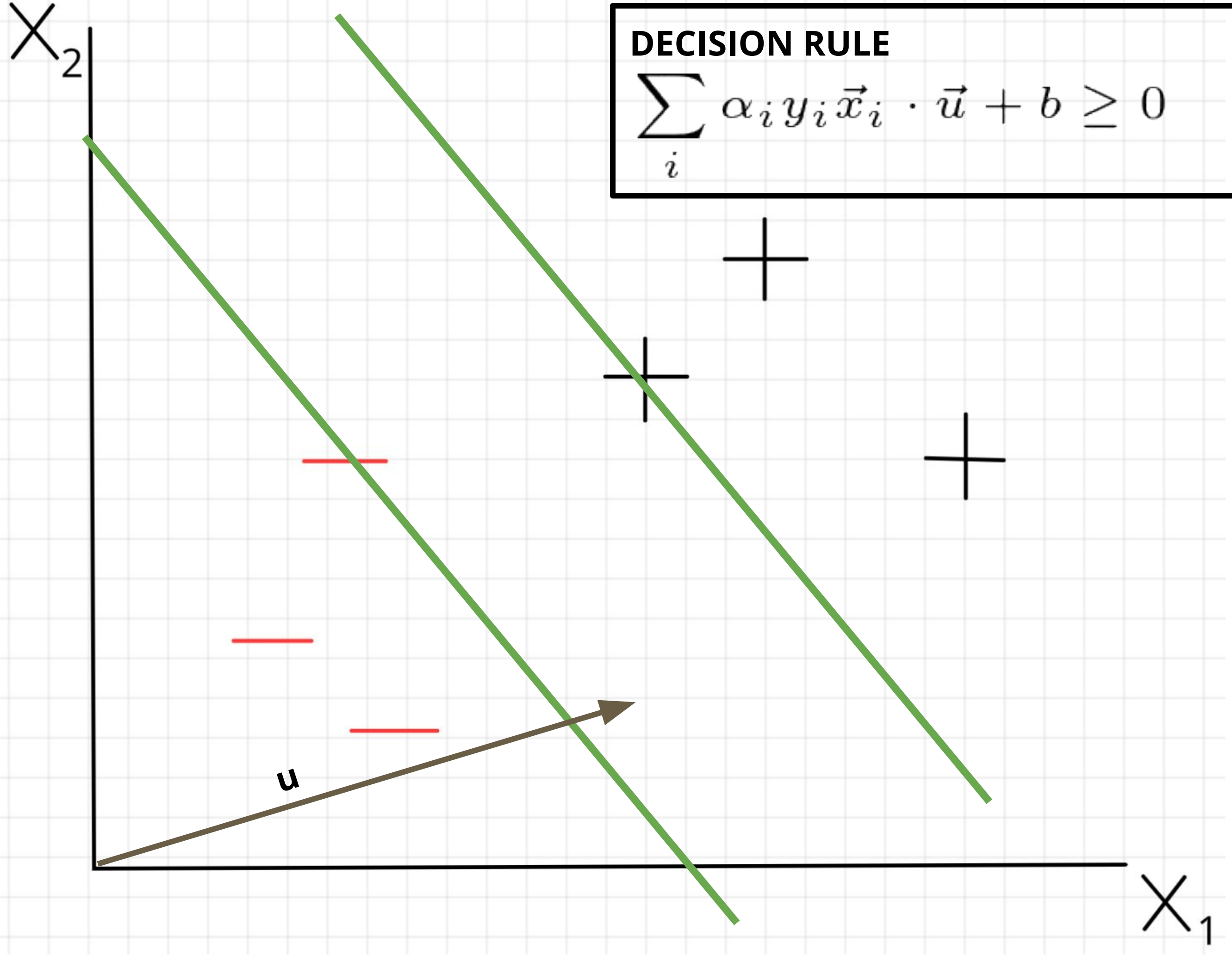
Let's plug these values back into L to see what happens to L at its extremum

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

Simplifying, we get:

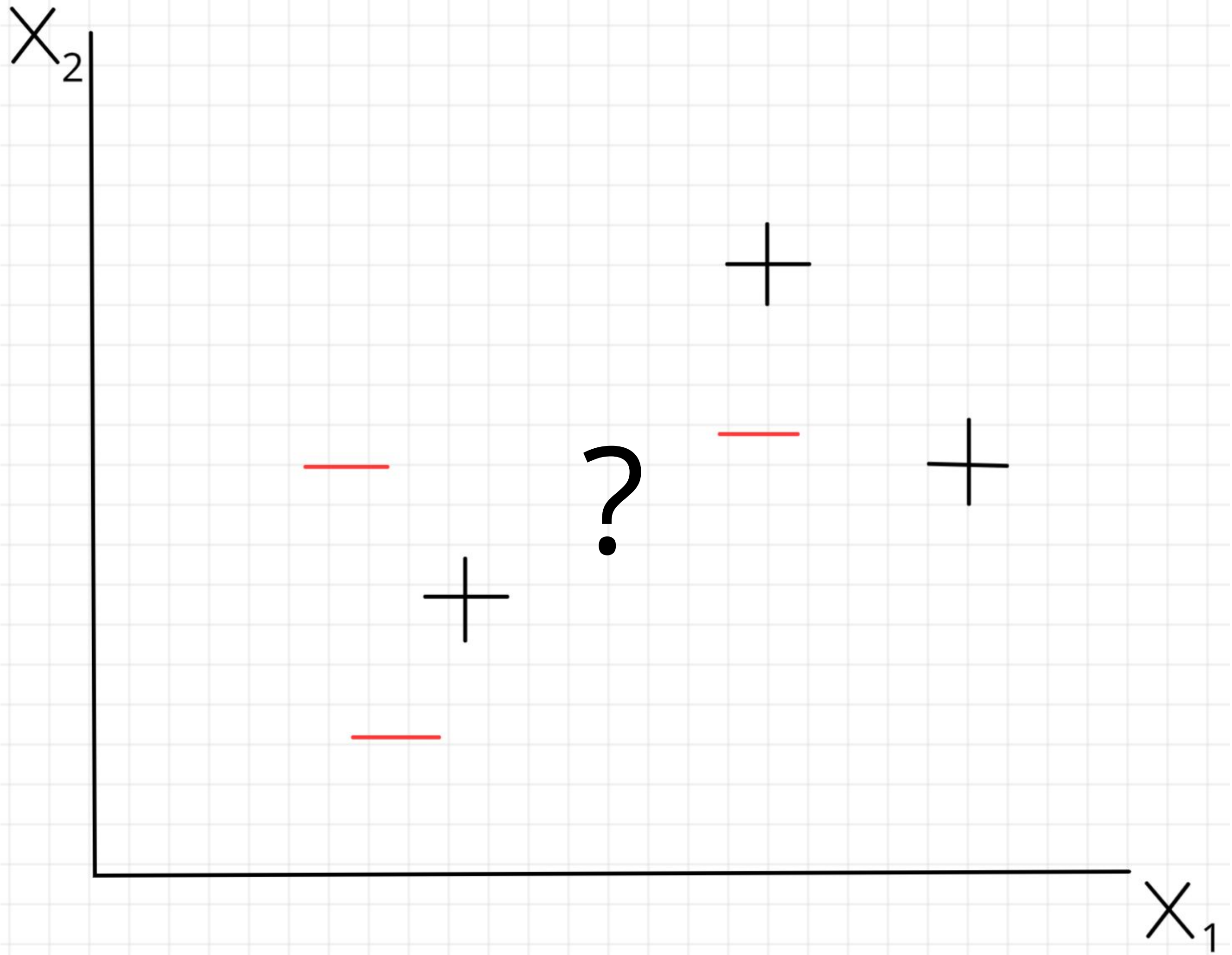
$$L = \sum_i \alpha_i - \frac{1}{2}\left(\sum_i \alpha_i y_i \vec{x}_i\right) \cdot \left(\sum_i \alpha_i y_i \vec{x}_i\right)$$

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



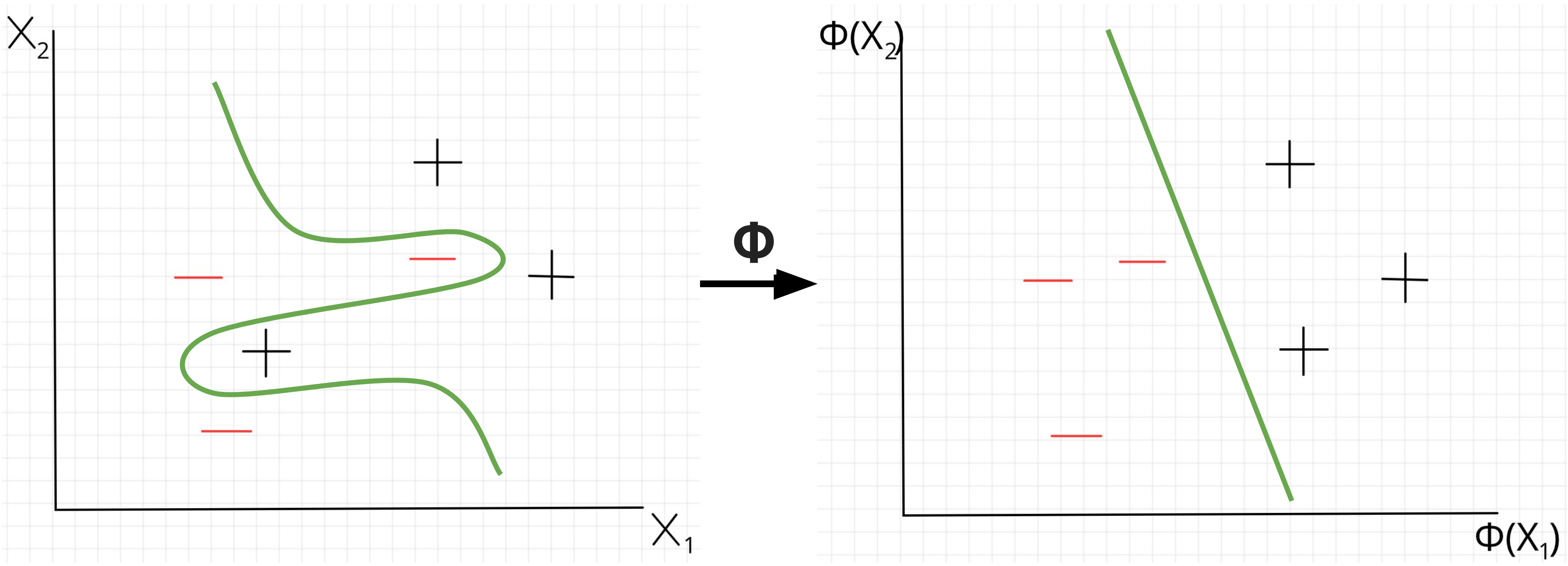
DECISION RULE

$$\sum_i \alpha_i y_i \vec{x}_i \cdot \vec{u} + b \geq 0 \quad \text{then } +$$



X_2
X_1
?

When stuck - change perspective



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".

Example Kernel Functions

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

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

DEMO