Machine Learning with Python

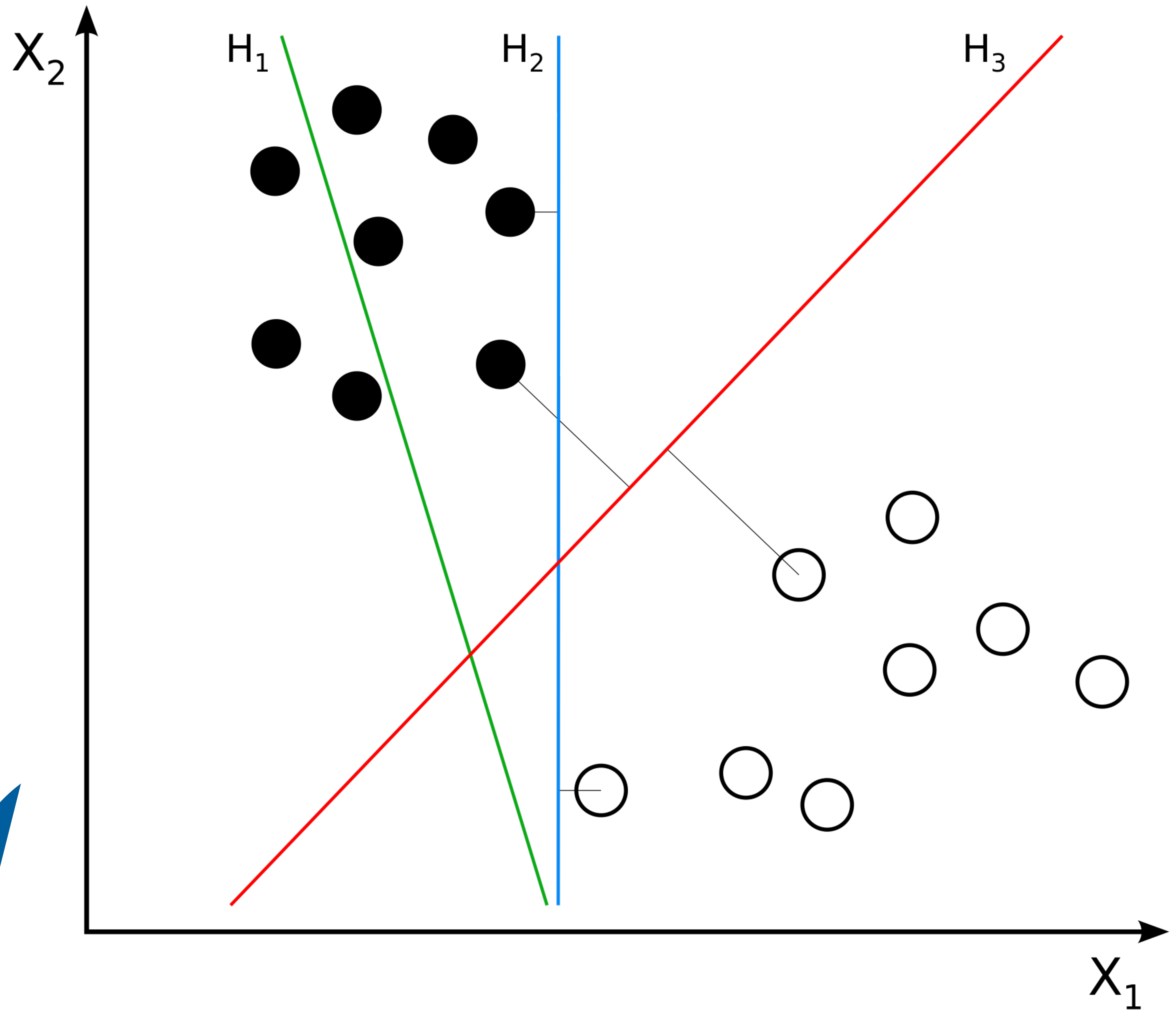
MTH786U/P 2023/24

Lecture 10: Support vector machines

Nicola Perra, Queen Mary University of London (QMUL)

n.perra@qmul.ac.uk

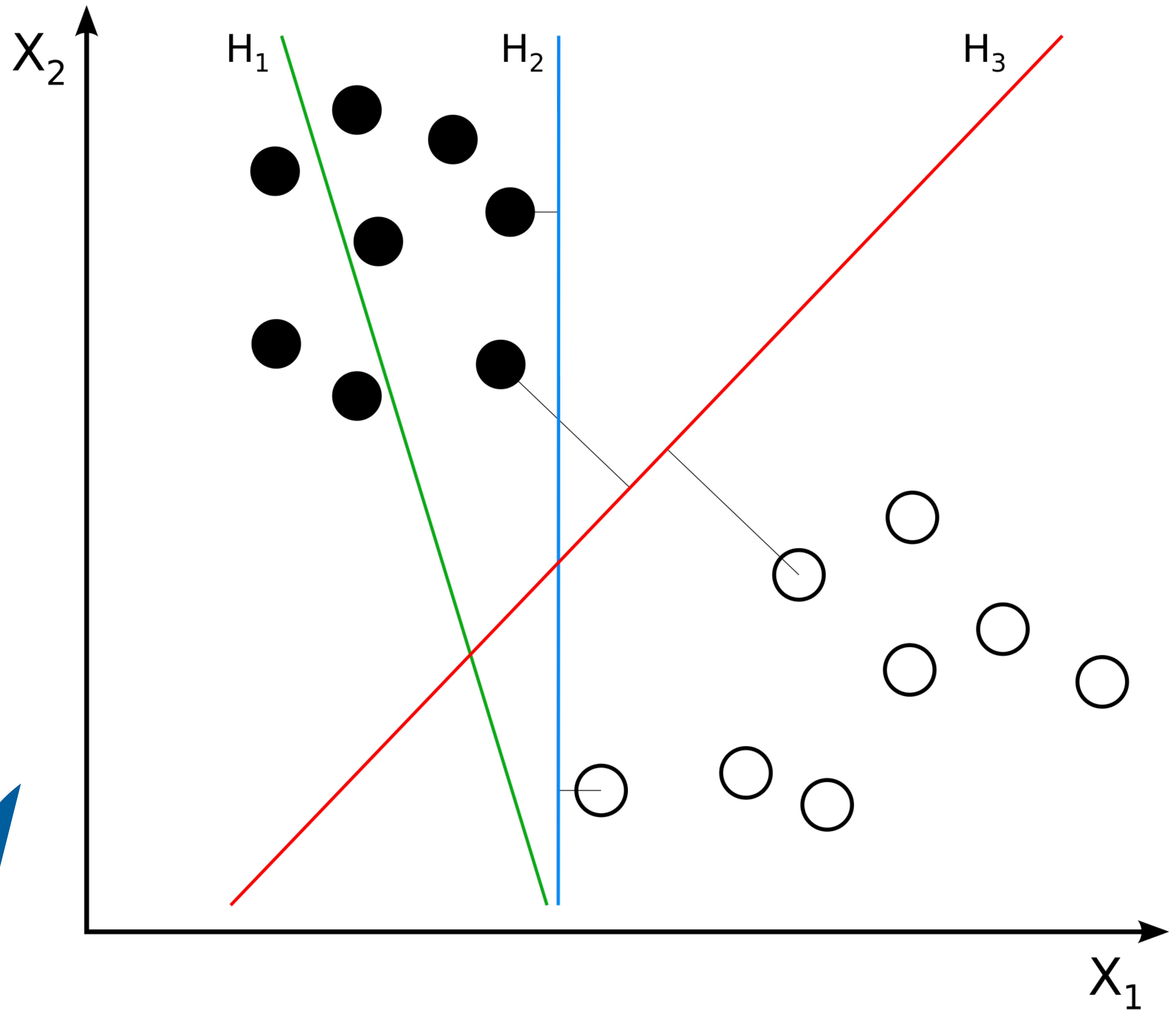
Support vector machines



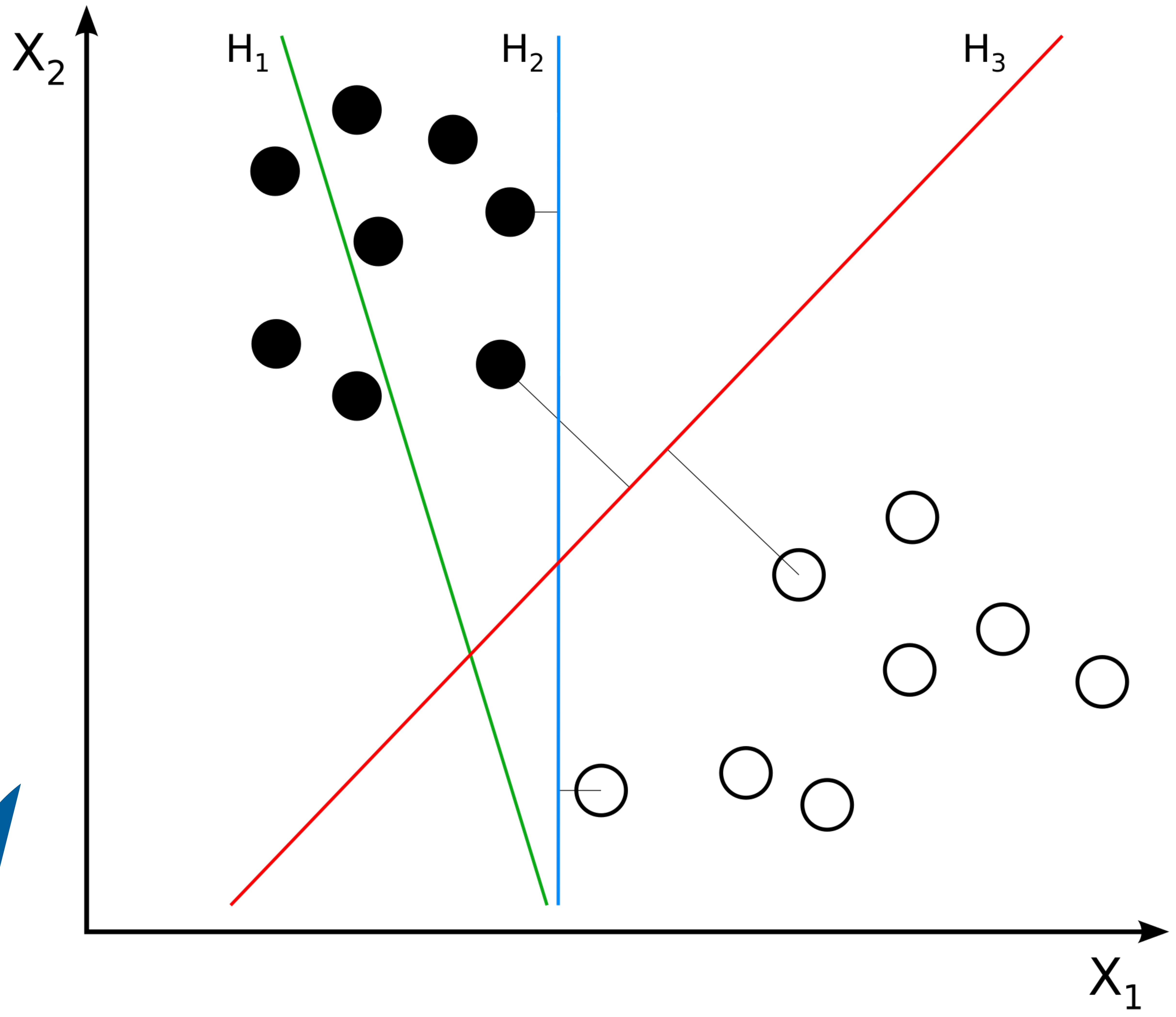
Suppose we would like to classify the data, splitting it in two categories

Which line does the best job?

Support vector machines



Support vector machines

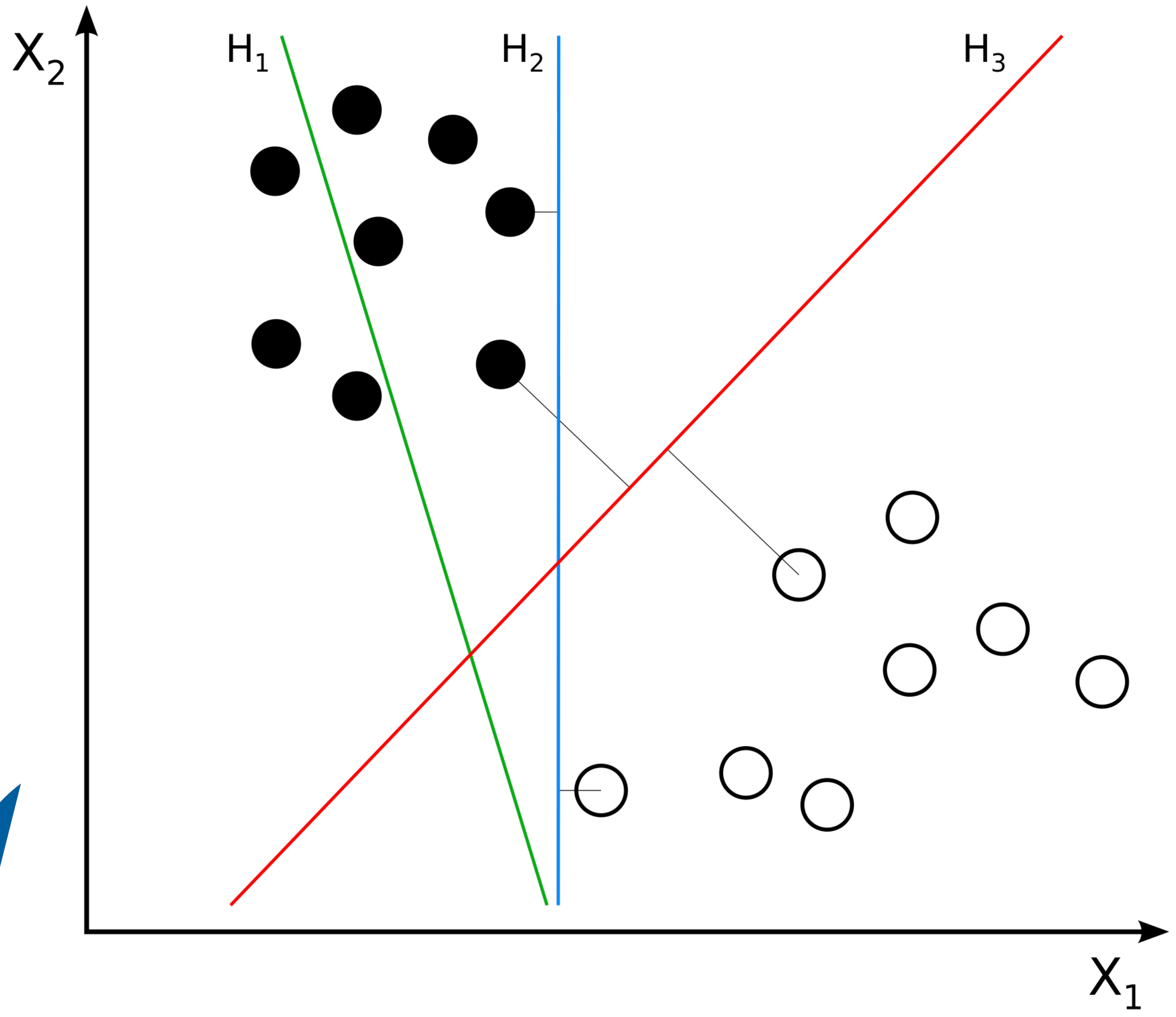


In this case H3 looks better than the other two

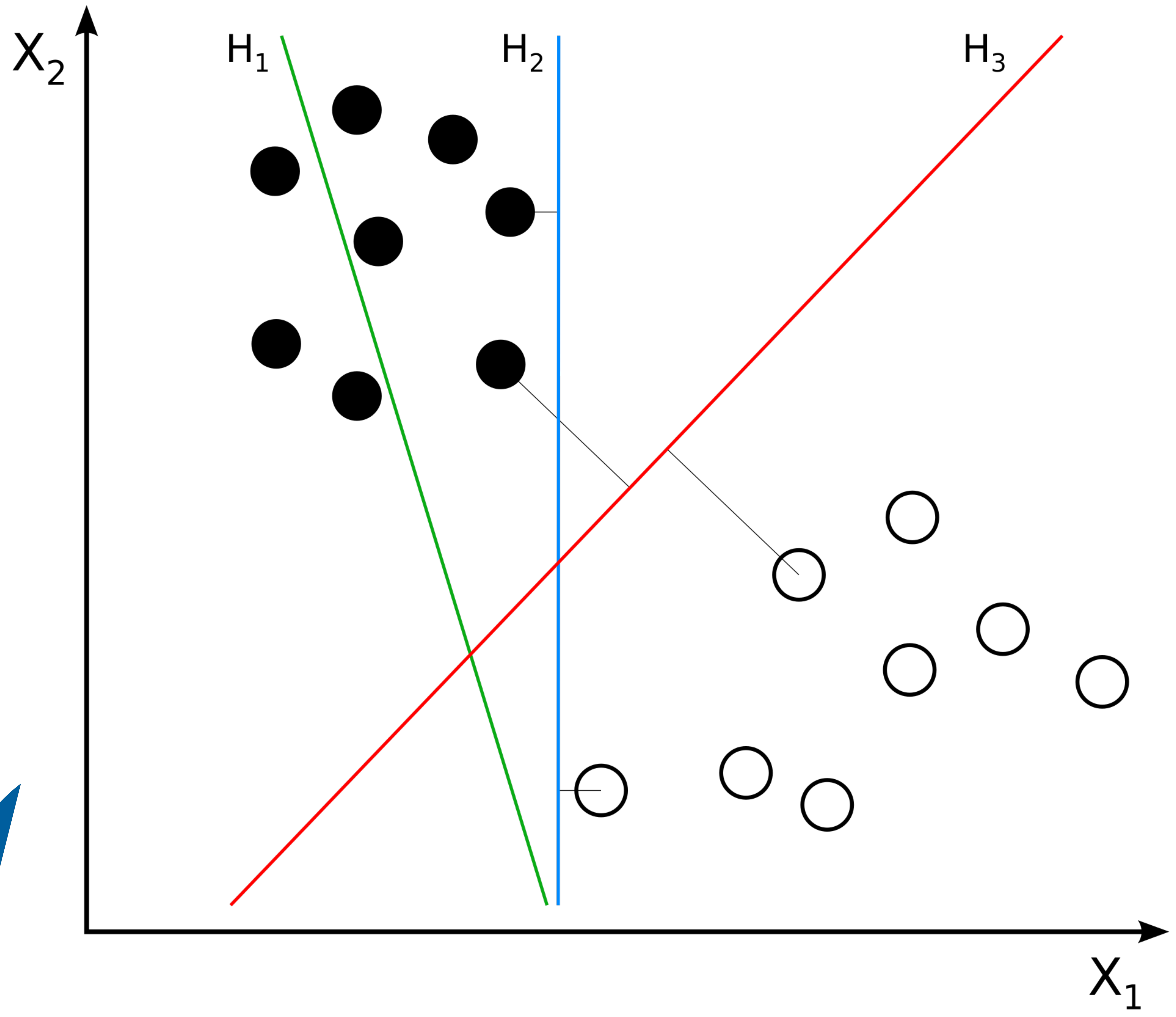
It is more separated from the data points. Farther with respect to the closest data point

Small perturbations of the data would not affect it as much

Support vector machines

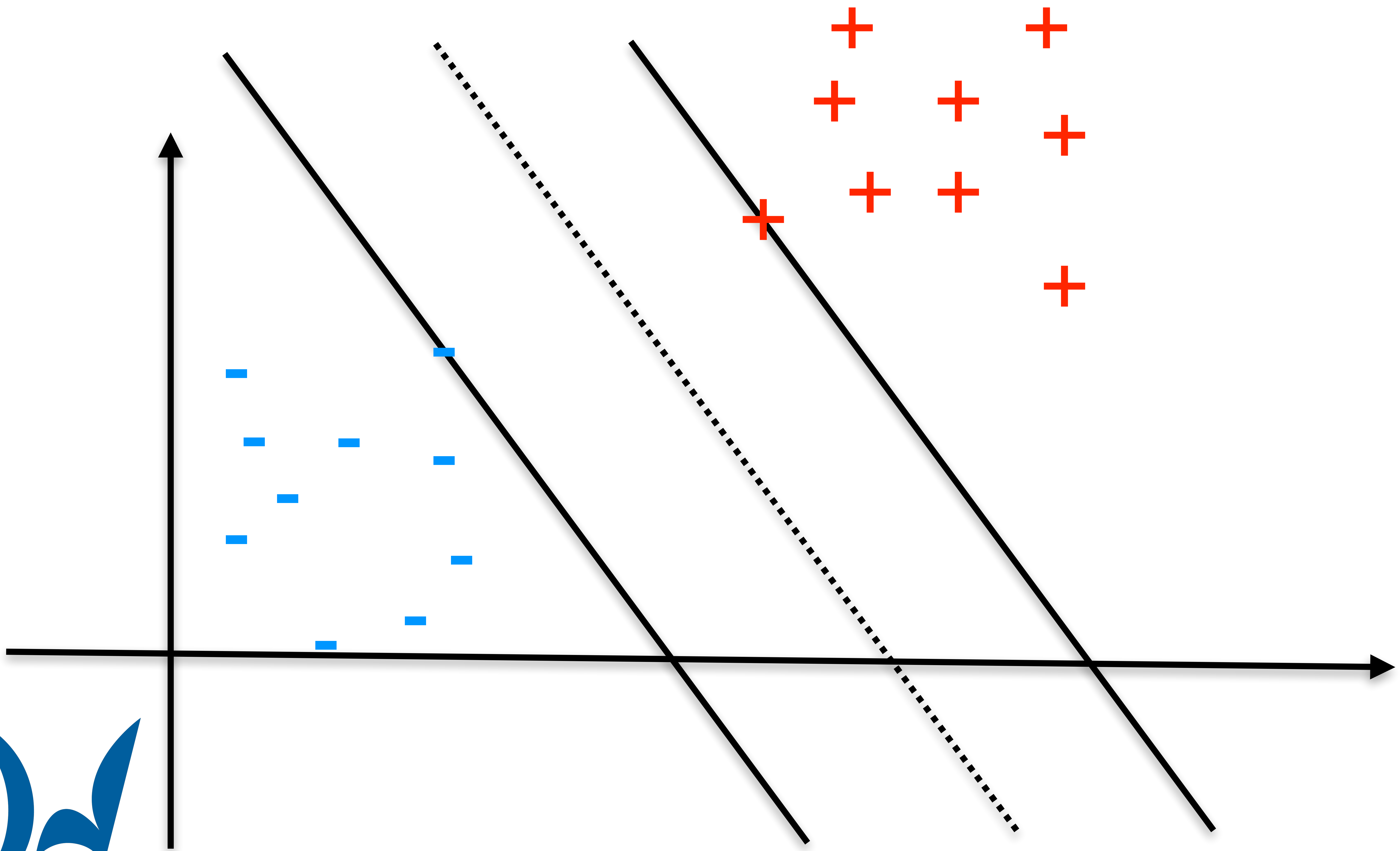


Support vector machines

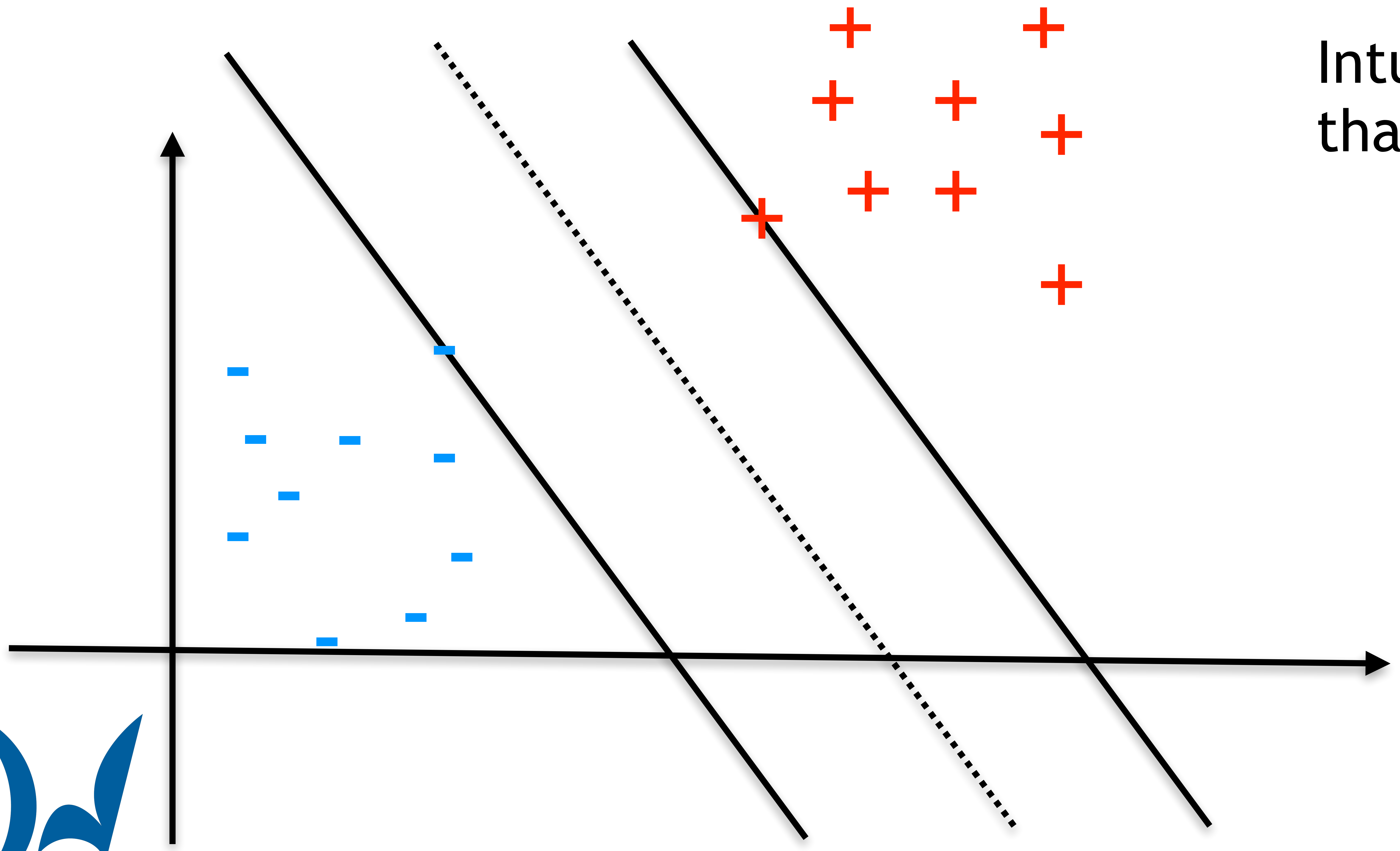


General question: how do we ensure an 'optimal' hyperplane?

Support vector machines

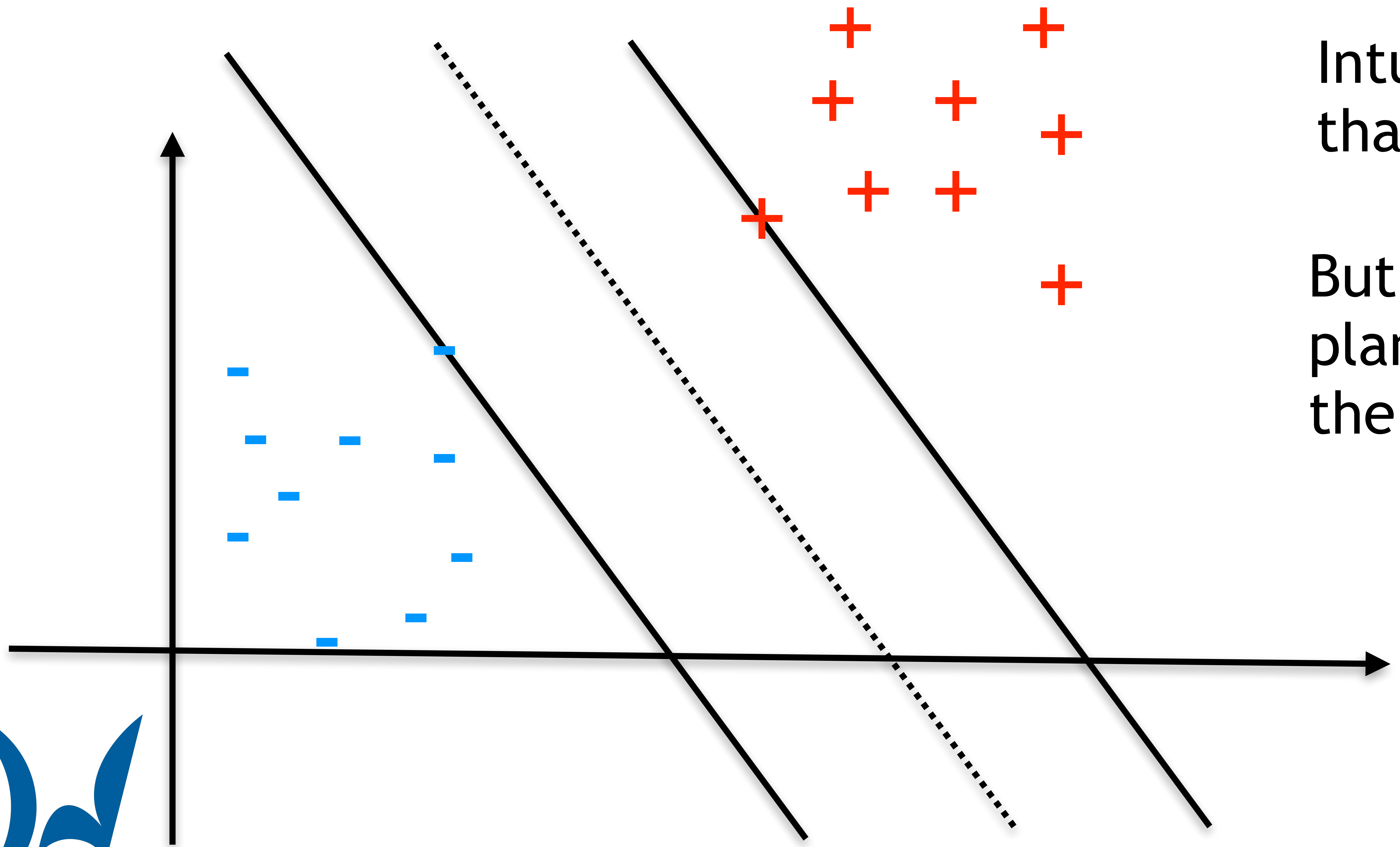


Support vector machines



Intuition is to find a vector/plane that separates the two

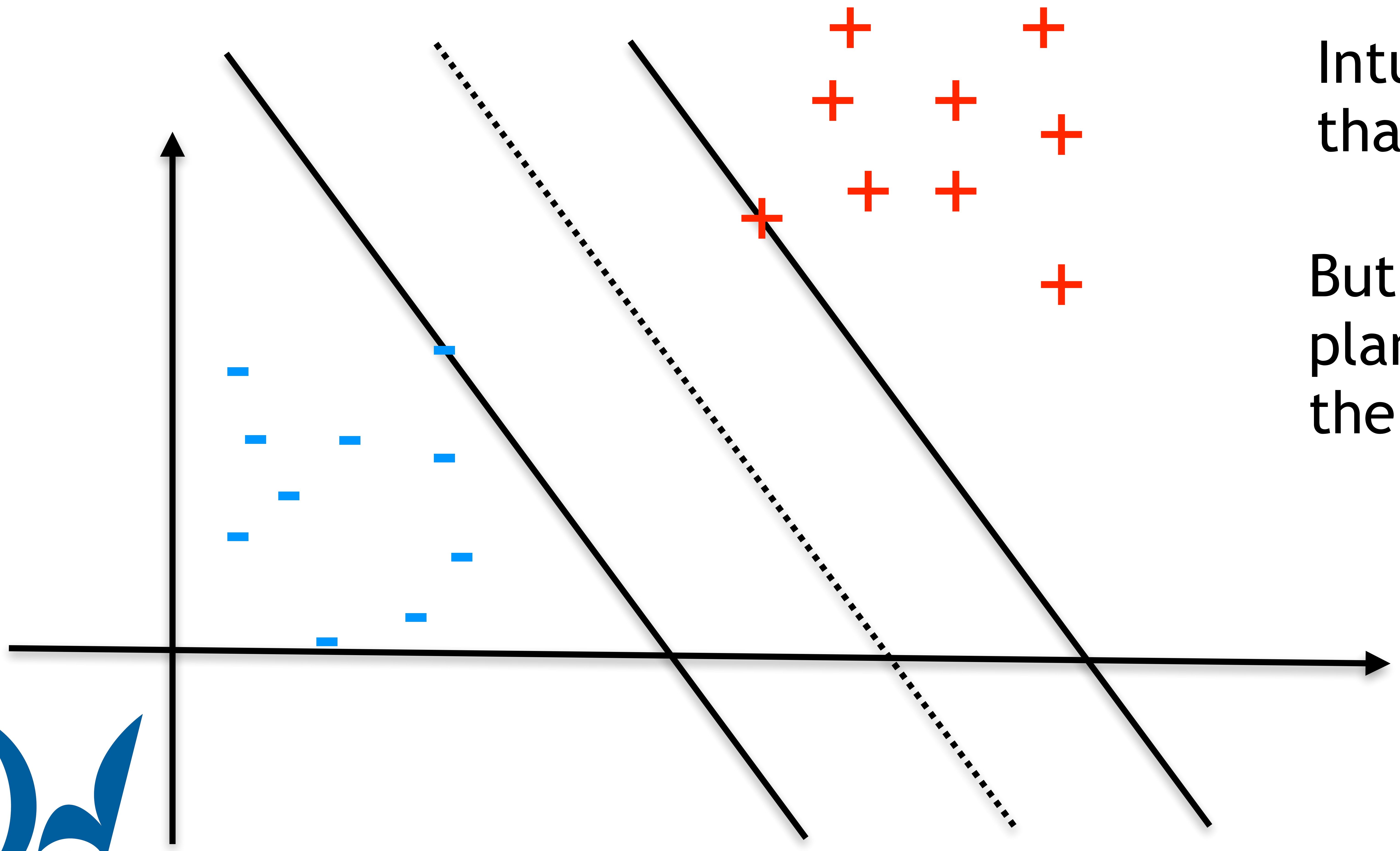
Support vector machines



Intuition is to find a vector/plane that separates the two

But also, two support vectors/ planes that define the width of the separation

Support vector machines

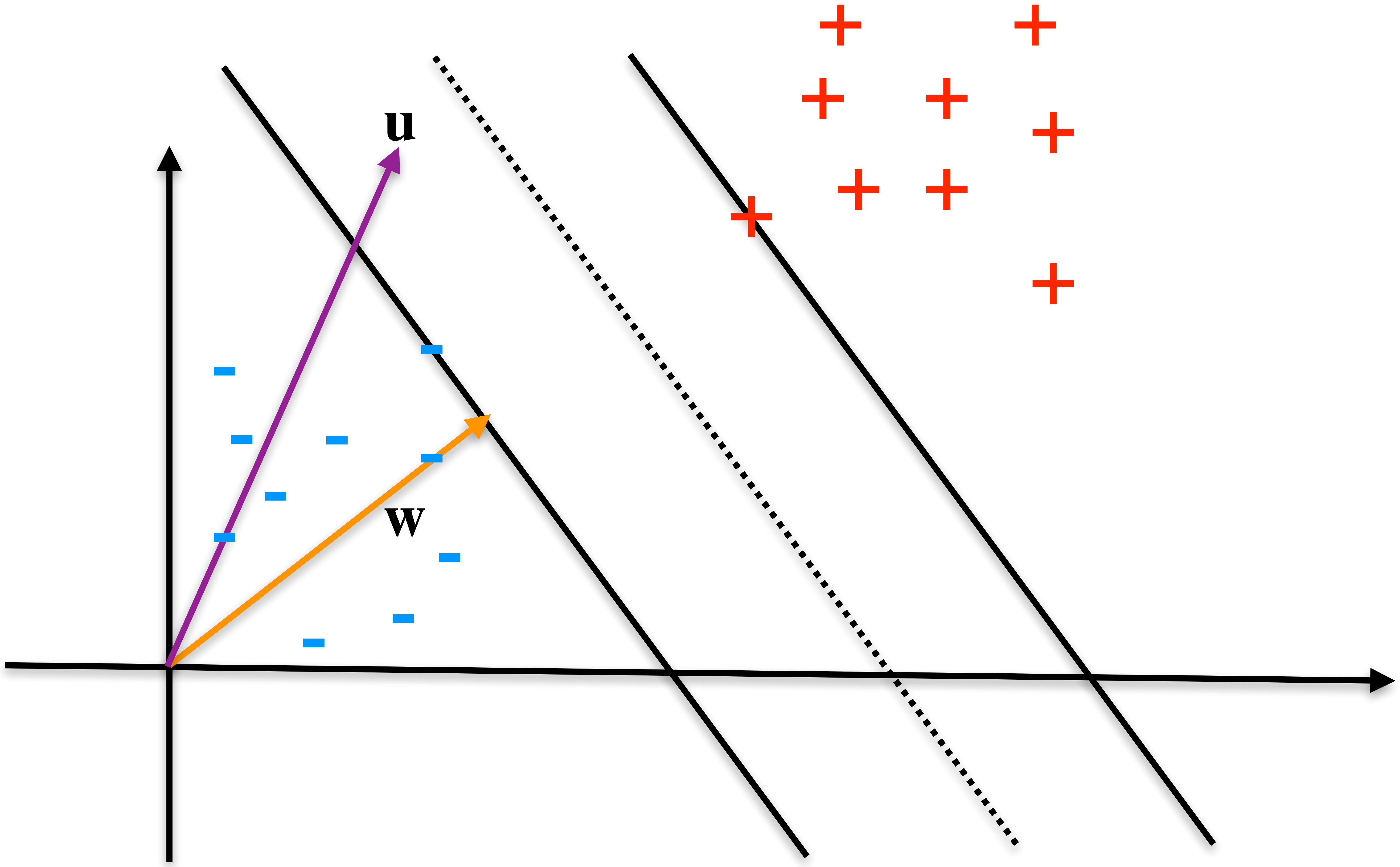


Intuition is to find a vector/plane that separates the two

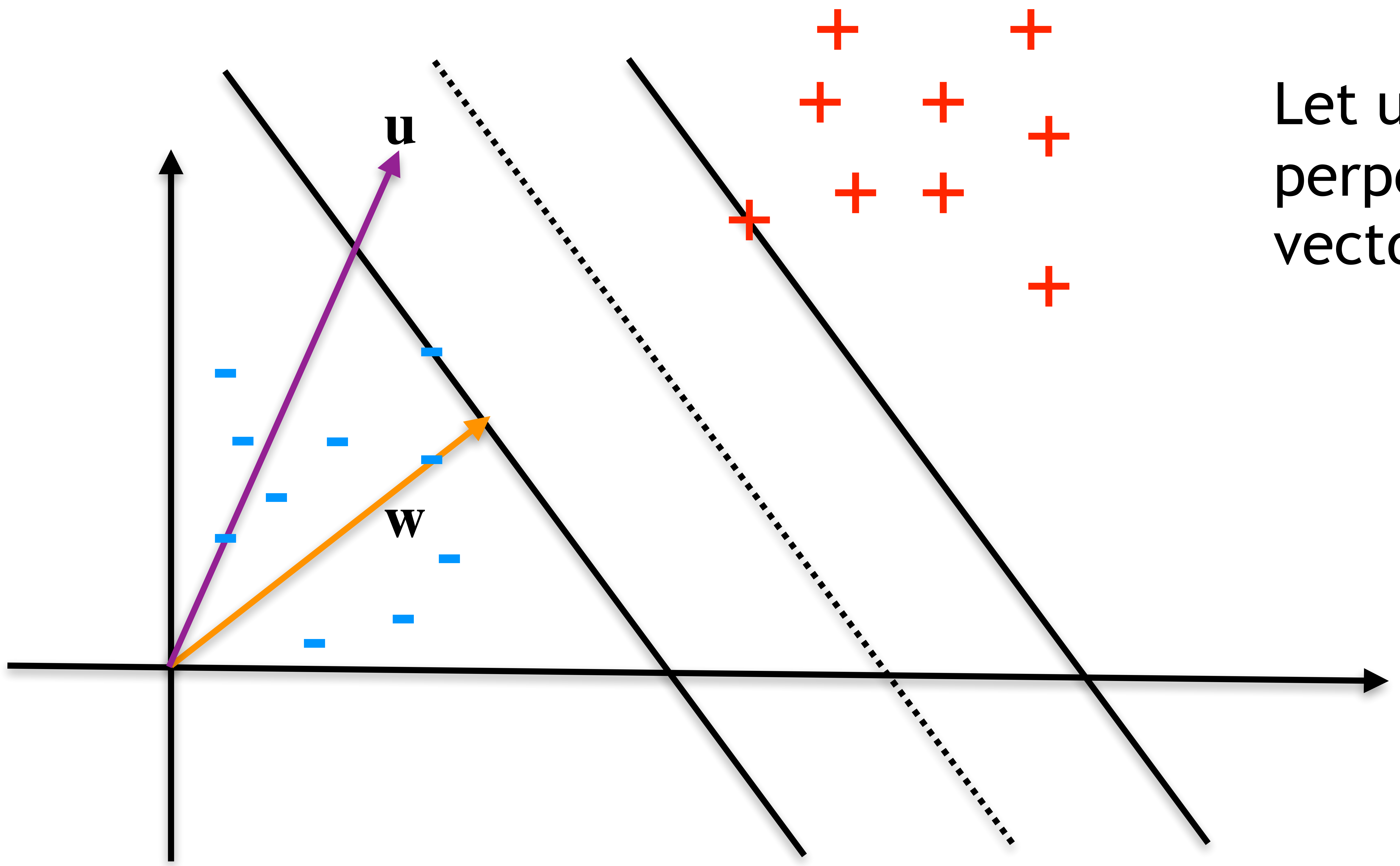
But also, two support vectors/ planes that define the width of the separation

This width should be as big as possible!

Support vector machines

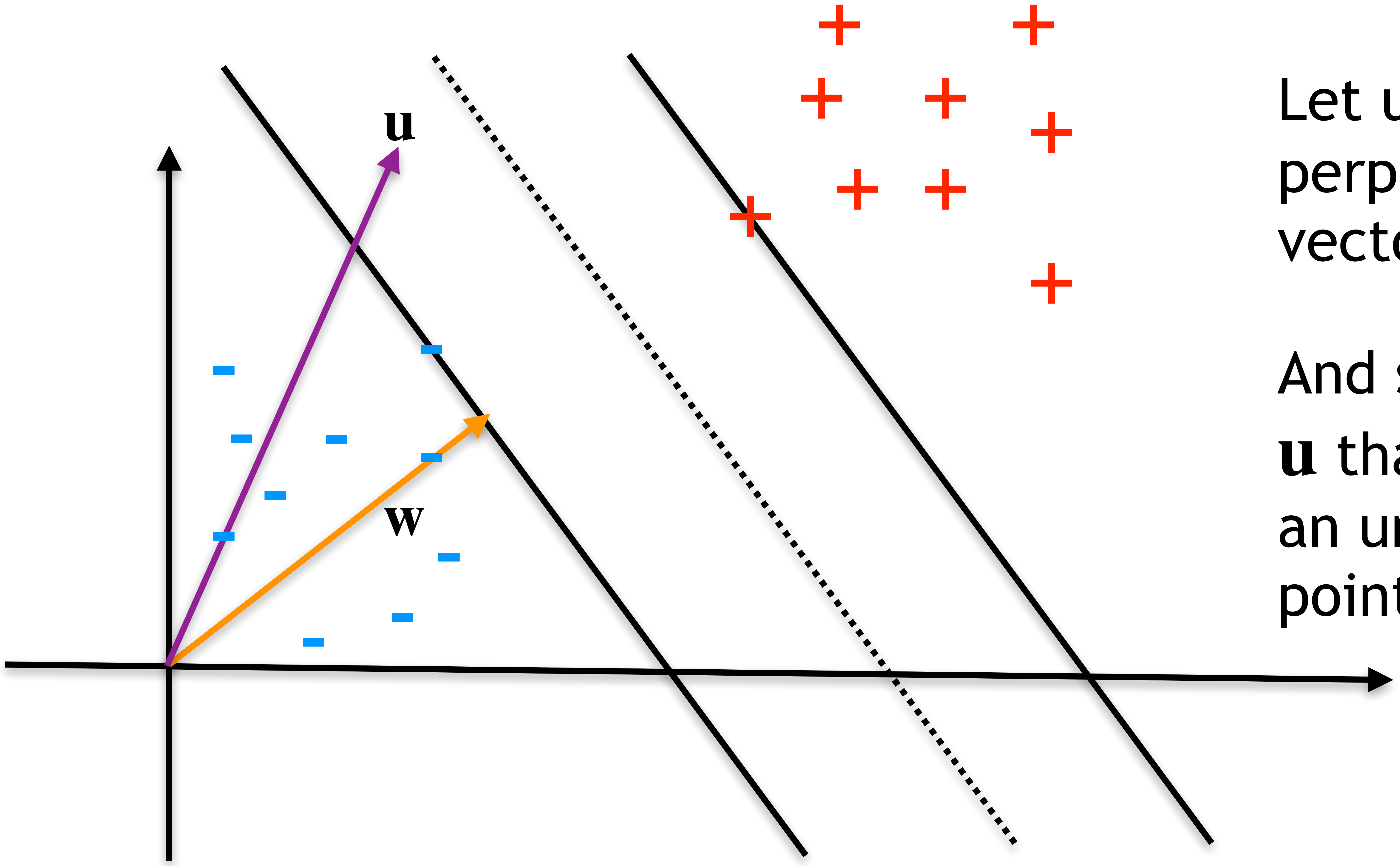


Support vector machines



Let us consider a vector $\mathbf{w}$ perpendicular to the vectors/planes

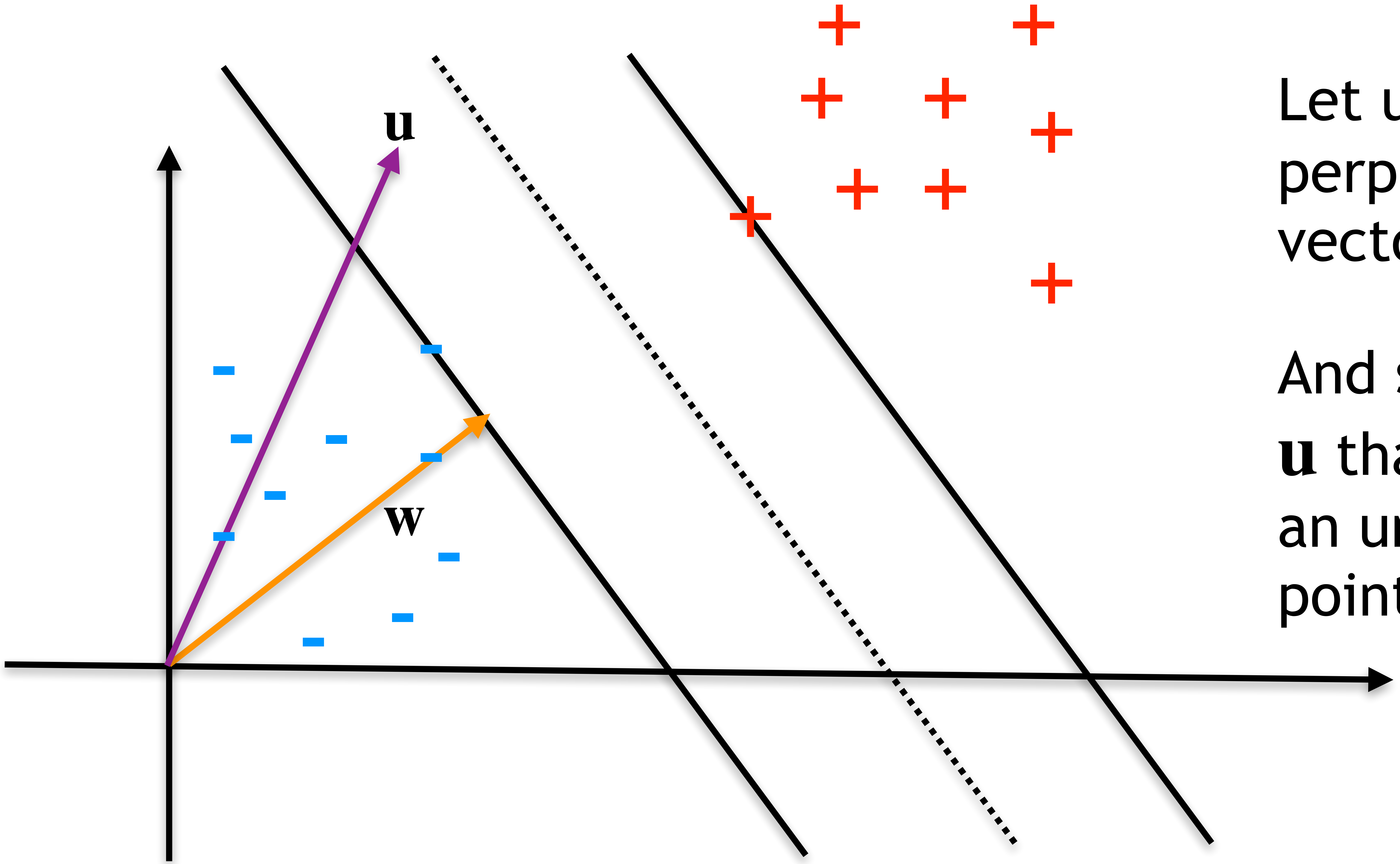
Support vector machines



Let us consider a vector $\mathbf{w}$ perpendicular to the vectors/planes

And some unknown vector $\mathbf{u}$ that we can consider as an unknown (unlabelled) point

Support vector machines

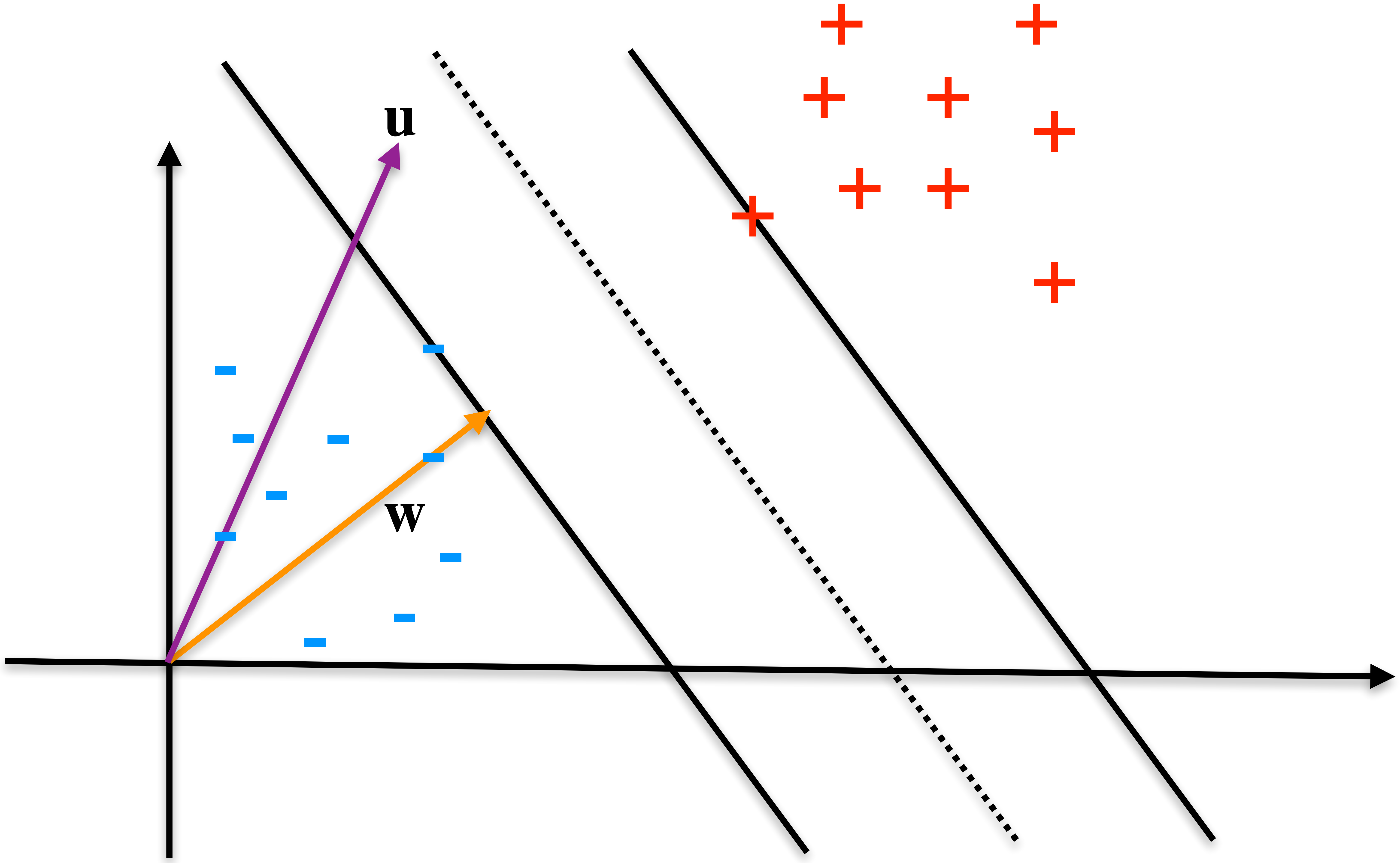


Let us consider a vector $\mathbf{w}$ perpendicular to the vectors/planes

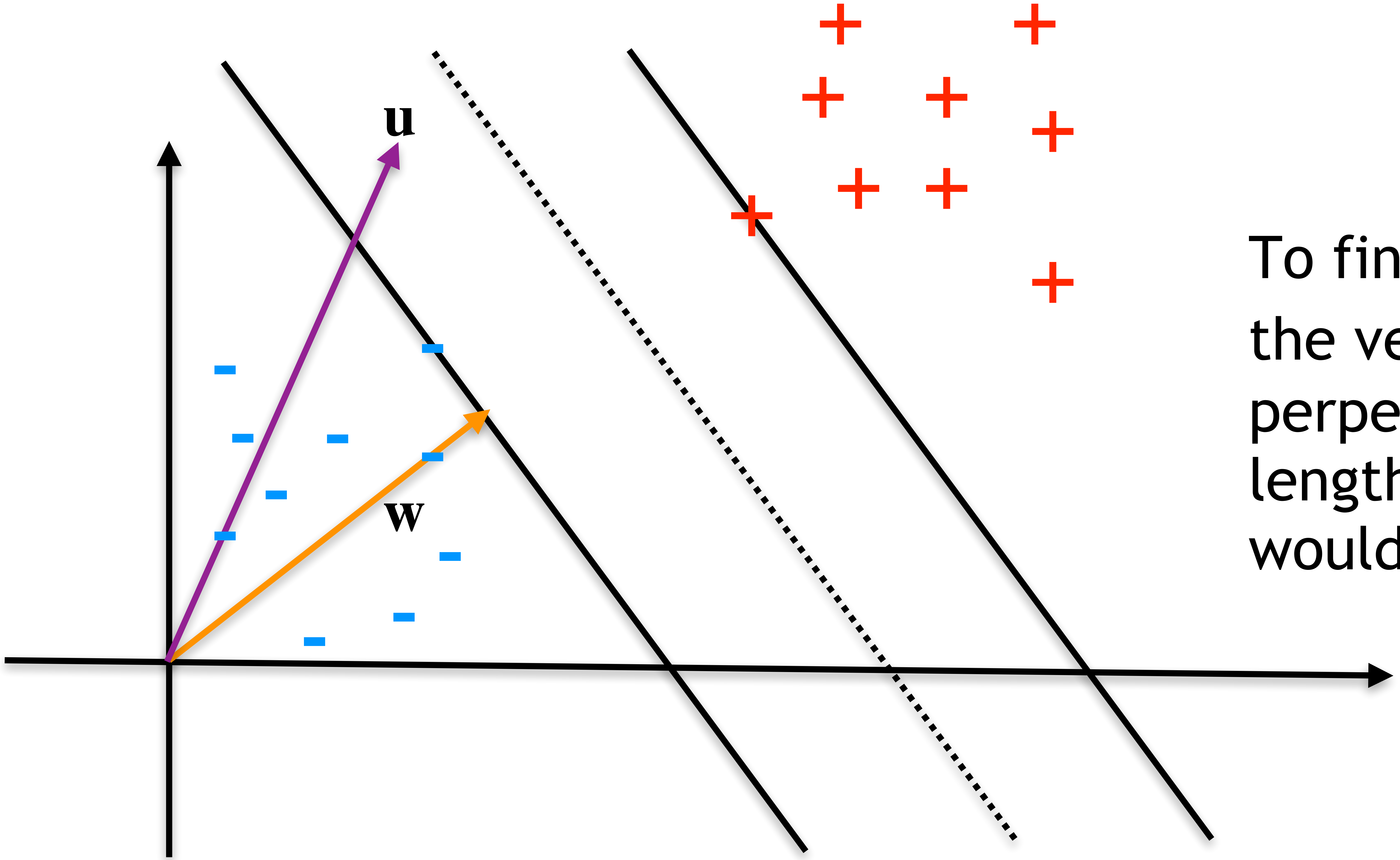
And some unknown vector $\mathbf{u}$ that we can consider as an unknown (unlabelled) point

We want to find out whether the point that defines the vector is on the left or right side of the central plane/vector (key for classification!)

Support vector machines

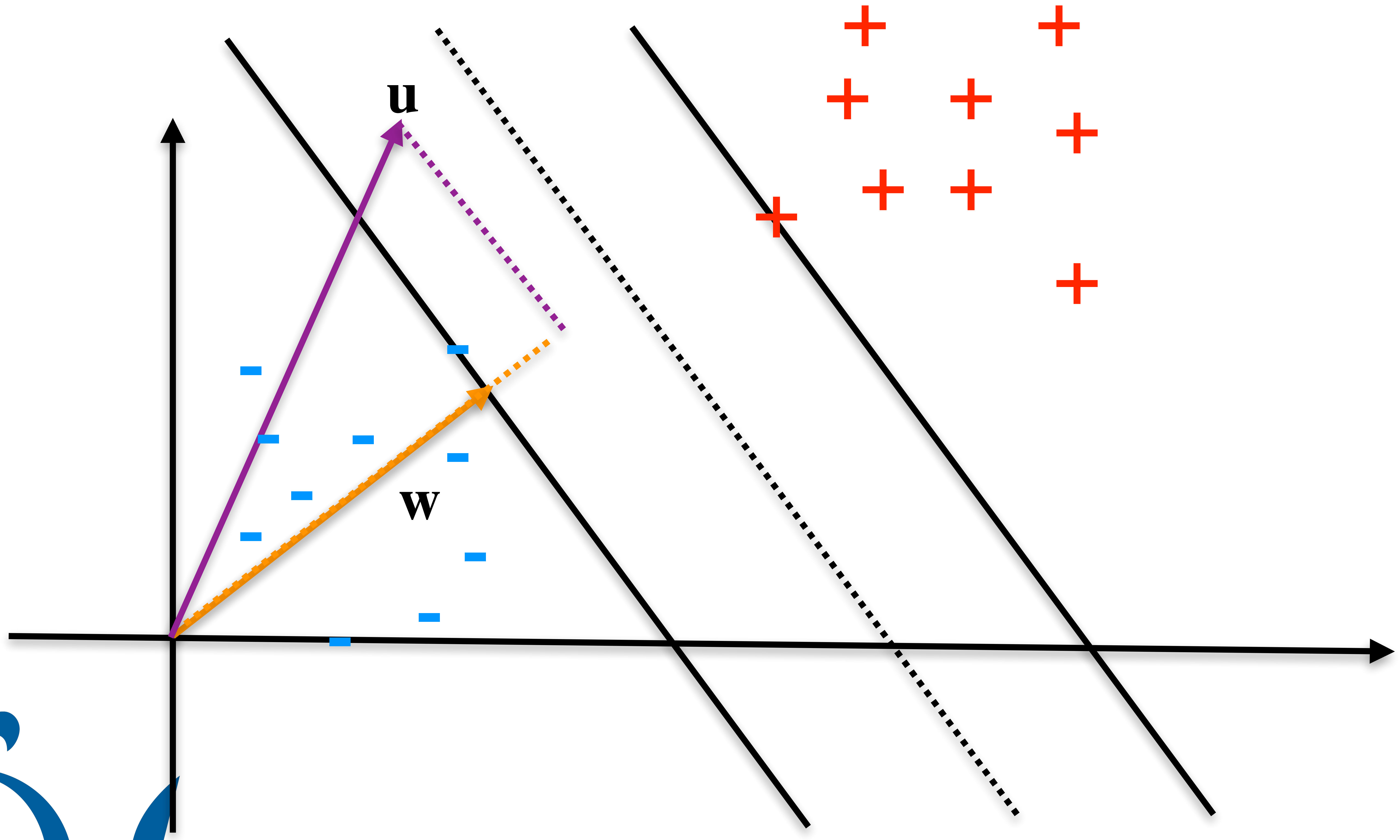


Support vector machines

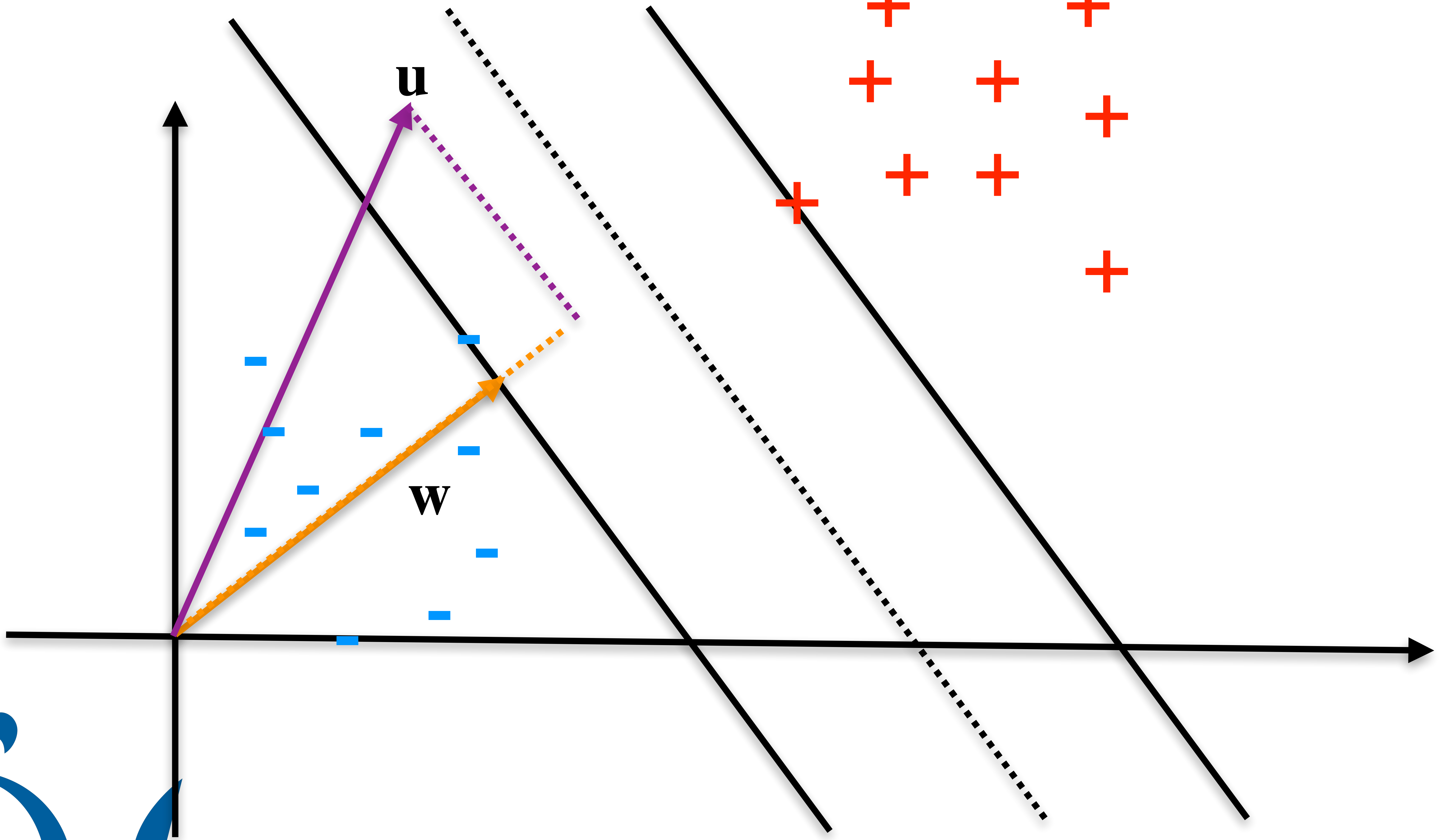


To find out, we can project the vector $\mathbf{u}$ to the perpendicular and the length of the projection would tell us

Support vector machines

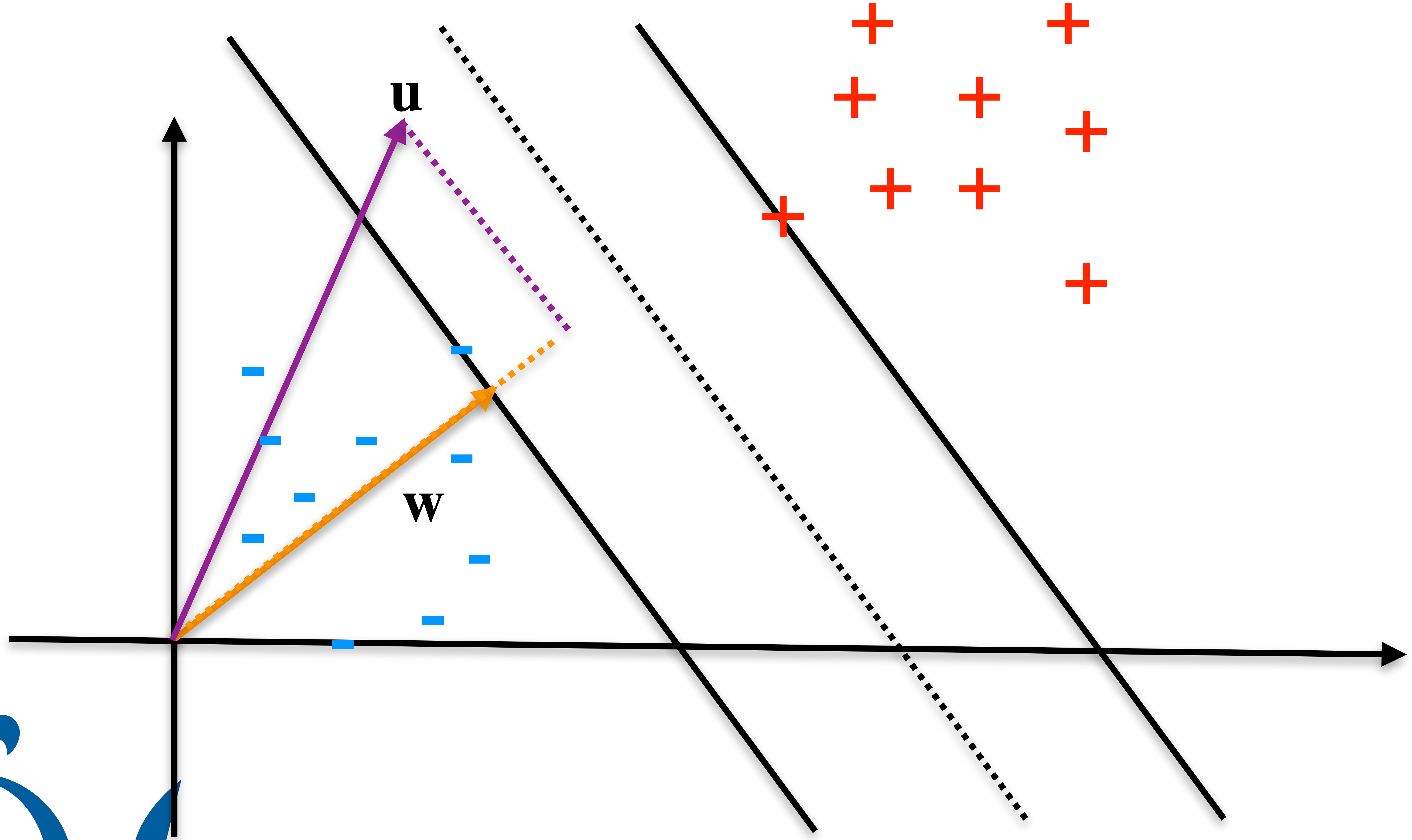


Support vector machines



How do we compute projections of vectors?

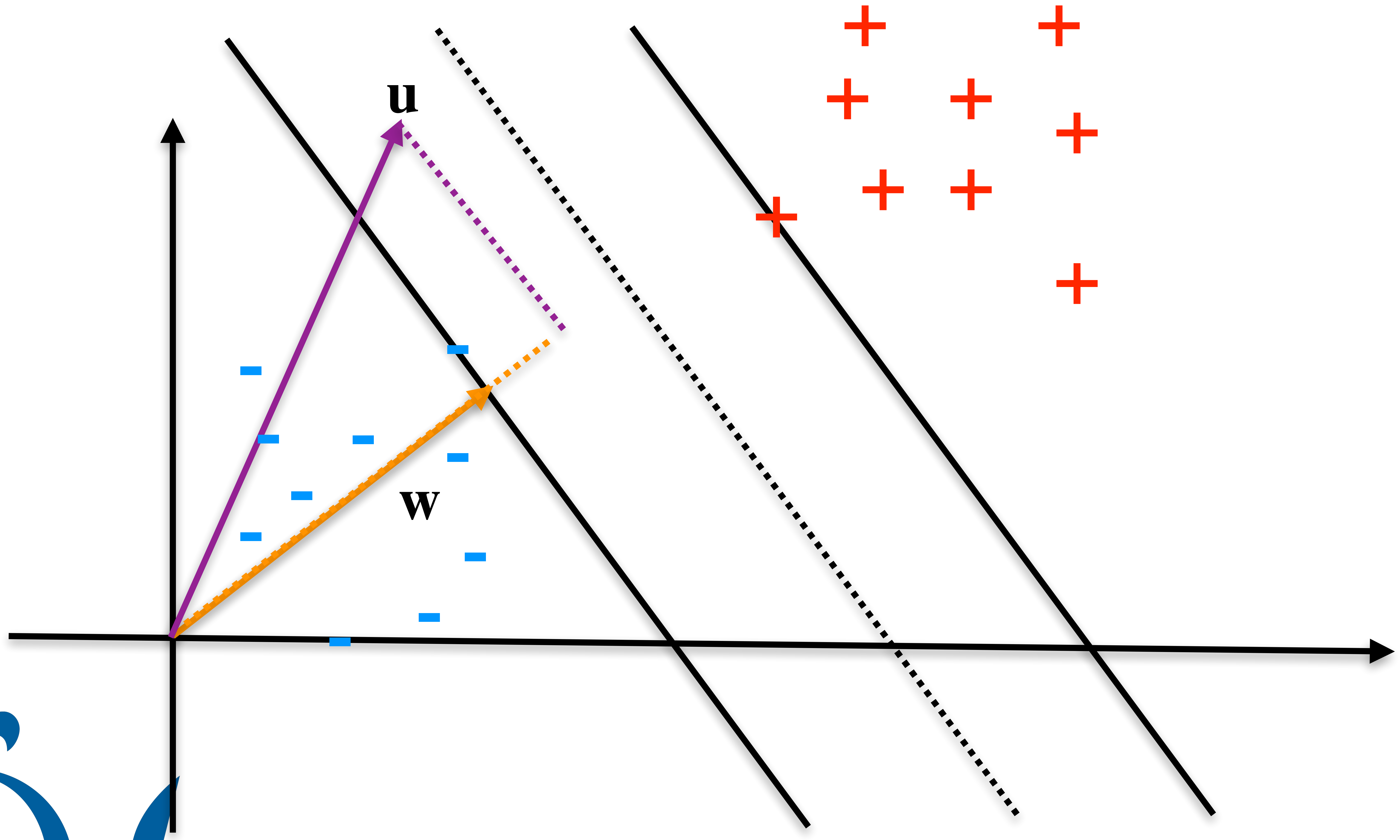
Support vector machines



How do we compute projections of vectors?

$$\langle \mathbf{w}, \mathbf{u} \rangle = \|\mathbf{w}\| \|\mathbf{u}\| \cos\theta$$

Support vector machines

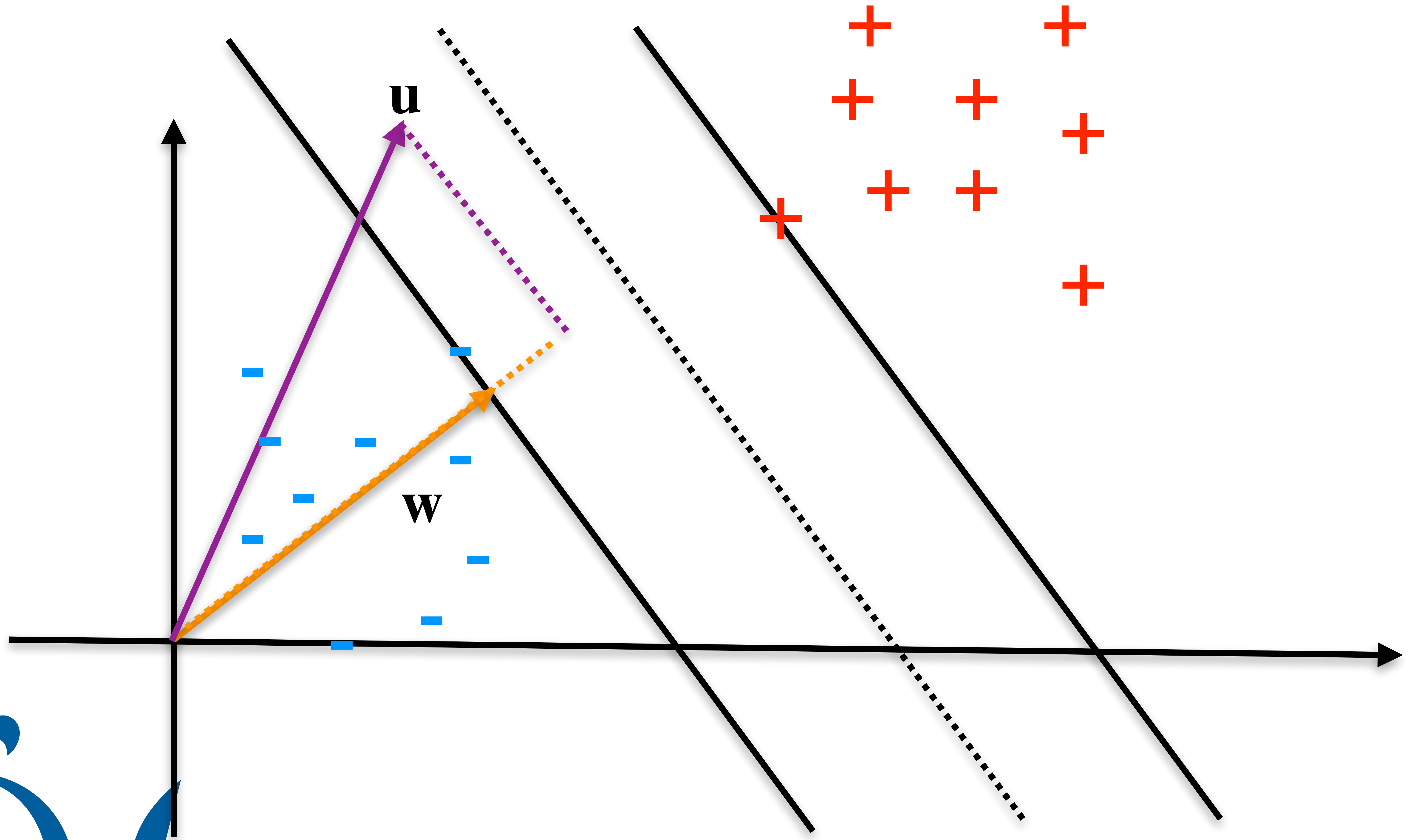


How do we compute projections of vectors?

$$\langle \mathbf{w}, \mathbf{u} \rangle = \|\mathbf{w}\| \|\mathbf{u}\| \cos \theta$$

$$\frac{\langle \mathbf{w}, \mathbf{u} \rangle}{\|\mathbf{w}\|} = \|\mathbf{u}\| \cos \theta$$

Support vector machines



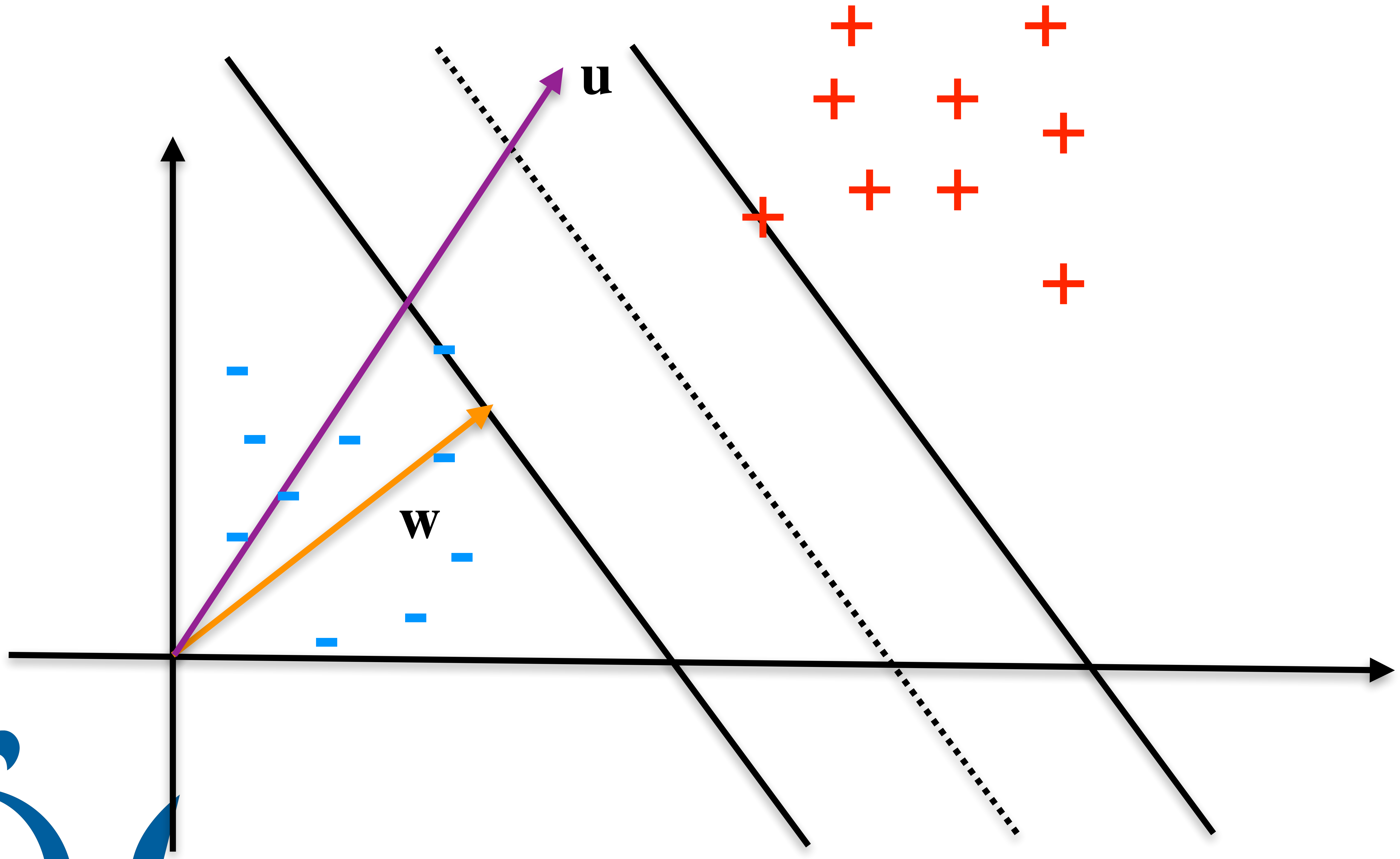
How do we compute projections of vectors?

$$\langle \mathbf{w}, \mathbf{u} \rangle = \|\mathbf{w}\| \|\mathbf{u}\| \cos\theta$$

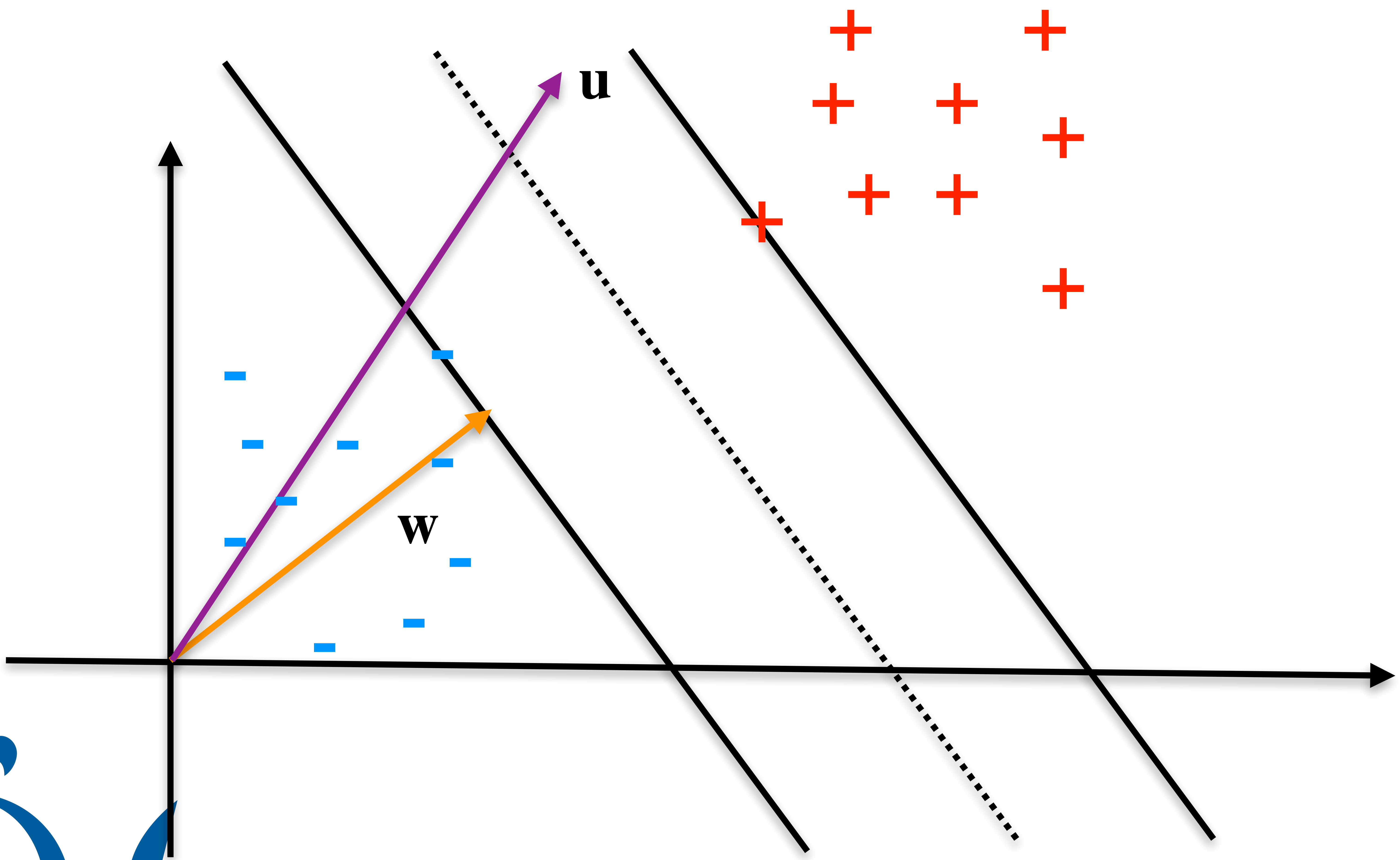
$$\frac{\langle \mathbf{w}, \mathbf{u} \rangle}{\|\mathbf{w}\|} = \|\mathbf{u}\| \cos\theta$$

Projection of u in w

Support vector machines

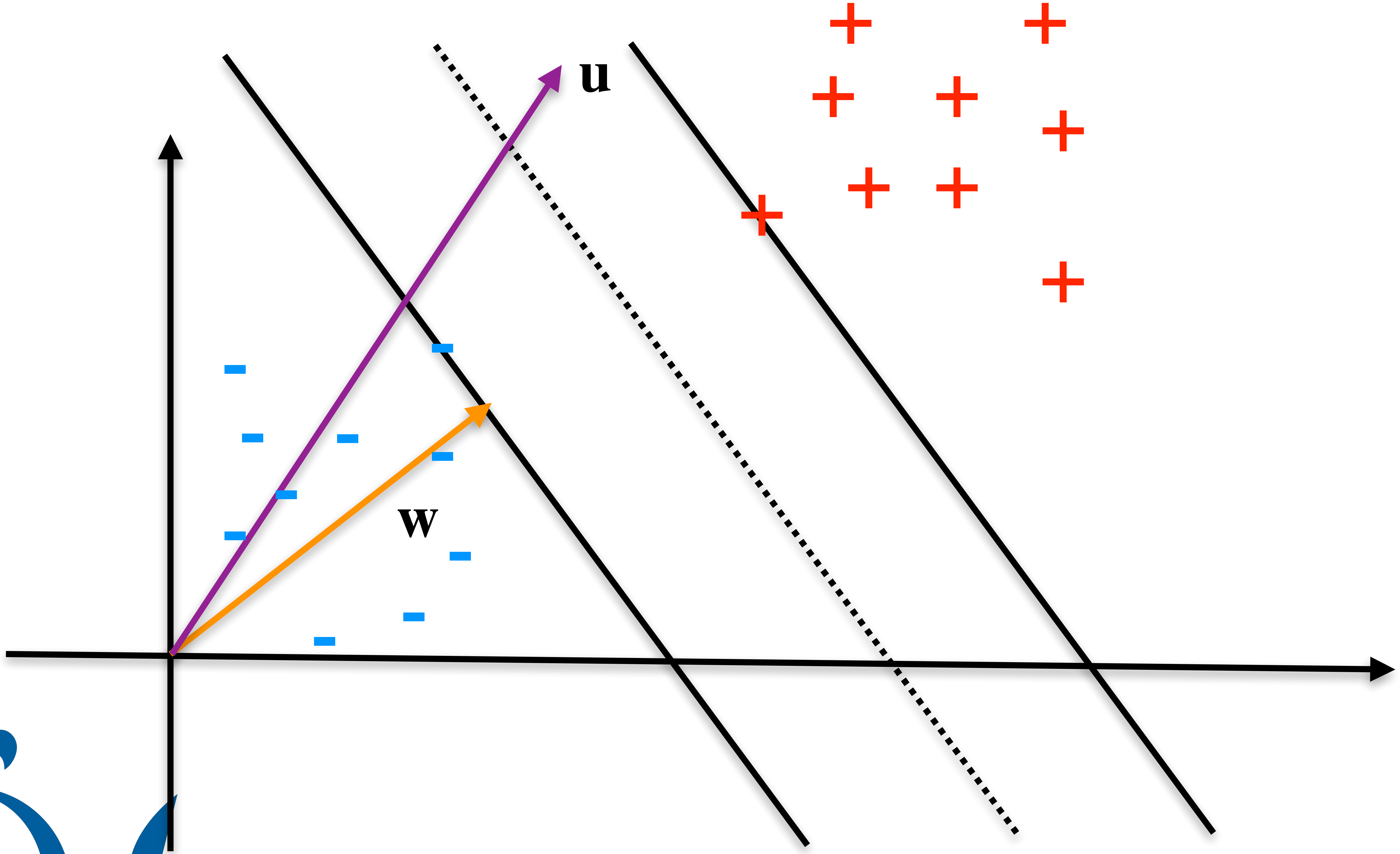


Support vector machines



So, we can say

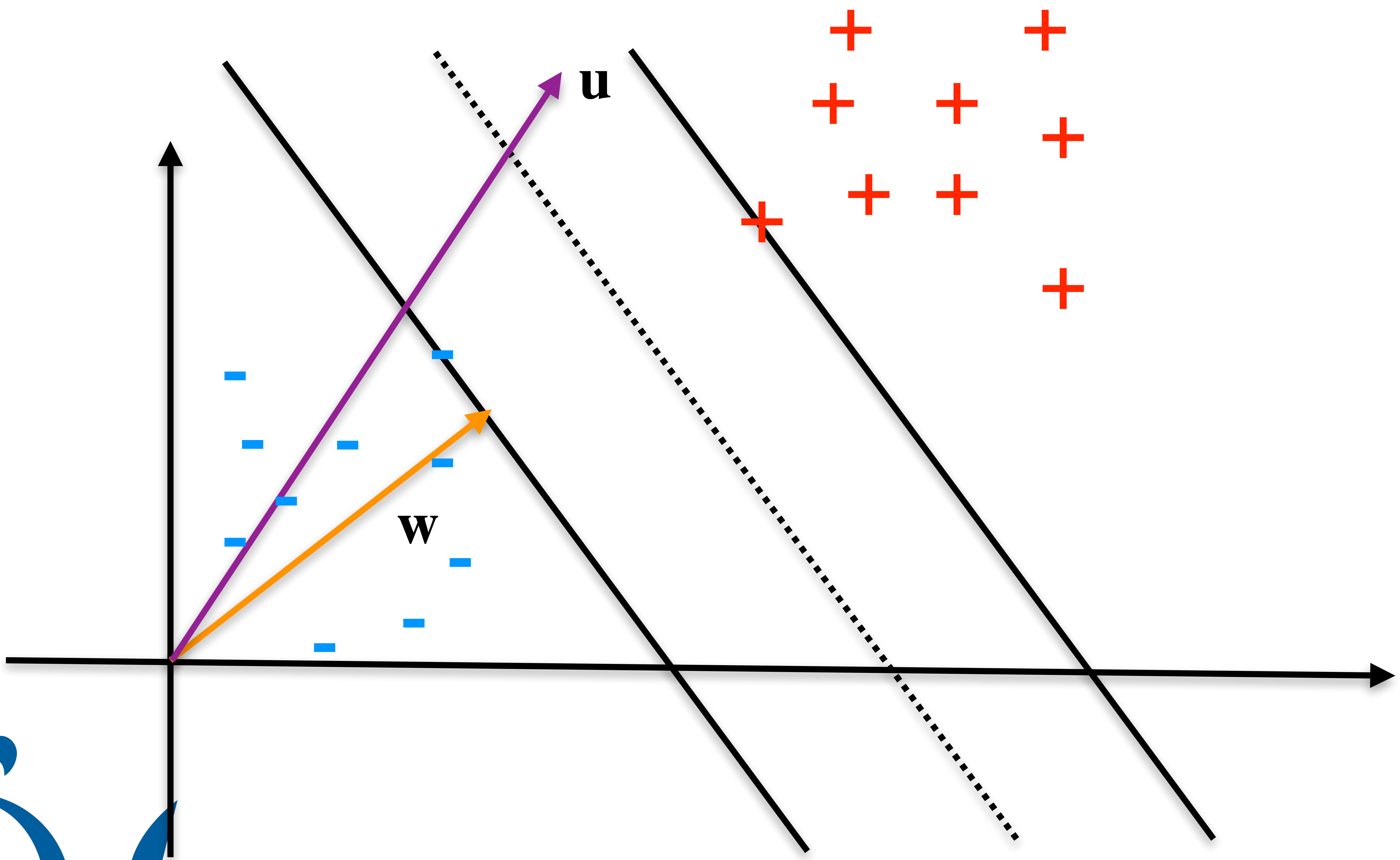
Support vector machines



So, we can say

$$\langle \mathbf{w}, \mathbf{u} \rangle \geq r$$

Support vector machines

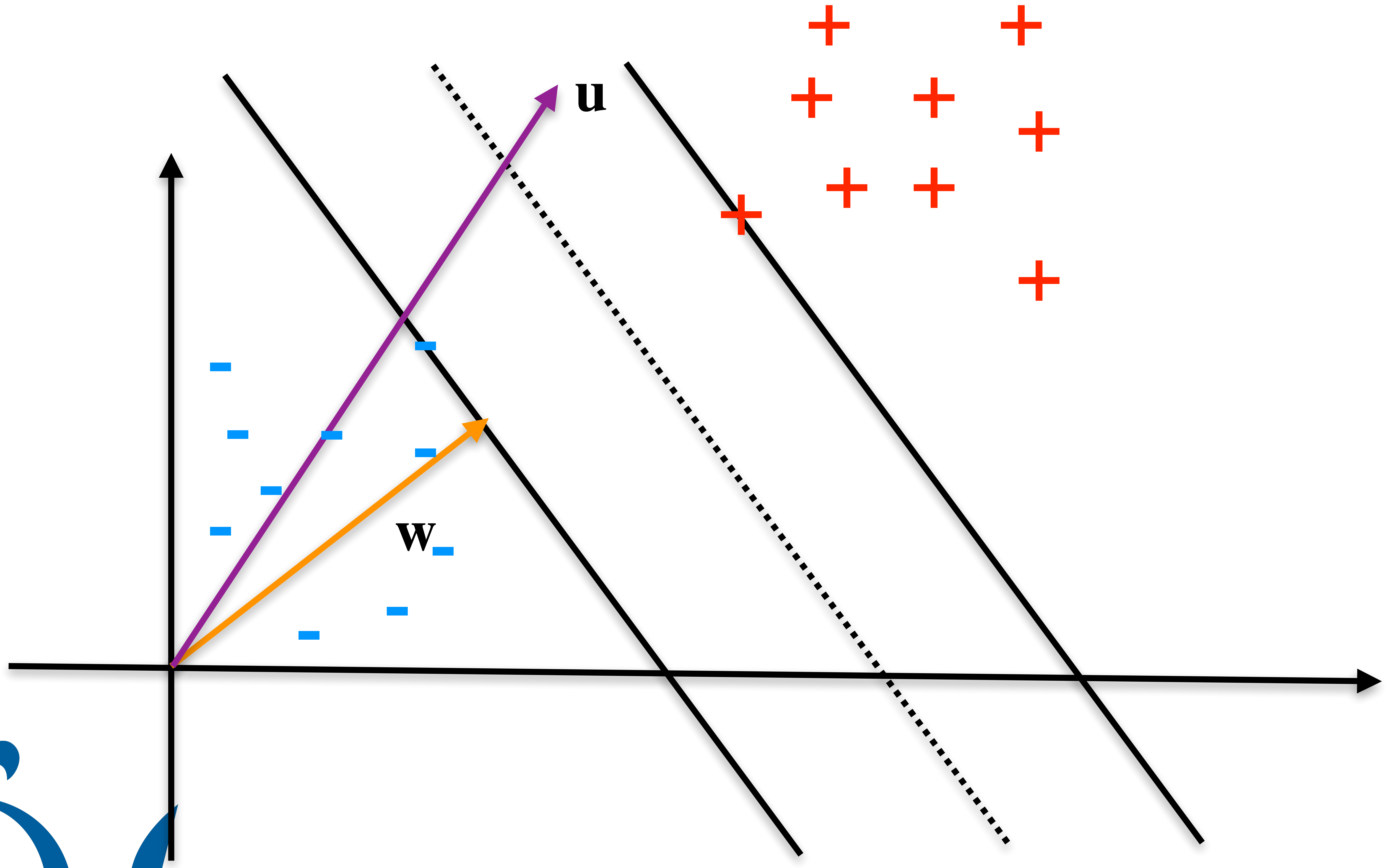


So, we can say

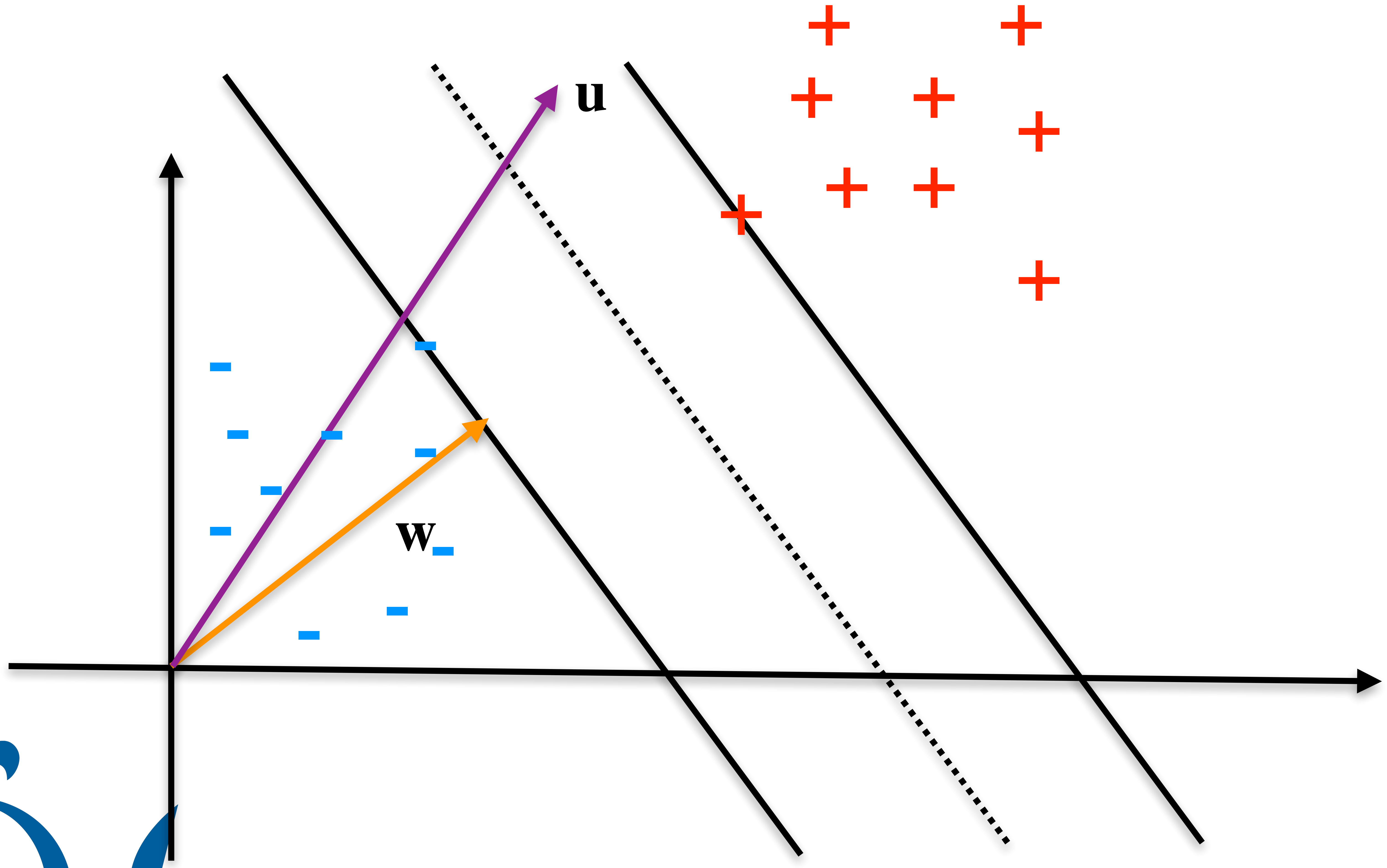
$$\langle \mathbf{w}, \mathbf{u} \rangle \geq r$$

and if r is large enough so that the projection is past the central line we can classify the point u with the “stars”

Support vector machines

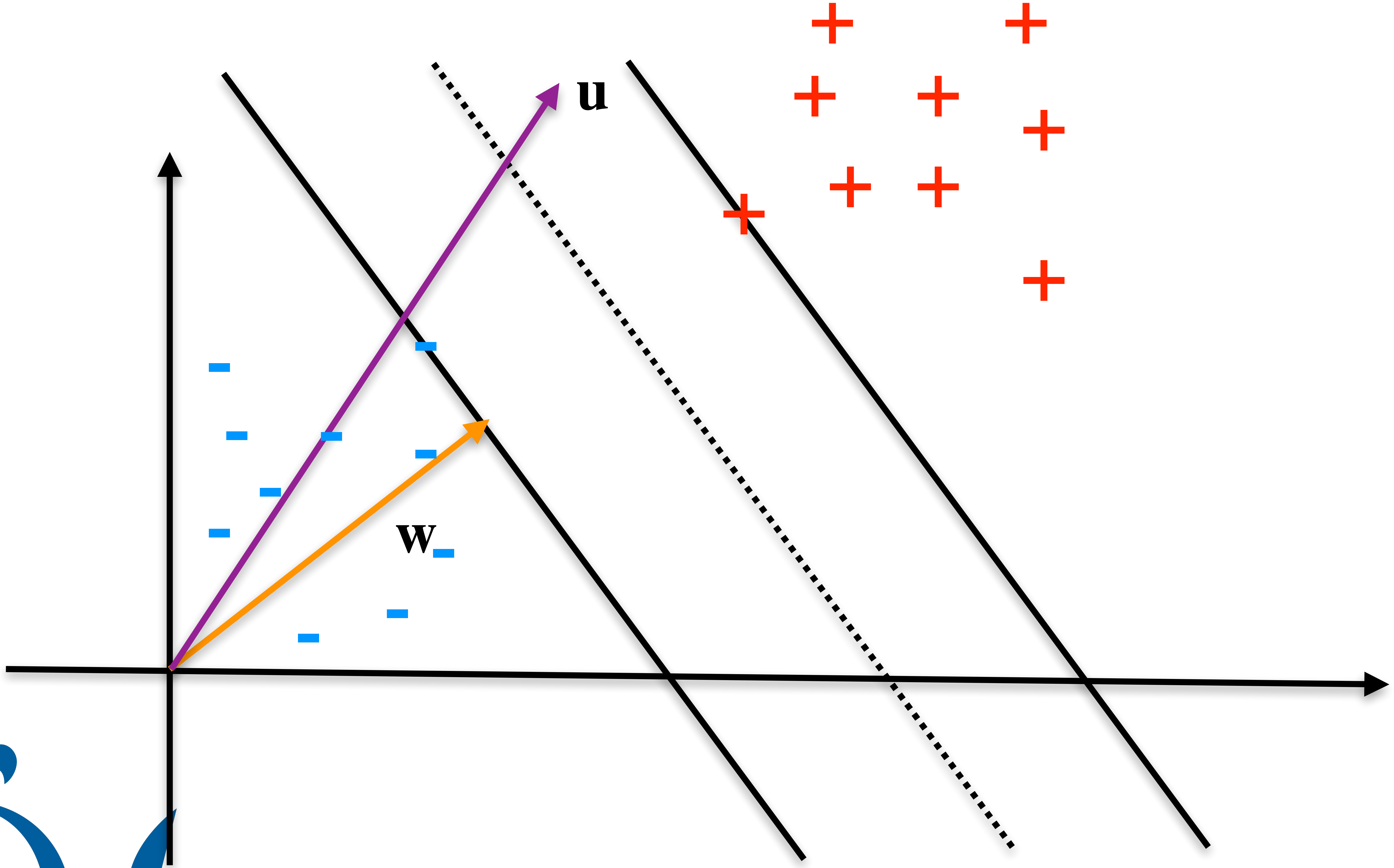


Support vector machines



More in general we can say, if

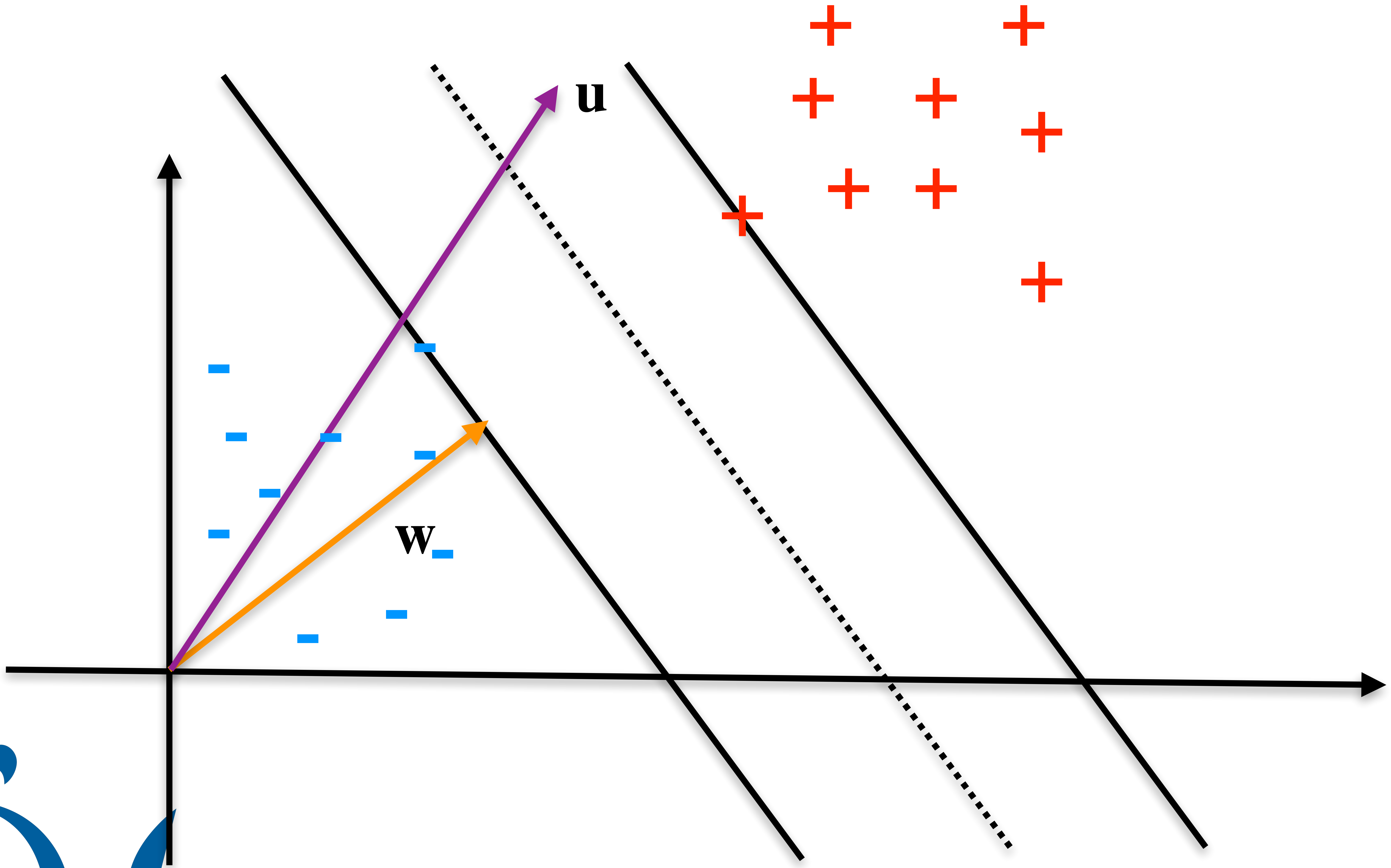
Support vector machines



More in general we can say, if

$$\langle \mathbf{w}, \mathbf{u} \rangle + b \geq 0$$

Support vector machines

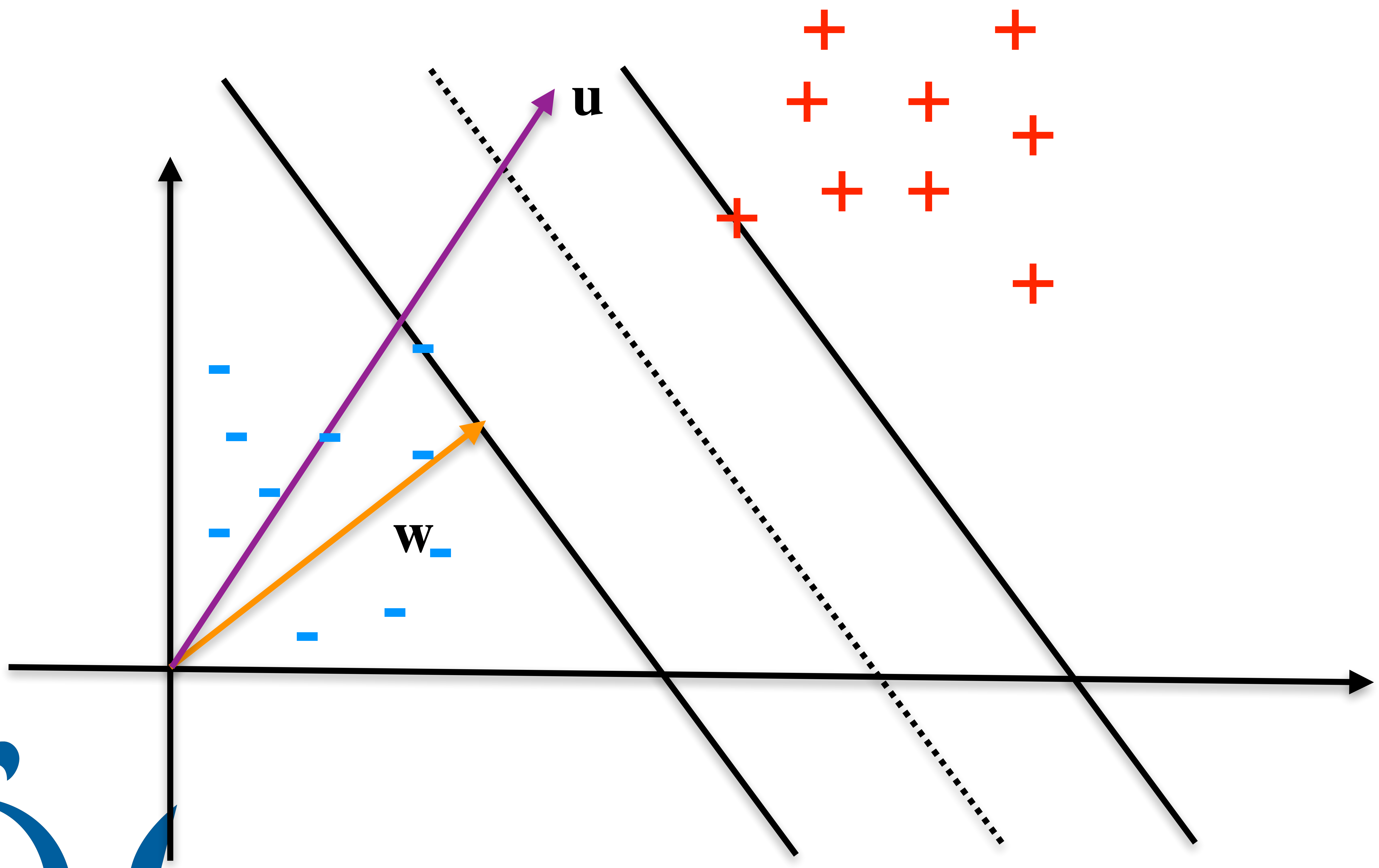


More in general we can say, if

$$\langle \mathbf{w}, \mathbf{u} \rangle + b \geq 0$$

Then the point u is a star

Support vector machines



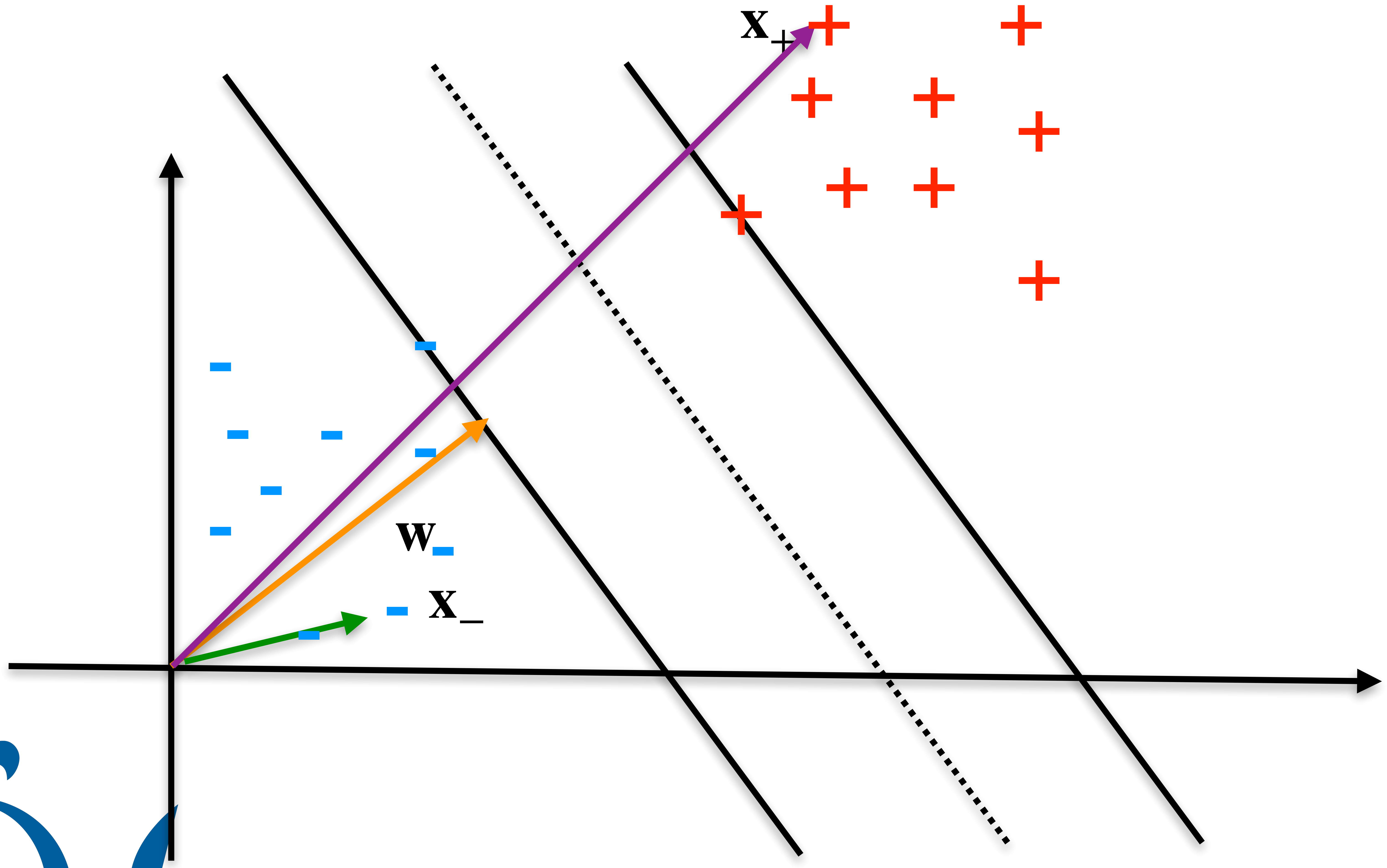
More in general we can say, if

$$\langle \mathbf{w}, \mathbf{u} \rangle + b \geq 0$$

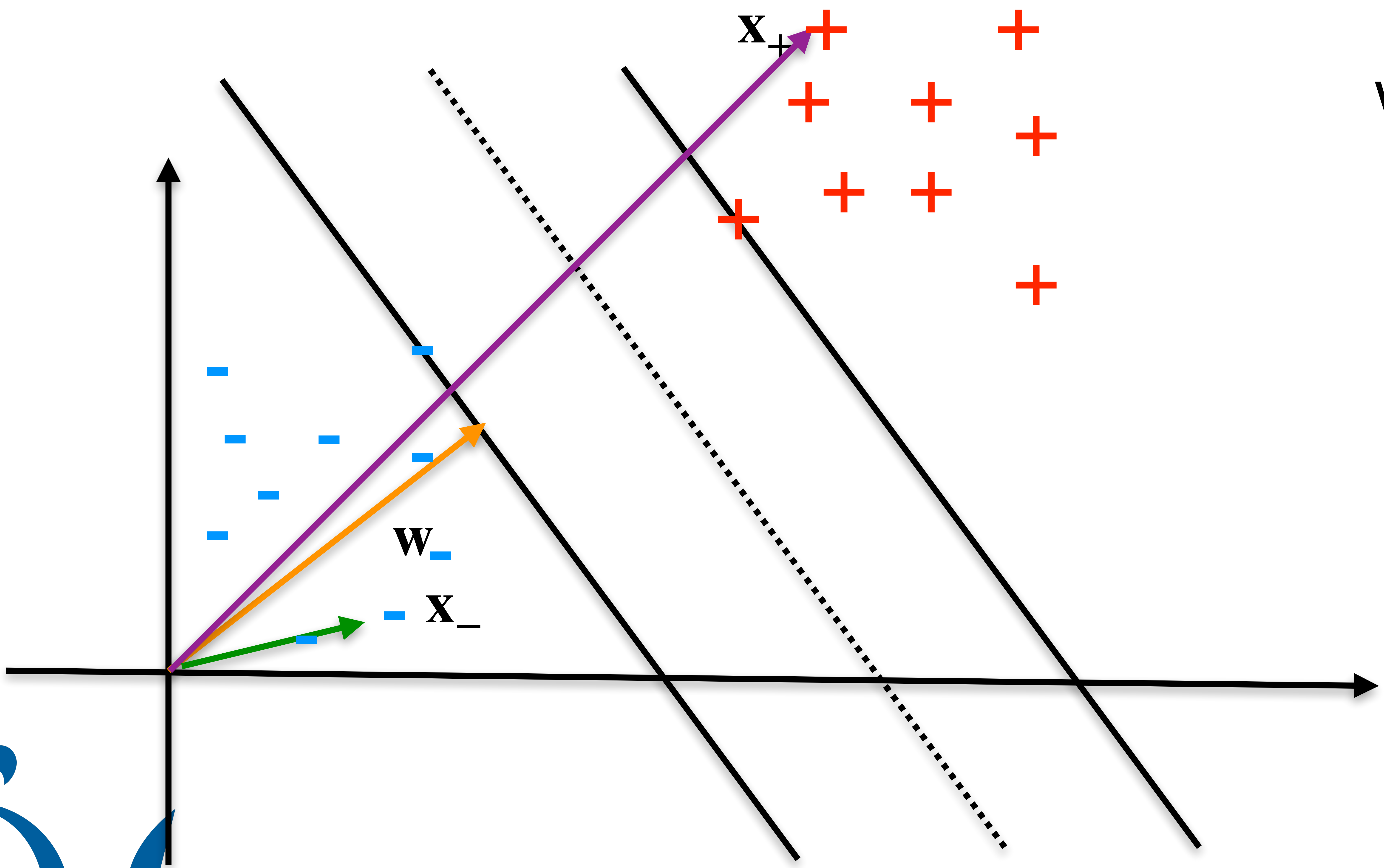
Then the point u is a star

We need to find w and b! How can we do that

Support vector machines

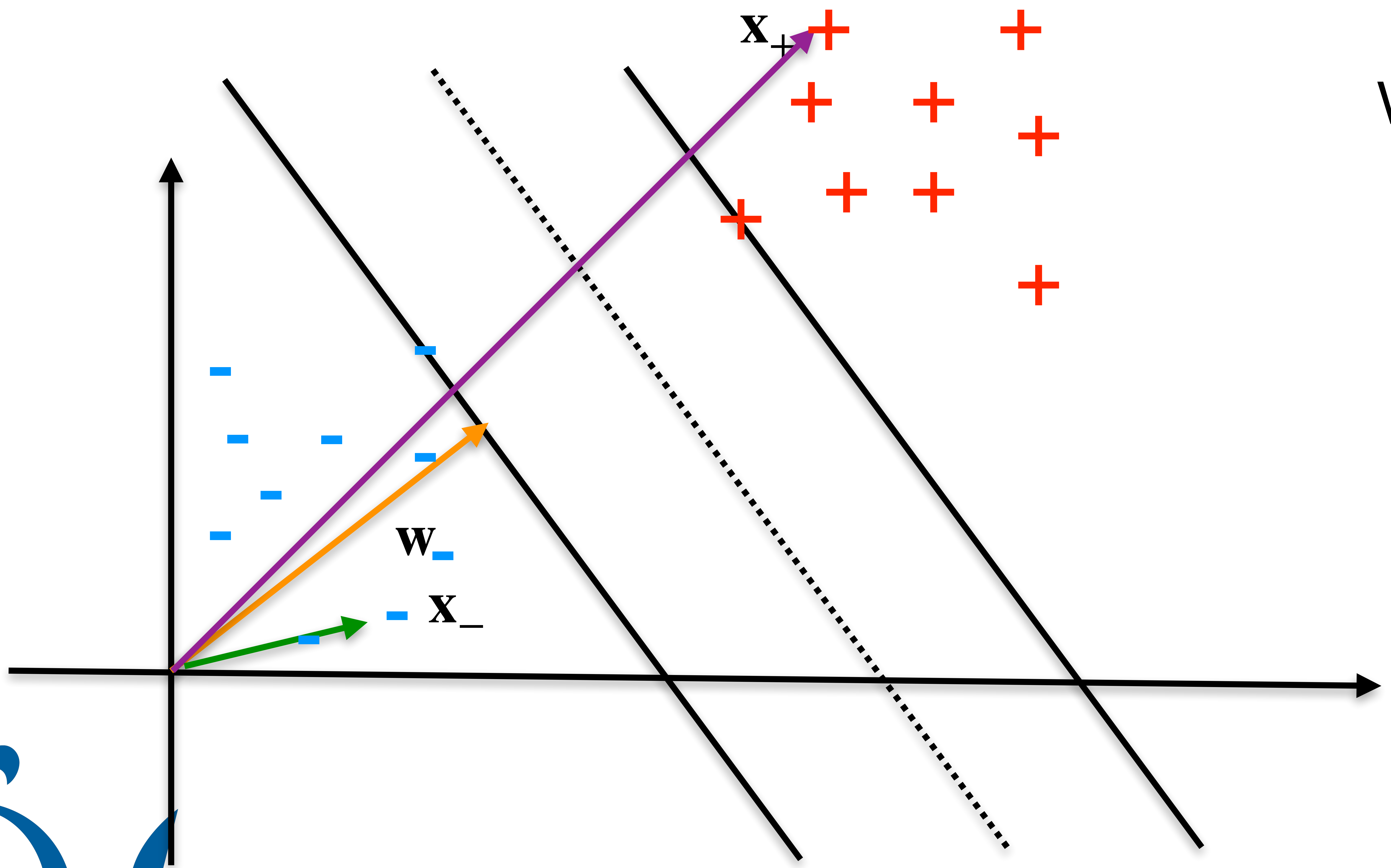


Support vector machines



We need to set more conditions

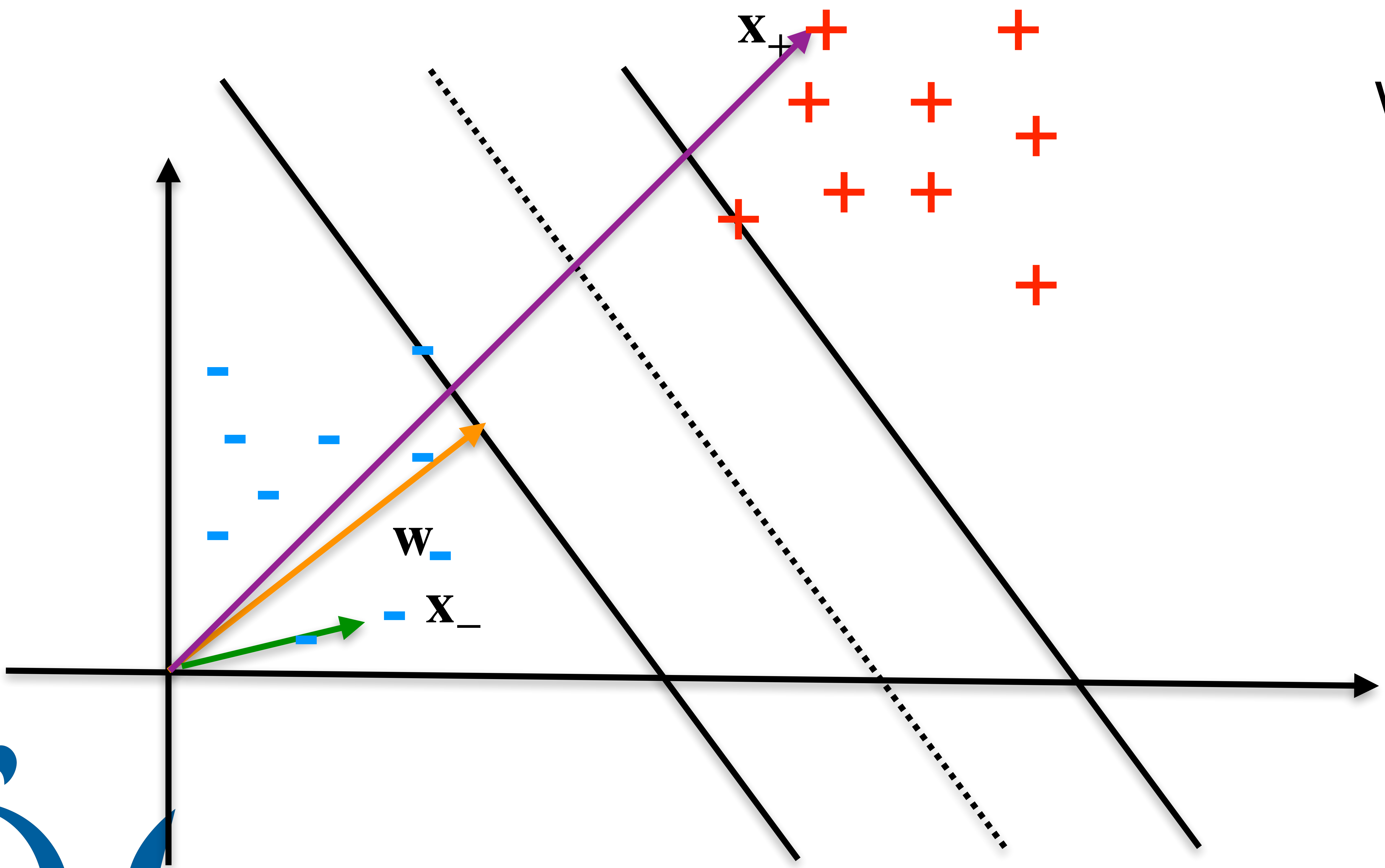
Support vector machines



We need to set more conditions

$$\langle \mathbf{w}, \mathbf{x}_+ \rangle + b \geq 1$$

Support vector machines



We need to set more conditions

$$\langle \mathbf{w}, \mathbf{x}_+ \rangle + b \geq 1$$

$$\langle \mathbf{w}, \mathbf{x}_- \rangle + b \leq -1$$

Support vector machines

Let us assume that the output (classification) variables are defined as

$$y_i = 1 \text{ for + samples}$$
$$y_i = -1 \text{ for - samples}$$

Support vector machines

Let us assume that the output (classification) variables are defined as

$$y_i = 1 \text{ for + samples}$$
$$y_i = -1 \text{ for - samples}$$

If we multiply the two conditions by ys we get the same expression

Support vector machines

Let us assume that the output (classification) variables are defined as

$$y_i = 1 \text{ for + samples}$$
$$y_i = -1 \text{ for - samples}$$

If we multiply the two conditions by ys we get the same expression

$$y_i \left(\langle \mathbf{x}_i, \mathbf{w} \rangle + b \right) \geq 1$$

Support vector machines

Let us assume that the output (classification) variables are defined as

$$y_i = 1 \text{ for + samples}$$
$$y_i = -1 \text{ for - samples}$$

If we multiply the two conditions by ys we get the same expression

$$y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) \geq 1 \quad \rightarrow y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 \geq 0$$

Support vector machines

Let us assume that the output (classification) variables are defined as

$$y_i = 1 \text{ for + samples}$$
$$y_i = -1 \text{ for - samples}$$

If we multiply the two conditions by ys we get the same expression

$$y_i\left(\langle \mathbf{x}_i, \mathbf{w} \rangle + b\right) \geq 1 \quad \rightarrow y_i\left(\langle \mathbf{x}_i, \mathbf{w} \rangle + b\right) - 1 \geq 0$$

And

Support vector machines

Let us assume that the output (classification) variables are defined as

$y_i = 1$ for + samples
$y_i = -1$ for - samples

If we multiply the two conditions by ys we get the same expression

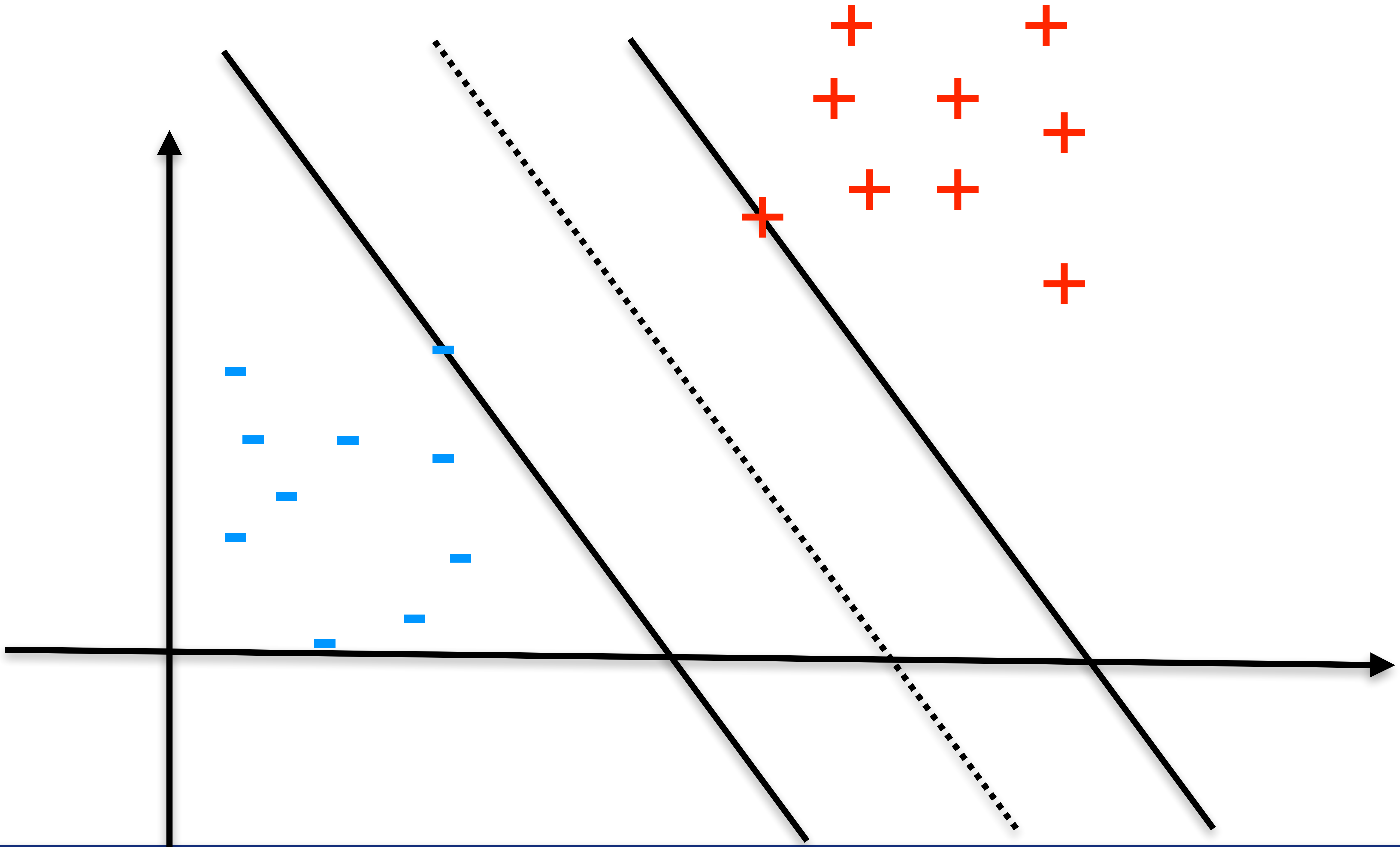
$$y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) \geq 1 \quad \rightarrow y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 \geq 0$$

And

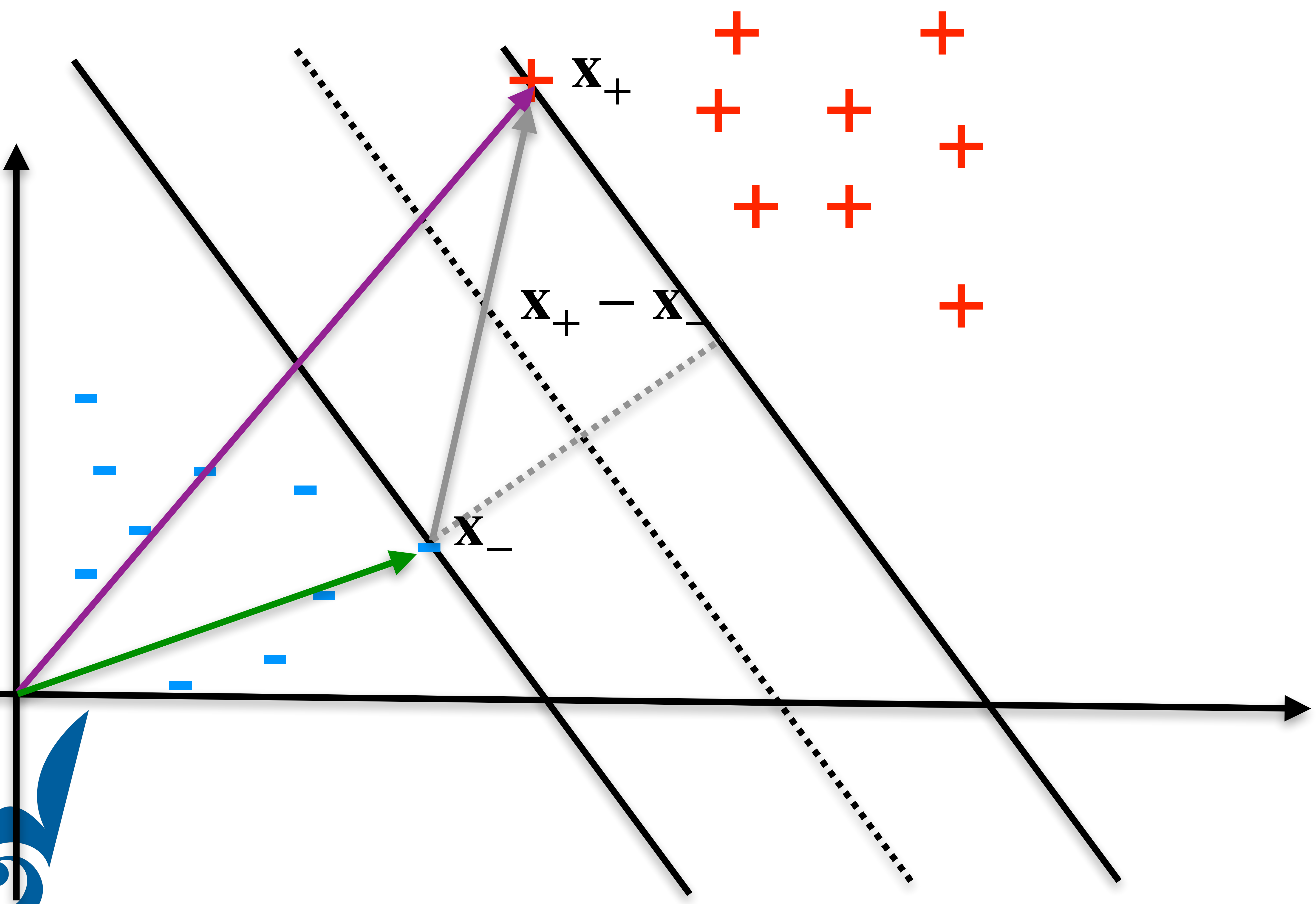
$y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 = 0$ for x_i in the support vectors

Support vector machines

$y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 = 0$ for $\mathbf{x}_i$ in the support vectors

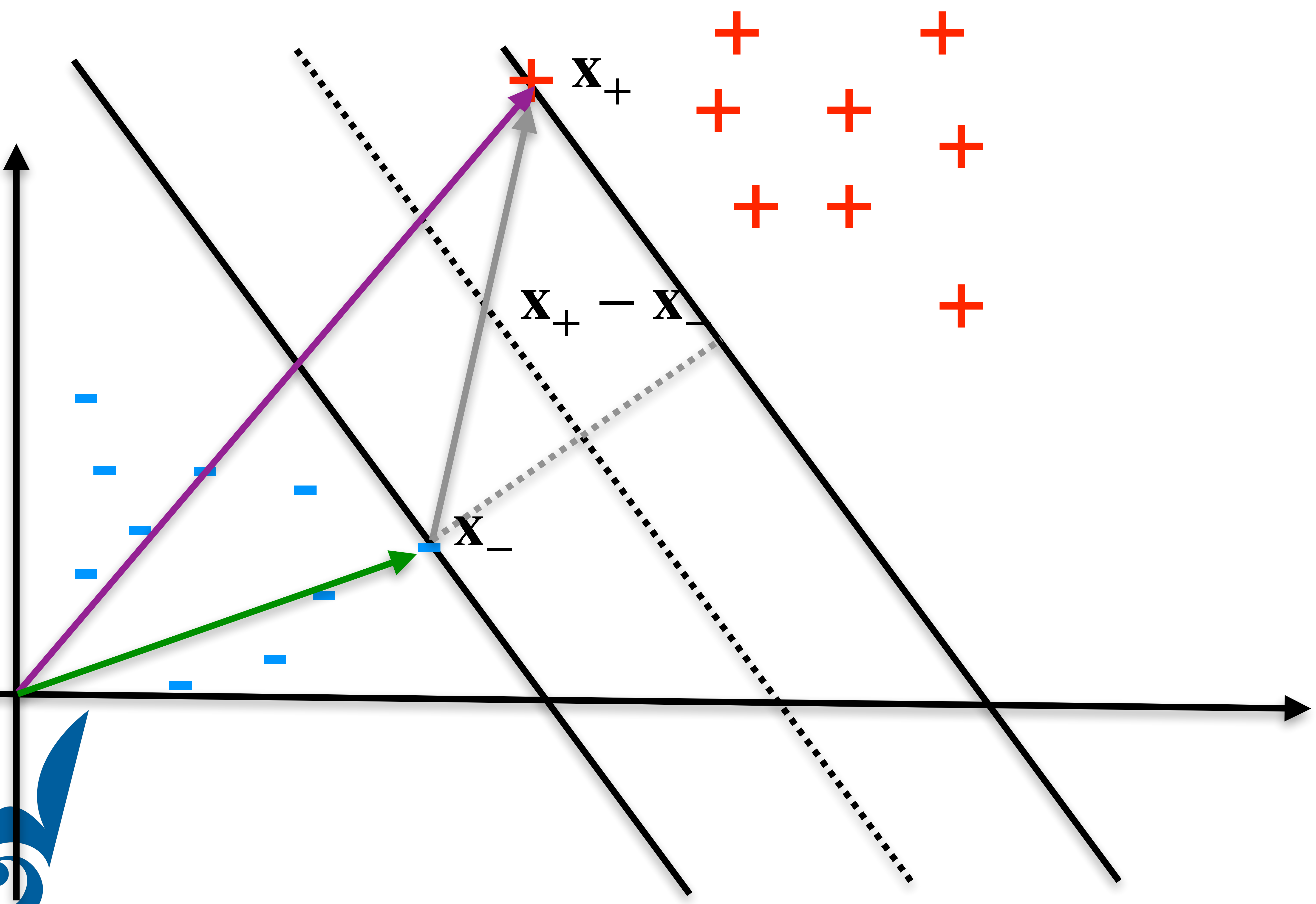


Support vector machines



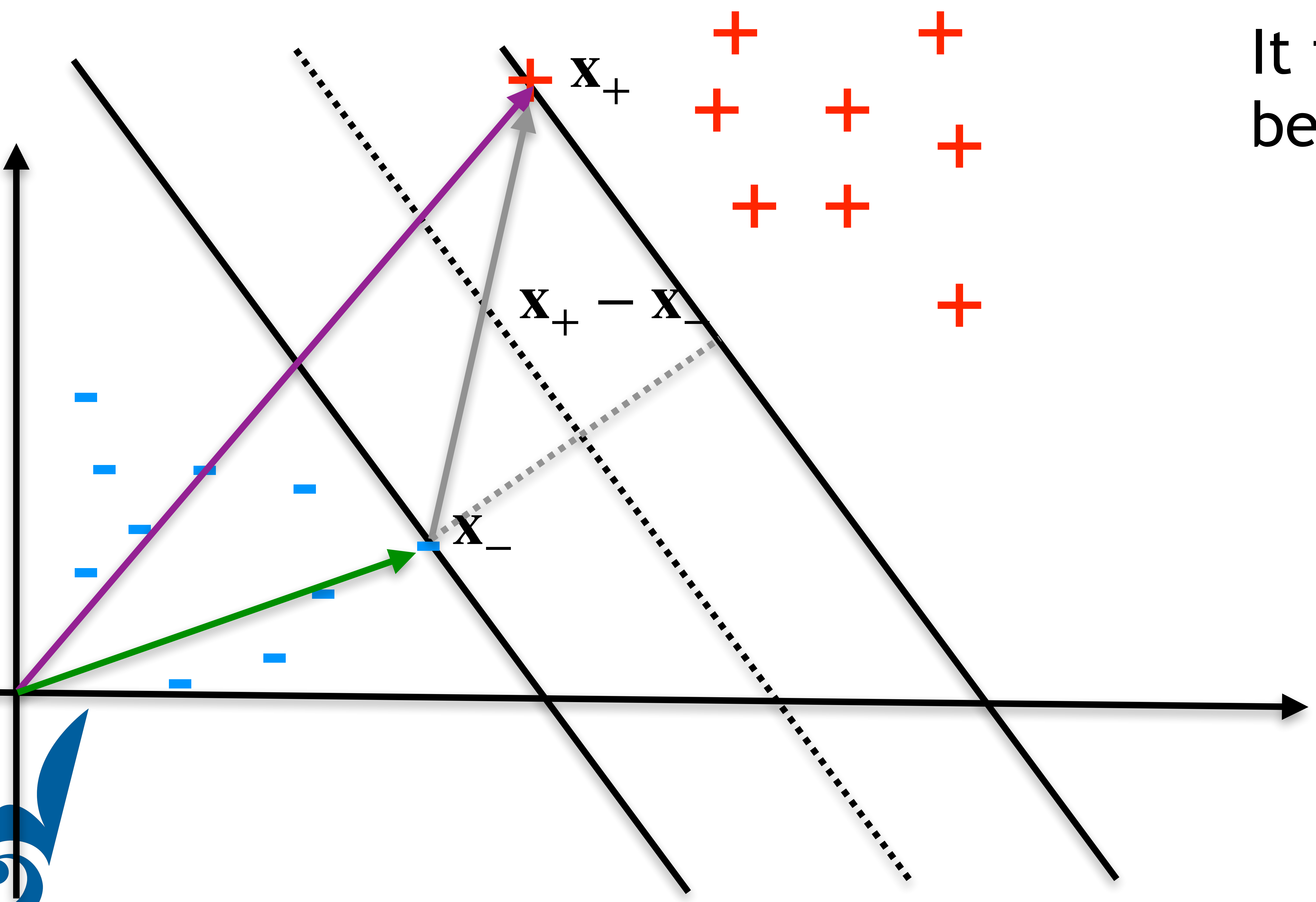
Support vector machines

How can we measure the width?



Support vector machines

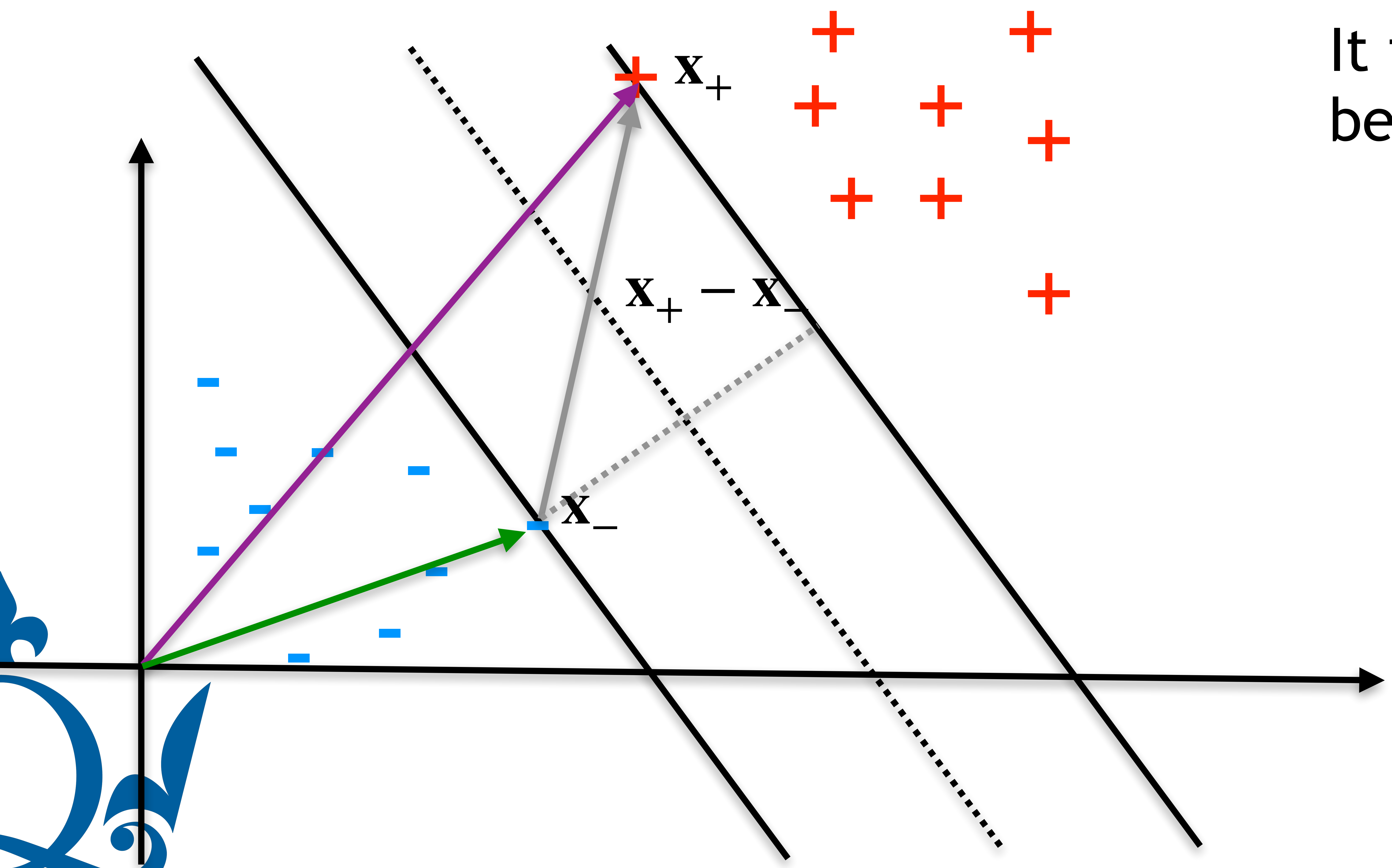
How can we measure the width?



It is the projection of the difference between the two vectors to w

Support vector machines

How can we measure the width?

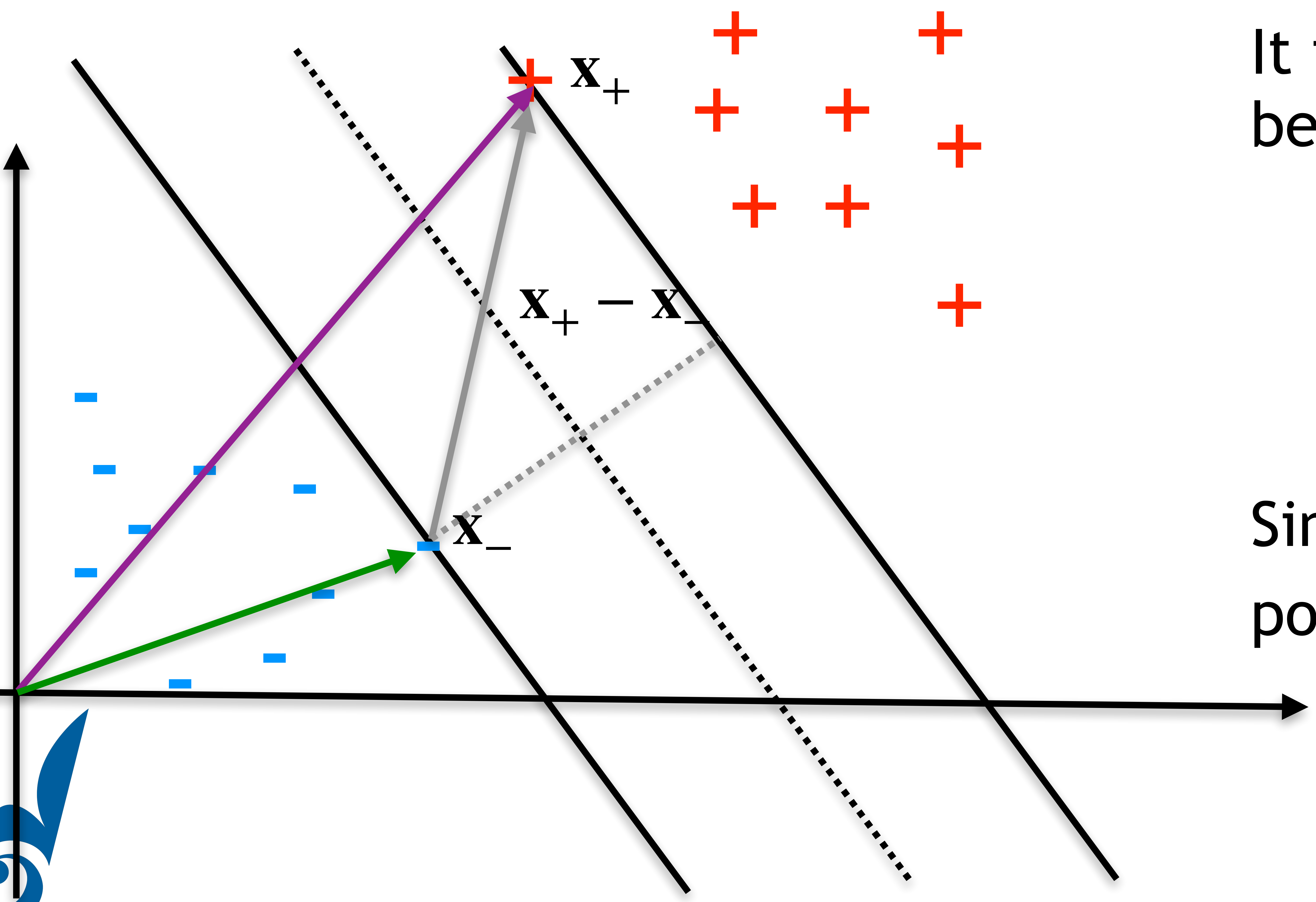


It is the projection of the difference between the two vectors to w

$$\langle \mathbf{x}_+ - \mathbf{x}_-, \frac{\mathbf{w}}{\|\mathbf{w}\|} \rangle$$

Support vector machines

How can we measure the width?



It is the projection of the difference between the two vectors to w

$$\langle \mathbf{x}_+ - \mathbf{x}_-, \frac{\mathbf{w}}{\|\mathbf{w}\|} \rangle$$

Since $\mathbf{w}\|\mathbf{w}\|^{-1}$ is a unit vector pointing in the direction of $\mathbf{w}$

Support vector machines

Let us remember that $y_i \left(\langle \mathbf{x}_i, \mathbf{w} \rangle + b \right) - 1 = 0$ for x_i in the support vectors

Support vector machines

Let us remember that $y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 = 0$ for x_i in the support vectors

$$y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 = 0 \;\rightarrow \langle \mathbf{x}_+, \mathbf{w}\rangle = 1 - b$$

Support vector machines

Let us remember that $y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 = 0$ for x_i in the support vectors

$$y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 = 0 \;\rightarrow \langle \mathbf{x}_+, \mathbf{w}\rangle = 1 - b$$

$$y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 = 0 \;\rightarrow -\langle \mathbf{x}_-, \mathbf{w}\rangle = 1 + b$$

Support vector machines

Let us remember that $y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 = 0$ for x_i in the support vectors

$$y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 = 0 \;\rightarrow \langle \mathbf{x}_+, \mathbf{w}\rangle = 1 - b$$

$$y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 = 0 \;\rightarrow -\langle \mathbf{x}_-, \mathbf{w}\rangle = 1 + b$$

$$\langle \mathbf{x}_+ - \mathbf{x}_-, \frac{\mathbf{w}}{\|\mathbf{w}\|}\rangle$$

Support vector machines

Let us remember that $y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 = 0$ for x_i in the support vectors

$$y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 = 0 \;\rightarrow \langle \mathbf{x}_+, \mathbf{w}\rangle = 1 - b$$

$$y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 = 0 \;\rightarrow -\langle \mathbf{x}_-, \mathbf{w}\rangle = 1 + b$$

$$\langle \mathbf{x}_+ - \mathbf{x}_-, \frac{\mathbf{w}}{\|\mathbf{w}\|}\rangle \qquad \rightarrow \langle \mathbf{x}_+, \frac{\mathbf{w}}{\|\mathbf{w}\|}\rangle - \langle \mathbf{x}_-, \frac{\mathbf{w}}{\|\mathbf{w}\|}\rangle = \frac{2}{\|\mathbf{w}\|}$$

Support vector machines

Let us remember that $y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 = 0$ for x_i in the support vectors

$$y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 = 0 \ \rightarrow \langle \mathbf{x}_+, \mathbf{w}\rangle = 1 - b$$

$$y_i\left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) - 1 = 0 \ \rightarrow -\langle \mathbf{x}_-, \mathbf{w}\rangle = 1 + b$$

$$\langle \mathbf{x}_+ - \mathbf{x}_-, \frac{\mathbf{w}}{\|\mathbf{w}\|}\rangle \qquad \rightarrow \langle \mathbf{x}_+, \frac{\mathbf{w}}{\|\mathbf{w}\|}\rangle - \langle \mathbf{x}_-, \frac{\mathbf{w}}{\|\mathbf{w}\|}\rangle = \frac{2}{\|\mathbf{w}\|}$$

We want to maximize this quantity!

Support vector machines

$$\langle \mathbf{x}_+, \frac{\mathbf{w}}{\|\mathbf{w}\|} \rangle - \langle \mathbf{x}_-, \frac{\mathbf{w}}{\|\mathbf{w}\|} \rangle = \frac{2}{\|\mathbf{w}\|}$$

Support vector machines

$$\langle \mathbf{x}_+, \frac{\mathbf{w}}{\|\mathbf{w}\|} \rangle - \langle \mathbf{x}_-, \frac{\mathbf{w}}{\|\mathbf{w}\|} \rangle = \frac{2}{\|\mathbf{w}\|}$$

Maximize that ratio is equivalent to minimize the denominator!

Support vector machines

$$\langle \mathbf{x}_+, \frac{\mathbf{w}}{\|\mathbf{w}\|} \rangle - \langle \mathbf{x}_-, \frac{\mathbf{w}}{\|\mathbf{w}\|} \rangle = \frac{2}{\|\mathbf{w}\|}$$

Maximize that ratio is equivalent to minimize the denominator!

But, we can actually compute

Support vector machines

$$\langle \mathbf{x}_+, \frac{\mathbf{w}}{\|\mathbf{w}\|} \rangle - \langle \mathbf{x}_-, \frac{\mathbf{w}}{\|\mathbf{w}\|} \rangle = \frac{2}{\|\mathbf{w}\|}$$

Maximize that ratio is equivalent to minimize the denominator!

But, we can actually compute

$$\arg\min_{\mathbf{w}} \left[\frac{\alpha}{2} \|\mathbf{w}\|^2 \right]$$

Support vector machines

$$\langle \mathbf{x}_+, \frac{\mathbf{w}}{\|\mathbf{w}\|} \rangle - \langle \mathbf{x}_-, \frac{\mathbf{w}}{\|\mathbf{w}\|} \rangle = \frac{2}{\|\mathbf{w}\|}$$

Maximize that ratio is equivalent to minimize the denominator!

But, we can actually compute

$$\arg\min_{\mathbf{w}} \left[\frac{\alpha}{2} \|\mathbf{w}\|^2 \right]$$

Why the one half and the square?

Support vector machines

So, first step is to

$$\arg\min_{\mathbf{w}} \left[\frac{\alpha}{2} \|\mathbf{w}\|^2 \right]$$

However we have to account for some constraints

$$y_i \left(\langle \mathbf{x}_i, \mathbf{w} \rangle + b \right) - 1 \geq 0 \qquad \text{for} \quad i = 1, \ldots, s$$

Support vector machines

So, first step is to

$$\arg\min_{\mathbf{w}} \left[\frac{\alpha}{2} \|\mathbf{w}\|^2 \right]$$

However we have to account for some constraints

$$y_i \left(\langle \mathbf{x}_i, \mathbf{w} \rangle + b \right) - 1 \geq 0 \qquad \text{for} \quad i = 1, \ldots, s$$

Hence, the problem can be written as

Support vector machines

So, first step is to

$$\arg\min_{\mathbf{w}} \left[\frac{\alpha}{2} \|\mathbf{w}\|^2 \right]$$

However we have to account for some constraints

$$y_i \left(\langle \mathbf{x}_i, \mathbf{w} \rangle + b \right) - 1 \geq 0 \qquad \text{for} \quad i = 1, \ldots, s$$

Hence, the problem can be written as $$\arg\min_{\mathbf{w}} \left[\sum_{i=1}^{s} \max\left(0, 1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right]$$

Support vector machines

So, first step is to

$$\arg\min_{\mathbf{w}} \left[\frac{\alpha}{2} \|\mathbf{w}\|^2 \right]$$

However we have to account for some constraints

$$y_i \left(\langle \mathbf{x}_i, \mathbf{w} \rangle + b \right) - 1 \geq 0 \qquad \text{for} \quad i = 1, \ldots, s$$

Hence, the problem can be written as $\arg\min_{\mathbf{w}} \left[\sum_{i=1}^{s} \max \left(0, 1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle \right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right]$

Since b can be included in w_0

Support vector machines

$$\arg\min_{\mathbf{w}} \left[\sum_{i=1}^{s} \max\left(0, 1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right]$$

Support vector machines

$$\arg\min_{\mathbf{w}} \left[\sum_{i=1}^{s} \max\left(0, 1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right]$$

Penalties for mislabelling

Support vector machines

$$\arg\min_{\mathbf{w}} \left[\sum_{i=1}^{s} \max\left(0, 1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right]$$

Penalties for mislabelling

If $y_i\langle \mathbf{x}_i, \mathbf{w}\rangle - 1 \geq 0$ we will end up with a correct labelling

This implies non positive values of $1 - y_i\langle \mathbf{x}_i, \mathbf{w}\rangle$

Support vector machines

$$\arg\min_{\mathbf{w}} \left[\sum_{i=1}^{s} \max\left(0, 1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right]$$

Penalties for mislabelling

Penalty for distance

If $\quad y_i \langle \mathbf{x}_i, \mathbf{w} \rangle - 1 \geq 0 \quad$ we will end up with a correct labelling

This implies non positive values of $1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle$

Support vector machines

$$\arg\min_{\mathbf{w}} \left[\sum_{i=1}^{s} \max\left(0, 1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right]$$

Penalties for mislabelling

Penalty for distance

If $y_i \langle \mathbf{x}_i, \mathbf{w} \rangle - 1 \geq 0$ we will end up with a correct labelling

This implies non positive values of $1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle$

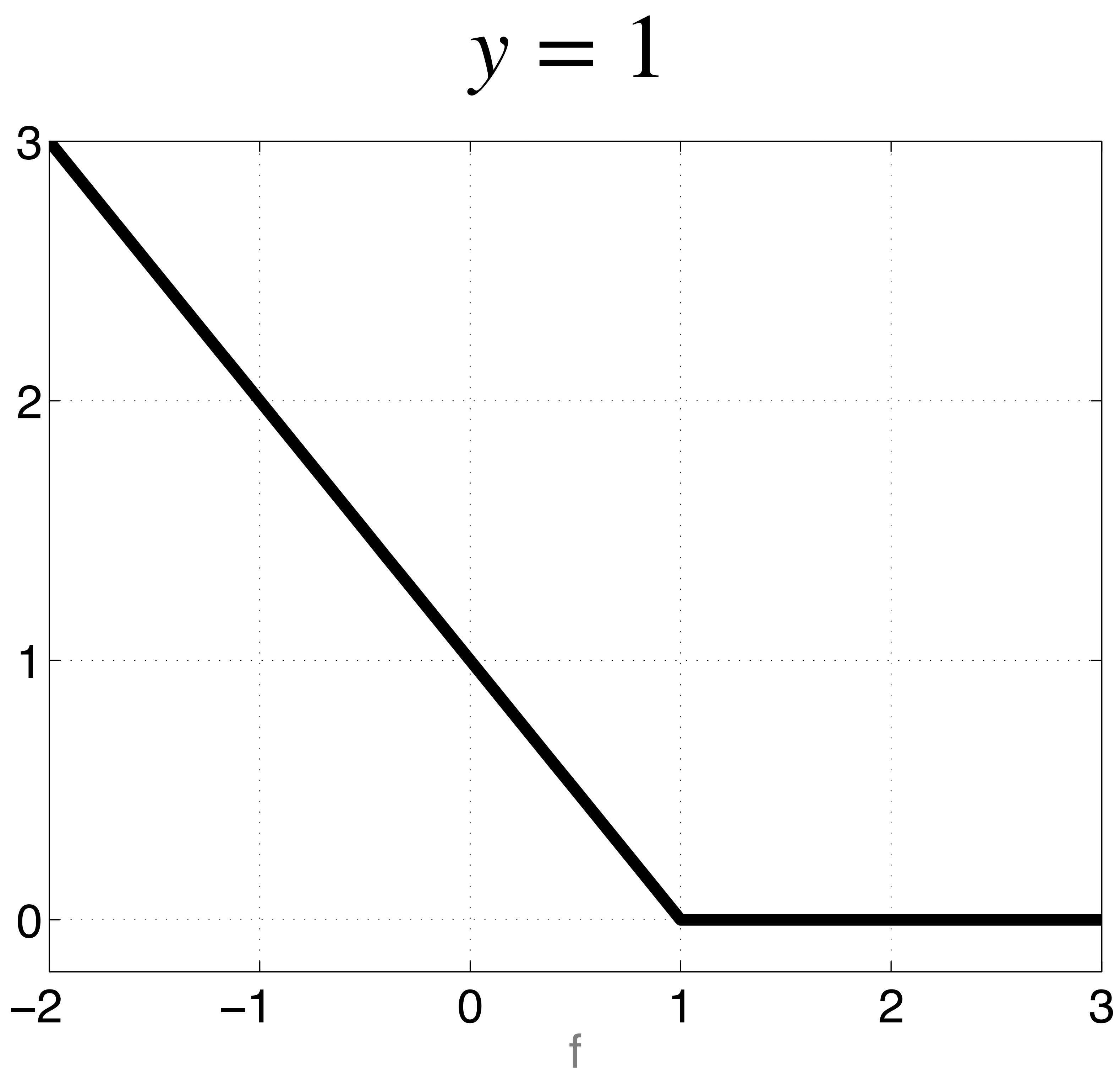
Note how the max here does not allow us to use the usual machinery as it is not differentiable

Support vector machines

The first term $L(\mathbf{w}) := \sum_{i=1}^{s} \max\left(0, 1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right)$ is also known as the *Hinge-loss*

that makes use of the Hinge function

$$\text{Hinge}(z) = \max(0, 1 - yz)$$
$$=: \left[1 - yz\right]_+$$



Looks like a door hinge, therefore the name

Support vector machines

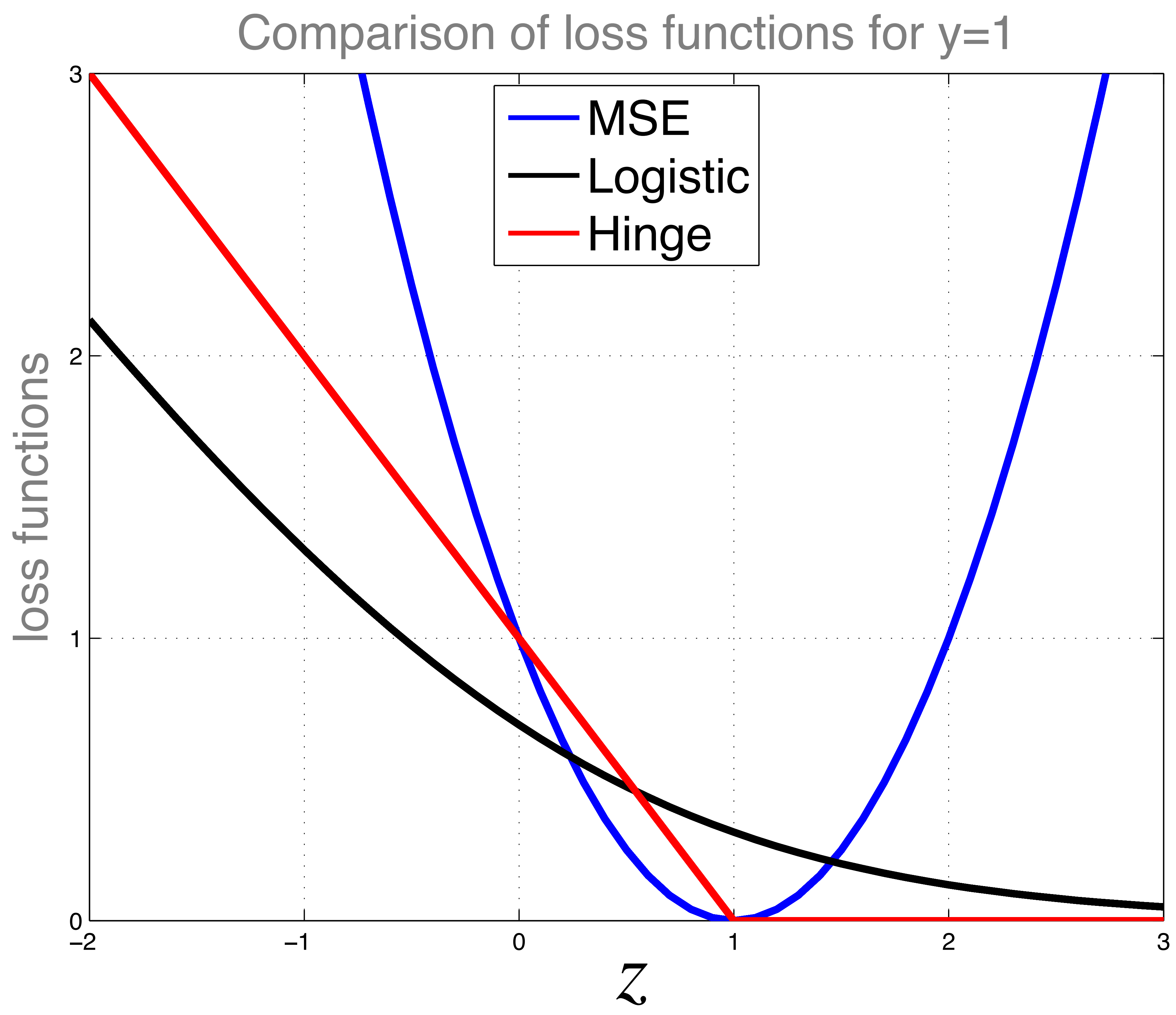
Consider $y \in \{-1, 1\}$

then the MSE, logistic regression and Hinge loss can be written as

$$\text{MSE}(z) = (1 - yz)^2$$

$$\text{LogisticLoss}(z) = \log\left(1 + e^{-yz}\right)$$

$$\text{Hinge}(z) = \max(0, 1 - yz)$$



Support vector machines

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \left\{ \sum_{i=1}^{s} \max\left(0, 1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\}$$

Support vector machines

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \left\{ \sum_{i=1}^{s} \max\left(0, 1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\}$$

$$= \arg\min_{\mathbf{w}} \left\{ \sum_{i=1}^{s} \max\left(0, \mathbf{1}_i - (\mathbf{Y}\mathbf{X}\mathbf{w})_i\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\}$$

for $\mathbf{Y} = \mathrm{diag}(\mathbf{y}) := \begin{pmatrix} y_1 & 0 & 0 & \cdots & 0 \\ 0 & y_2 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & & & y_s \end{pmatrix}$ and $\mathbf{1} = (1, 1, \cdots, 1)^T$

Support vector machines

How can we solve this optimisation problem?

$$\min_{\mathbf{w}} \left\{ \sum_{i=1}^{s} \max\left(0, 1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\}$$

Support vector machines

How can we solve this optimisation problem? Note that we can reformulate

$$\min_{\mathbf{w}} \left\{ \sum_{i=1}^{s} \max\left(0, 1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\}$$

to

$$\min_{\mathbf{w}} \left\{ \max_{\lambda \in [0,1]^s} \sum_{i=1}^{s} \lambda_i \left(1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\}$$

because of $\max(0,z) = \max\limits_{\lambda \in [0,1]} \lambda z$

Support vector machines

How can we solve this optimisation problem? Note that we can reformulate

$$\min_{\mathbf{w}} \left\{ \sum_{i=1}^{s} \max\left(0, \mathbf{1}_i - (\mathbf{YXw})_i\right) + \frac{\alpha}{2}\|\mathbf{w}\|^2 \right\}$$

to

$$\min_{\mathbf{w}} \left\{ \max_{\lambda \in [0,1]^s} \sum_{i=1}^{s} \left(\boldsymbol{\Lambda}\,(1 - \mathbf{YXw})\right)_i + \frac{\alpha}{2}\|\mathbf{w}\|^2 \right\}$$

because of $\max(0,z) = \max_{\lambda \in [0,1]} \lambda z$ for $\boldsymbol{\Lambda} := \mathrm{diag}(\lambda)$

Support vector machines

Assume for the moment that we can swap min and max, *i.e.*

$$\min_{\mathbf{w}} \left\{ \max_{\lambda \in [0,1]^s} \sum_{i=1}^{s} \lambda_i \left(1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\} = \max_{\lambda \in [0,1]^s} \left\{ \min_{\mathbf{w}} \sum_{i=1}^{s} \lambda_i \left(1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\}$$

Support vector machines

Assume for the moment that we can swap min and max, *i.e.*

$$\min_{\mathbf{w}} \left\{ \max_{\lambda \in [0,1]^s} \sum_{i=1}^{s} \lambda_i \left(1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\} = \max_{\lambda \in [0,1]^s} \left\{ \min_{\mathbf{w}} \sum_{i=1}^{s} \lambda_i \left(1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\}$$

Then the (new) inner optimisation problem becomes differentiable

$$\min_{w} \left\{ L(\mathbf{w}, \boldsymbol{\lambda}) := \sum_{i=1}^{s} \lambda_i \left(1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\}$$

Support vector machines

Assume for the moment that we can swap min and max, *i.e.*

$$\min_{\mathbf{w}} \left\{ \max_{\lambda \in [0,1]^s} \sum_{i=1}^{s} \lambda_i \left(1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\} = \max_{\lambda \in [0,1]^s} \left\{ \min_{\mathbf{w}} \sum_{i=1}^{s} \lambda_i \left(1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\}$$

Then the (new) inner optimisation problem becomes differentiable

$$\min_{w} \left\{ L(\mathbf{w}, \boldsymbol{\lambda}) := \sum_{i=1}^{s} \lambda_i \left(1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\}$$

The function is convex! We can just compute the gradient and set it to zero

Support vector machines

We can re-write the function as

$$L(\mathbf{w}, \boldsymbol{\lambda}) = \sum_{i=1}^{s} \lambda_i \left(1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle \right) + \frac{\alpha}{2} \|\mathbf{w}\|^2$$

(Simplified notation lambdas and ys are diagonal matrices)

Support vector machines

We can re-write the function as

$$L(\mathbf{w}, \boldsymbol{\lambda}) = \sum_{i=1}^{s} \lambda_i \left(1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2$$

$$= \sum_{i=1}^{s} \lambda_i \left(1 - y_i \sum_j x_{ij} w_j\right) + \frac{\alpha}{2} \sum_j w_j^2$$

(Simplified notation lambdas and ys are diagonal matrices)

Support vector machines

We can re-write the function as

$$L(\mathbf{w},\boldsymbol{\lambda}) = \sum_{i=1}^{s} \lambda_i \left(1 - y_i\langle \mathbf{x}_i, \mathbf{w}\rangle\right) + \frac{\alpha}{2}\|\mathbf{w}\|^2$$

$$= \sum_{i=1}^{s} \lambda_i \left(1 - y_i \sum_j x_{ij} w_j\right) + \frac{\alpha}{2} \sum_j w_j^2$$

(Simplified notation lambdas and ys are diagonal matrices)

Hence

$$\nabla L(\mathbf{w},\boldsymbol{\lambda})_p = \frac{\partial}{\partial w_p} \sum_{i=1}^{s} \lambda_i \left(1 - y_i \sum_j x_{ij} w_j\right) + \frac{\alpha}{2} \frac{\partial}{\partial w_p} \sum_j w_j^2$$

Support vector machines

$$\nabla L(\mathbf{w}, \boldsymbol{\lambda})_p = \frac{\partial}{\partial w_p} \sum_{i=1}^{s} \lambda_i \left(1 - y_i \sum_j x_{ij} w_j \right) + \frac{\alpha}{2} \frac{\partial}{\partial w_p} \sum_j w_j^2$$

Support vector machines

$$\nabla L(\mathbf{w}, \boldsymbol{\lambda})_p = \frac{\partial}{\partial w_p} \sum_{i=1}^{s} \lambda_i \left(1 - y_i \sum_j x_{ij} w_j \right) + \frac{\alpha}{2} \frac{\partial}{\partial w_p} \sum_j w_j^2$$

$$= - \sum_{i=1}^{s} \lambda_i y_i x_{ip} + \alpha w_p$$

$$= - \sum_{i=1}^{s} x_{pi}^{\top} \lambda_i y_i + \alpha w_p$$

$$\rightarrow \hat{\mathbf{w}} = \frac{1}{\alpha} \mathbf{X}^{\top} \mathbf{Y} \boldsymbol{\lambda} \qquad \rightarrow \hat{\mathbf{w}} = \frac{1}{\alpha} \mathbf{X}^{\top} \mathbf{Y} \boldsymbol{\Lambda}$$

Support vector machines

$$\min_{\mathbf{w}} \left\{ L(\mathbf{w}, \lambda) := \sum_{i=1}^{s} \Big(\boldsymbol{\Lambda} \left(\mathbf{1} - \mathbf{Y}\mathbf{X}\mathbf{w} \right) \Big)_i + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\}$$

Support vector machines

$$\min_{\mathbf{w}} \left\{ L(\mathbf{w}, \lambda) := \sum_{i=1}^{s} \Big(\boldsymbol{\Lambda} \left(\mathbf{1} - \mathbf{YXw} \right) \Big)_i + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\}$$

Let us plug the solution we just found and try to solve the outer problem

Support vector machines

$$\min_{\mathbf{w}} \left\{ L(\mathbf{w}, \lambda) := \sum_{i=1}^{s} \Big(\boldsymbol{\Lambda} \left(\mathbf{1} - \mathbf{YXw} \right) \Big)_i + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\}$$

Let us plug the solution we just found and try to solve the outer problem

$$= \left\{ \sum_{i=1}^{s} \left[\lambda \left(1 - \frac{1}{\alpha} \mathbf{YXX}^\top \mathbf{Y} \lambda \right) \right]_i + \frac{1}{2\alpha} \|\mathbf{X}^\top \mathbf{Y} \lambda\|^2 \right\}$$

Support vector machines

$$\min_{\mathbf{w}} \left\{ L(\mathbf{w}, \lambda) := \sum_{i=1}^{s} \Big(\mathbf{\Lambda} \left(\mathbf{1} - \mathbf{YXw} \right) \Big)_i + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\}$$

Let us plug the solution we just found and try to solve the outer problem

$$= \left\{ \sum_{i=1}^{s} \left[\lambda \left(1 - \frac{1}{\alpha} \mathbf{YXX}^\top \mathbf{Y} \lambda \right) \right]_i + \frac{1}{2\alpha} \|\mathbf{X}^\top \mathbf{Y} \lambda\|^2 \right\}$$

$$= \langle \lambda, \mathbf{1} \rangle - \frac{1}{2\alpha} \|\mathbf{X}^\top \mathbf{Y} \lambda\|^2$$

Support vector machines

$$\hat{\boldsymbol{\lambda}} = \arg \max_{\boldsymbol{\lambda} \in [0,1]^s} \left\{ \langle \boldsymbol{\lambda}, \mathbf{1} \rangle - \frac{1}{2\alpha} \|\mathbf{X}^\top \mathbf{Y} \boldsymbol{\lambda}\|^2 \right\}$$

Support vector machines

$$\hat{\lambda} = \arg \max_{\lambda \in [0,1]^s} \left\{ \langle \lambda, \mathbf{1} \rangle - \frac{1}{2\alpha} \|\mathbf{X}^\top \mathbf{Y} \lambda\|^2 \right\}$$

$$= \arg \max_{\lambda} \left\{ \langle \lambda, \mathbf{1} \rangle - \frac{1}{2\alpha} \|\mathbf{X}^T \mathbf{Y} \lambda\|^2 - \chi_{[0,1]^s}(\lambda) \right\}$$

Support vector machines

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\lambda \in [0,1]^s} \left\{ \langle \boldsymbol{\lambda}, \mathbf{1} \rangle - \frac{1}{2\alpha} \|\mathbf{X}^\top \mathbf{Y} \boldsymbol{\lambda}\|^2 \right\}$$

$$= \arg\max_{\lambda} \left\{ \langle \boldsymbol{\lambda}, \mathbf{1} \rangle - \frac{1}{2\alpha} \|\mathbf{X}^T \mathbf{Y} \boldsymbol{\lambda}\|^2 - \chi_{[0,1]^s}(\boldsymbol{\lambda}) \right\}$$

$\chi_{[0,1]^s}(\lambda) = 0$ if $\lambda \in [0,1]$

Support vector machines

$$\hat{\boldsymbol{\lambda}} = \arg \max_{\lambda \in [0,1]^s} \left\{ \langle \boldsymbol{\lambda}, \mathbf{1} \rangle - \frac{1}{2\alpha} \| \mathbf{X}^\top \mathbf{Y} \boldsymbol{\lambda} \|^2 \right\}$$

$$= \arg \max_{\lambda} \left\{ \langle \boldsymbol{\lambda}, \mathbf{1} \rangle - \frac{1}{2\alpha} \| \mathbf{X}^T \mathbf{Y} \boldsymbol{\lambda} \|^2 - \chi_{[0,1]^s}(\boldsymbol{\lambda}) \right\}$$

$\chi_{[0,1]^s}(\lambda) = 0$ if $\lambda \in [0,1]$

$\chi_{[0,1]^s}(\lambda) = \infty$ if $\lambda \notin [0,1]$

Support vector machines

We can solve this problem for example via projected gradient **ascent**

$$\lambda^{k+1} = \mathrm{proj}_{[0,1]^s}\left[\lambda^k + \tau\left(\mathbf{1} - \frac{1}{\alpha}\mathbf{Y}\mathbf{X}\mathbf{X}^\top\mathbf{Y}\lambda^k\right)\right]$$

Support vector machines

We can solve this problem for example via projected gradient **ascent**

$$\lambda^{k+1} = \text{proj}_{[0,1]^s}\left[\lambda^k + \tau\left(\mathbf{1} - \frac{1}{\alpha}\mathbf{Y}\mathbf{X}\mathbf{X}^\top\mathbf{Y}\lambda^k\right)\right]$$

Why gradient ascent?

Support vector machines

Note that once we have computed a numerical approximation for $\hat{\lambda}$, we can compute $\hat{\mathbf{w}}$ *via*

$$\hat{\mathbf{w}} = \frac{1}{\alpha}\mathbf{X}^\top \mathbf{Y}\hat{\lambda}$$

Support vector machines

Note that once we have computed a numerical approximation for $\hat{\lambda}$, we can compute $\hat{\mathbf{w}}$ *via*

$$\hat{\mathbf{w}} = \frac{1}{\alpha}\mathbf{X}^\top \mathbf{Y}\hat{\lambda}$$

But the question that we have to ask ourselves now is: is this actually a solution of

$$\hat{w} = \arg\min_{\mathbf{w}} \left\{ \sum_{i=1}^{s} \max\left(0, 1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2}\|\mathbf{w}\|^2 \right\} \qquad ?$$

Duality

This question boils down to when can we guarantee

$$\min_x \max_y f(x, y) = \max_y \min_x f(x, y) \qquad ?$$

Duality

This question boils down to when can we guarantee

$$\min_x \max_y f(x, y) = \max_y \min_x f(x, y) \qquad ?$$

Max-min inequality tells us $\max_y \min_x f(x, y) \leq \min_x \max_y f(x, y)$

Duality

This question boils down to when can we guarantee

$$\min_x \max_y f(x, y) = \max_y \min_x f(x, y) \qquad ?$$

Max-min inequality tells us $\max_y \min_x f(x, y) \leq \min_x \max_y f(x, y)$

so equality is possible, but we can have

$$\max_y \min_x f(x, y) < \min_x \max_y f(x, y)$$

Duality

Example: $f(x,y) = \sin(x+y)$

Across all x the min is -1. After I have reached the min in x with y mute, the max is unchanged!

Duality

Example: $f(x, y) = \sin(x + y)$

$$\Rightarrow \quad -1 = \max_y \min_x \sin(x + y) < \min_x \max_y \sin(x + y) = 1$$

Across all x the min
is -1. After I have
reached the min in x
with y mute, the max is
unchanged!

Duality

Recall: definition of convexity

A function $f : C \to \mathbb{R}$ over a convex set C is called convex if

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$$

is satisfied for all $x, y \in C$ and $\lambda \in [0,1]$.

Duality

Recall: definition of convexity

A function $f : C \to \mathbb{R}$ over a convex set C is called convex if

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$$

is satisfied for all $x, y \in C$ and $\lambda \in [0,1]$.

Similarly we can define concavity:

A function $f : C \to \mathbb{R}$ over a convex set C is called *concave* if

$$f(\lambda x + (1-\lambda)y) \geq \lambda f(x) + (1-\lambda)f(y)$$

is satisfied for all $x, y \in C$ and $\lambda \in [0,1]$.

Duality

Minimax Theorem (von Neumann 1928)

Let $X \subset \mathbb{R}^m$ and $Y \subset \mathbb{R}^n$ be compact, convex sets.

If $f : X \times Y \to \mathbb{R}$ is a continuous function that is convex-concave, i.e.

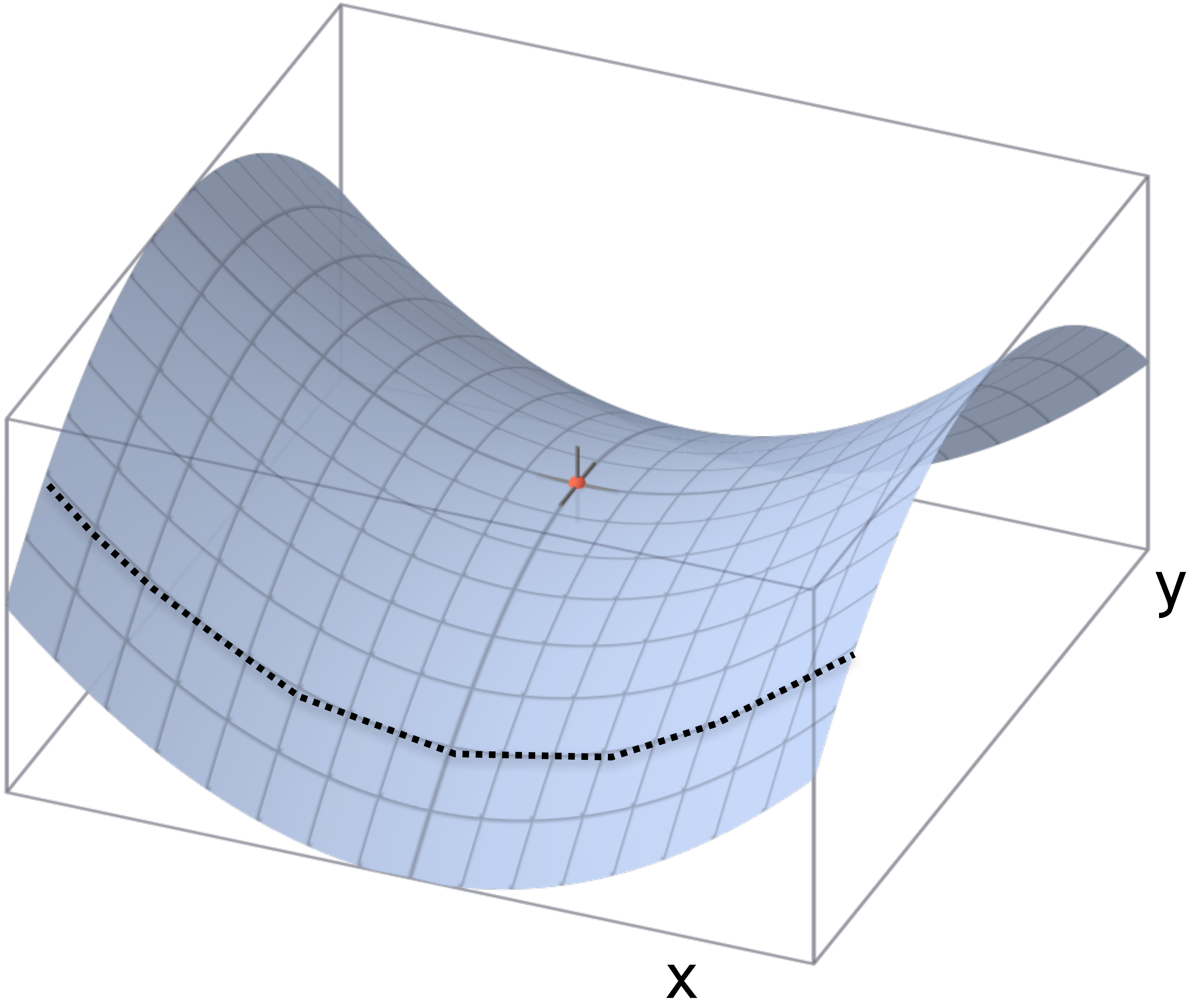
$$f(\,\cdot\,, y) : X \to \mathbb{R} \text{ is convex for fixed } y$$
$$f(x, \cdot\,) : Y \to \mathbb{R} \text{ is concave for fixed } x$$

Then the max-min inequality is an equality, i.e.

$$\min_{x \in X} \max_{y \in Y} f(x, y) = \max_{y \in Y} \min_{x \in X} f(x, y)\,.$$

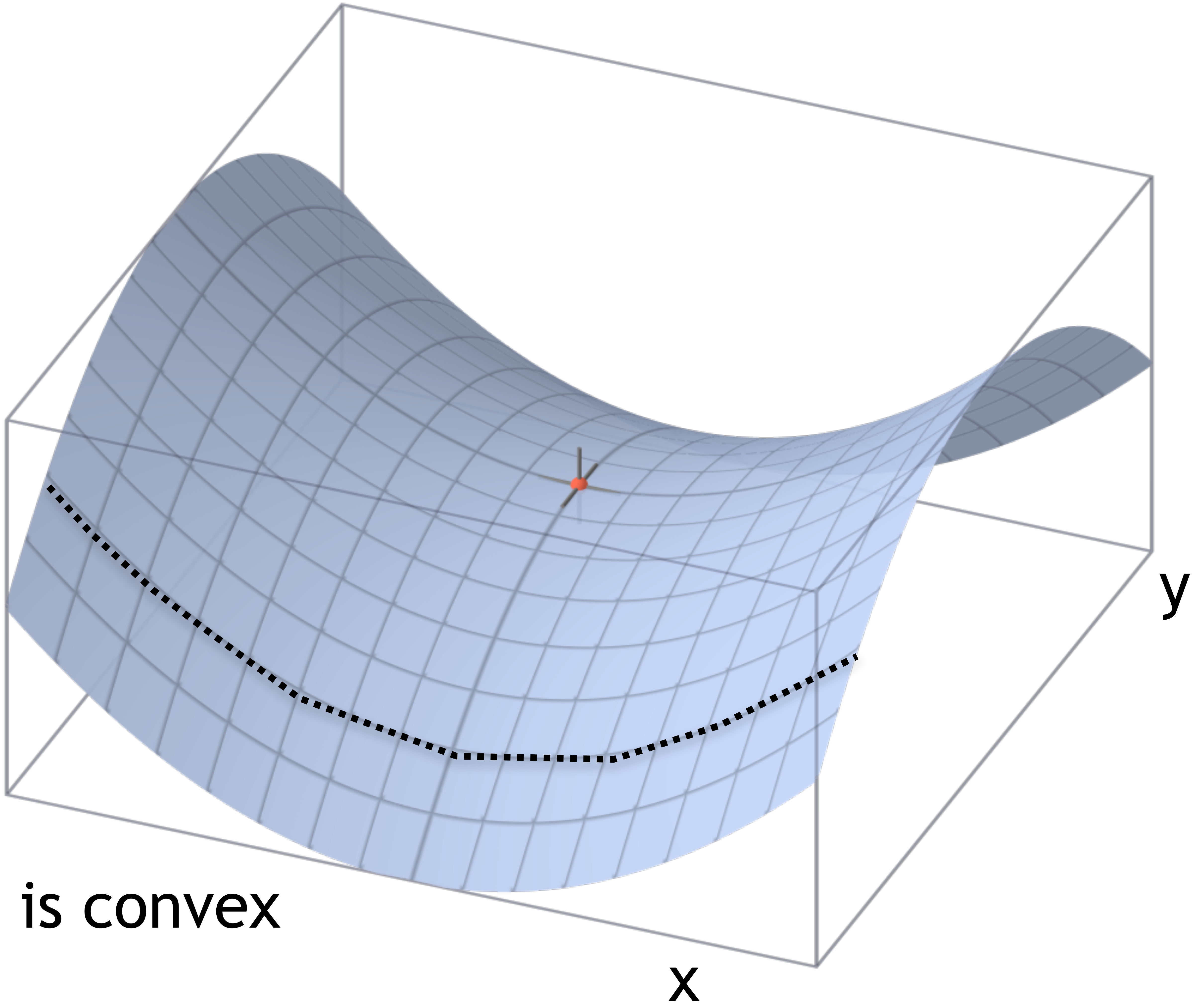
Duality

Convex-concave
saddle-point problem



Duality

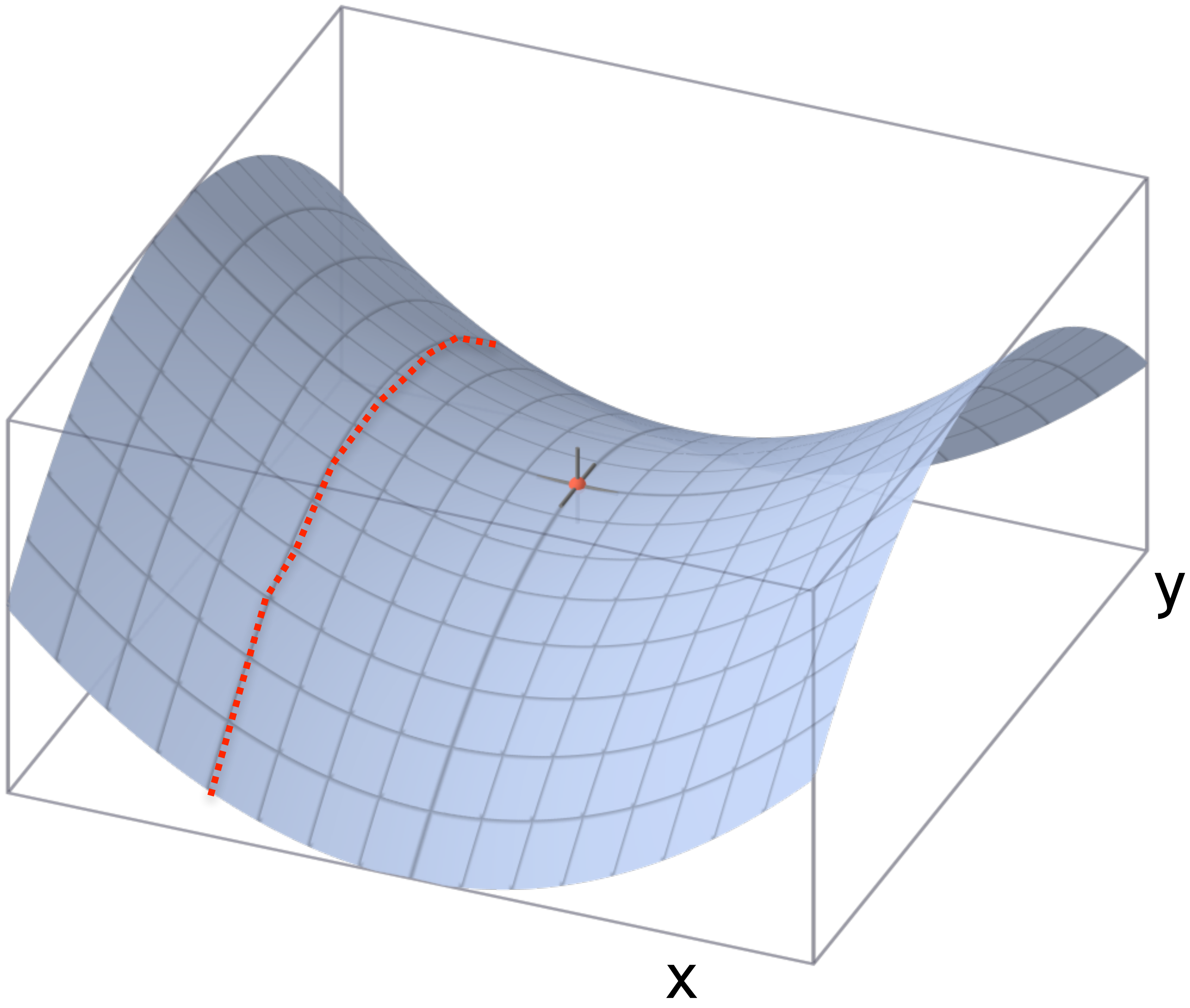
Convex-concave
saddle-point problem



Function is convex
in x

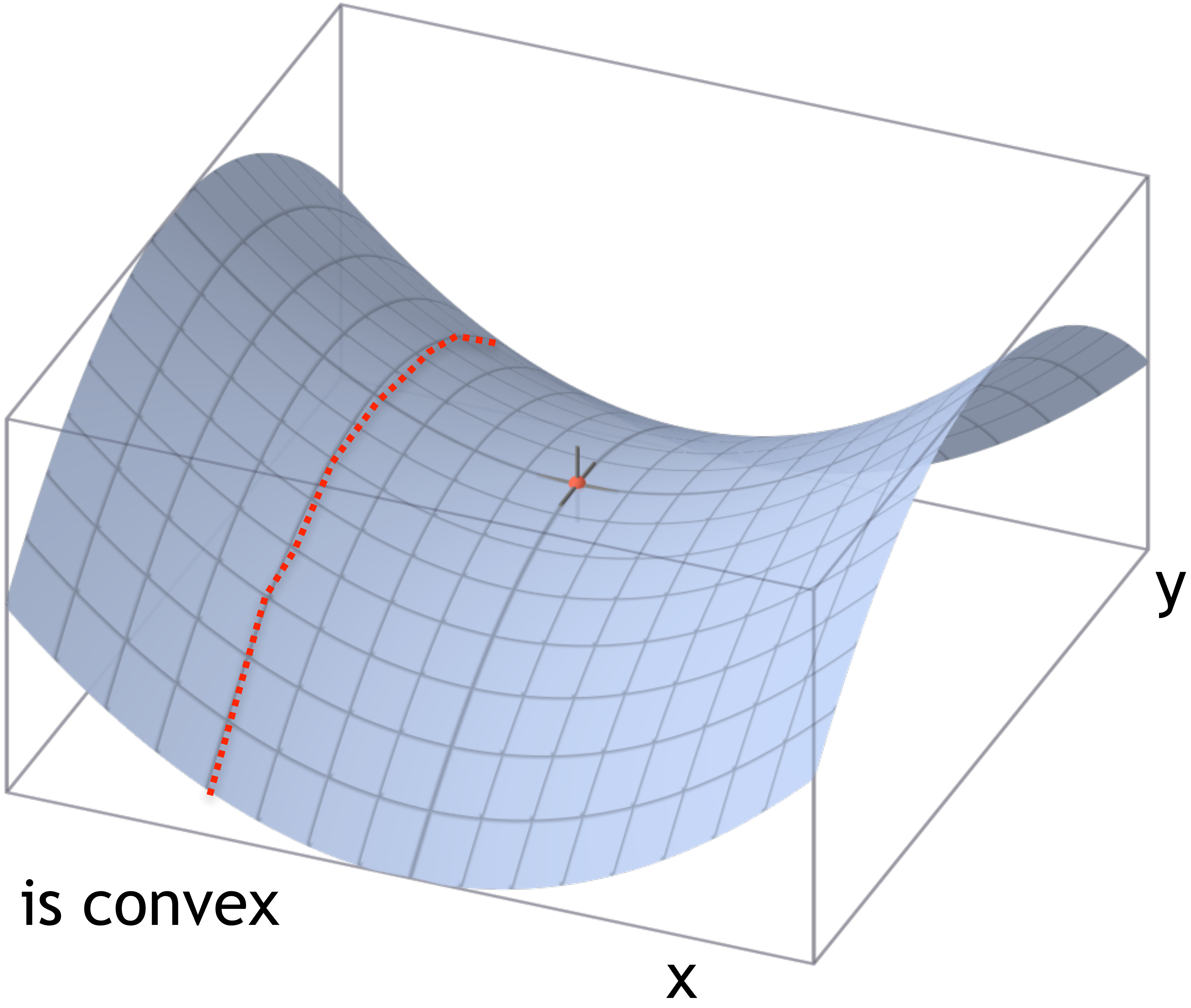
Duality

Convex-concave
saddle-point problem



Duality

Convex-concave
saddle-point problem

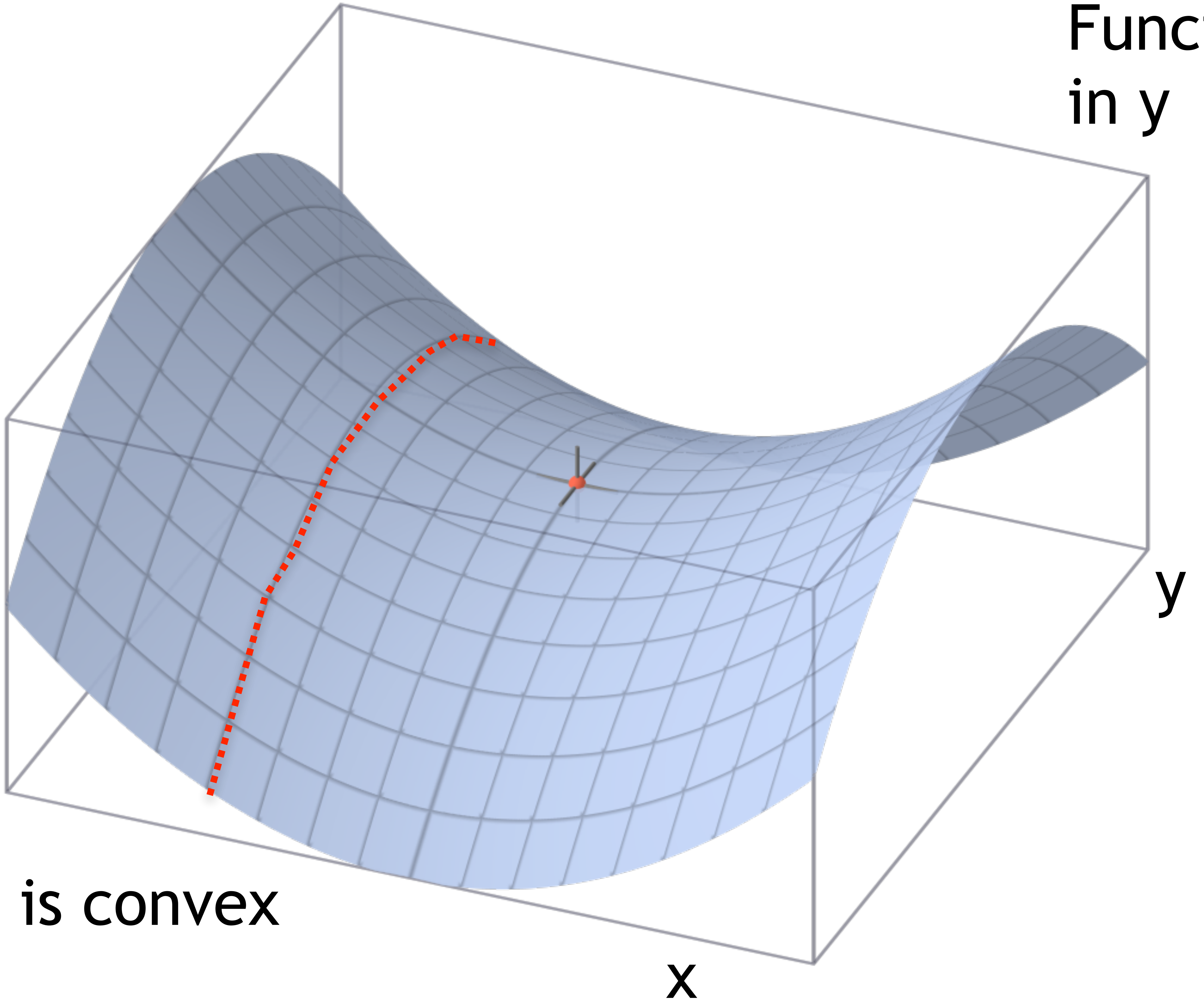


Function is convex
in x

Duality

Convex-concave
saddle-point problem

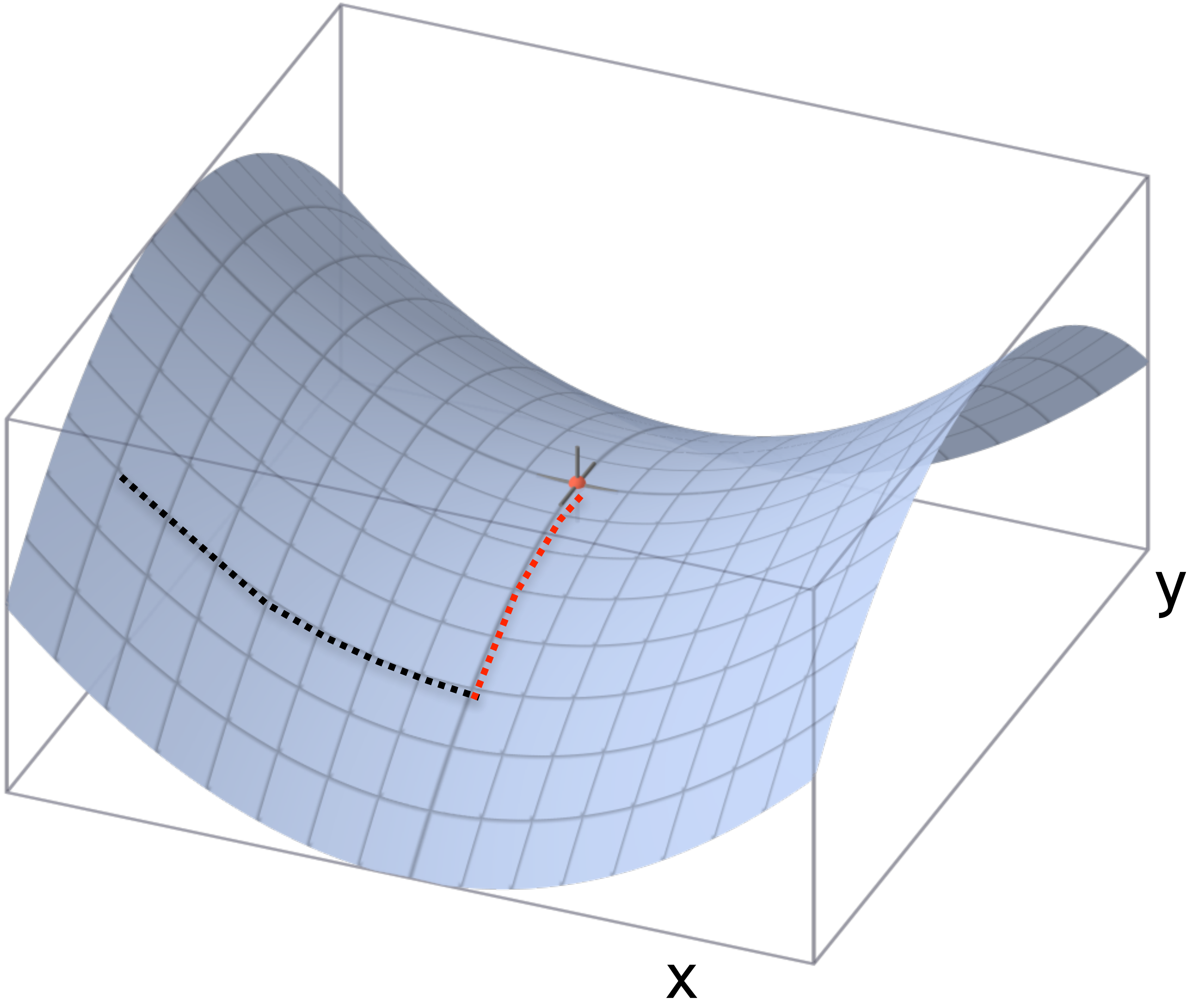
Function is concave
in y



Function is convex
in x

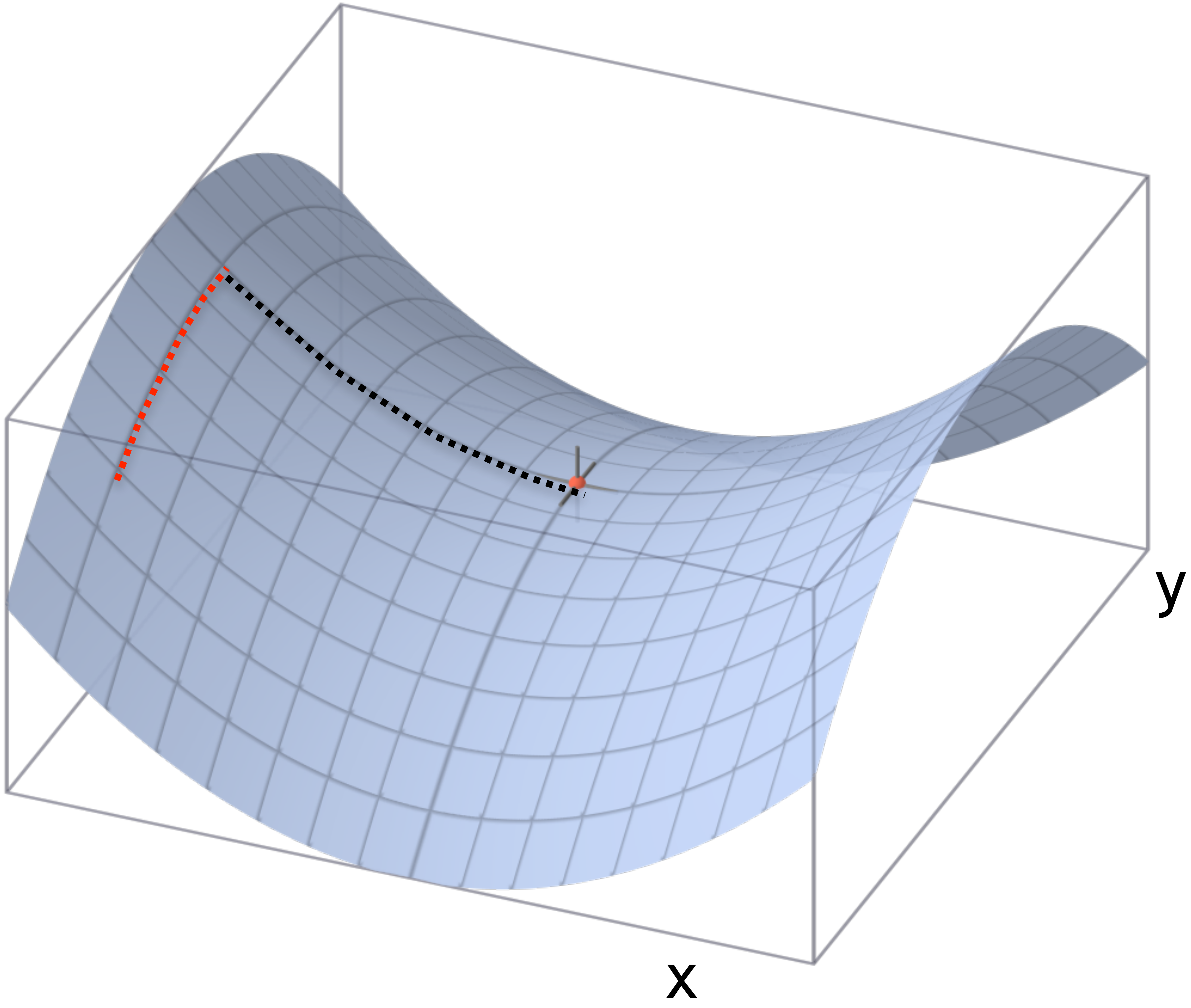
Duality

What happens if we start from a point x,y we minimize in x and then maximize in y?



Duality

What happens if we revert? First maximize in y and then minimize in x?



Duality

So, you can switch min and max if you are minimizing a convex function and maximizing a concave function!

Duality

$$L(\mathbf{w}, \boldsymbol{\lambda}) = \sum_{i=1}^{s} \lambda_i \left(1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2$$

$L : \mathbb{R}^n \times [0,1]^s \to \mathbb{R}$ is convex in $\mathbf{w} \in \mathbb{R}^n$ and concave in $\boldsymbol{\lambda} \in [0,1]^s$

Duality

$$L(\mathbf{w}, \boldsymbol{\lambda}) = \sum_{i=1}^{s} \lambda_i \left(1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2$$

$L : \mathbb{R}^n \times [0,1]^s \to \mathbb{R}$ is convex in $\mathbf{w} \in \mathbb{R}^n$ and concave in $\boldsymbol{\lambda} \in [0,1]^s$

Hence, $$\min_{\mathbf{w} \in \mathbb{R}^n} \max_{\boldsymbol{\lambda} \in [0,1]^s} L(\mathbf{w}, \boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda} \in [0,1]^s} \min_{\mathbf{w} \in \mathbb{R}^n} L(\mathbf{w}, \boldsymbol{\lambda}) .$$

Duality

$$L(\mathbf{w}, \boldsymbol{\lambda}) = \sum_{i=1}^{s} \lambda_i \left(1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2$$

$L : \mathbb{R}^n \times [0,1]^s \to \mathbb{R}$ is convex in $\mathbf{w} \in \mathbb{R}^n$ and concave in $\boldsymbol{\lambda} \in [0,1]^s$

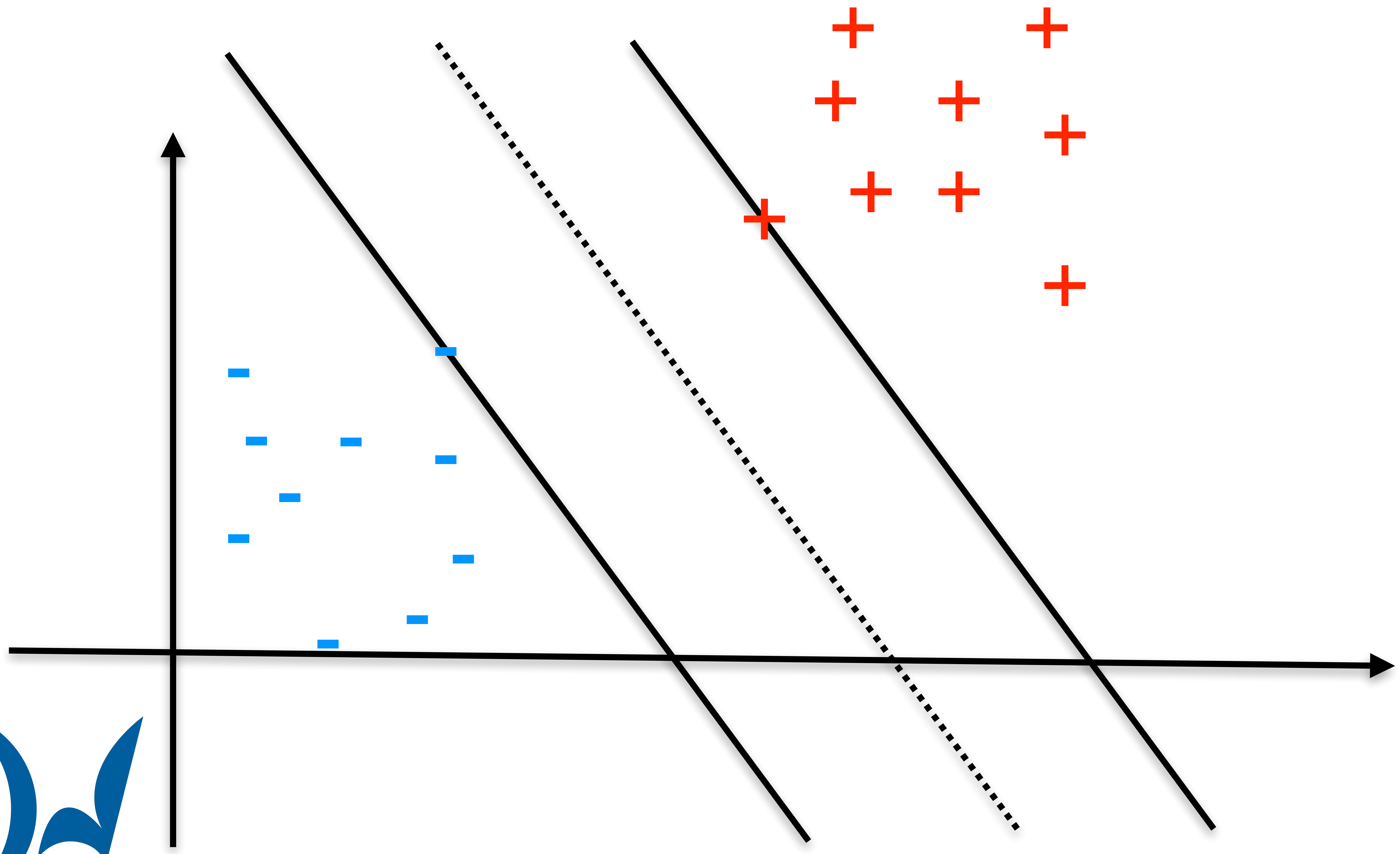
Hence,

$$\min_{\mathbf{w}\in\mathbb{R}^n} \max_{\boldsymbol{\lambda}\in[0,1]^s} L(\mathbf{w}, \boldsymbol{\lambda}) = \max_{\boldsymbol{\lambda}\in[0,1]^s} \min_{\mathbf{w}\in\mathbb{R}^n} L(\mathbf{w}, \boldsymbol{\lambda}) .$$

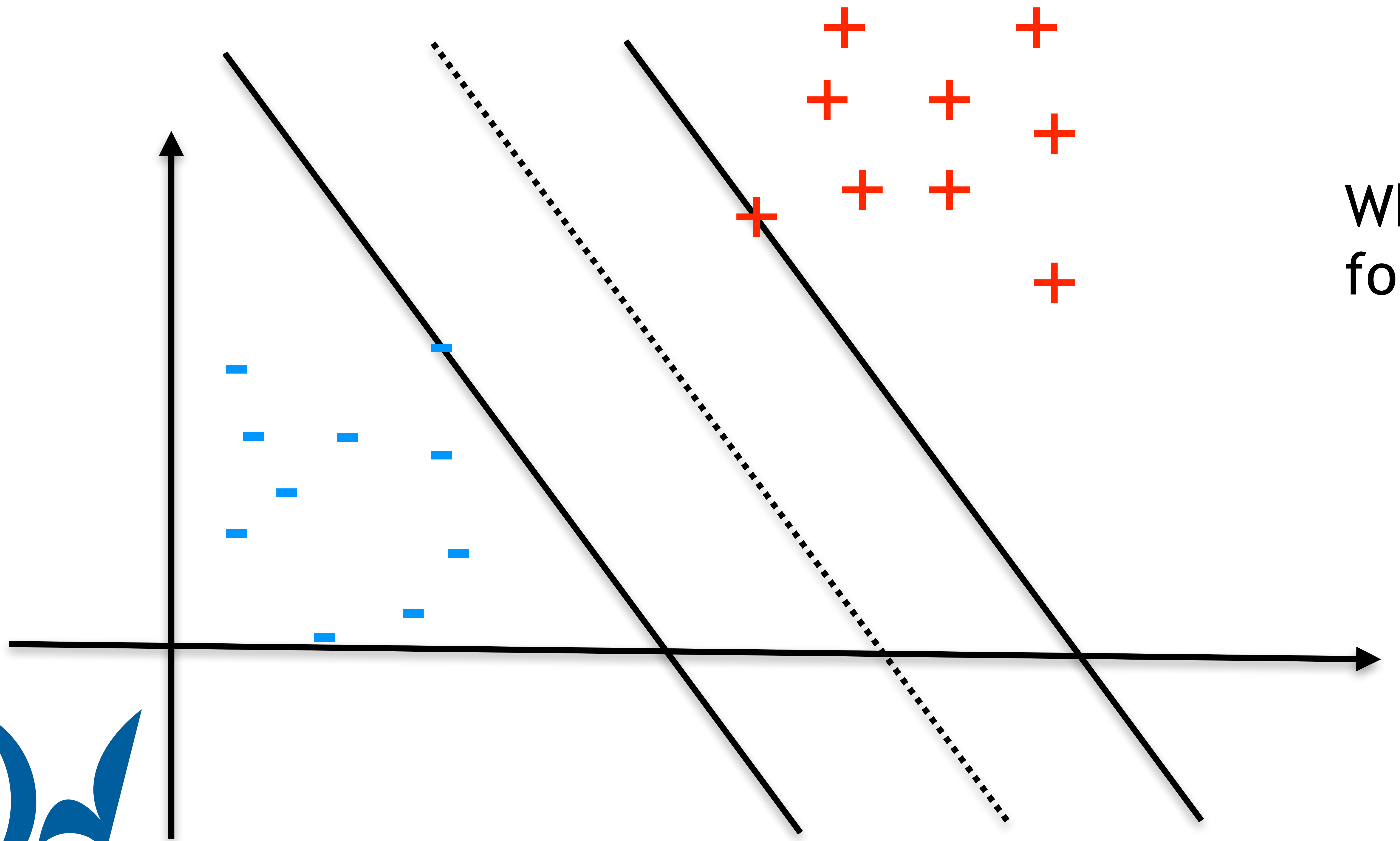
$$\|$$

$$\min_{\mathbf{w}\in\mathbb{R}^n} \left\{ \sum_{i=1}^{s} \max\left(0, 1 - y_i \langle \mathbf{x}_i, \mathbf{w} \rangle\right) + \frac{\alpha}{2} \|\mathbf{w}\|^2 \right\}$$

Support vector machines

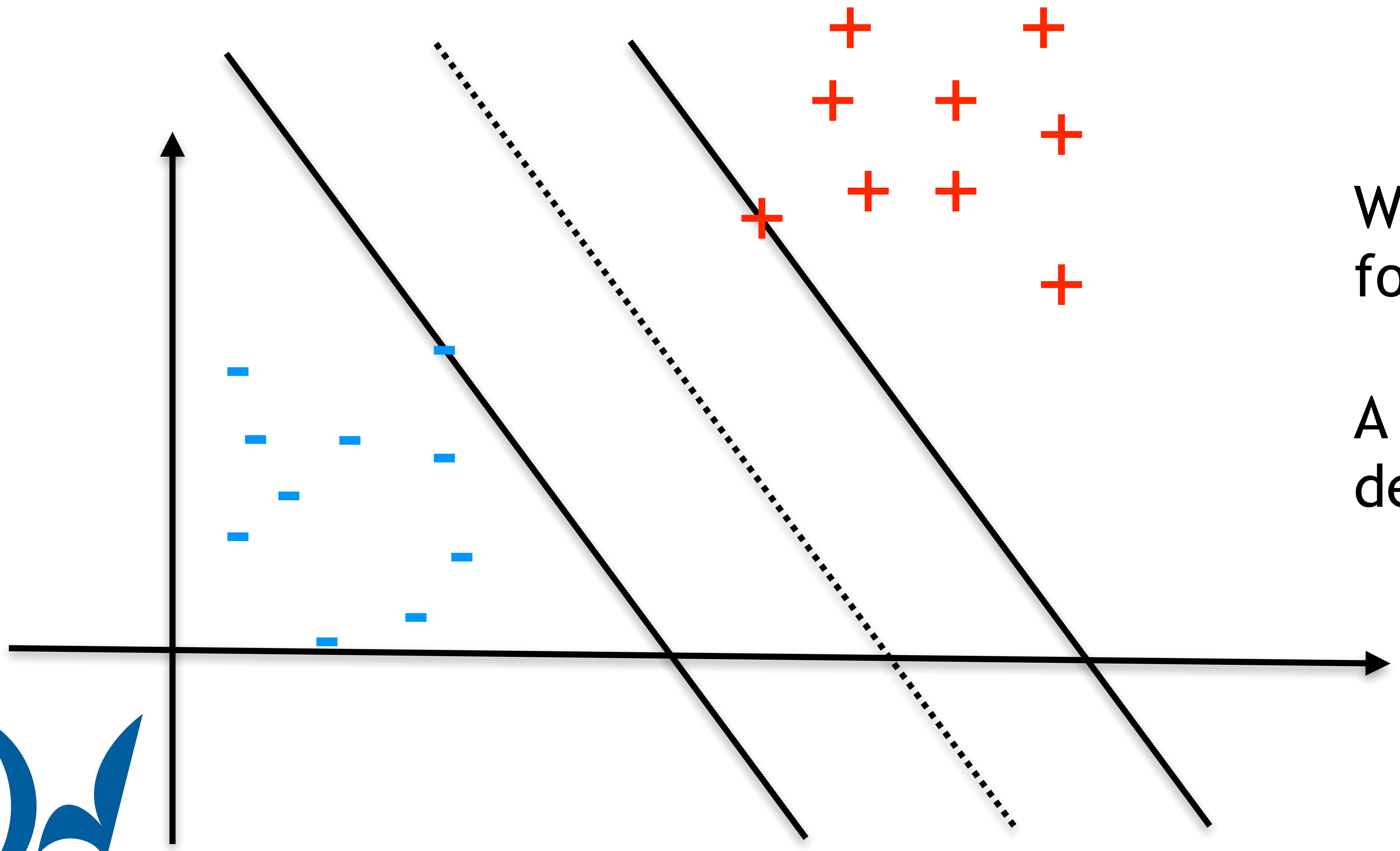


Support vector machines



What we have done so far works for data that looks like this

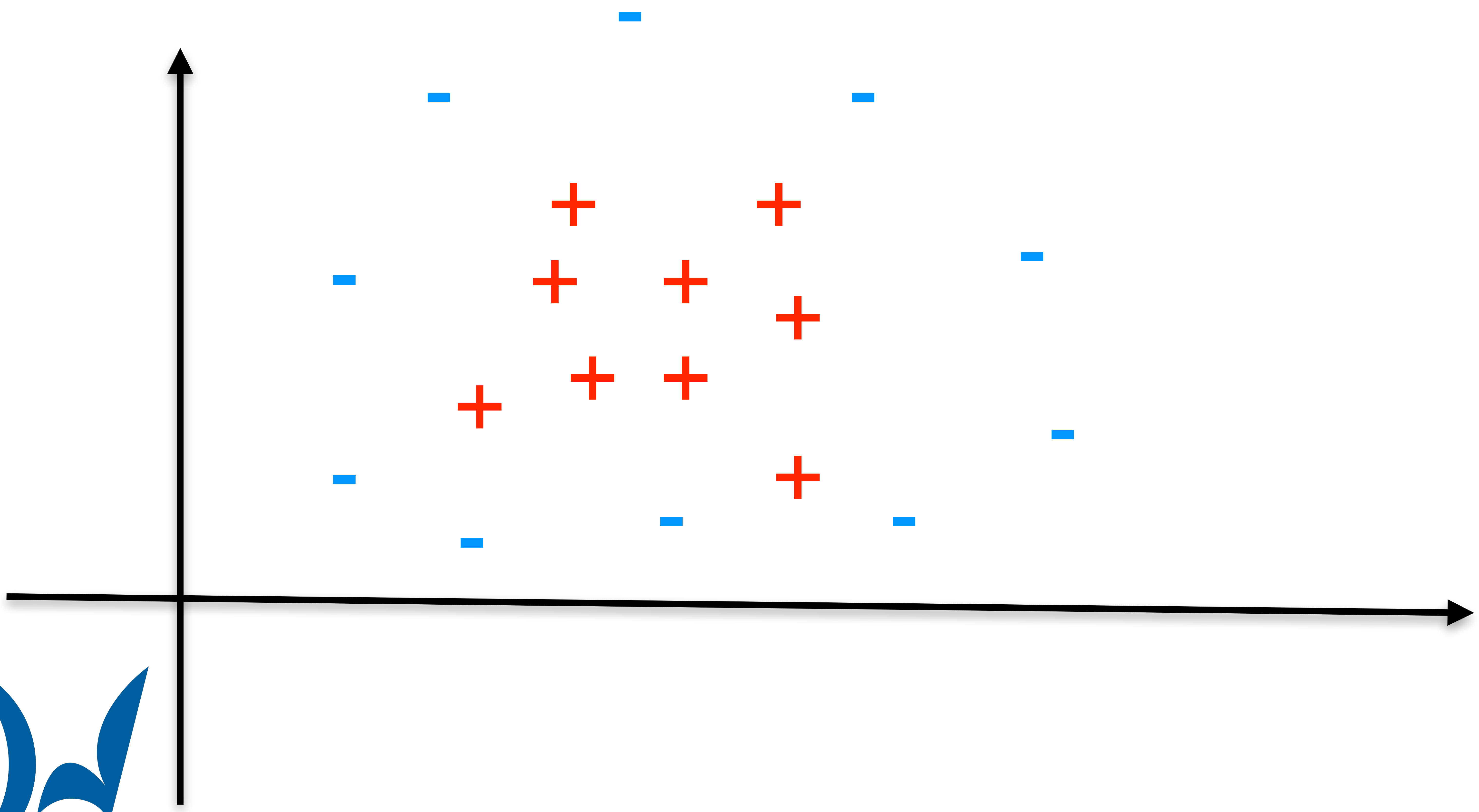
Support vector machines



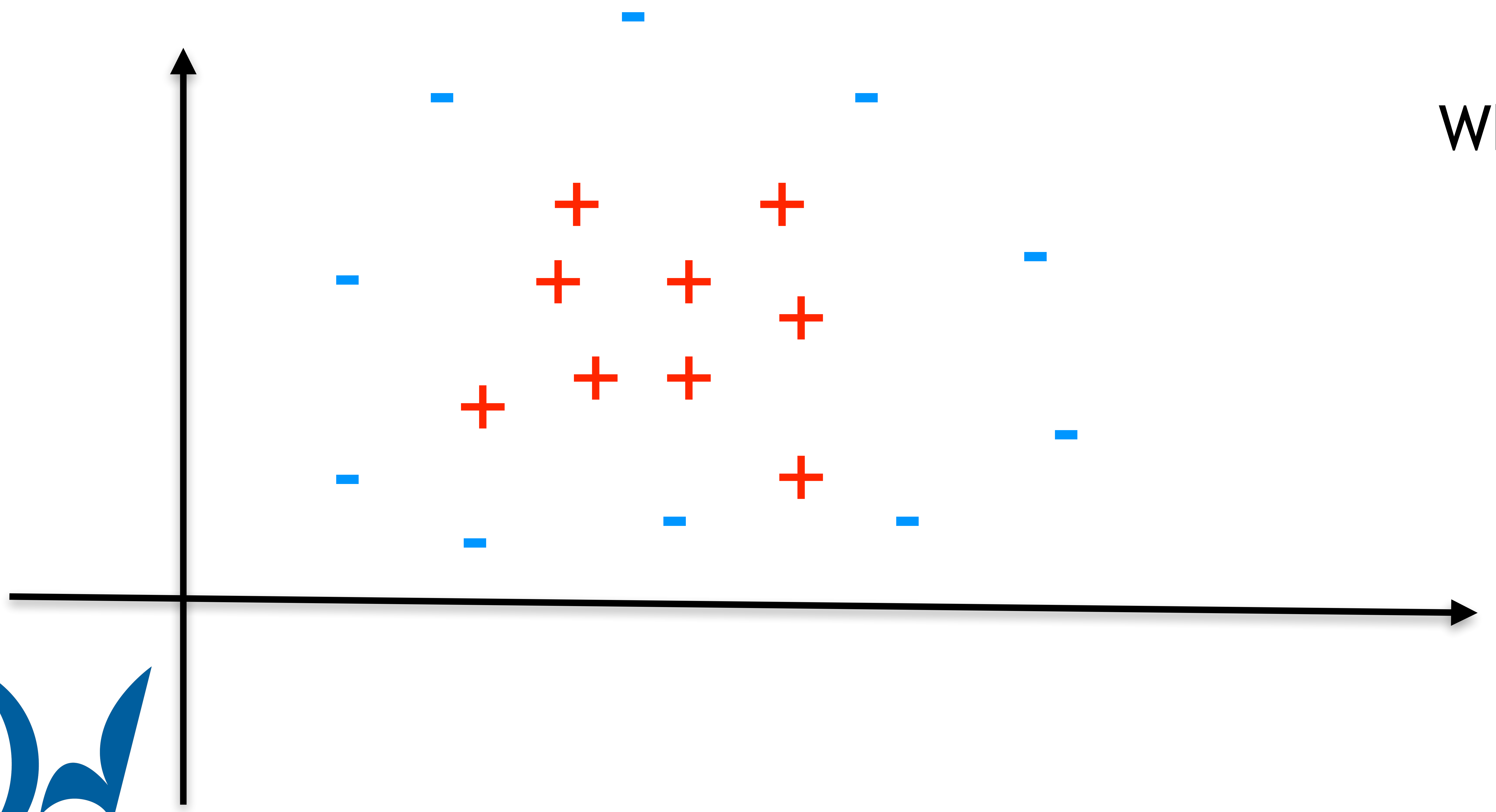
What we have done so far works for data that looks like this

A linear function can be used as decision boundary

Support vector machines



Support vector machines



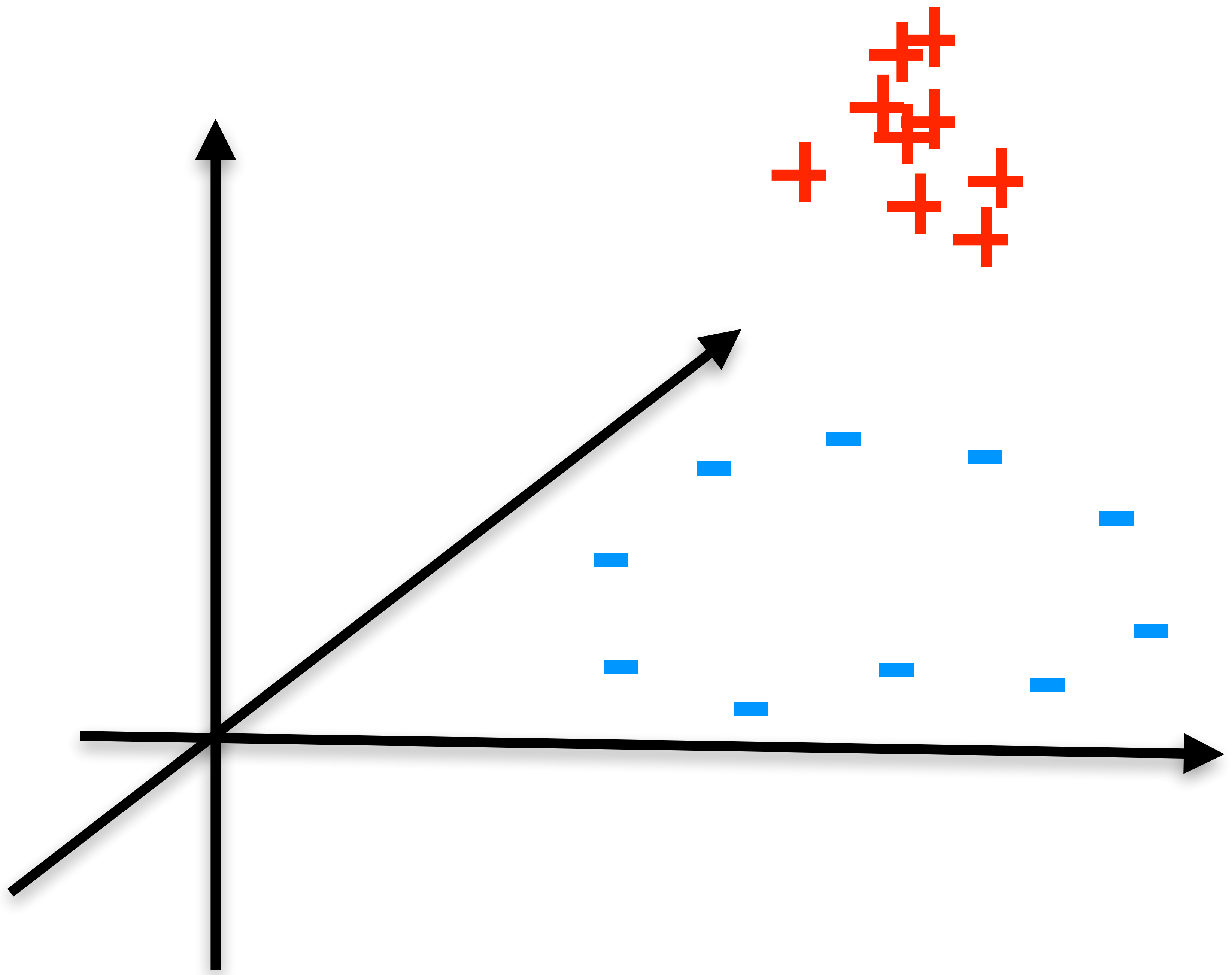
What if the data is like this?!

Support vector machines

Idea: we can project the data into an higher dimensional space where the separation might be clear!

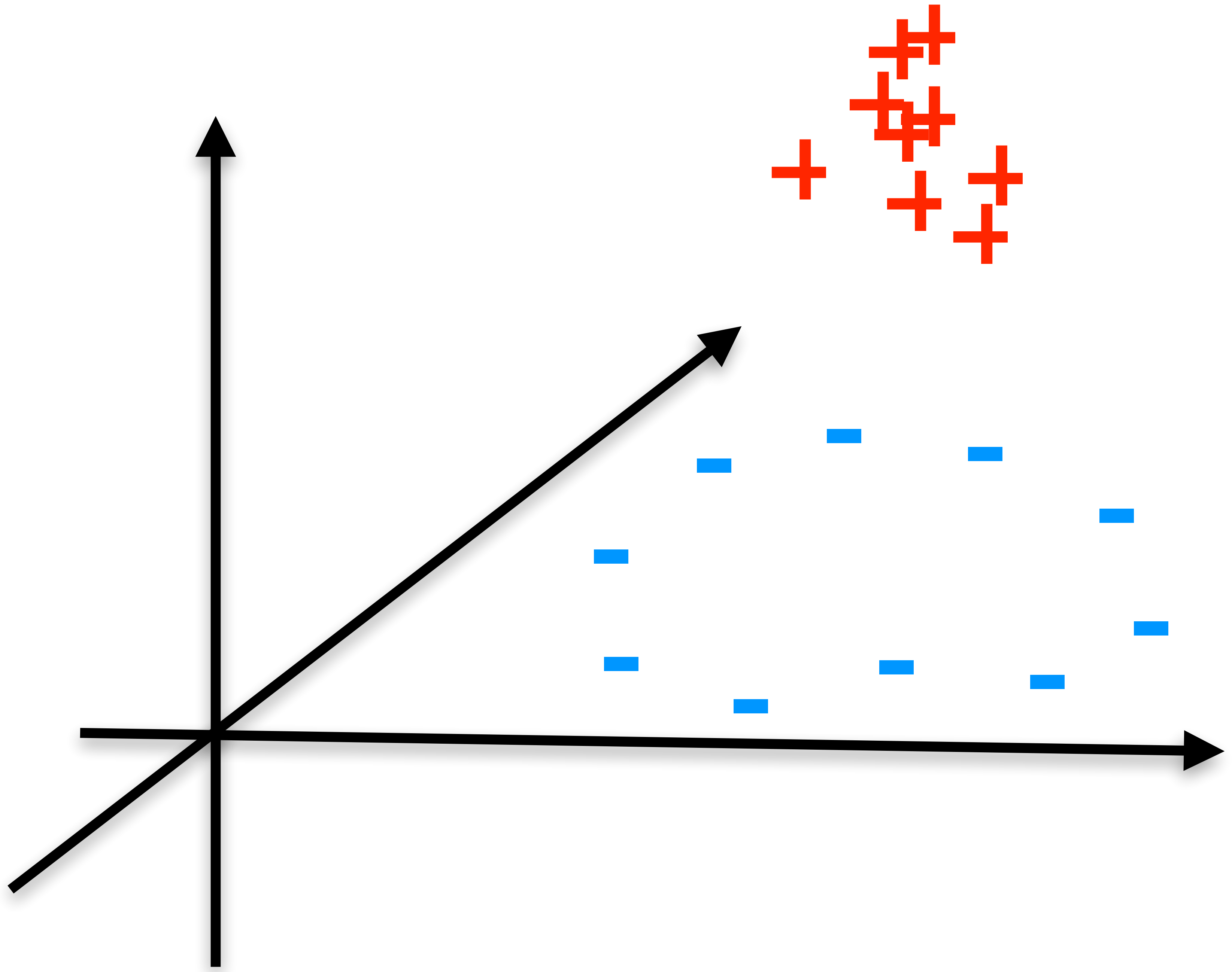
Support vector machines

Idea: we can project the data into an higher dimensional space where the separation might be clear!



Support vector machines

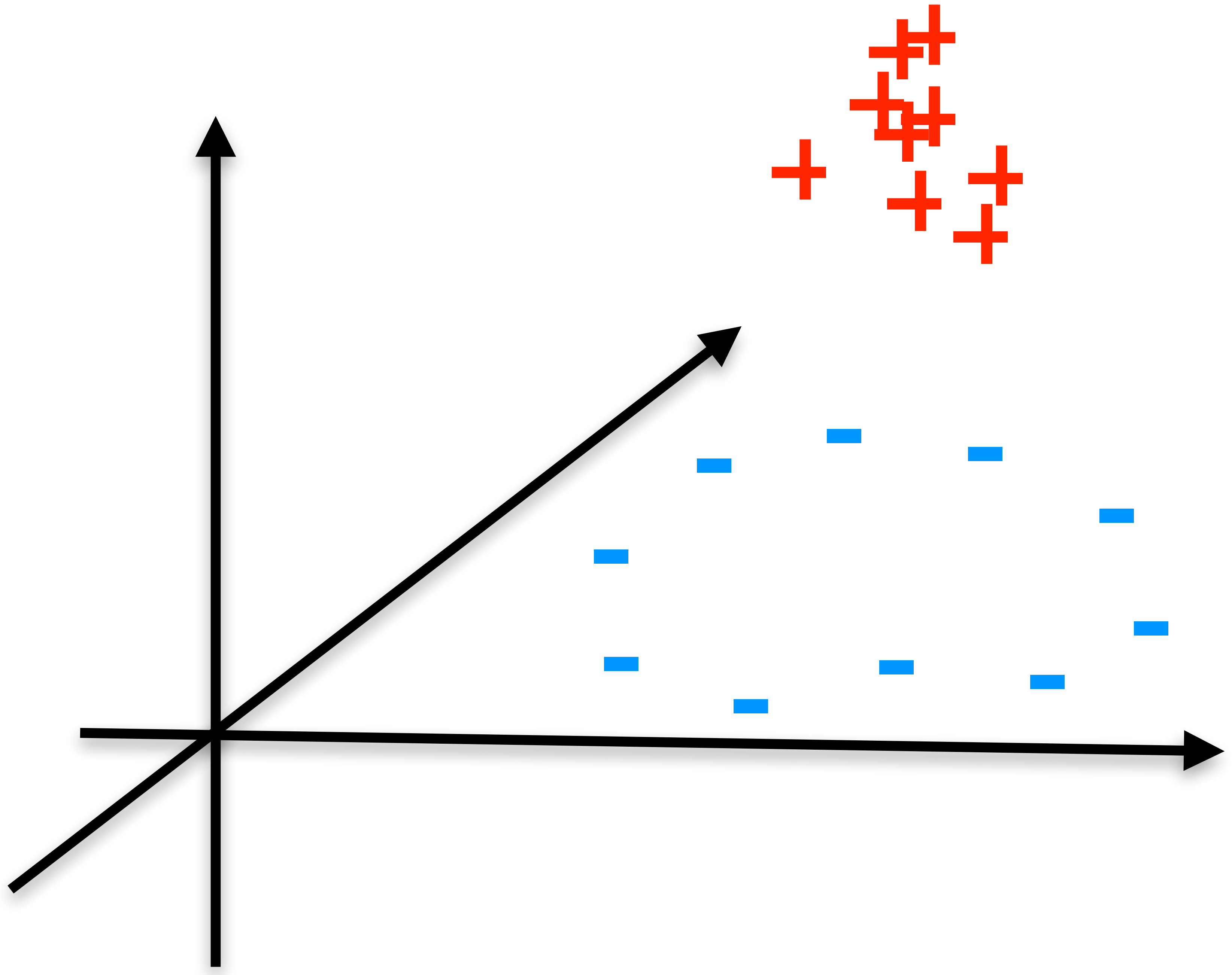
In this case, we moved from a two dimensional space (i.e., $\mathbf{x}_i \in \mathbb{R}^2$) to a three dimensional space (i.e., $\mathbf{x}_i' \in \mathbb{R}^3$)



Support vector machines

In this case, we moved from a two dimensional space (i.e., $\mathbf{x}_i \in \mathbb{R}^2$) to a three dimensional space (i.e., $\mathbf{x}'_i \in \mathbb{R}^3$)

The shift is done thanks to the so-called ***feature map***

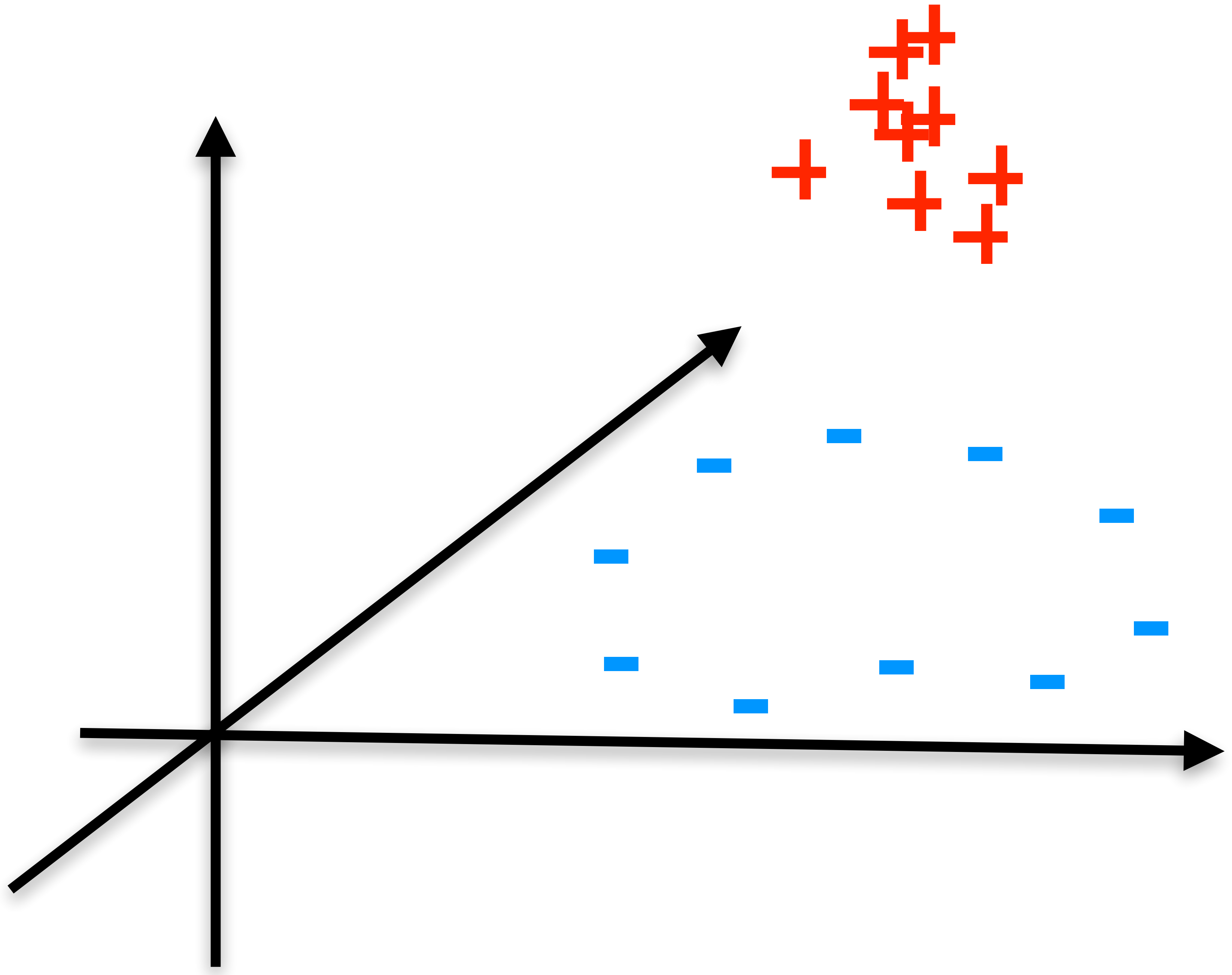


Support vector machines

In this case, we moved from a two dimensional space (i.e., $\mathbf{x}_i \in \mathbb{R}^2$) to a three dimensional space (i.e., $\mathbf{x}'_i \in \mathbb{R}^3$)

The shift is done thanks to the so-called ***feature map***

$$\mathbf{x}_i \rightarrow \phi(\mathbf{x}_i) = \mathbf{x}'_i$$



Feature maps

We already saw an example of feature map…

Feature maps

We already saw an example of feature map…

The polynomial basis!

Feature maps

We already saw an example of feature map…

The polynomial basis!

$$\mathbf{x}_i = (1, x_i)$$

Feature maps

We already saw an example of feature map…

The polynomial basis!

$$\mathbf{x}_i = (1, x_i)$$

$$\langle \mathbf{x}_i, \mathbf{w} \rangle = w_0 + w_1 x_i$$

Feature maps

We already saw an example of feature map…

The polynomial basis!

$$\mathbf{x}_i = (1, x_i) \qquad \phi(\mathbf{x}_i) = (1, x_i, x_i^2, \dots, x_i^k)$$

$$\langle \mathbf{x}_i, \mathbf{w} \rangle = w_0 + w_1 x_i$$

Feature maps

We already saw an example of feature map…

The polynomial basis!

$$\mathbf{x}_i = (1, x_i) \qquad \phi(\mathbf{x}_i) = (1, x_i, x_i^2, \ldots, x_i^k)$$

$$\langle \mathbf{x}_i, \mathbf{w} \rangle = w_0 + w_1 x_i \qquad \langle \phi(\mathbf{x}_i), \mathbf{w} \rangle = w_0 + w_1 x_i + \ldots + w_d x_i^k$$

Feature maps

Feature maps can be anything

For example

Feature maps

Feature maps can be anything

For example

$$\mathbf{x} = (1, x_1, x_2)$$

Feature maps

Feature maps can be anything

For example

$$\mathbf{x} = (1, x_1, x_2)$$

$$\langle \mathbf{x}, \mathbf{w} \rangle = w_0 + w_1 x_1 + w_2 x_2$$

Feature maps

Feature maps can be anything

For example

$$\mathbf{x} = (1, x_1, x_2) \qquad \phi(\mathbf{x}) = (1, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

$$\langle \mathbf{x}, \mathbf{w} \rangle = w_0 + w_1x_1 + w_2x_2$$

Feature maps

Feature maps can be anything

For example

$$\mathbf{x} = (1, x_1, x_2) \qquad \phi(\mathbf{x}) = (1, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

$$\langle \mathbf{x}, \mathbf{w} \rangle = w_0 + w_1x_1 + w_2x_2 \qquad \langle \phi(\mathbf{x}), \mathbf{w} \rangle = w_0 + w_1x_1^2 + w_2x_2^2 + \sqrt{2}w_3x_1x_2$$

Feature maps

Given a feature map, like for example $\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$

Feature maps

Given a feature map, like for example $\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$

We can compute the scalar product in the feature space

Feature maps

Given a feature map, like for example $\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$

We can compute the scalar product in the feature space

$$\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle = (x_1^2, x_2^2, \sqrt{2}x_1x_2)\begin{pmatrix} z_1^2 \\ z_2^2 \\ \sqrt{2}z_1z_2 \end{pmatrix} = (x_1z_1 + x_2z_2)^2 = \langle \mathbf{x}, \mathbf{z} \rangle^2$$

Feature maps

Given a feature map, like for example $\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$

We can compute the scalar product in the feature space

$$\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle = (x_1^2, x_2^2, \sqrt{2}x_1x_2) \begin{pmatrix} z_1^2 \\ z_2^2 \\ \sqrt{2}z_1z_2 \end{pmatrix} = (x_1z_1 + x_2z_2)^2 = \langle \mathbf{x}, \mathbf{z} \rangle^2$$

This is called kernel function $\kappa(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^2$

Feature maps

Given a feature map, like for example $\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$

We can compute the scalar product in the feature space

$$\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle = (x_1^2, x_2^2, \sqrt{2}x_1x_2) \begin{pmatrix} z_1^2 \\ z_2^2 \\ \sqrt{2}z_1z_2 \end{pmatrix} = (x_1z_1 + x_2z_2)^2 = \langle \mathbf{x}, \mathbf{z} \rangle^2$$

This is called kernel function $\kappa(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle^2$

Note how to compute the kernel function we do not need to know expression of phi! This is the kernel trick

Kernel trick

A big advantage of the kernel trick is that we do not need to specify $\phi(x)$ explicitly

Kernel trick

A big advantage of the kernel trick is that we do not need to specify $\phi(x)$ explicitly

For a *kernel function* $\kappa(\mathbf{x}, \mathbf{z})$ we can define a matrix $\mathbf{K}$ as $K_{ij} := \kappa(x_i, z_j)$

Kernel trick

A big advantage of the kernel trick is that we do not need to specify $\phi(x)$ explicitly

For a *kernel function* $\kappa(\mathbf{x}, \mathbf{z})$ we can define a matrix $\mathbf{K}$ as $K_{ij} := \kappa(x_i, z_j)$

Examples:

Kernel trick

A big advantage of the kernel trick is that we do not need to specify $\phi(x)$ explicitly

For a *kernel function* $\kappa(\mathbf{x}, \mathbf{z})$ we can define a matrix $\mathbf{K}$ as $K_{ij} := \kappa(x_i, z_j)$

Examples: $\phi(\mathbf{x}) = \mathbf{x}$ $\Rightarrow$ $\kappa(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x})^T \phi(\mathbf{z}) = \mathbf{x}^T \mathbf{z}$

Kernel trick

A big advantage of the kernel trick is that we do not need to specify $\phi(x)$ explicitly

For a *kernel function* $\kappa(\mathbf{x}, \mathbf{z})$ we can define a matrix $\mathbf{K}$ as $K_{ij} := \kappa(x_i, z_j)$

Examples:

$$\phi(\mathbf{x}) = \mathbf{x} \quad \Rightarrow \quad \kappa(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x})^T \phi(\mathbf{z}) = \mathbf{x}^T \mathbf{z}$$

$$\phi(\mathbf{x}) = \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_3^2 \\ \sqrt{2} x_1 x_2 \\ \sqrt{2} x_1 x_3 \\ \sqrt{2} x_2 x_3 \end{pmatrix} \quad \Rightarrow \quad \kappa(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x})^T \phi(\mathbf{z}) = \left(x_1 z_1 + x_2 z_2 + x_3 z_3\right)^2$$

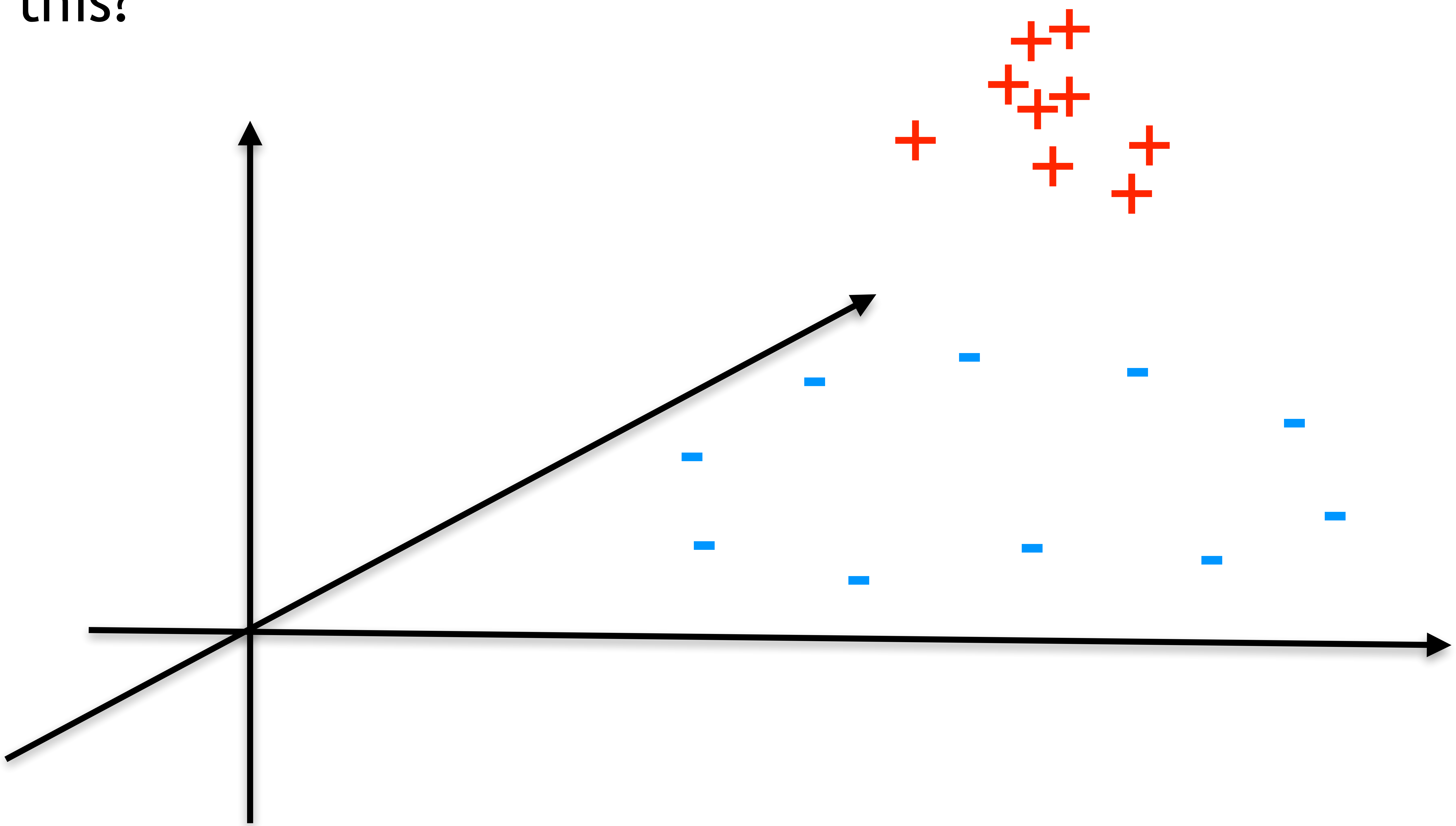
Kernel trick

Working with $\mathbf{K}$ instead of $\phi(\mathbf{X})$ is known as the *kernel* trick

$$\mathbf{K} = \Phi(\mathbf{X})^T \Phi(\mathbf{X}) = \begin{pmatrix} \|\phi(\mathbf{x}_1)\|^2 & \langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \rangle & \cdots & \langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_s) \rangle \\ \langle \phi(\mathbf{x}_2), \phi(\mathbf{x}_1) \rangle & \|\phi(\mathbf{x}_2)\|^2 & \cdots & \langle \phi(\mathbf{x}_2), \phi(\mathbf{x}_s) \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \phi(\mathbf{x}_s), \phi(\mathbf{x}_1) \rangle & \langle \phi(\mathbf{x}_s), \phi(\mathbf{x}_2) \rangle & \cdots & \|\phi(\mathbf{x}_s)\|^2 \end{pmatrix}$$

Kernel trick

How can we get this?



Kernel trick

By using a *radial basis function* kernel

$$\kappa(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{z})^T(\mathbf{x} - \mathbf{z})\right)$$

In summary, we can just define the transformation needed, considering the kernel function, without needing to explicitly define the feature map that does that!

Kernel trick

Kernel trick

When does there exist a corresponding feature-map?

Kernel trick

When does there exist a corresponding feature-map?

1.) $\mathbf{K}$ with $K_{ij} := \kappa(x_i, z_j)$ should be symmetric, *i.e.*

$$\kappa(\mathbf{x}, \mathbf{z}) = \kappa(\mathbf{z}, \mathbf{x}) \qquad \forall \mathbf{x}, \mathbf{z} \in \mathbb{R}^n$$

2.) $\mathbf{K}$ should be positive semi-definite, *i.e.*

$$\mathbf{x}^T \mathbf{K} \mathbf{x} \geq 0 \qquad \forall \mathbf{x} \in \mathbb{R}^n$$

Kernel SVM

Recall the SVM problem

$$\hat{\lambda} = \arg \max_{\lambda \in [0,1]^s} \left\{ \langle \boldsymbol{\lambda}, \mathbf{1} \rangle - \frac{1}{2\alpha} \|\mathbf{X}^T \mathbf{Y} \boldsymbol{\lambda}\|^2 \right\}$$

Kernel SVM

Recall the SVM problem

$$\hat{\lambda} = \arg \max_{\lambda \in [0,1]^s} \left\{ \langle \boldsymbol{\lambda}, \mathbf{1} \rangle - \frac{1}{2\alpha} \|\mathbf{X}^T \mathbf{Y} \boldsymbol{\lambda}\|^2 \right\}$$

Gradient of differentiable part $L(\lambda) = \langle \boldsymbol{\lambda}, \mathbf{1} \rangle - \frac{1}{2\alpha} \|\mathbf{X}^T \mathbf{Y} \boldsymbol{\lambda}\|^2$:

$$\nabla L(\lambda) = \mathbf{1} - \frac{1}{\alpha} \mathbf{Y}^T \mathbf{X} \mathbf{X}^T \mathbf{Y} \boldsymbol{\lambda} \qquad \rightarrow \nabla L(\lambda) = \mathbf{1} - \frac{1}{\alpha} \mathbf{Y}^T \mathbf{K} \mathbf{Y} \boldsymbol{\lambda}$$

Kernel SVM

Recall the SVM problem

$$\hat{\lambda} = \arg \max_{\lambda \in [0,1]^s} \left\{ \langle \lambda, \mathbf{1} \rangle - \frac{1}{2\alpha} \|\mathbf{X}^T \mathbf{Y} \lambda\|^2 \right\}$$

Gradient of differentiable part $L(\lambda) = \langle \lambda, \mathbf{1} \rangle - \frac{1}{2\alpha} \|\mathbf{X}^T \mathbf{Y} \lambda\|^2$:

$$\nabla L(\lambda) = \mathbf{1} - \frac{1}{\alpha} \mathbf{Y}^T \mathbf{X} \mathbf{X}^T \mathbf{Y} \lambda \qquad \rightarrow \nabla L(\lambda) = \mathbf{1} - \frac{1}{\alpha} \mathbf{Y}^T \mathbf{K} \mathbf{Y} \lambda$$

Hence, any SVM-algorithm that works with this gradient can be *kernelised*