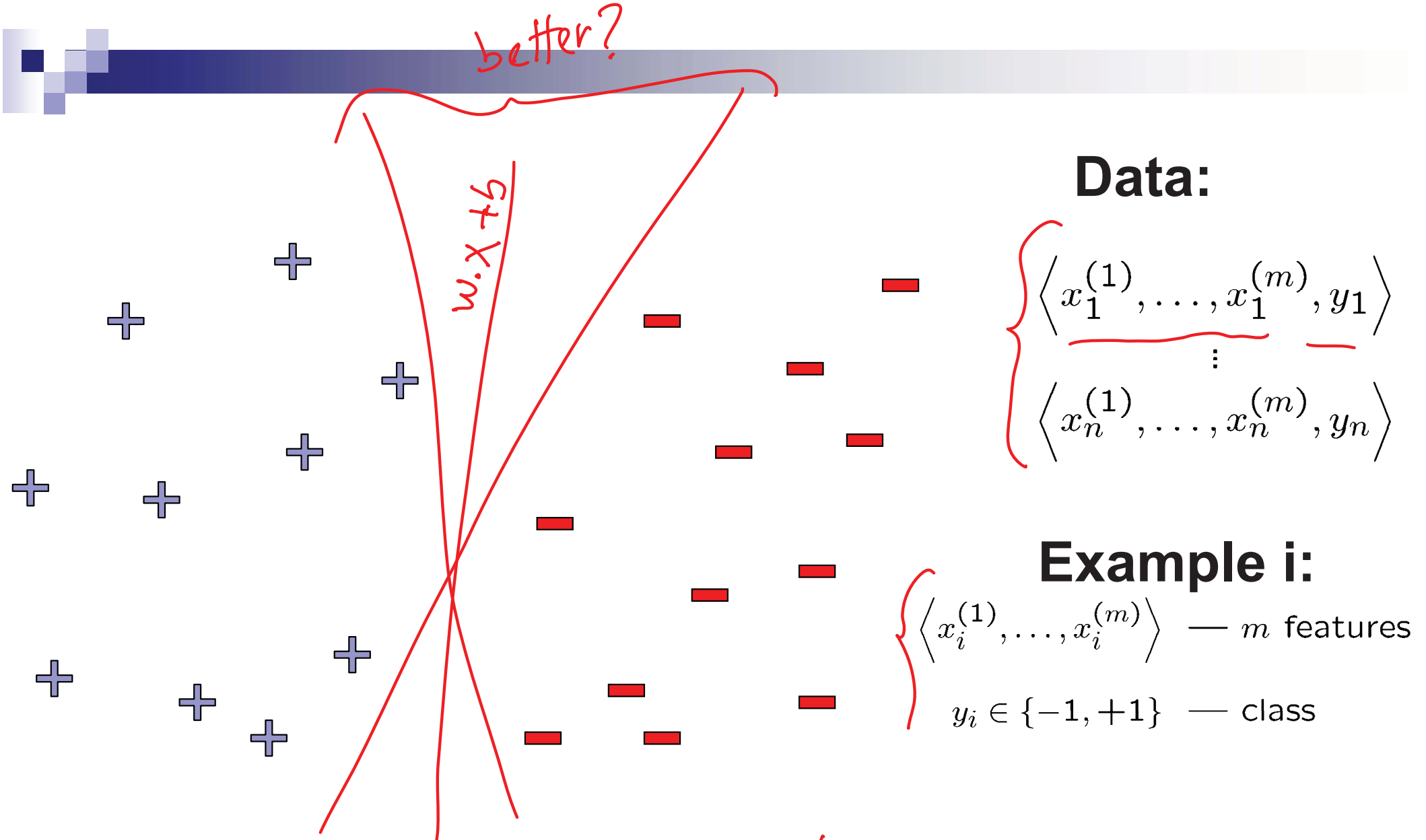


CS 175: Project in Artificial Intelligence

Support Vector Machine

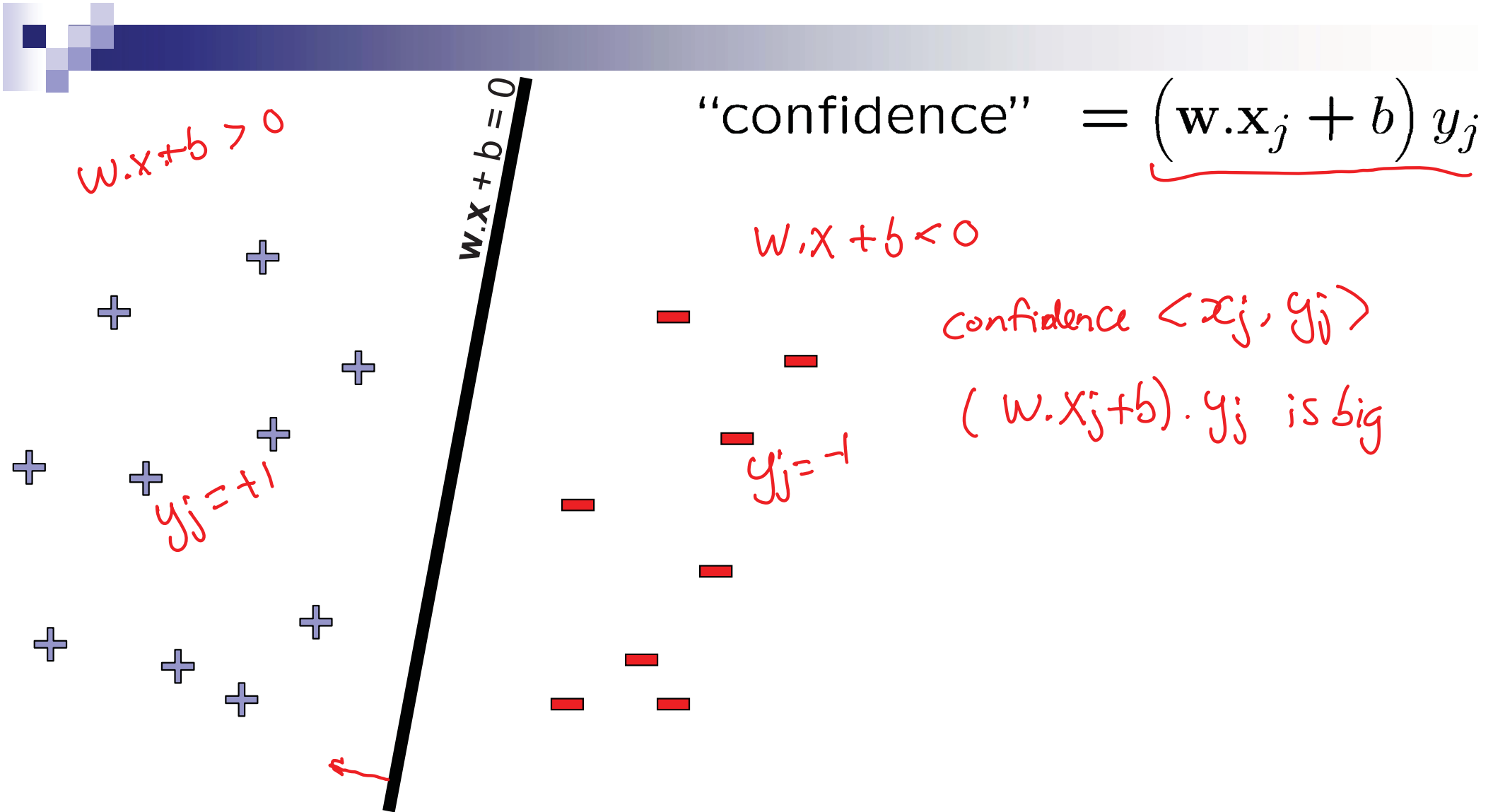
Combined slides from [Carlos Guestrin](#) and [Chih-Jen Lin](#)

Linear classifiers – Which line is better?



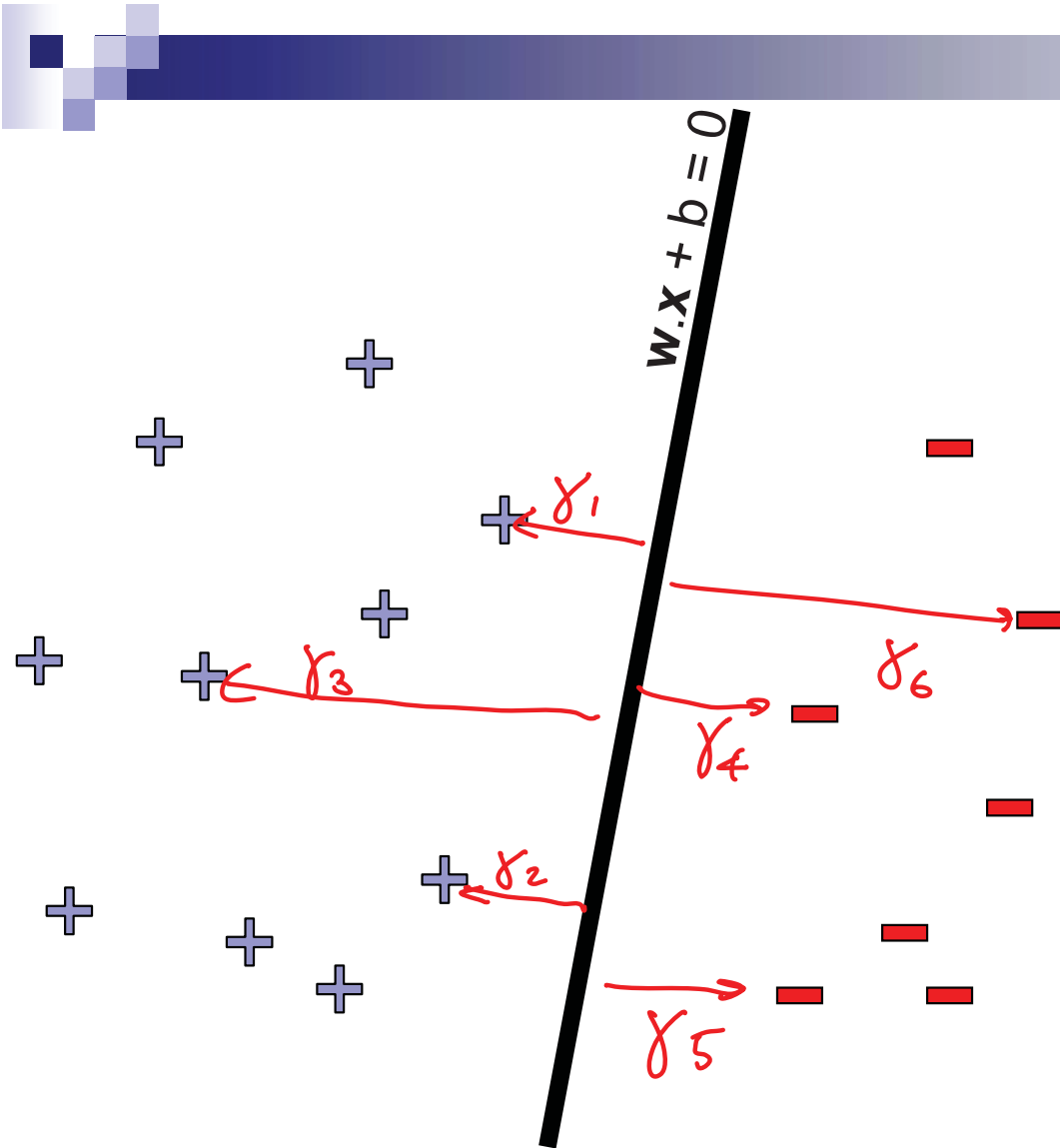
$$\mathbf{w} \cdot \mathbf{x} = \sum_j w^{(j)} x^{(j)} \leftarrow \text{dot product}$$

Pick the one with the largest margin!



$$w \cdot x = \sum_j w^{(j)} x^{(j)}$$

Maximize the margin

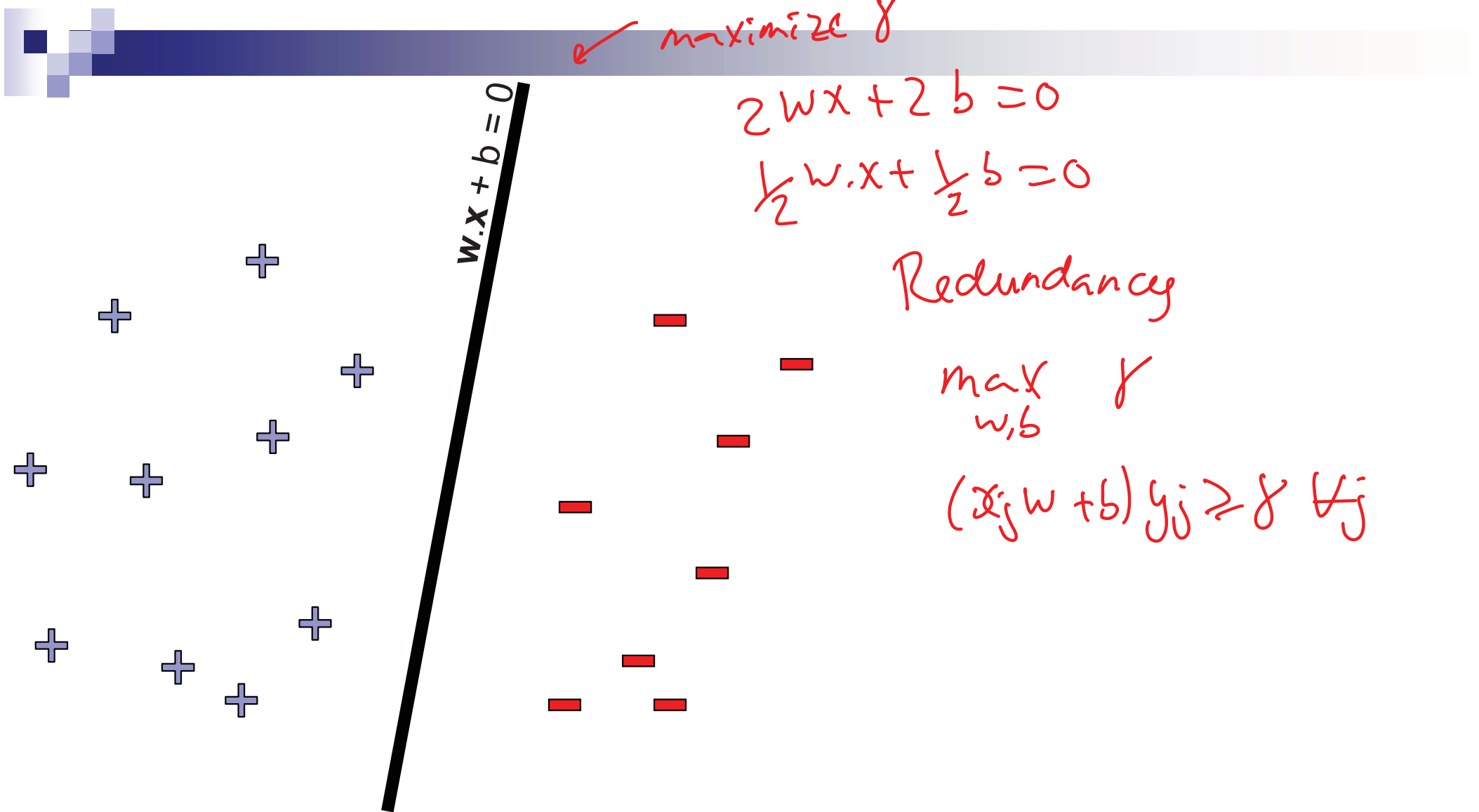


$$(x_1 \cdot w + b) y_1 = \delta_1$$
$$(x_j \cdot w + b) \cdot y_j = \delta_j$$
$$(x_n \cdot w + b) y_n = \delta_n$$

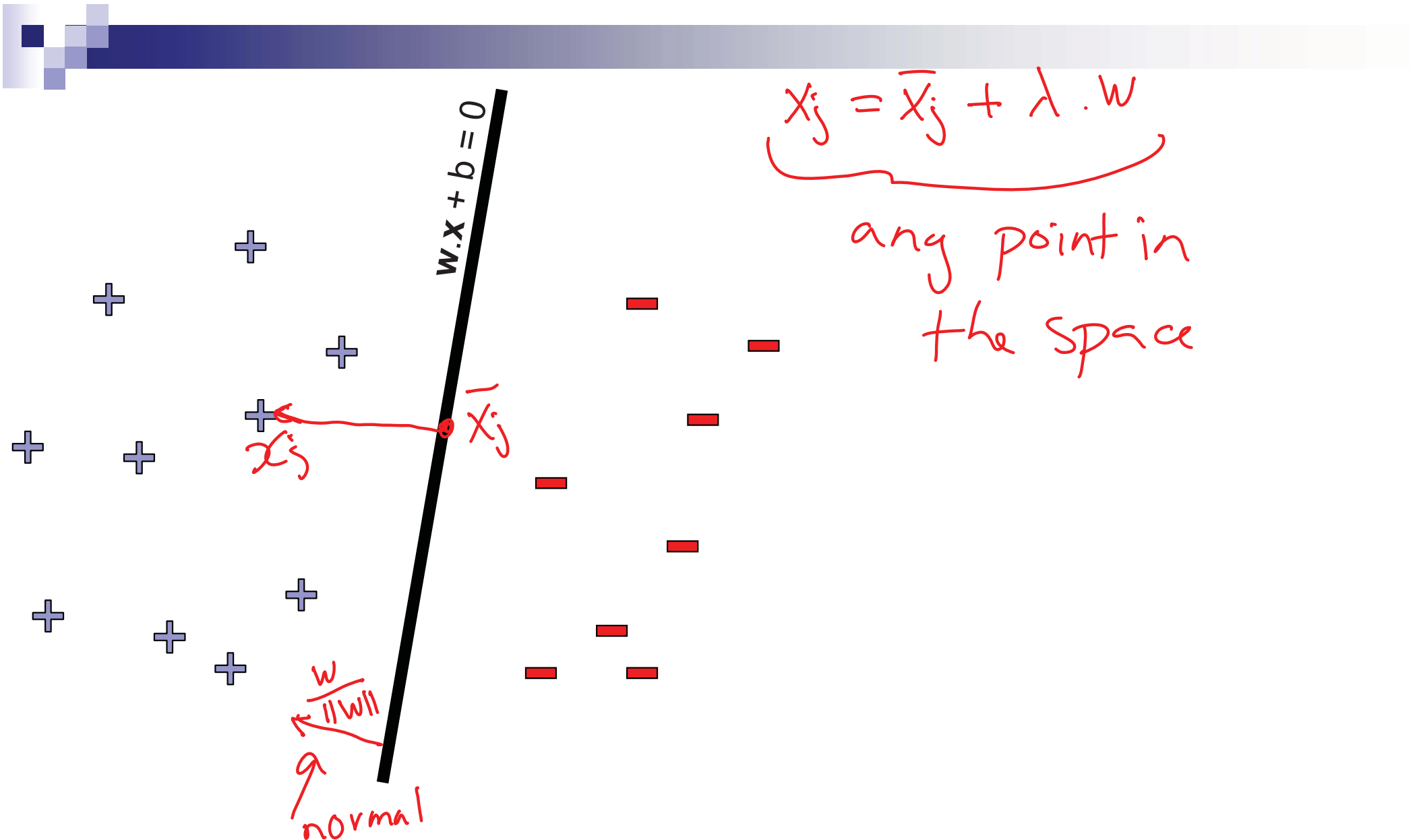
$$\delta = \min_j \delta_j$$

$$\boxed{\begin{array}{l} \text{maximize } \delta \\ w, b \end{array}}$$
$$(x_j w + b) y_j \geq \delta, \forall j$$

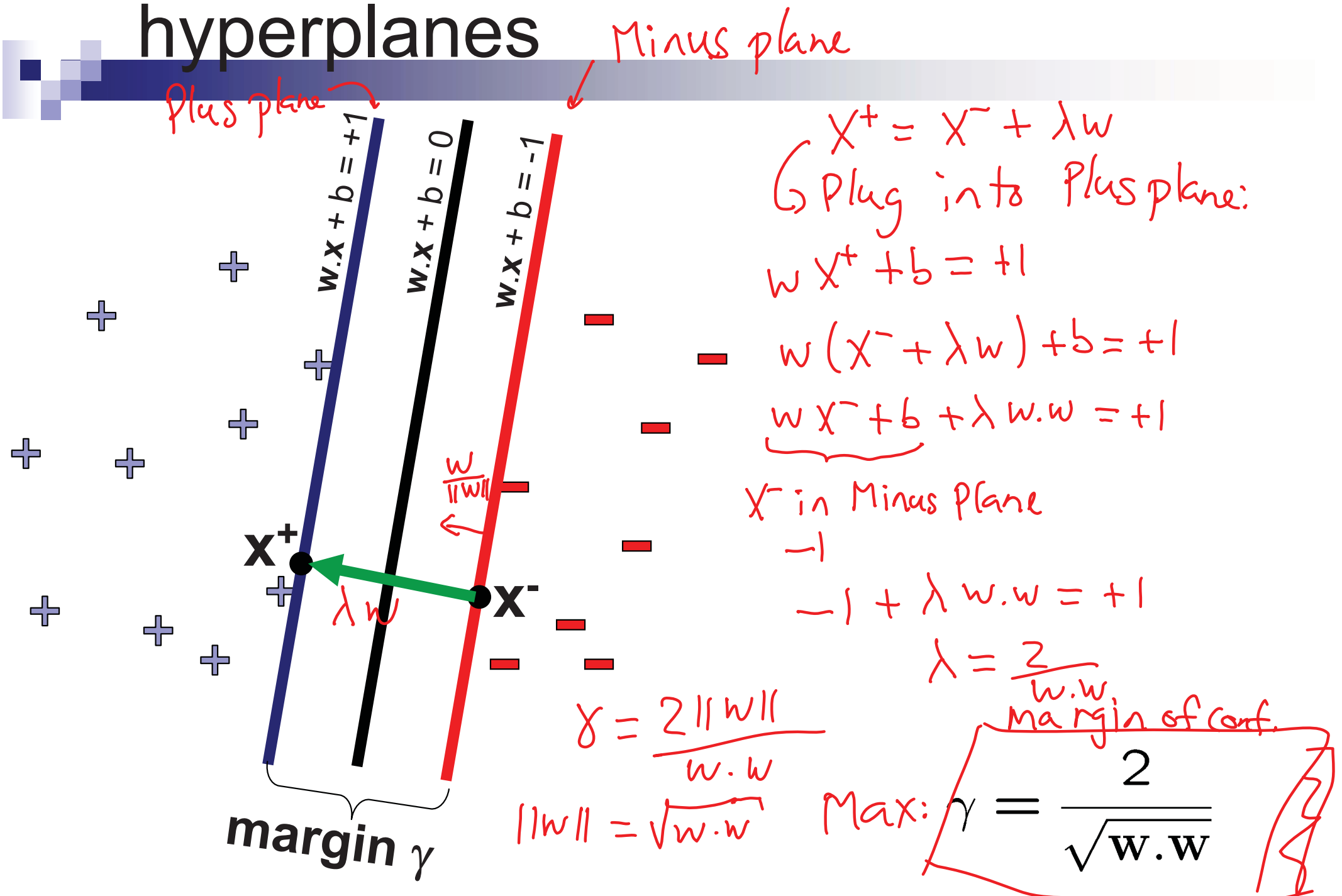
But there are a many planes...



Review: Normal to a plane



Normalized margin – Canonical hyperplanes



- Distance between $\mathbf{w}^T \mathbf{x} + b = 1$ and -1 :

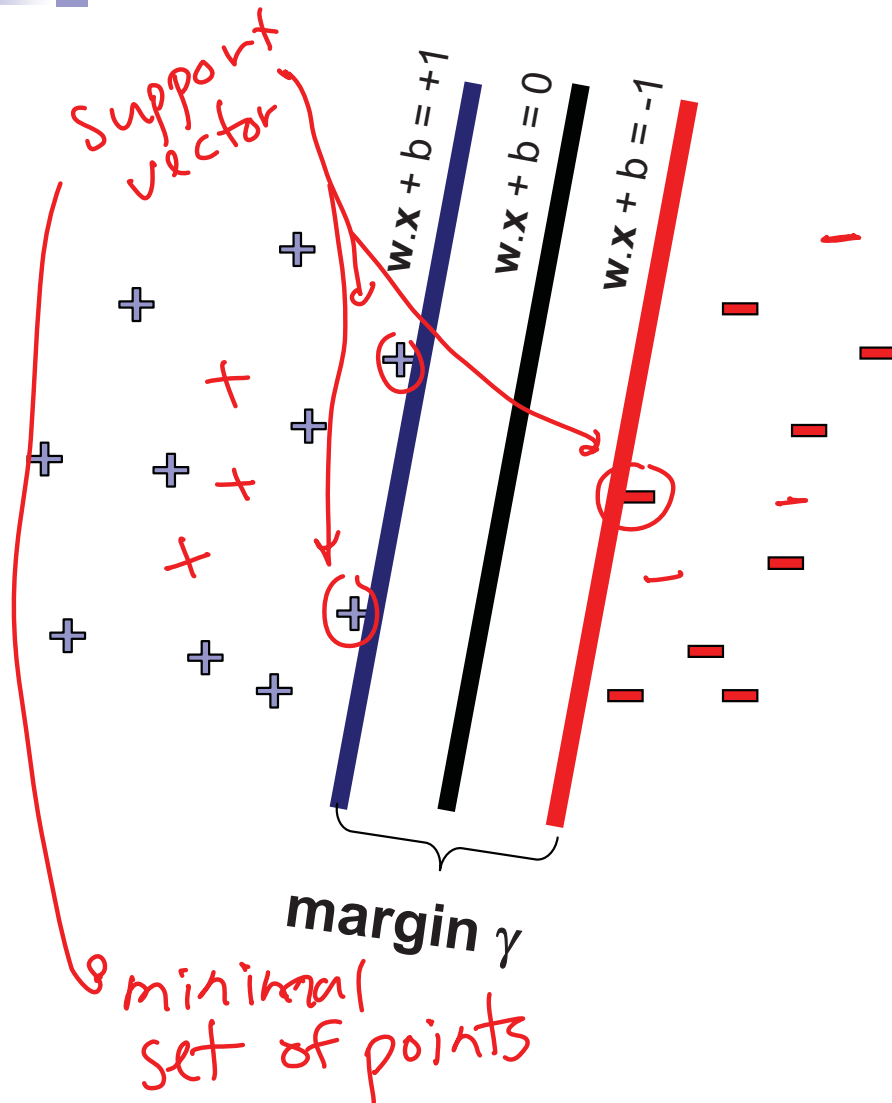
$$2/\|\mathbf{w}\| = 2/\sqrt{\mathbf{w}^T \mathbf{w}}$$

- $\max 2/\|\mathbf{w}\| \equiv \min \mathbf{w}^T \mathbf{w}/2$

$$\begin{array}{ll} \min_{\mathbf{w}, b} & \frac{1}{2} \mathbf{w}^T \mathbf{w} \\ \text{subject to} & y_i((\mathbf{w}^T \mathbf{x}_i) + b) \geq 1, \\ & i = 1, \dots, l. \end{array}$$

Primal Problem

Support vector machines (SVMs)

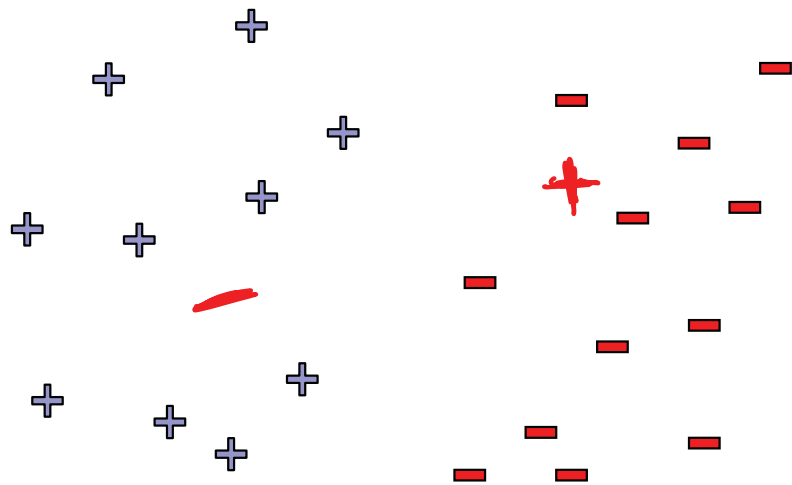


$$\text{minimize}_w \quad w \cdot w$$
$$(w \cdot x_j + b) y_j \geq 1, \quad \forall j$$

- Solve efficiently by quadratic programming (QP)
 - Well-studied solution algorithms
- Hyperplane defined by support vectors

What if the data is not linearly separable?

Use features of features of features....



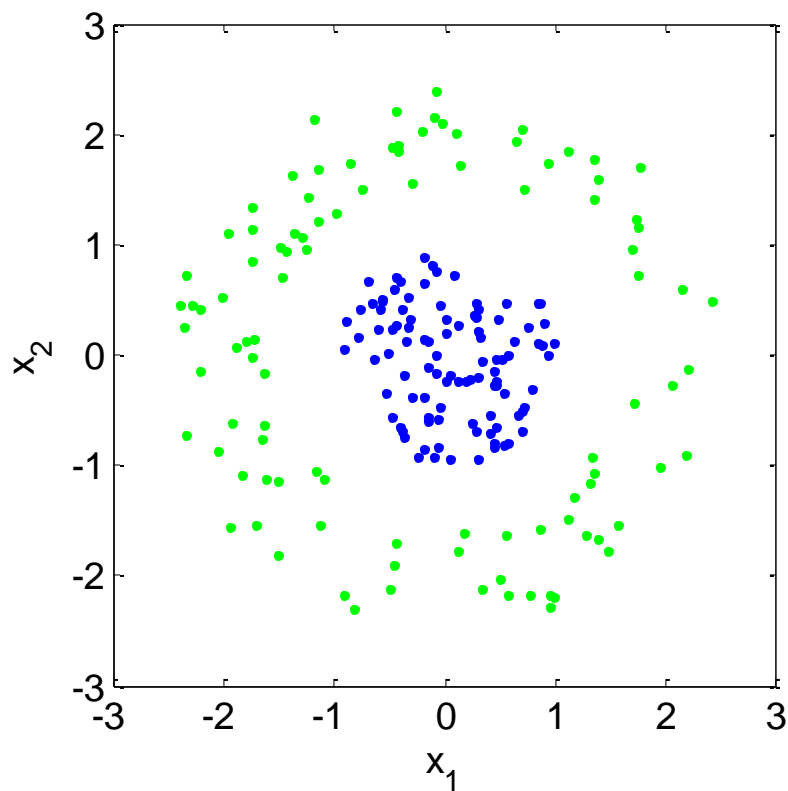
not linearly separable
 $\Rightarrow \nexists$ hyperplane
 $\gamma > 0$

2d $\langle x^{(1)}, x^{(2)}, y \rangle$

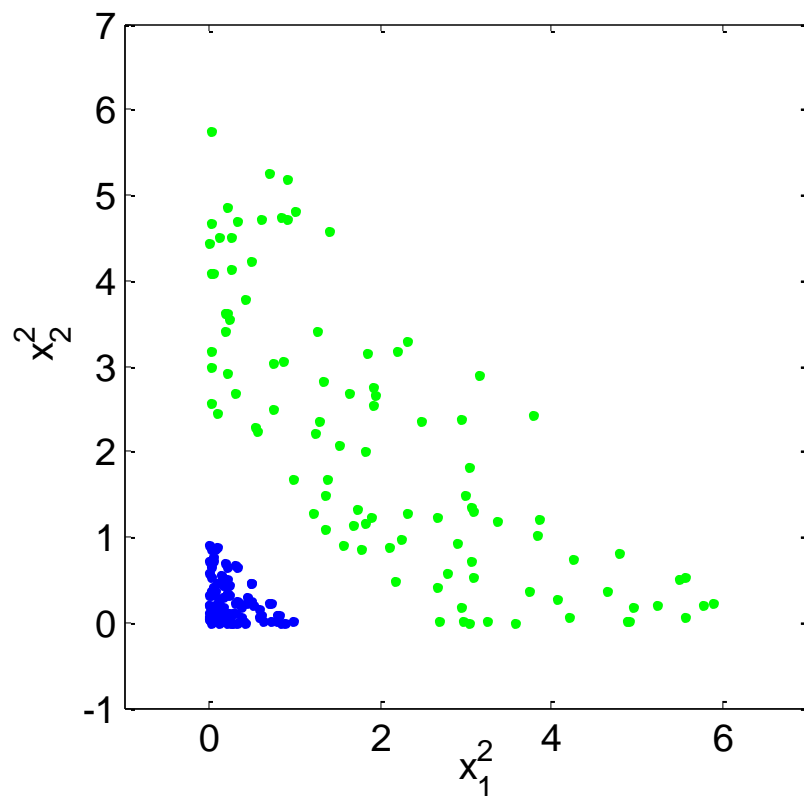
↓ Feed SVM:

$\langle x^{(1)}, x^{(2)}, (x^{(1)})^2, (x^{(2)})^2, x^{(1)} x^{(2)}, (x^{(1)})^3, (x^{(2)})^3, \dots, y \rangle$

polynomial features



$$\Phi = \langle x_1, x_2 \rangle$$



$$\Phi = \langle x_1, x_2, x_1^2, x_2^2, x_1x_2 \rangle$$

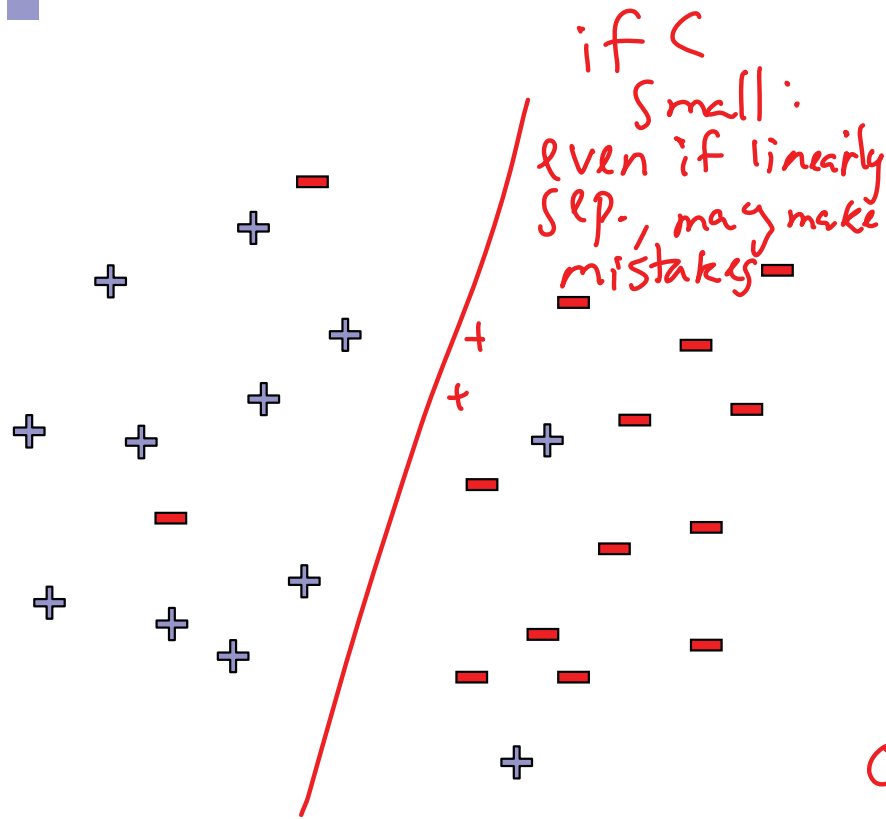
What if the data is still not linearly separable?



$$\text{minimize}_{\mathbf{w}} \quad \mathbf{w} \cdot \mathbf{w} + C (\# \text{ mistakes})$$
$$(\mathbf{w} \cdot \mathbf{x}_j + b) y_j \geq 1, \forall j$$

- Minimize $\mathbf{w} \cdot \mathbf{w}$ and number of training mistakes
 - Tradeoff two criteria?
- Tradeoff $\#(\text{mistakes})$ and $\mathbf{w} \cdot \mathbf{w}$
 - 0/1 loss
 - Slack penalty C
 - Not QP anymore
 - Also doesn't distinguish near misses and really bad mistakes

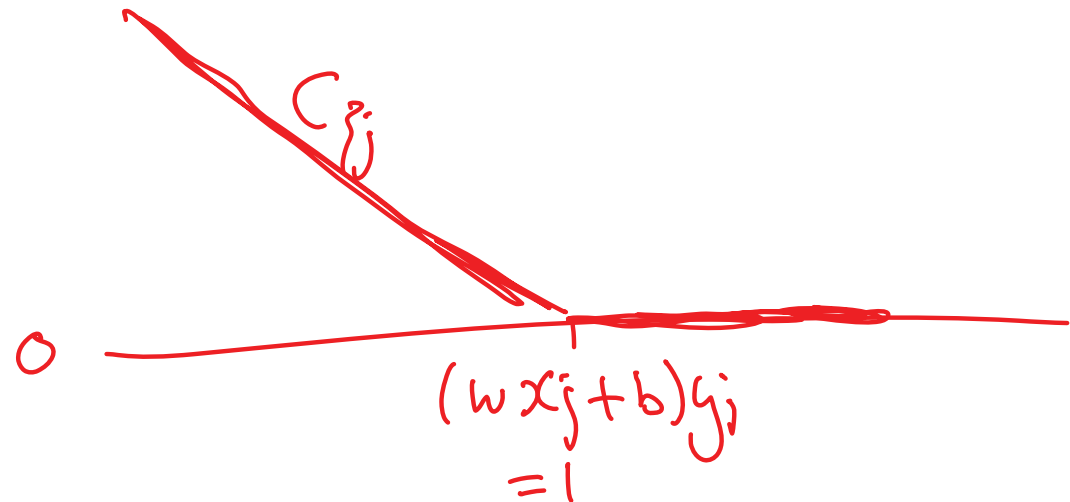
Slack variables – Hinge loss



$$\text{minimize}_w \quad w \cdot w + C \sum_j \xi_j$$

$$(w \cdot x_j + b) y_j \geq 1 - \xi_j, \forall j$$

$$\xi_j \geq 0$$



- If margin ≥ 1 , don't care $\Rightarrow \xi_j = 0$, pay nothing
- If margin < 1 , pay linear $\Rightarrow \xi_j > 0$, and pay $C \cdot \xi_j$ penalty

- Non-separable case: **allow training errors**

$$\min_{\mathbf{w}, b, \xi} \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^l \xi_i$$

$$y_i((\mathbf{w}^T \mathbf{x}_i) + b) \geq 1 - \xi_i,$$

$$\xi_i \geq 0, \quad i = 1, \dots, l$$

- $\xi_i > 1$, \mathbf{x}_i **not on the correct side** of the separating plane
- C : **large** penalty parameter, **most ξ_i are zero**

Side note: What's the difference between SVMs and logistic regression?

SVM:

$$\begin{aligned} \text{minimize}_{\mathbf{w}} \quad & \mathbf{w} \cdot \mathbf{w} + C \sum_j \xi_j \\ & (\mathbf{w} \cdot \mathbf{x}_j + b) y_j \geq 1 - \xi_j, \quad \forall j \\ & \xi_j \geq 0, \quad \forall j \end{aligned}$$

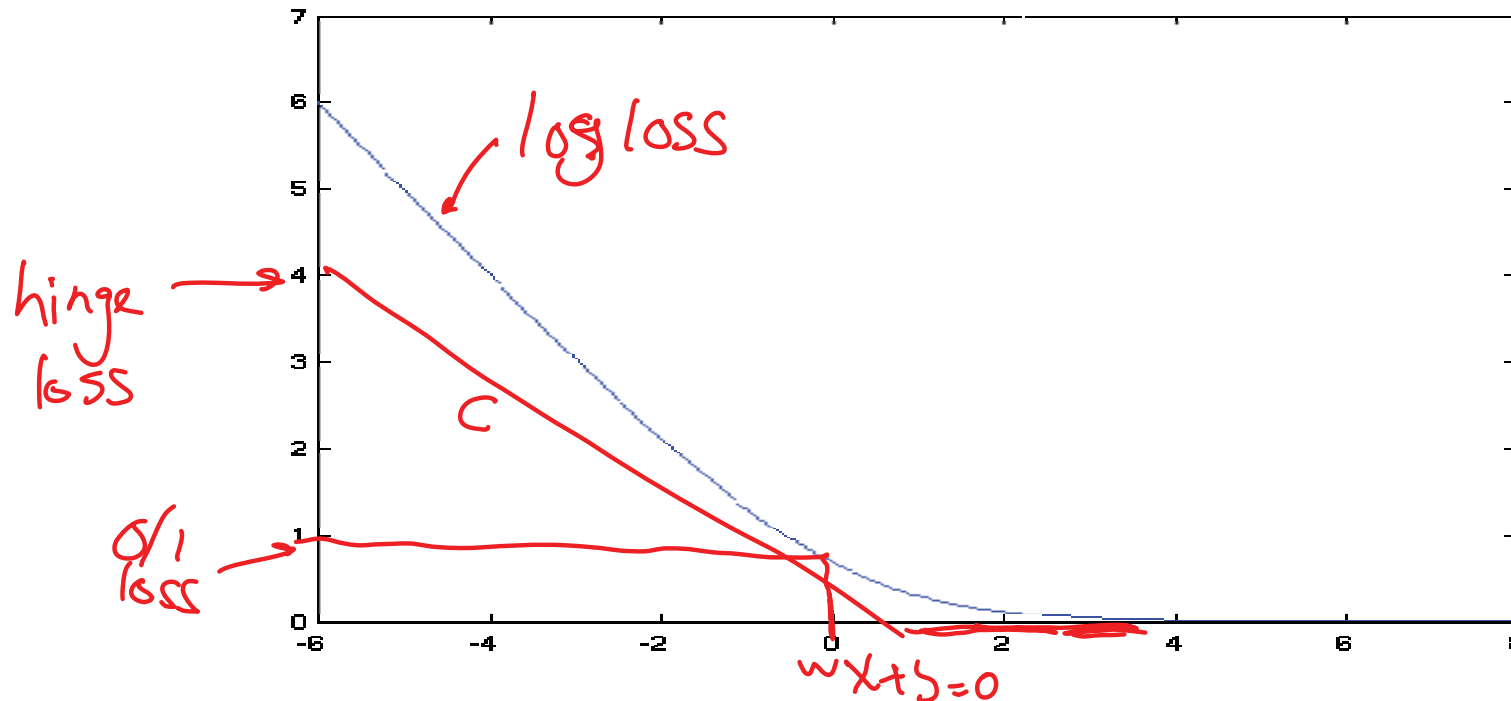
Hinge Loss:

Logistic regression:

$$P(Y = 1 | x, \mathbf{w}) = \frac{1}{1 + e^{-(\mathbf{w} \cdot \mathbf{x} + b)}}$$

Log loss:

$$\min -\ln P(Y = 1 | x, \mathbf{w}) = \ln(1 + e^{-(\mathbf{w} \cdot \mathbf{x} + b)})$$



Finding the Decision Function

- \mathbf{w} : a vector in a high dimensional space \Rightarrow maybe **infinite** variables
- The **dual** problem

$$\begin{array}{ll}\min_{\alpha} & \frac{1}{2} \alpha^T Q \alpha - \mathbf{e}^T \alpha \\ \text{subject to} & 0 \leq \alpha_i \leq C, i = 1, \dots, l \\ & \mathbf{y}^T \alpha = 0,\end{array}$$

where $Q_{ij} = y_i y_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ and $\mathbf{e} = [1, \dots, 1]^T$

$$\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \phi(\mathbf{x}_i)$$

- **Primal and dual**: optimization theory. Not trivial.

Infinite dimensional programming.

- A **finite** problem:

#variables = #training data

- $Q_{ij} = y_i y_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ needs a **closed** form

Efficient calculation of **high dimensional inner products**

Kernel trick, $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$

Decision function

- \mathbf{w} : maybe an **infinite** vector
- At optimum

$$\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \phi(\mathbf{x}_i)$$

- Decision function

$$\begin{aligned} & \mathbf{w}^T \phi(\mathbf{x}) + b \\ &= \sum_{i=1}^l \alpha_i y_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b \\ &= \sum_{i=1}^l \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \end{aligned}$$

No need to have \mathbf{w}

● > 0 : 1st class, < 0 : 2nd class

● Only $\phi(\mathbf{x}_i)$ of $\alpha_i > 0$ used

$\alpha_i > 0 \Rightarrow$ support vectors

● **Example:** $\mathbf{x}_i \in R^3, \phi(\mathbf{x}_i) \in R^{10}$

$$\begin{aligned} \phi(\mathbf{x}_i) = & (1, \sqrt{2}(x_i)_1, \sqrt{2}(x_i)_2, \sqrt{2}(x_i)_3, (x_i)_1^2, \\ & (x_i)_2^2, (x_i)_3^2, \sqrt{2}(x_i)_1(x_i)_2, \sqrt{2}(x_i)_1(x_i)_3, \sqrt{2}(x_i)_2(x_i)_3) \end{aligned}$$

Then $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^2$.

● **Popular methods:** $K(\mathbf{x}_i, \mathbf{x}_j) =$

$$e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}, \text{ (Radial Basis Function)}$$

$$(\mathbf{x}_i^T \mathbf{x}_j / a + b)^d \text{ (Polynomial kernel)}$$

Kernel Tricks

- Kernel: $K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^T \phi(\mathbf{y})$
- **No need** to explicitly know $\phi(\mathbf{x})$
- Common kernels $K(\mathbf{x}_i, \mathbf{x}_j) =$

$$e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}, \text{ (Radial Basis Function)}$$

$$(\mathbf{x}_i^T \mathbf{x}_j / a + b)^d \text{ (Polynomial kernel)}$$

- They can be inner product in **infinite** dimensional space
- Assume $x \in R^1$ and $\gamma > 0$.

$$\begin{aligned}
e^{-\gamma\|x_i-x_j\|^2} &= e^{-\gamma(x_i-x_j)^2} = e^{-\gamma x_i^2 + 2\gamma x_i x_j - \gamma x_j^2} \\
&= e^{-\gamma x_i^2 - \gamma x_j^2} \left(1 + \frac{2\gamma x_i x_j}{1!} + \frac{(2\gamma x_i x_j)^2}{2!} + \frac{(2\gamma x_i x_j)^3}{3!} + \dots \right) \\
&= e^{-\gamma x_i^2 - \gamma x_j^2} \left(1 \cdot 1 + \sqrt{\frac{2\gamma}{1!}} x_i \cdot \sqrt{\frac{2\gamma}{1!}} x_j + \sqrt{\frac{(2\gamma)^2}{2!}} x_i^2 \cdot \sqrt{\frac{(2\gamma)^2}{2!}} x_j^2 \right. \\
&\quad \left. + \sqrt{\frac{(2\gamma)^3}{3!}} x_i^3 \cdot \sqrt{\frac{(2\gamma)^3}{3!}} x_j^3 + \dots \right) \\
&= \phi(x_i)^T \phi(x_j),
\end{aligned}$$

where

$$\phi(x) = e^{-\gamma x^2} \left[1, \sqrt{\frac{2\gamma}{1!}} x, \sqrt{\frac{(2\gamma)^2}{2!}} x^2, \sqrt{\frac{(2\gamma)^3}{3!}} x^3, \dots \right]^T.$$

SVM^{light}

- <http://svmlight.joachims.org/>
- `svm_learn` [options] example_file model_file
- `svm_classify` [options] example_file model_file output_file
- Input file format:

<line> .=. <target> <feature>:<value> <feature>:<value> ...

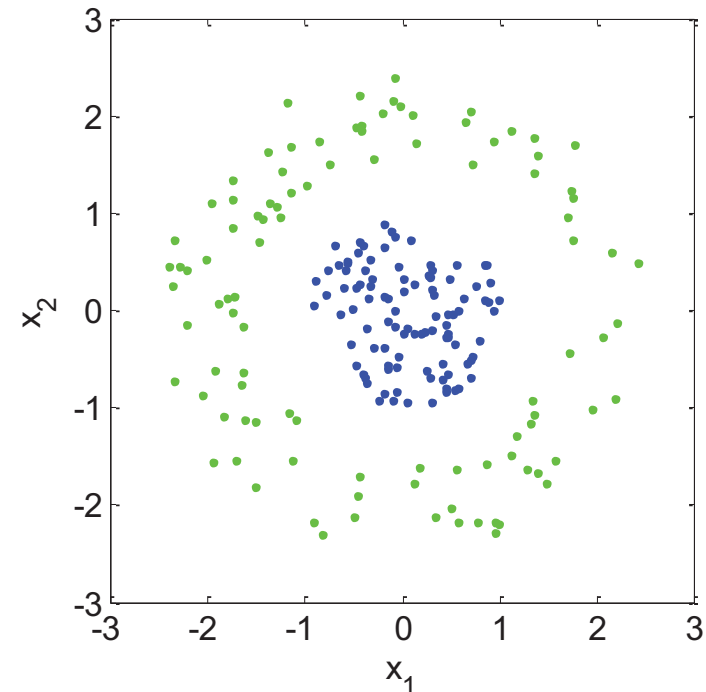
 <feature>:<value> # <info>

<target> .=. +1 | -1 | 0 | <float>

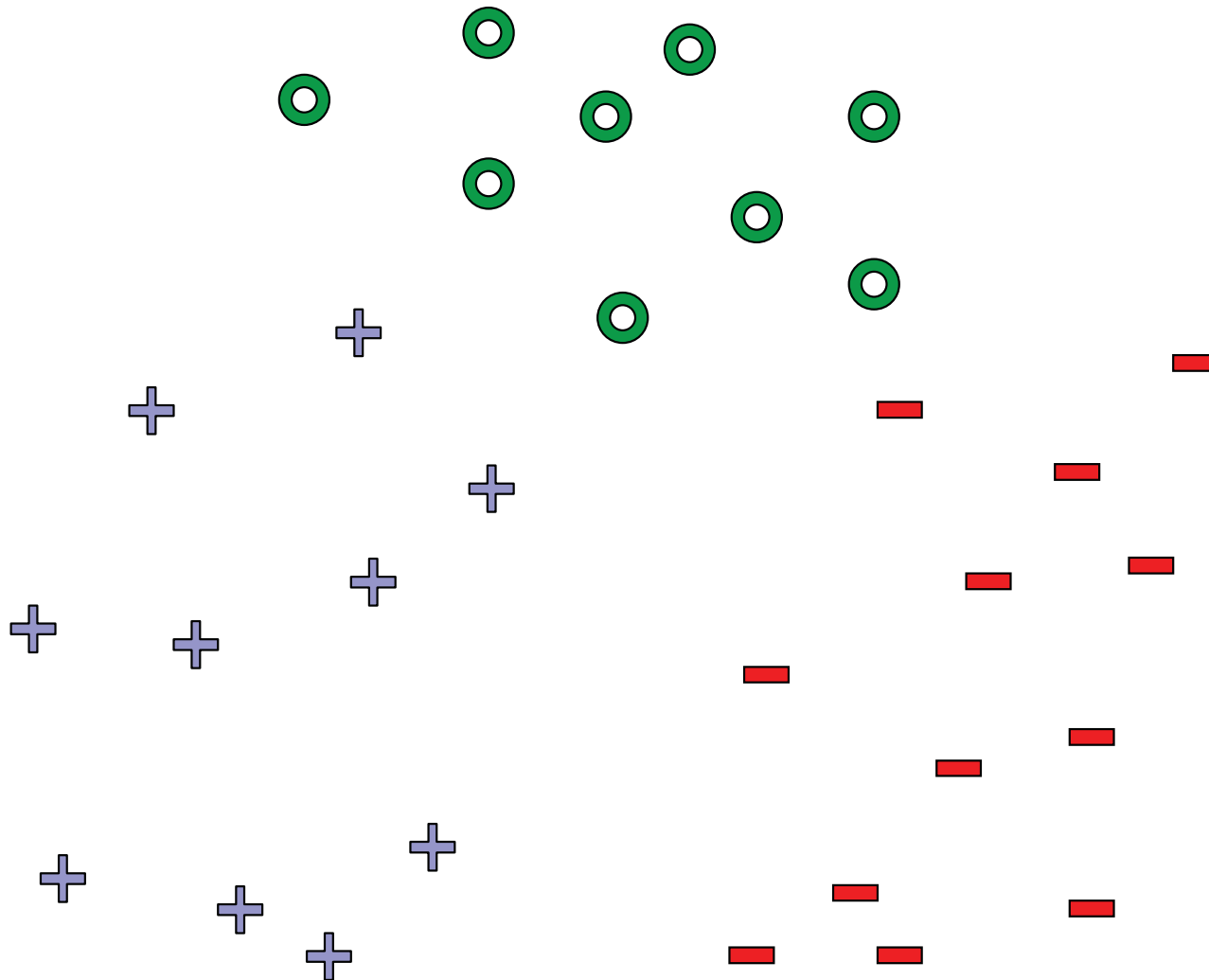
<feature> .=. <integer> | "qid"

<value> .=. <float>

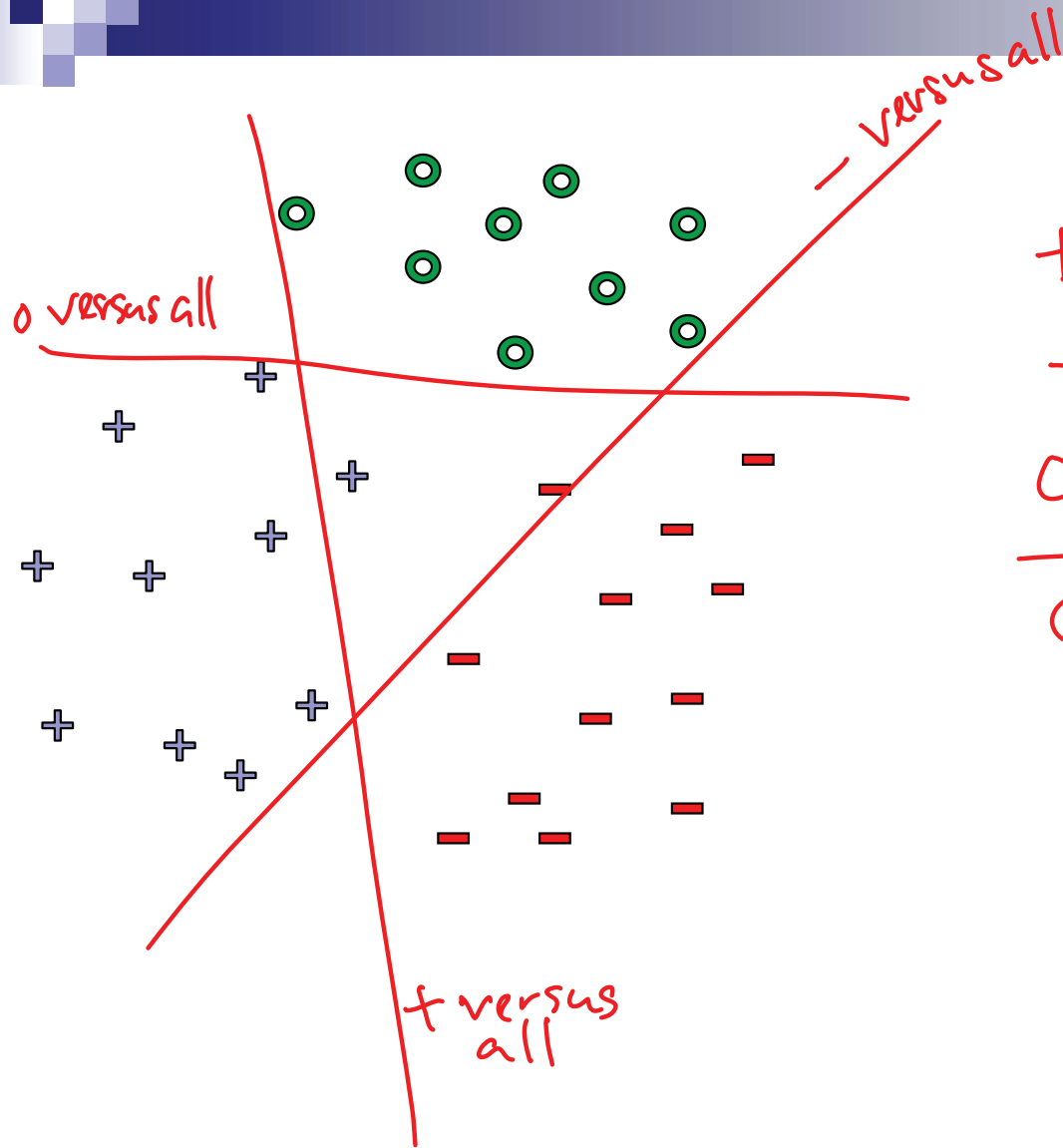
<info> .=. <string>



What about multiple classes?



One against All



Learn 3 classifiers:

+ versus $\{0, -\}$

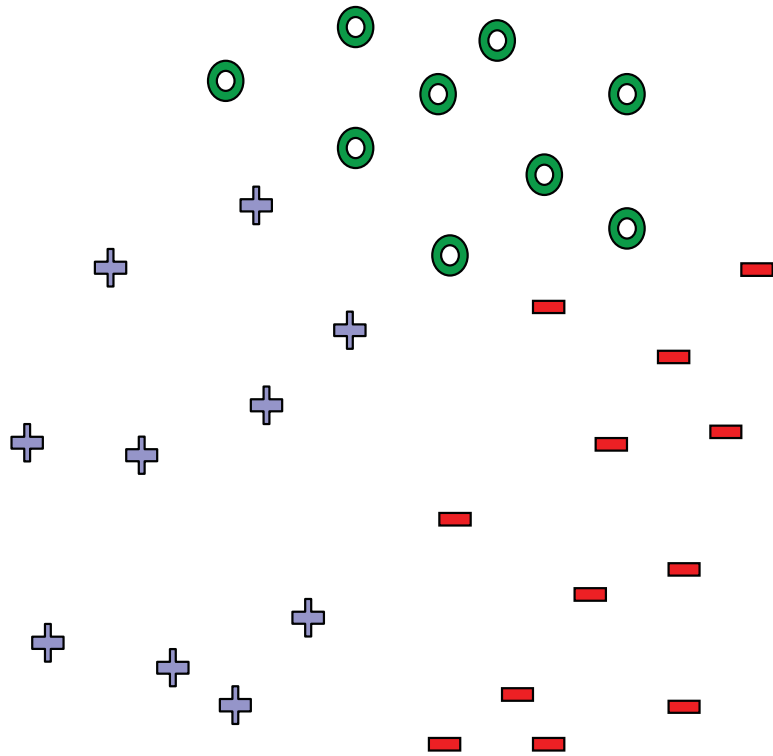
- versus $\{0, +\}$

0 versus $\{+, -\}$

classifier x
classifier with
highest confidence

Learn 1 classifier: Multiclass SVM

Simultaneously learn 3 sets of weights



For + examples:

$$(w^{(+)} x_j + b^{(+)}) \geq 1 + w^{(-)} x_j + b^{(-)}$$

$$w^{(+)} x_j + b^{(+)} \geq 1 + w^{(o)} x_j + b^{(o)}$$

For - examples

$w^{(-)}, b^{(-)}$ win

For o examples

$w^{(o)}, b^{(o)}$ win

$$w^{(y_j)} \cdot x_j + b^{(y_j)} \geq w^{(y')} \cdot x_j + b^{(y')} + 1, \quad \forall y' \neq y_j, \quad \forall j$$

Learn 1 classifier: Multiclass SVM



possible classes

$$\text{minimize}_{\mathbf{w}} \quad \sum_y \mathbf{w}^{(y)} \cdot \mathbf{w}^{(y)} + C \sum_j \xi_j$$

$$\mathbf{w}^{(y_j)} \cdot \mathbf{x}_j + b^{(y_j)} \geq \mathbf{w}^{(y')} \cdot \mathbf{x}_j + b^{(y')} + 1 - \xi_j, \quad \forall y' \neq y_j, \quad \forall j$$

$$\xi_j \geq 0, \quad \forall j$$

