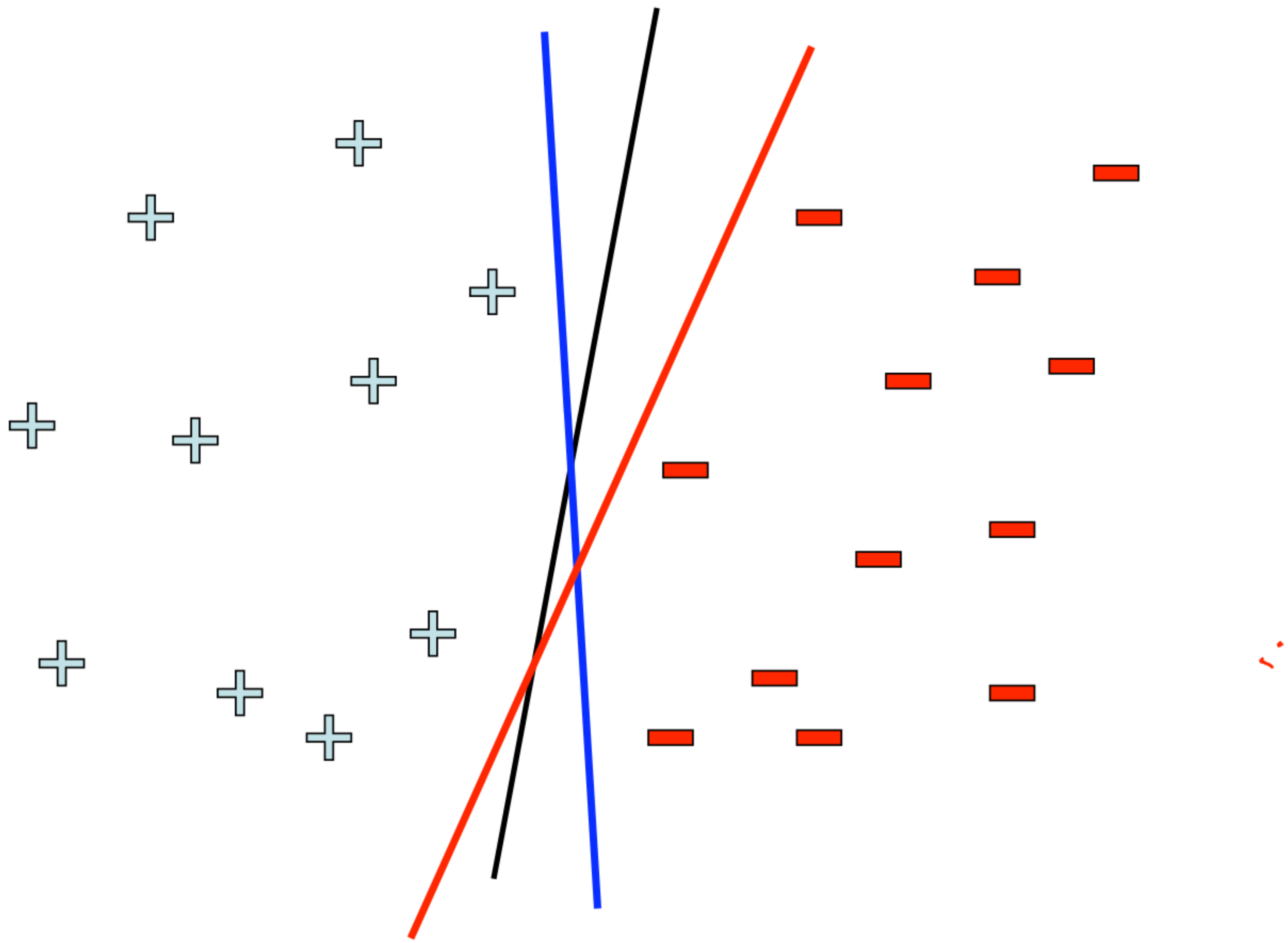
CS 4824/ECE 4424: Support Vector Machine

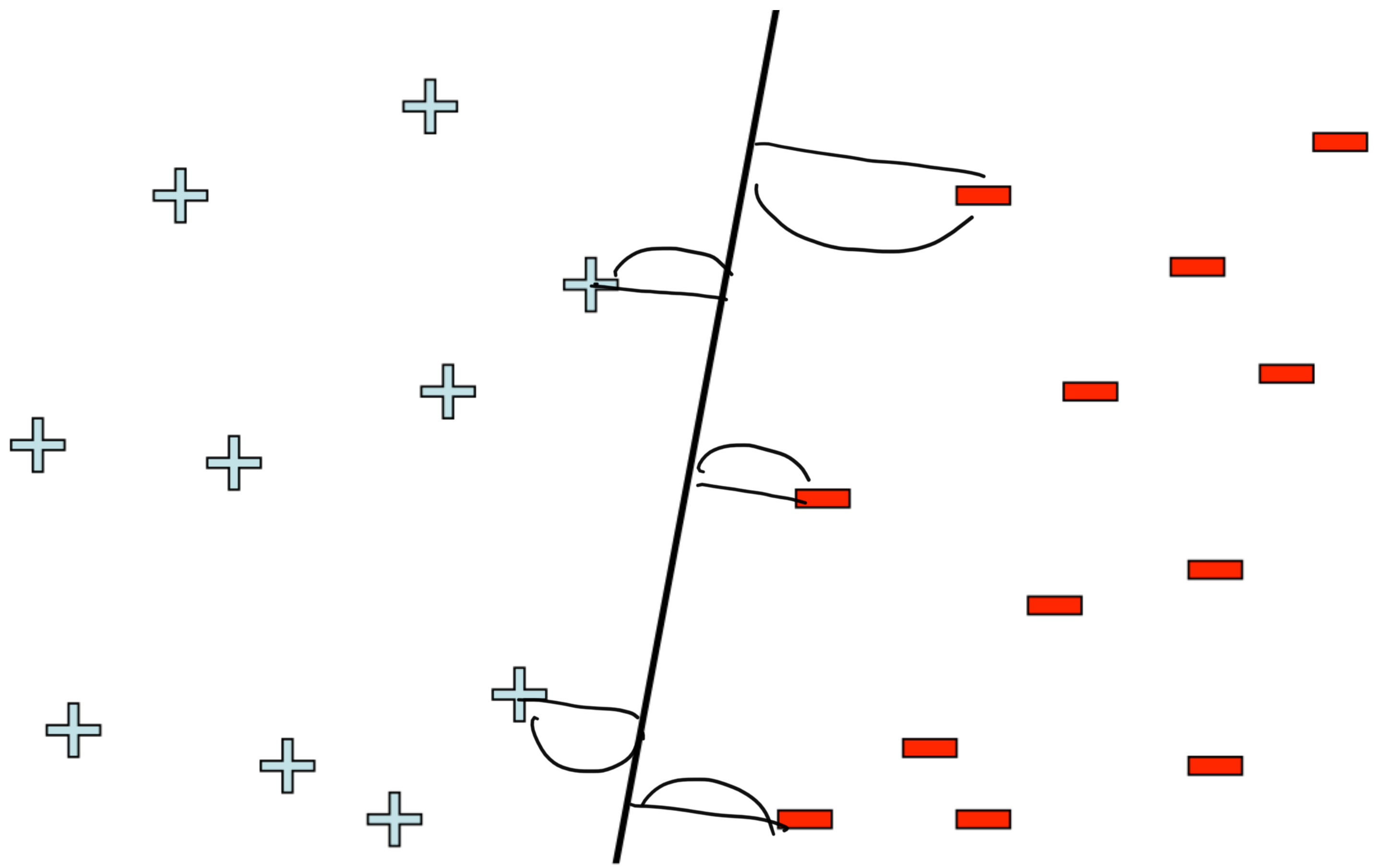
Acknowledgement:
Many of these slides are derived from Tom Mitchell, Pascal Poupart, Pieter Abbeel, Eric Eaton, Carlos Guestrin, William Cohen, and Andrew Moore.

Linear classifiers – multiple possibilities



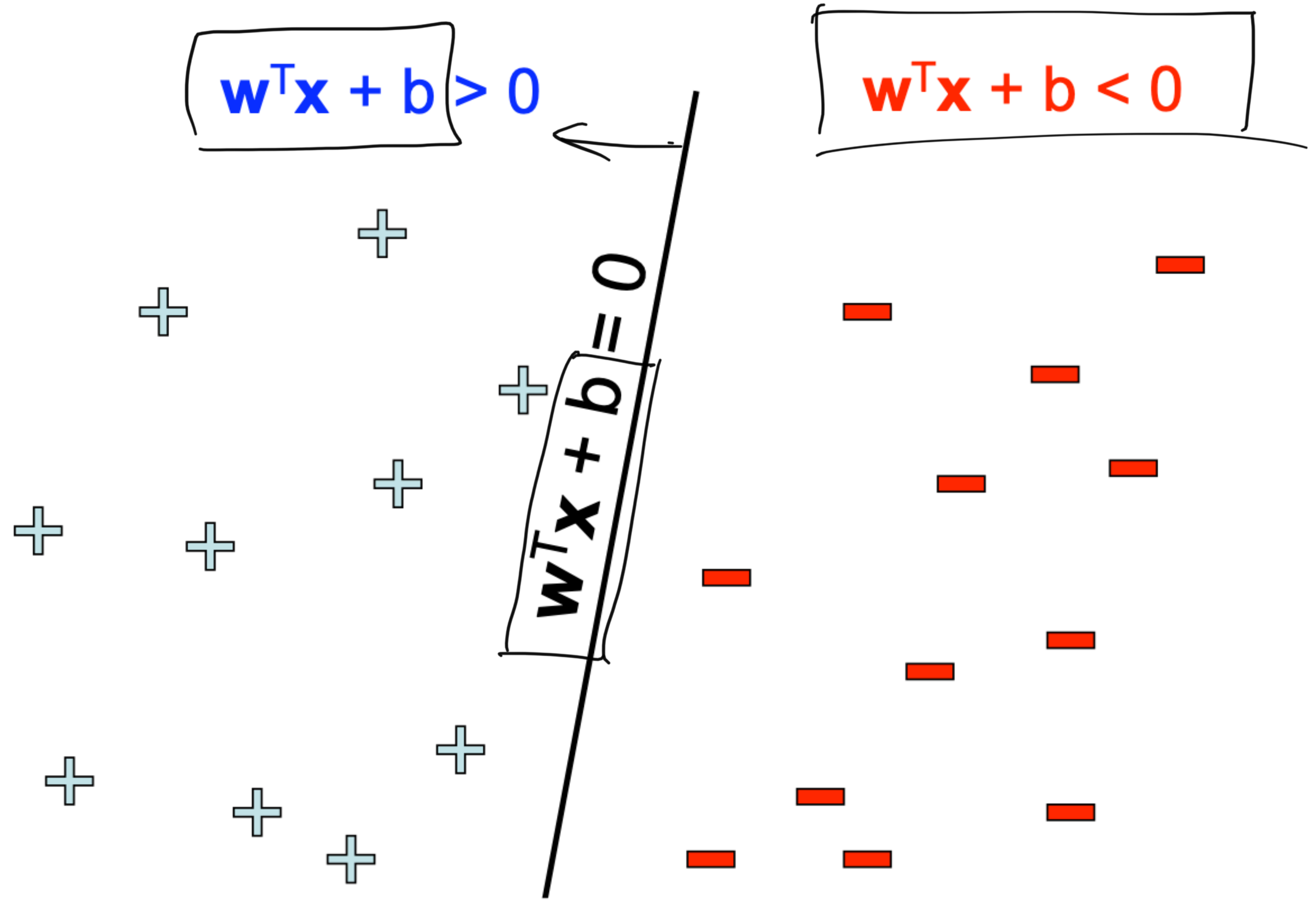
- **Challenge**: How to pick the best classifier?

Pick the one with the largest margin!



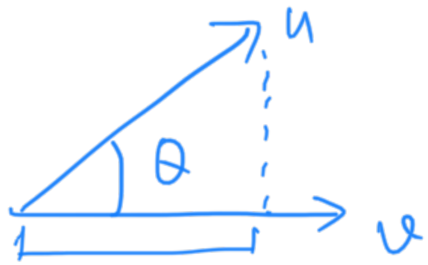
Parameterizing the decision boundary

$\hat{y} = \boxed{w^T x + b}$



Labels $y_i \in \{-1, +1\}$ — class

Maximizing the margin

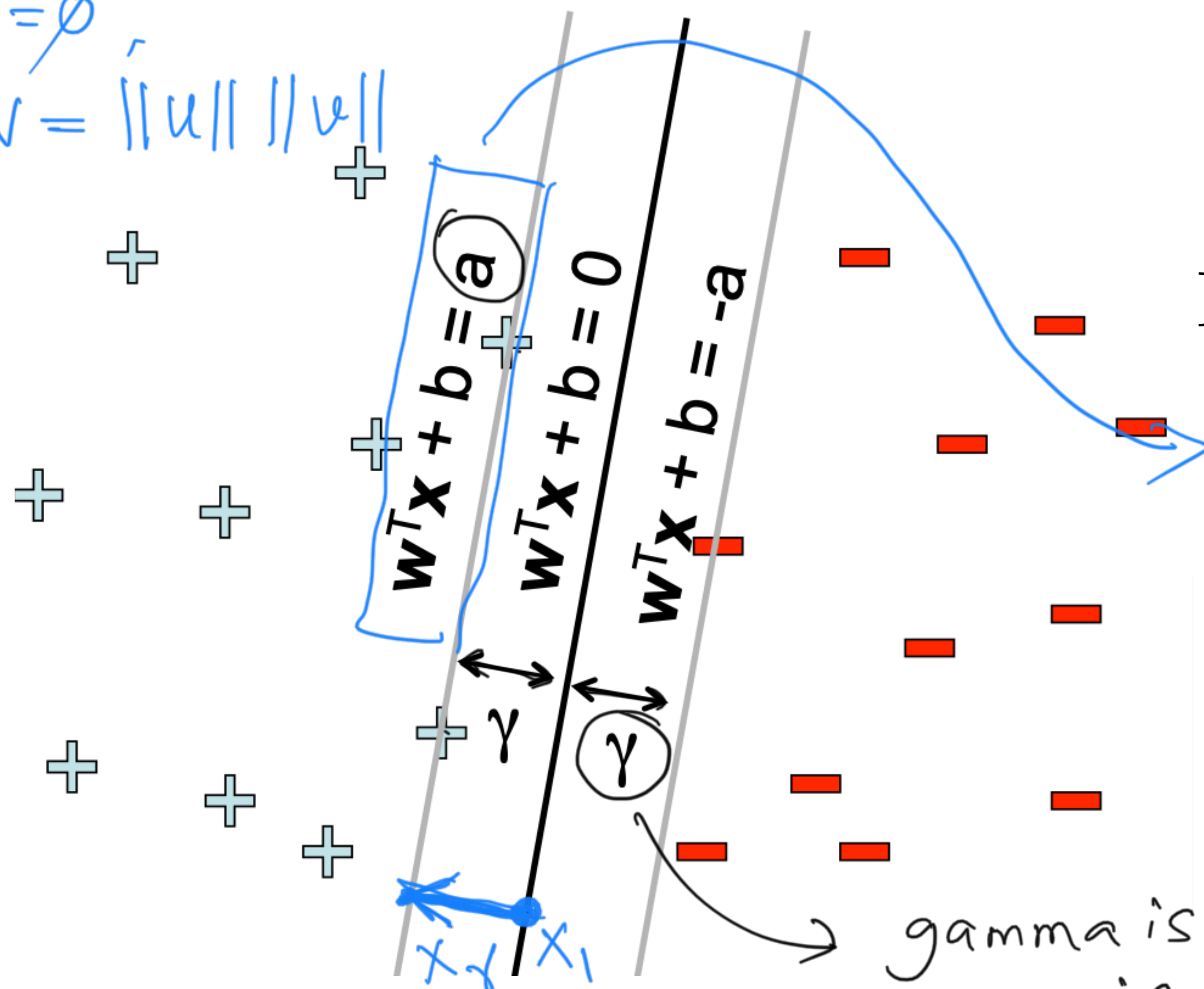


$u \cdot v = \|u\| \, \|v\| \cos\theta$

if $\theta = \emptyset$,

$u \cdot v = \|u\| \, \|v\|$

D.S. sits at the middle



gamma is the margin

- Margin = Distance of closest examples from the decision line/ hyperplane

How to find the Max Margin = ?

$w^T \boxed{(x_1 + x_\gamma)} + b = a$

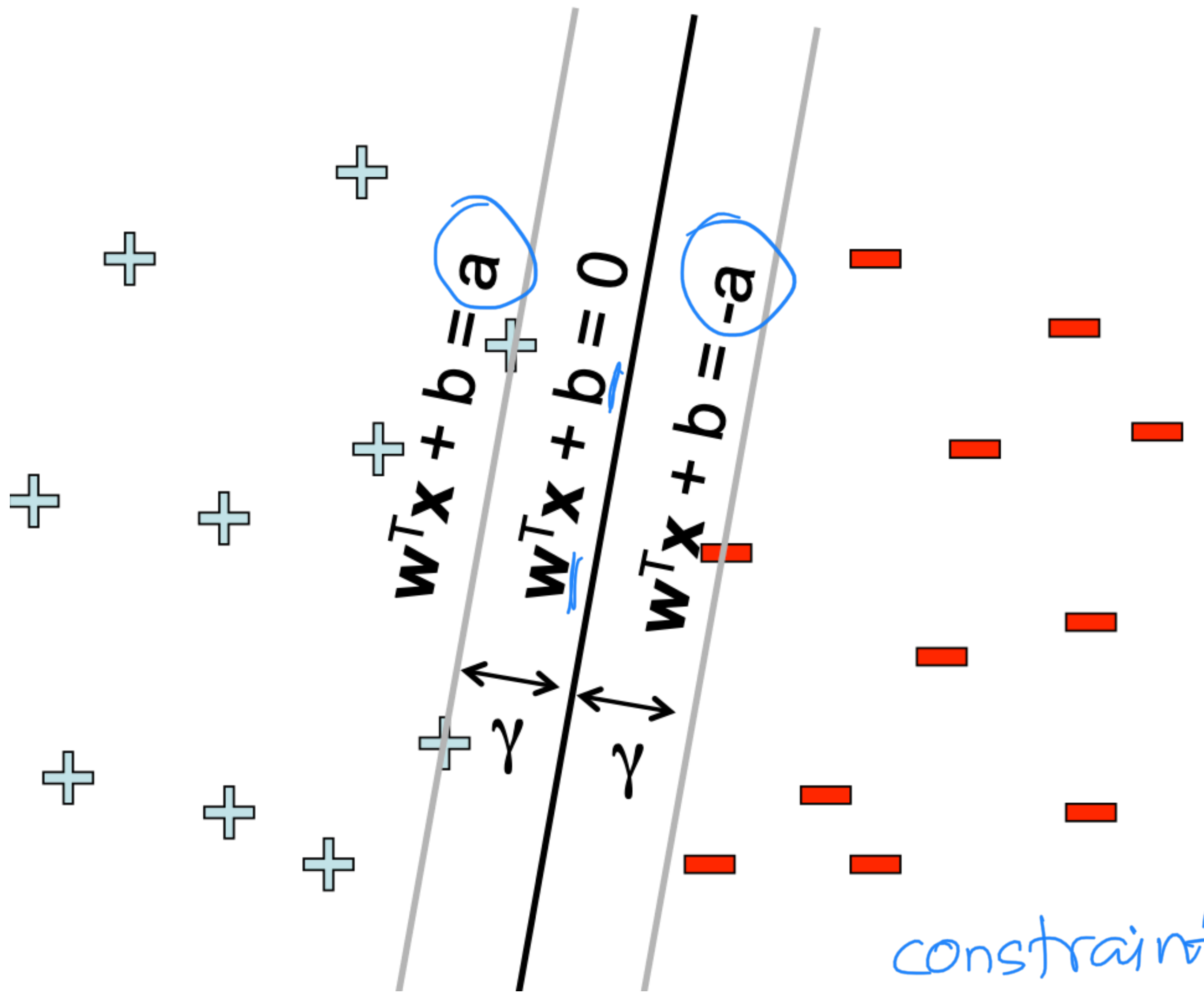
$\underbrace{w^T x_1 + b}_{0} + w^T x_\gamma = a$

$0 + w^T x_\gamma = a$

$\|w\| \, \|x_\gamma\| = a$

$\|w\| \, \gamma = a \Rightarrow \boxed{\gamma = \frac{a}{\|w\|}}$

Maximizing the margin



- Margin = Distance of closest examples from the decision line/ hyperplane

$$\text{Margin} = \boxed{\gamma = \frac{a}{\|\mathbf{w}\|}}$$

$$\arg\max_{\mathbf{w},b} \frac{a}{\|\mathbf{w}\|}$$

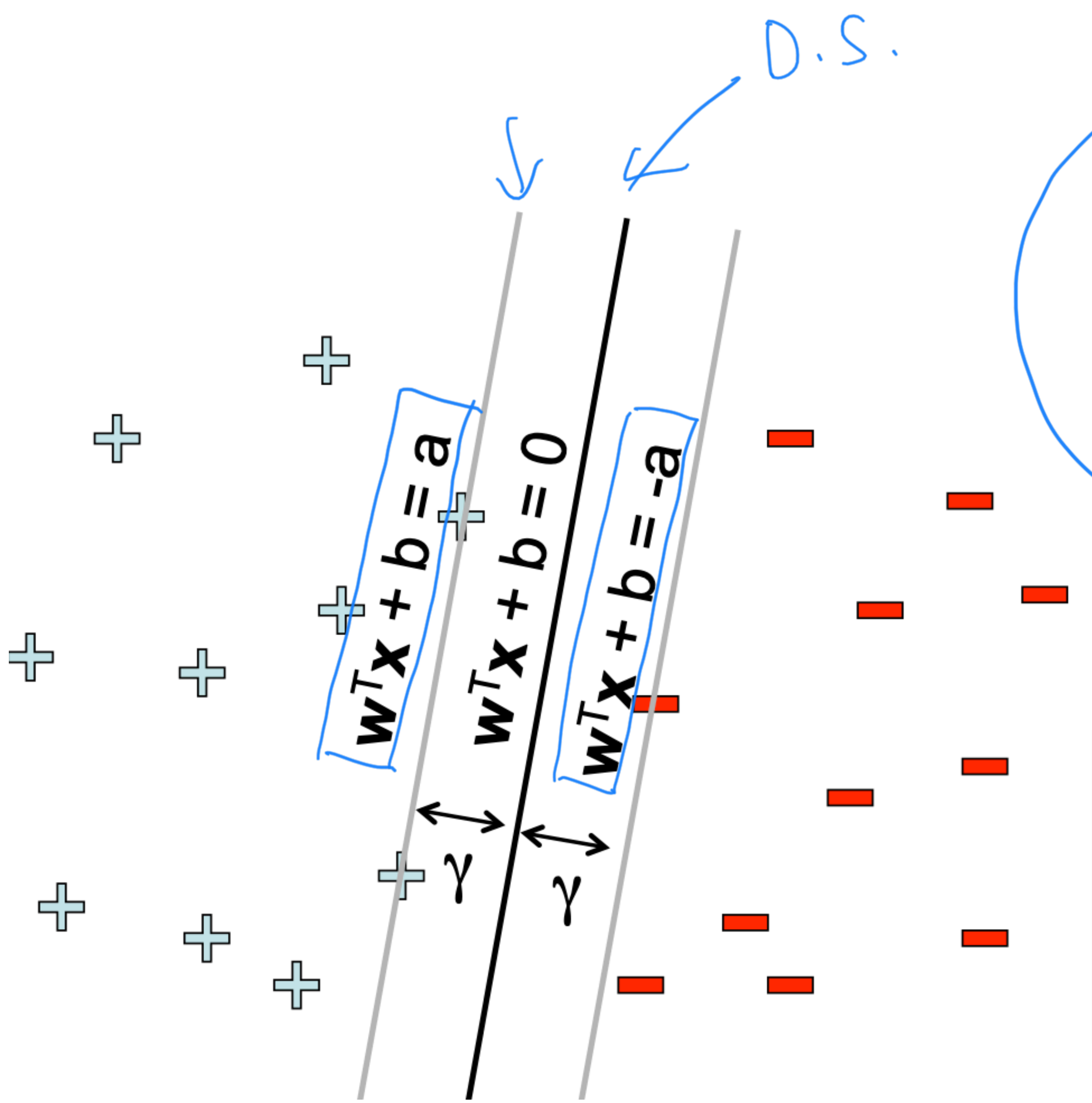
$$s.t. \; (\mathbf{W}^T\mathbf{X}_j + b)y_j \geq a \; \forall j$$

constraint

subject to

$\hat{y}_j$ $\quad$ y_j

Support Vector Machine



$$\arg\max_{\mathbf{w},b} \frac{a}{\|\mathbf{w}\|}$$

$$s.t.\ (\mathbf{W}^T\mathbf{X}_j + b)y_j \geq a \,\forall j$$

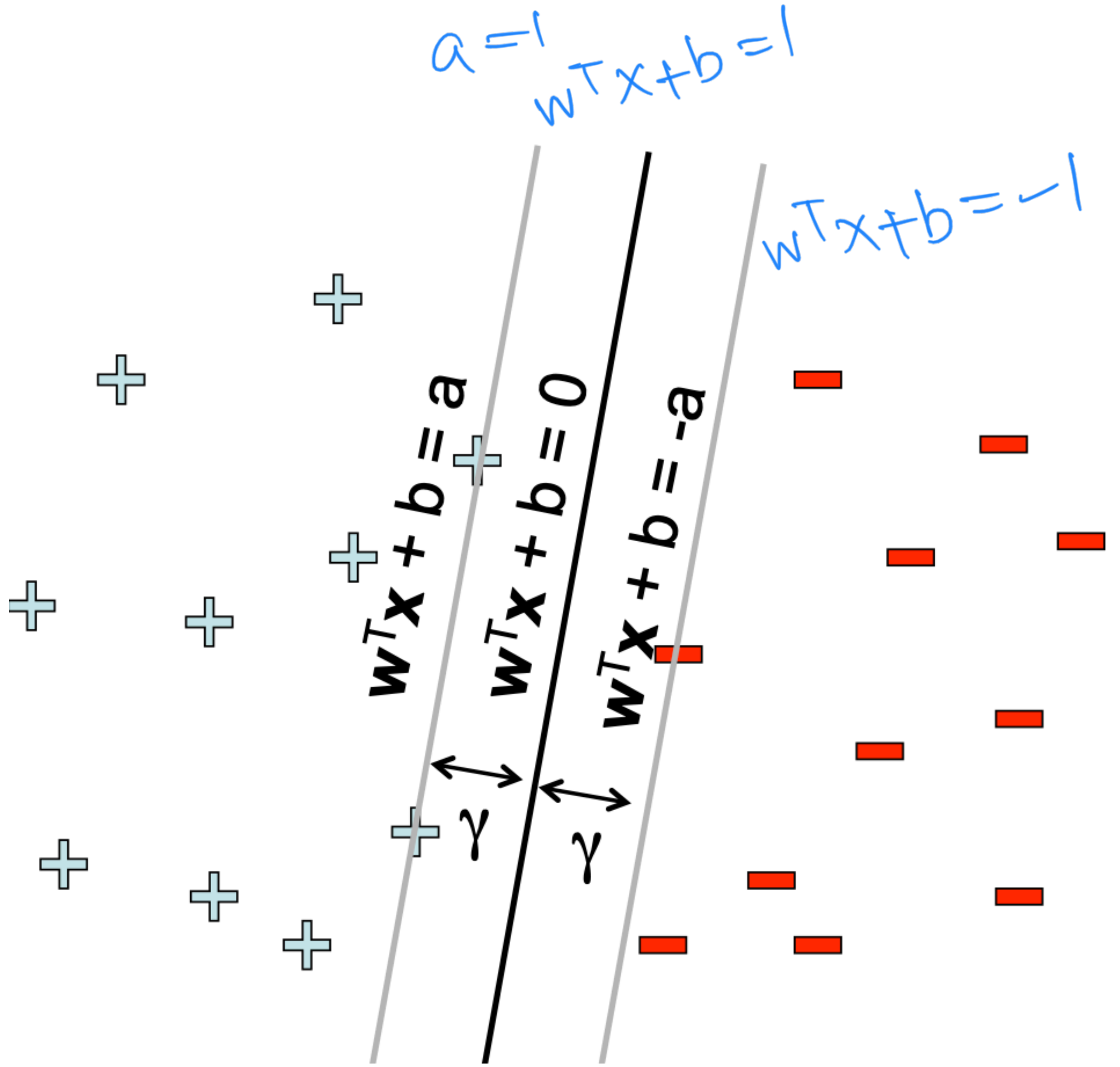
$$\arg\min_{\mathbf{w},b} \mathbf{W}^T\mathbf{W}$$

$$s.t.\ (\mathbf{W}^T\mathbf{X}_j + b)y_j \geq a \,\forall j$$

Solve efficiently by quadratic programming (QP) - well studied

Note: a is arbitrary (can normalize equations by a)

Support Vector Machine



$$\arg\max_{\mathbf{w},b} \frac{1}{||\mathbf{w}||}$$

$$s.t. (\mathbf{W}^T\mathbf{X}_j + b)y_j \geq 1 \,\forall j$$

$$\arg\min_{\mathbf{w},b} \mathbf{W}^T\mathbf{W}$$

W.W

$$s.t. (\mathbf{W}^T\mathbf{X}_j + b)y_j \geq 1 \,\forall j$$

Solve efficiently by quadratic programming (QP) - well studied

Note: a is arbitrary (can normalize equations by a)

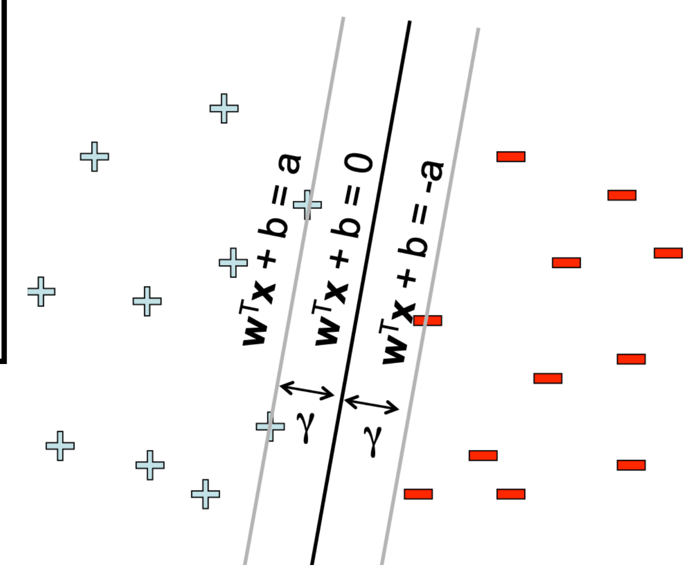
SVM — primal and dual forms

Primal form: solve for $\mathbf{w}, b$

$$\arg\min_{\mathbf{w},b} \boxed{\mathbf{W}^T\mathbf{W}}$$

$$s.t. \boxed{y_l(\mathbf{W}^T\mathbf{X}_l + b) \geq 1} \forall l \in \text{training examples}$$

Classification for new $\mathbf{X}$: $(\mathbf{W}^T\mathbf{X} + b) > 0$



SVM — primal and dual forms

Primal form: solve for $\mathbf{w}, b$

$$\underset{\mathbf{w},b}{\arg\min}\ \mathbf{W}^T\mathbf{W}$$

$$s.t.\ y_l(\mathbf{W}^T\mathbf{X}_l + b) \geq 1\ \forall l \in \text{training examples}$$

Classification for new $\mathbf{X}$: $\boxed{(\mathbf{W}^T\mathbf{X} + b) > 0}$

Dual form: solve for $\alpha_1, \ldots, \alpha_n$

$\phi(x_j)$ $\phi(x_k)$

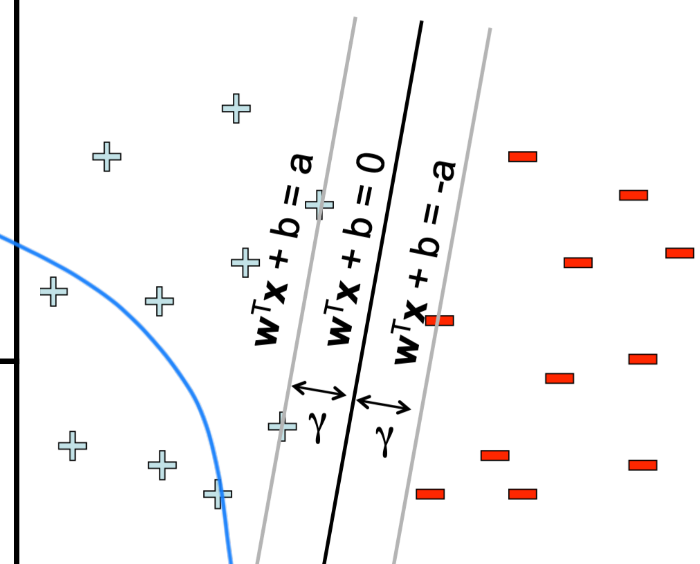
$$\underset{\alpha_1\ldots\alpha_n}{\arg\max} \sum_{l=1}^{M} \alpha_l - \frac{1}{2}\sum_{j=1}^{M}\sum_{k=1}^{M} \alpha_j\alpha_k y_j y_k \boxed{\left\langle \mathbf{X}_j, \mathbf{X}_k \right\rangle}$$

$$s.t.\ \alpha_l > 0\ \forall l \in \text{training examples} \quad \sum_{l=1}^{M} \alpha_l y_l = 0$$

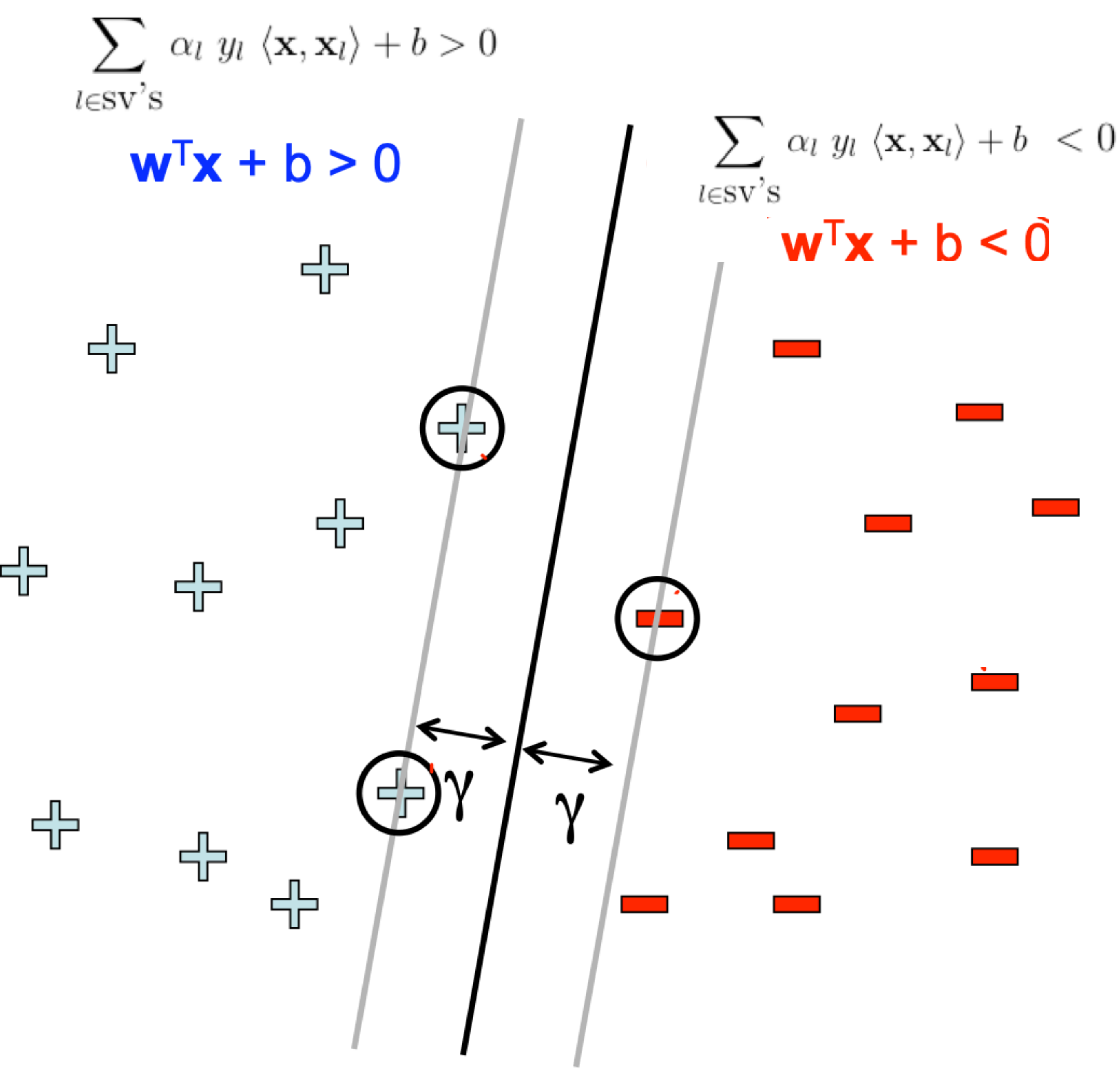
Classification for new $\mathbf{X}$: $\boxed{\sum_{l \in SV's} \alpha_l y_l \boxed{\left\langle \mathbf{X}, \mathbf{X}_l \right\rangle} + b > 0}$

$\langle x, x_l \rangle$

$= x \cdot x_l$



Support Vectors



- The linear hyperplane is defined by "**support vectors**"
- Moving other points a little doesn't effect the decision boundary
- Only need to store the support vectors to predict labels of new points

Kernel SVM — primal and dual forms

Primal form: solve for $\mathbf{w}, b$

$$\arg\min_{\mathbf{w},b} \mathbf{W}^T\mathbf{W}$$

$$s.t.\ y_l(\mathbf{W}^T \boxed{\phi(\mathbf{X}_l)} + b) \geq 1\ \forall l \in \text{training examples}$$

Classification for new $\mathbf{X}$: $(\mathbf{W}^T \boxed{\phi(\mathbf{X})} + b) > 0$

Dual form: solve for $\alpha_1, \ldots, \alpha_n$

$\phi(x_j) \cdot \phi(x_k)$

$$\arg\max_{\alpha_1 \ldots \alpha_n} \sum_{l=1}^{M} \alpha_l - \frac{1}{2} \sum_{j=1}^{M} \sum_{k=1}^{M} \alpha_j \alpha_k y_j y_k \boxed{K(\mathbf{X}_j, \mathbf{X}_k)}$$

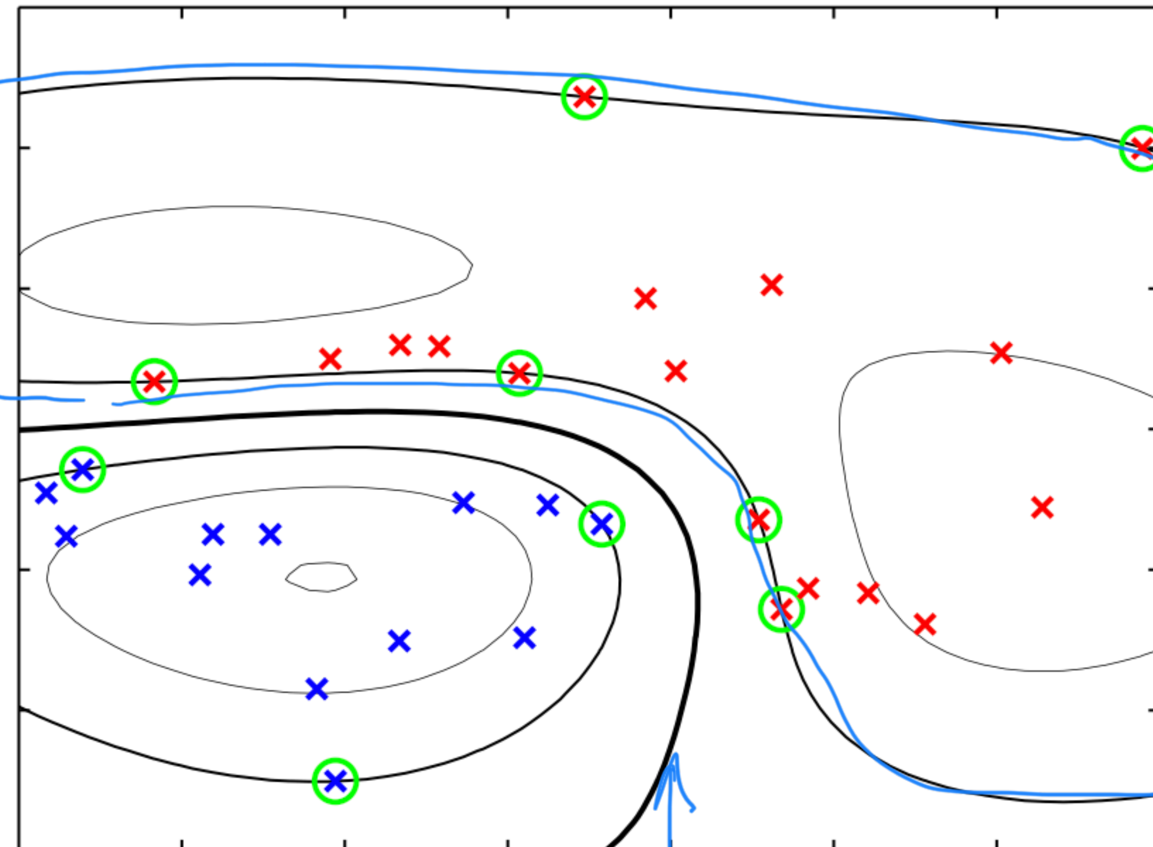
$$s.t.\ \alpha_l > 0\ \forall l \in \text{ training examples} \quad \sum_{l=1}^{M} \alpha_l y_l = 0$$

Classification for new $\mathbf{X}$: $\boxed{\sum_{l \in SV's}} \alpha_l y_l \boxed{K(\mathbf{X}, \mathbf{X}_l)} > + b > 0$

- Since the dual form depends only on inner products, we can apply the kernel trick to work in a (virtual) projected higher-dimensional space

$K(x, y) = \phi(x) \cdot \phi(y)$

SVM Decision Surface using Gaussian Kernel



D.S

$$\boxed{\hat{f}(\mathbf{x})} = \boxed{\mathbf{w}^T \Phi(\mathbf{x}) + b}$$

- Circled points are the *support vectors*: training examples with non-zero α_l
- Points plotted in original 2-D space
- Contour lines show constant $\hat{f}(\mathbf{x})$

$$\hat{f}(\mathbf{x}) = b + \sum_{l=1}^{M} \alpha_l \, y_l \boxed{\kappa(\mathbf{x}, \mathbf{x}_l)} = b + \sum_{l=1}^{M} \alpha_l \, y_l \boxed{\exp(-\|\mathbf{x} - \mathbf{x}_l\|^2 / 2\sigma^2)}$$

SVM Summary

- **Objective**: maximize margin between decision surface and data

- Primal and dual formulations
  - dual represents classifier decision in terms of *support vectors*

- Kernel SVM's
  - learn linear decision surface in high dimension space, working in original low dimension space

- SVM algorithm: Quadratic Program optimization
  - single global minimum