Support Vector Machines (SVMs)

Aarti Singh & Geoff Gordon

Machine Learning 10-701
Mar 22, 2021

Discriminative Classifiers

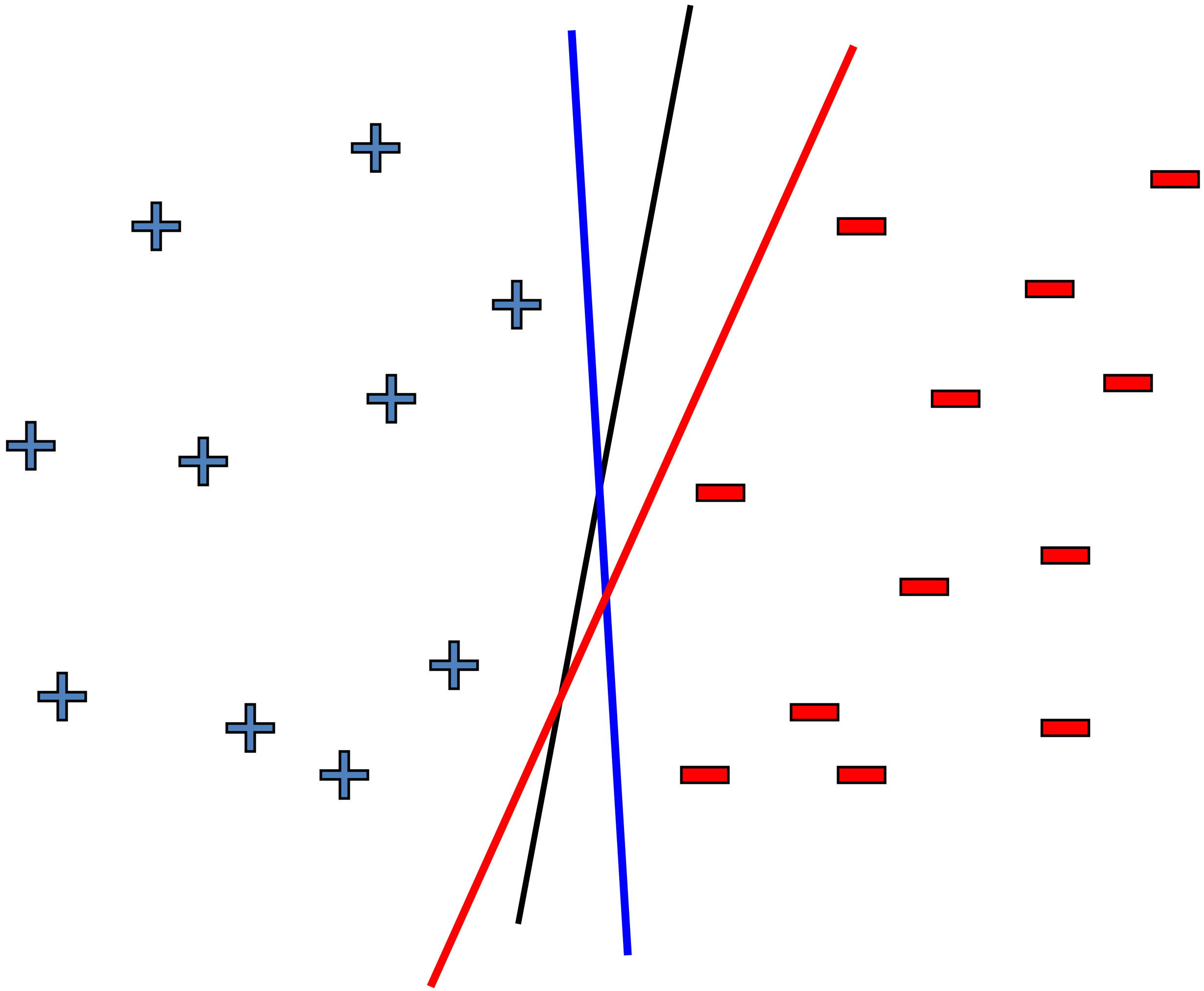
Optimal Classifier:

$$f^*(x) = \arg\max_{Y=y} P(Y = y|X = x)$$
$$= \arg\max_{Y=y} P(X = x|Y = y)P(Y = y)$$

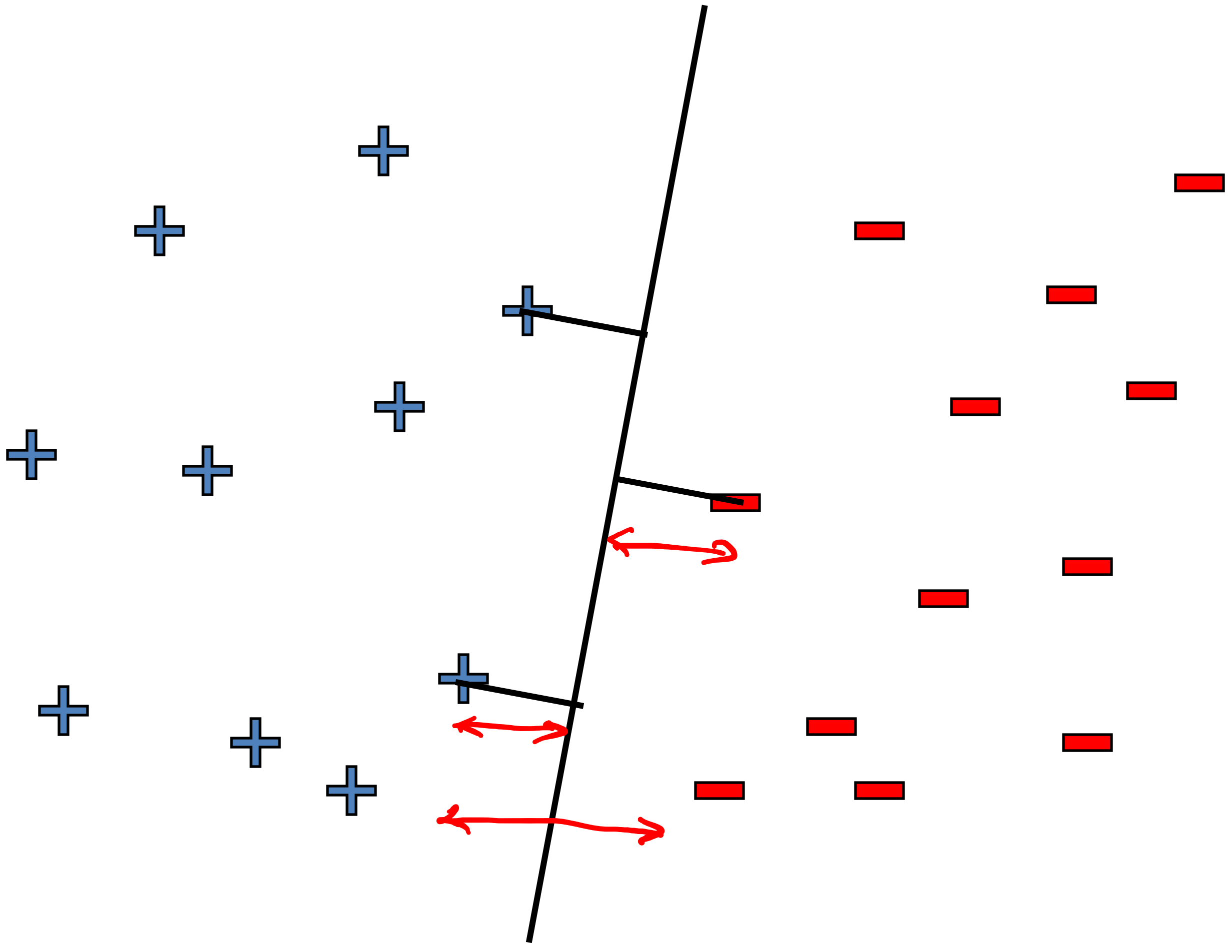
Why not learn P(Y|X) directly? Or better yet, why not learn the decision boundary directly?

- Assume some functional form for P(Y|X) (e.g. Logistic Regression) or for the decision boundary (e.g. Neural nets, SVMs - today)
- Estimate parameters of functional form directly from training data

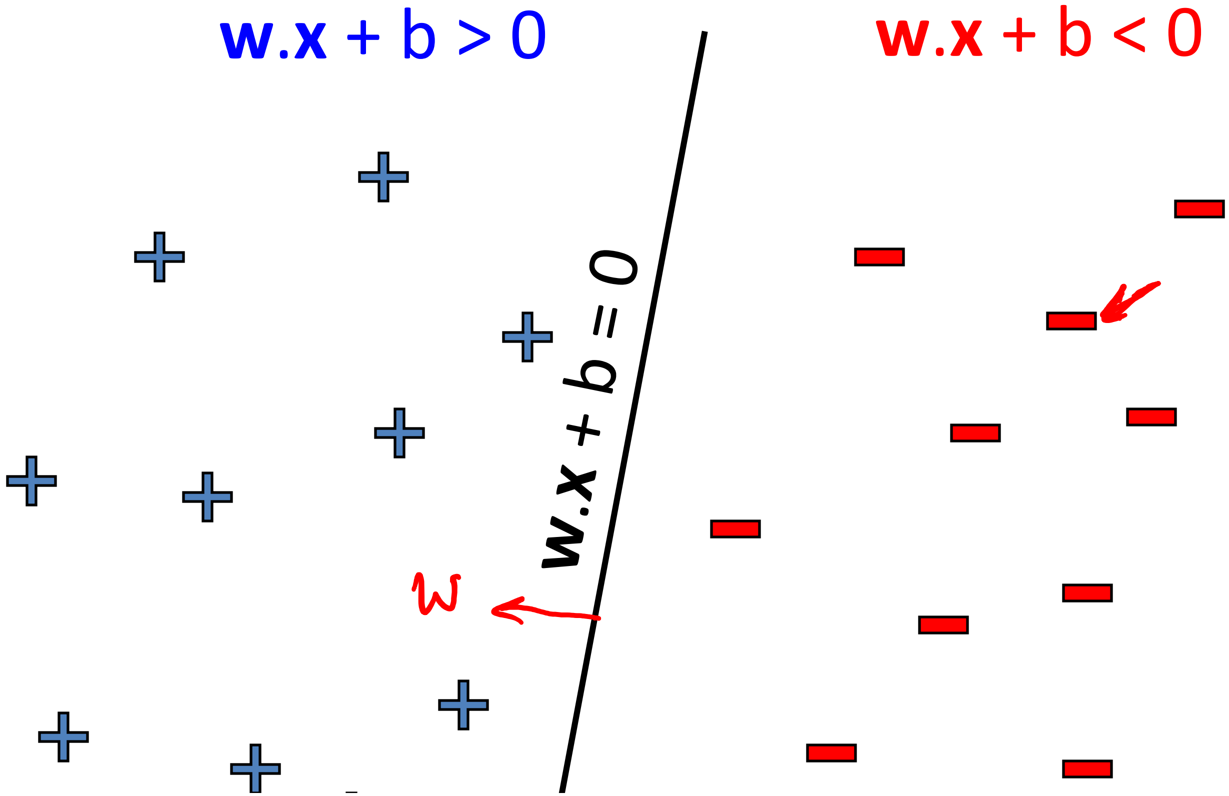
Linear classifiers – which line is better?



Pick the one with the largest margin!



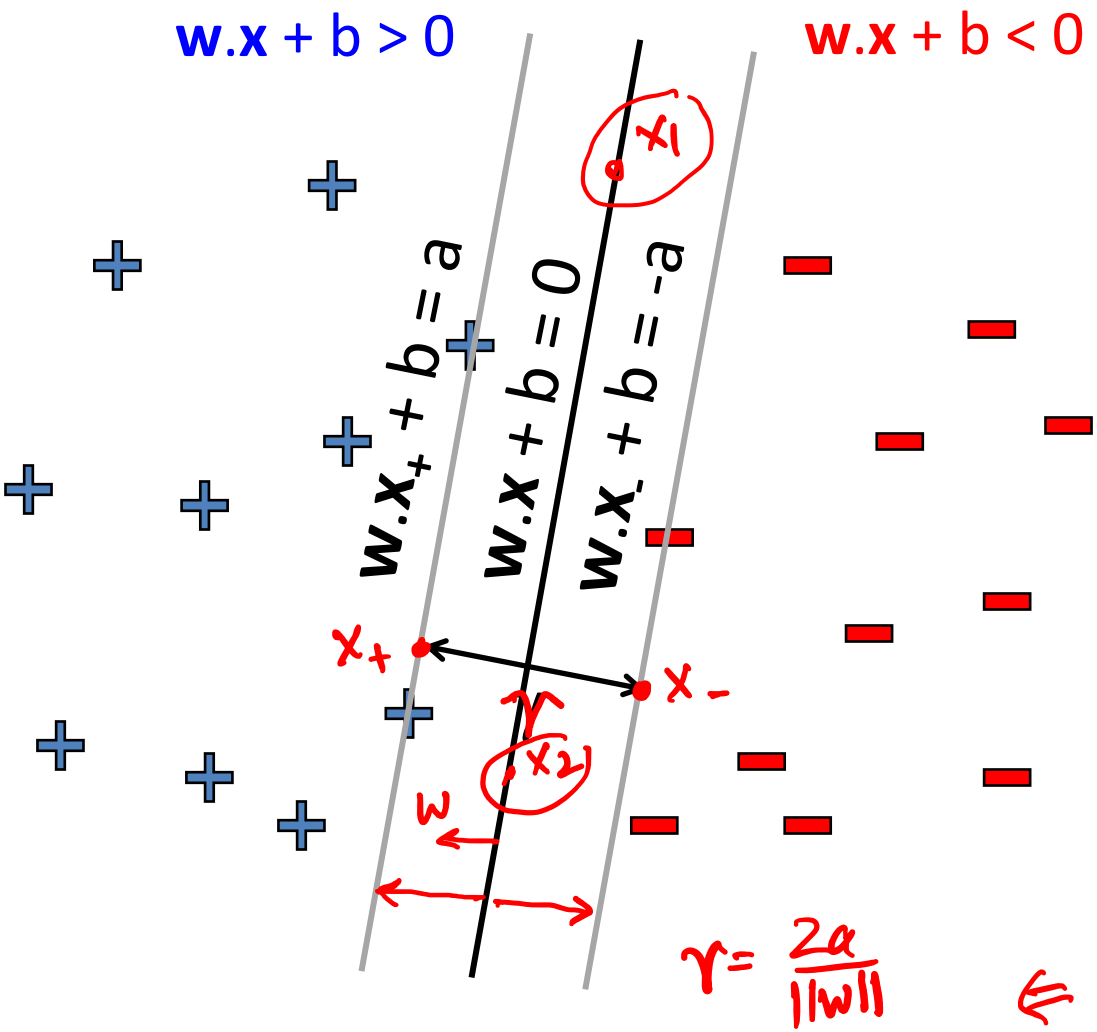
Parameterizing the decision boundary



$y_j \in \{-1, +1\}$ — class

"confidence" $= \left(\mathbf{w}.\mathbf{x}_j + b\right) y_j$

Maximizing the margin



$w \cdot x_1 + b = 0$

$w \cdot x_2 + b = 0$

$w \cdot (x_1 - x_2) = 0$

Distance of closest examples from the line/hyperplane

margin = γ = 2a/‖w‖

1) w ⊥ decision boundary

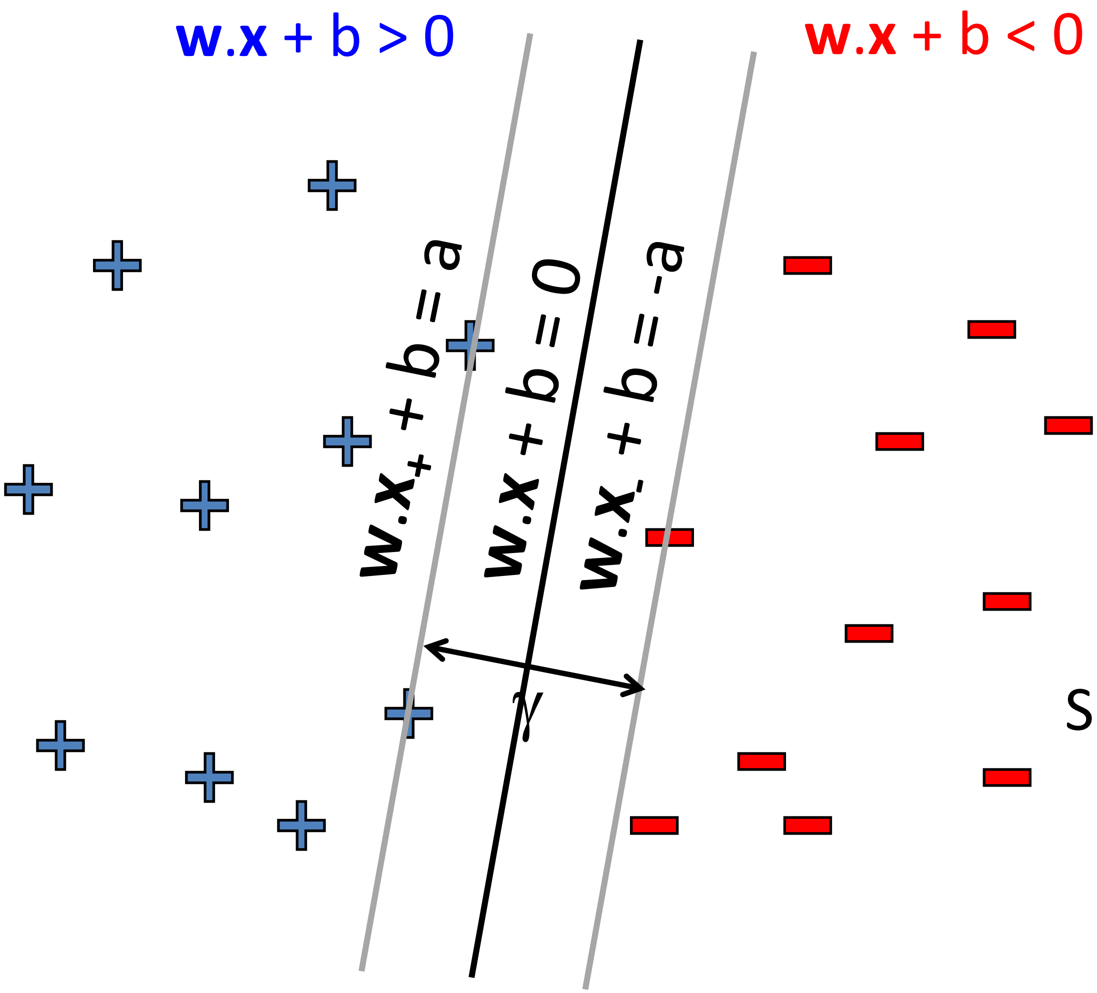
$w \cdot (x_1 - x_2) = 0$

2) $x_- + \gamma \frac{w}{\|w\|} = x_+$

$w \gamma = \|w\| (x_+ - x_-)$

$\Leftarrow \underbrace{w \cdot w}_{\|w\|^2} \gamma = \|w\| \underbrace{(w \cdot x_+ - w \cdot x_-)}_{2a}$

Maximizing the margin

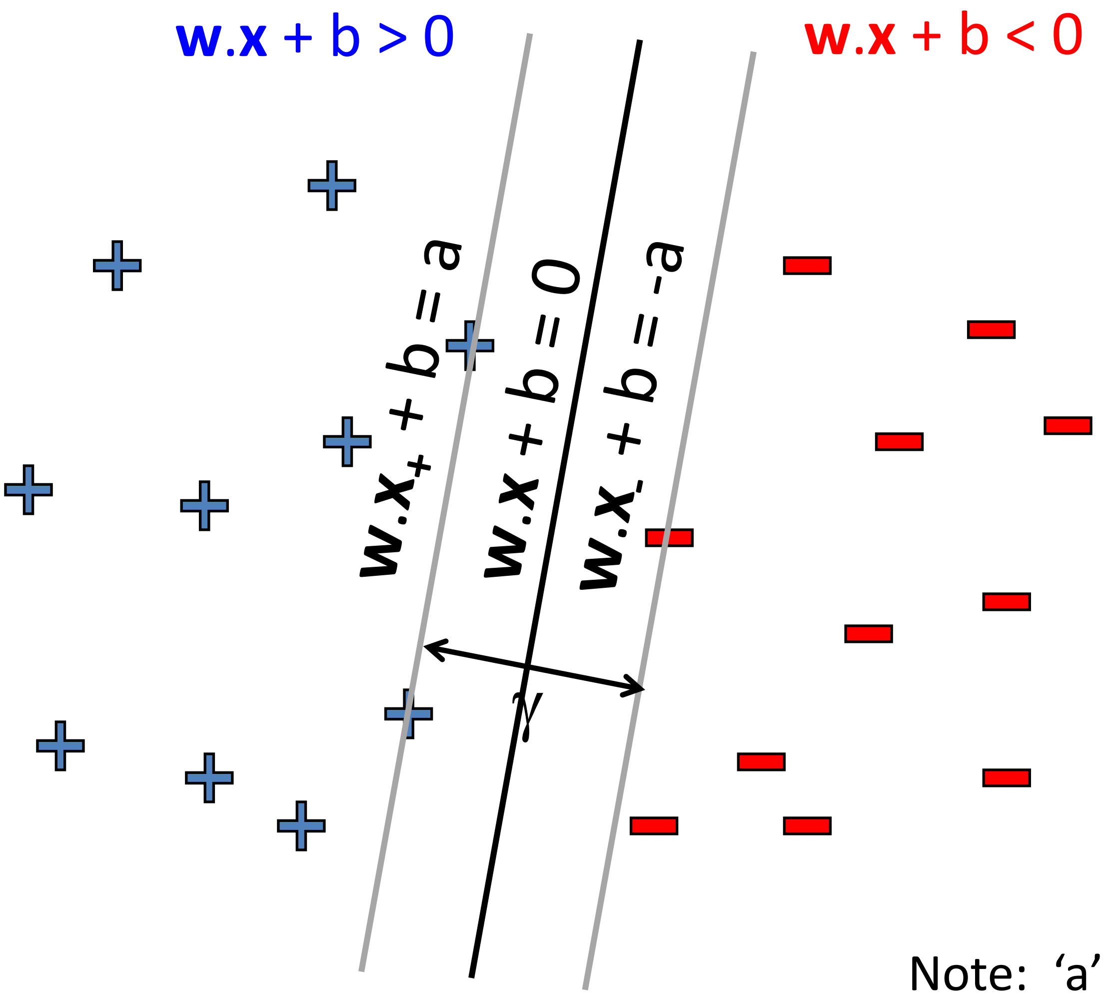


Distance of closest examples from the line/hyperplane

$$\text{margin} = \gamma = 2a/\|w\|$$

Smaller margin ⇔ larger ‖w‖

Maximizing the margin



Distance of closest examples from the line/hyperplane

$$\text{margin} = \gamma = 2a/\|w\|$$

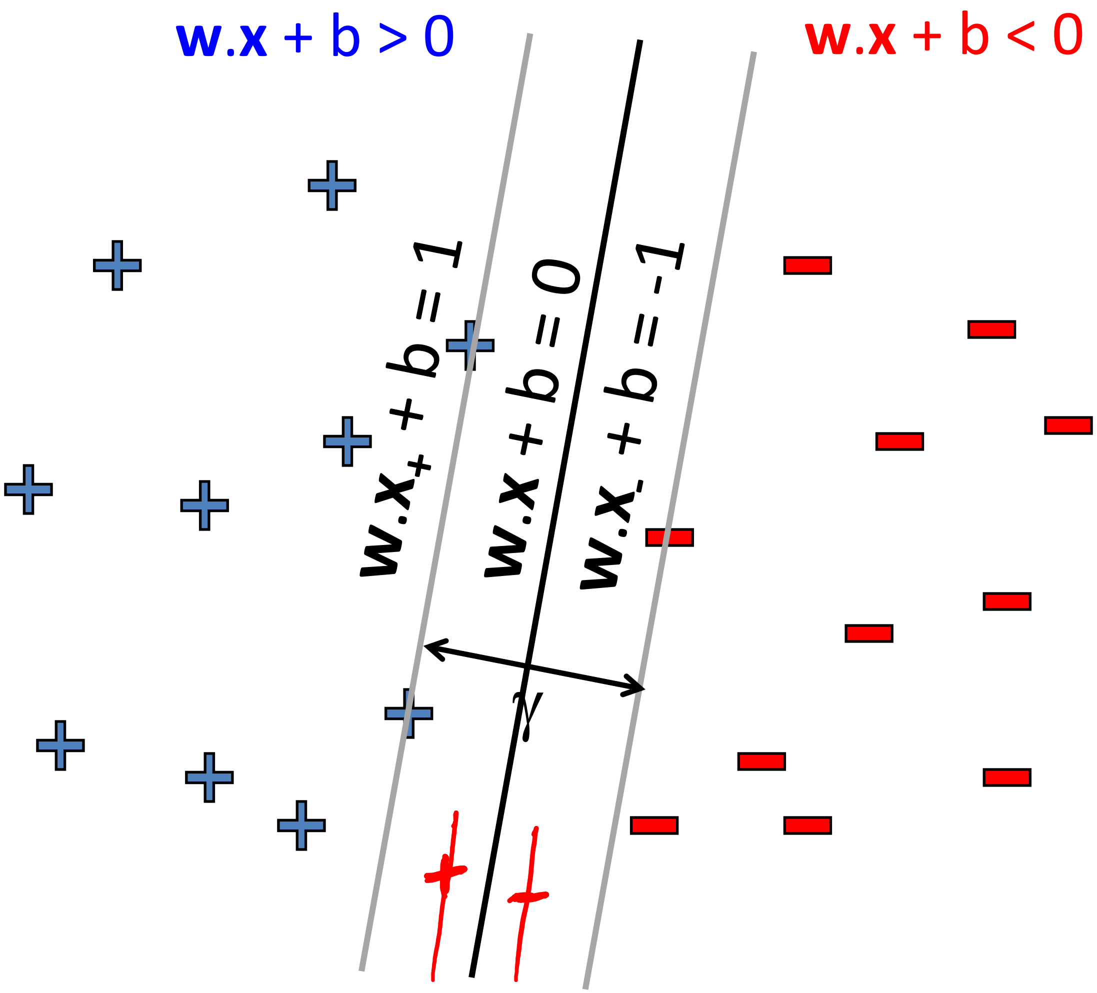
$$\max_{\mathbf{w},b} \ \gamma = 2a/\|w\|$$

$$\text{s.t. } (\mathbf{w}.\mathbf{x}_j + b)\, y_j \geq a \quad \forall j$$

confidence

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

Support Vector Machines



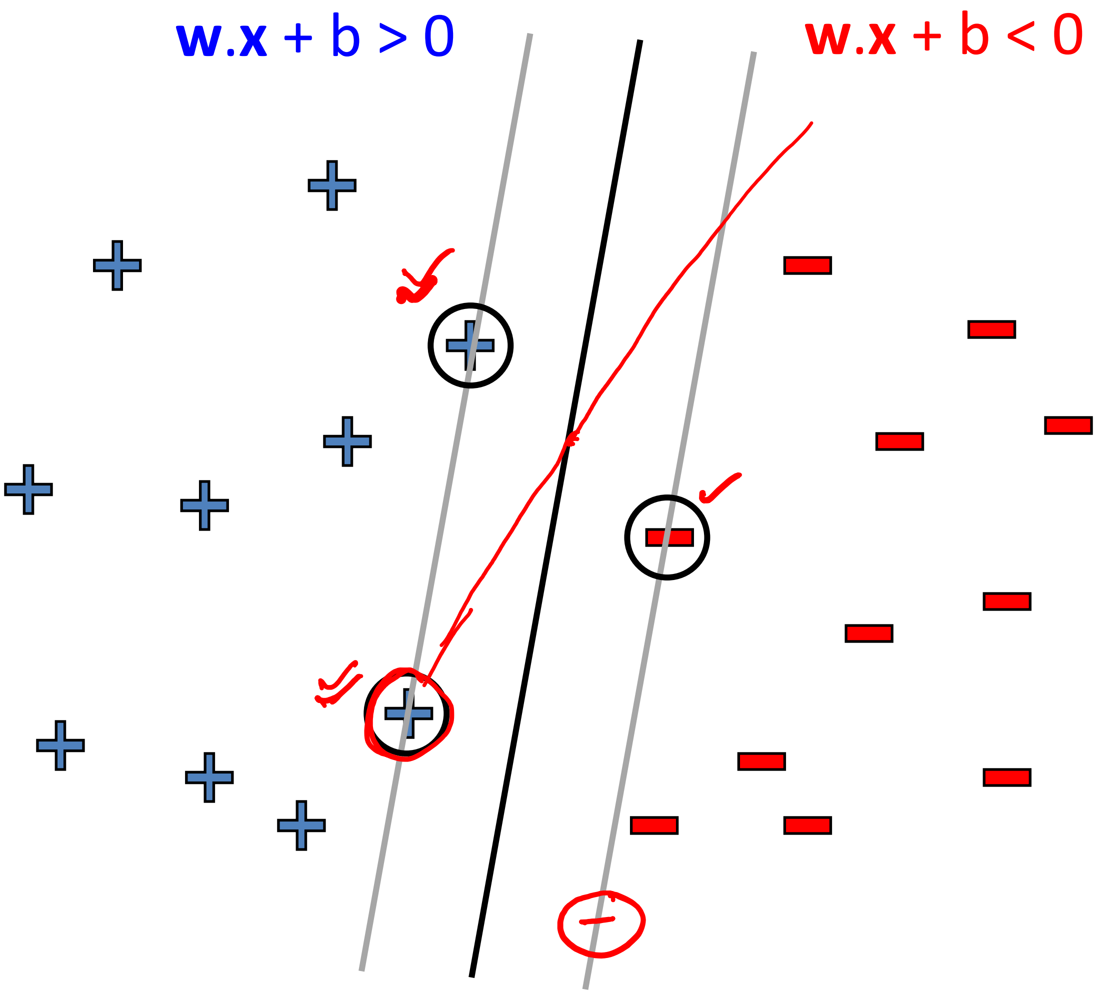
$$\min_{\mathbf{w},b} \ \mathbf{w}.\mathbf{w}$$

$$\text{s.t. } (\mathbf{w}.\mathbf{x}_j + b)\, y_j \geq 1 \quad \forall j$$

Solve efficiently by quadratic programming (QP)

- Quadratic objective, linear constraints
- Well-studied solution algorithms

Support Vectors



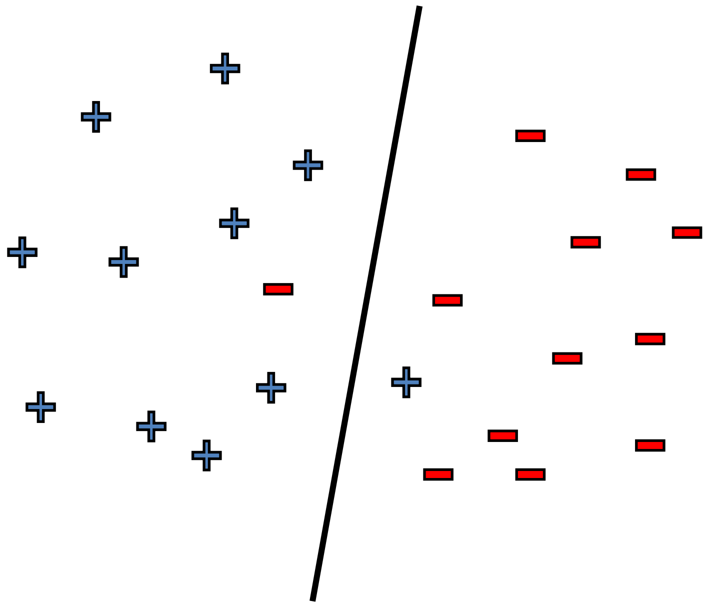
Linear hyperplane defined by "support vectors"

Moving other points a little doesn't effect the decision boundary

only need to store the support vectors to predict labels of new points

For support vectors
$(\mathbf{w}.\mathbf{x}_j + b)\, y_j = 1$

What if data is not linearly separable?



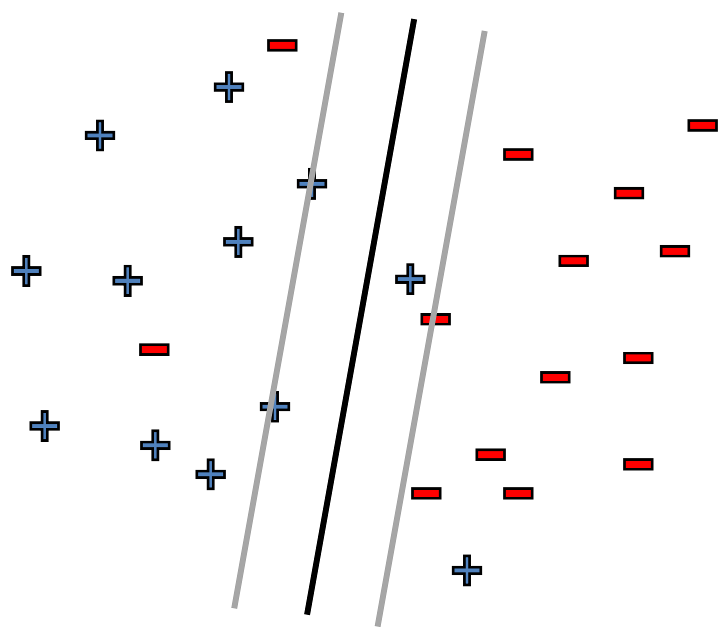
Use features of features of features of features....

x_1^2, x_2^2, x_1x_2, ..., $\exp(x_1)$

But run risk of overfitting!

What if data is still not linearly separable?

Allow "error" in classification



Smaller margin ⇔ larger ‖w‖

maximizing margin

$$\min_{\mathbf{w},b} \ \mathbf{w}.\mathbf{w} + C \ \#\text{mistakes}$$

$$\text{s.t. } (\mathbf{w}.\mathbf{x}_j + b)\, y_j \geq 1 \quad \forall j$$

Maximize margin and minimize # mistakes on training data

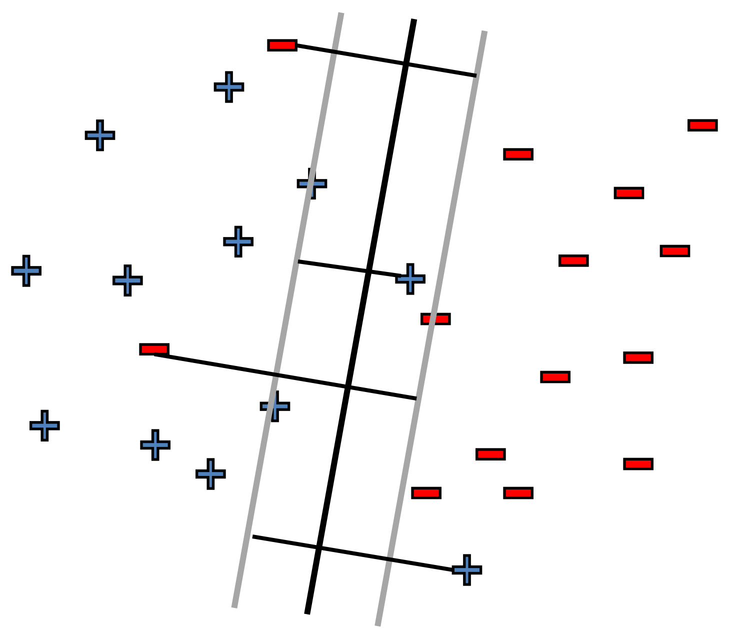
C - tradeoff parameter

Not QP ☹

0/1 loss (doesn't distinguish between near miss and bad mistake)

What if data is still not linearly separable?

Allow "error" in classification



Soft margin approach

$$\min_{\mathbf{w},b,\{\xi_j\}} \mathbf{w}.\mathbf{w} + C\sum_j \xi_j$$

$$\text{s.t. } (\mathbf{w}.\mathbf{x}_j + b)\, y_j \geq 1 - \xi_j \quad \forall j$$

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

ξ_j - "slack" variables

= (>1 if x_j misclassifed)

pay linear penalty if mistake

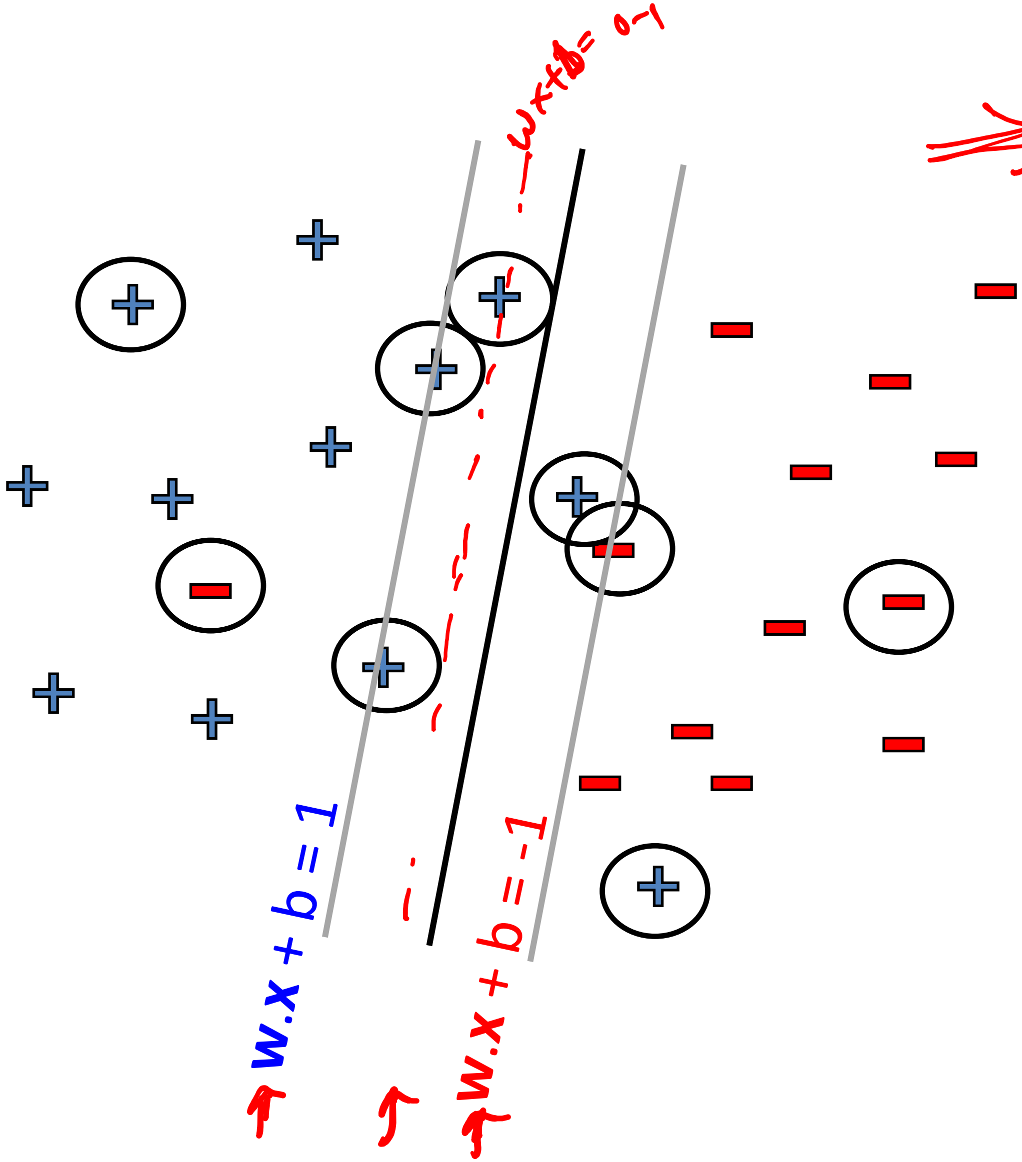
C - tradeoff parameter (C = ∞ recovers hard margin SVM)

Still QP ☺

Slack variables – Hinge loss

$\sum_i C \xi_j$

$\xi_j \geq 0$



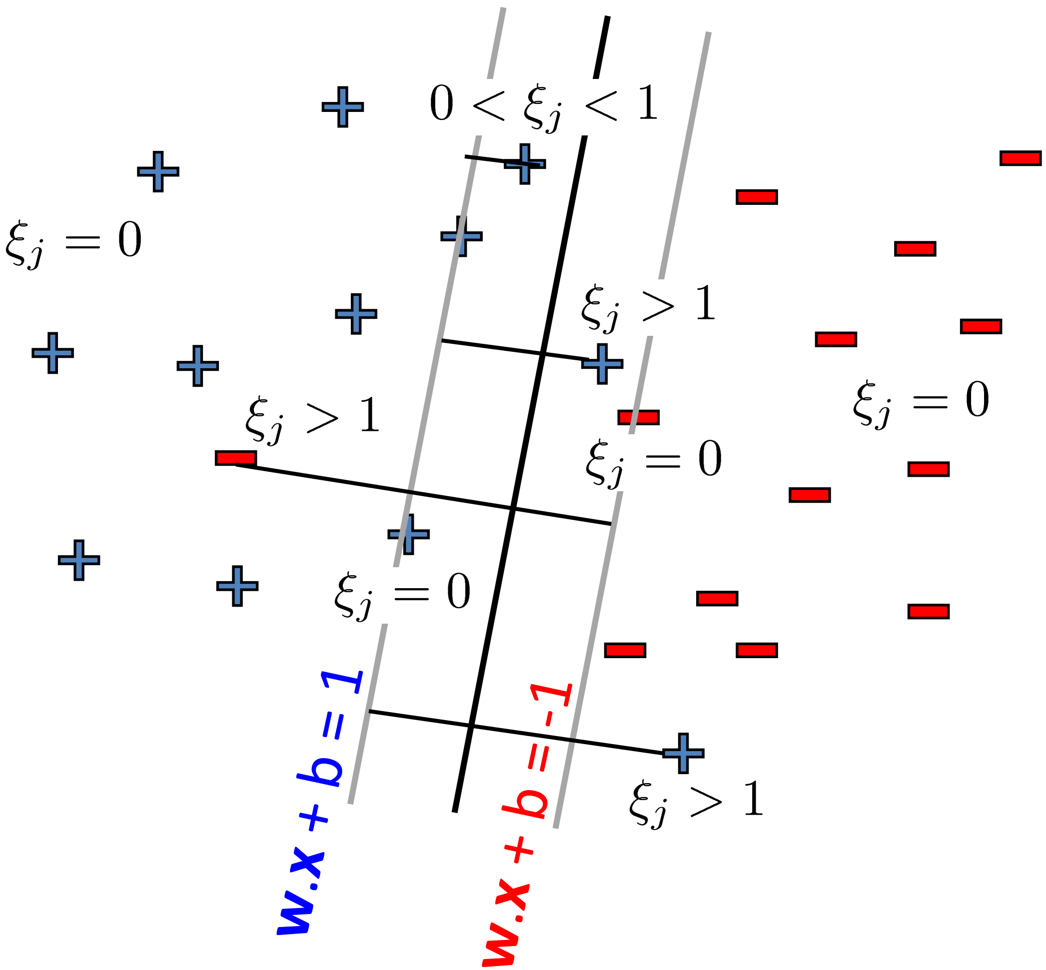
confidence

$(\mathbf{w}.\mathbf{x}_j+b)\, y_j \geq 1-\xi_j \quad \forall j$

What is the slack ξ_j for the following points?

Confidence	Slack
1	0
>1	0
0-1	0-1
<0	>1

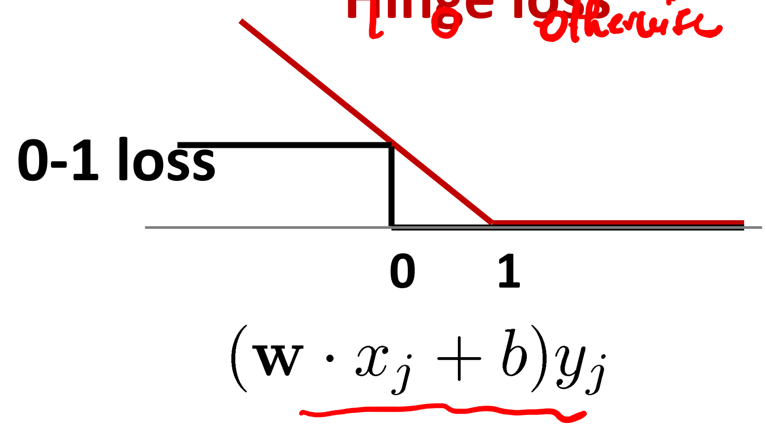
Slack variables – Hinge loss



Notice that

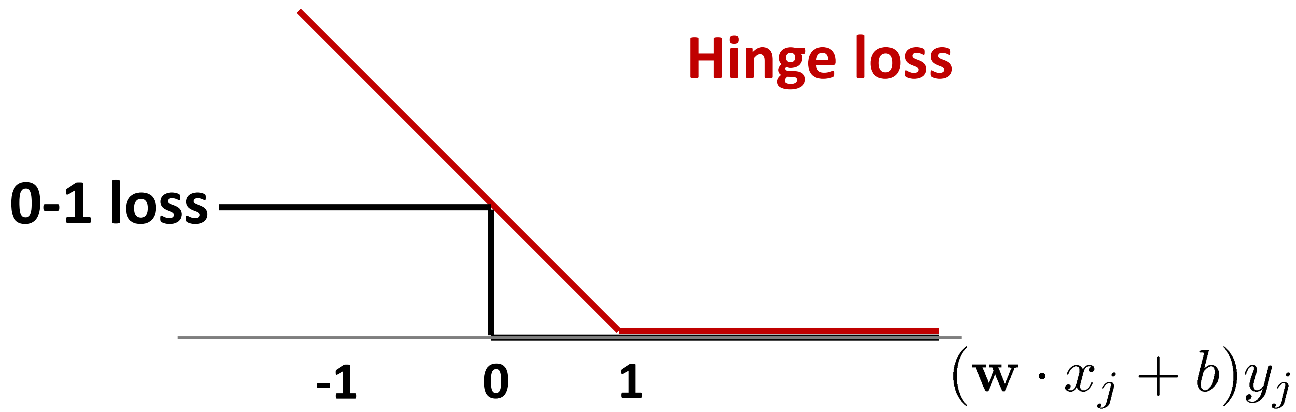
$$\xi_j = (1 - (\mathbf{w} \cdot x_j + b) y_j))_+$$

$$(a)_+ = \begin{cases} a & a > 0 \\ 0 & \text{otherwise} \end{cases}$$



Slack variables – Hinge loss

$$\xi_j = (1 - (\mathbf{w} \cdot x_j + b) y_j))_+$$



$$\min_{\mathbf{w}, b, \{\xi_j\}} \mathbf{w}.\mathbf{w} + C \sum_j \xi_j$$

$$\text{s.t. } (\mathbf{w}.\mathbf{x}_j + b)\, y_j \geq 1 - \xi_j \quad \forall j$$

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

⇔

Regularized hinge loss

$$\min_{\mathbf{w}, b} \mathbf{w}.\mathbf{w} + C \sum_j (1 - (\mathbf{w}.x_j + b) y_j)_+$$

$\|w\|^2$ loss