Support Vector Machines (SVMs)

Aarti Singh & Geoff Gordon

Machine Learning 10-701
Mar 22, 2021

Discriminative Classifiers

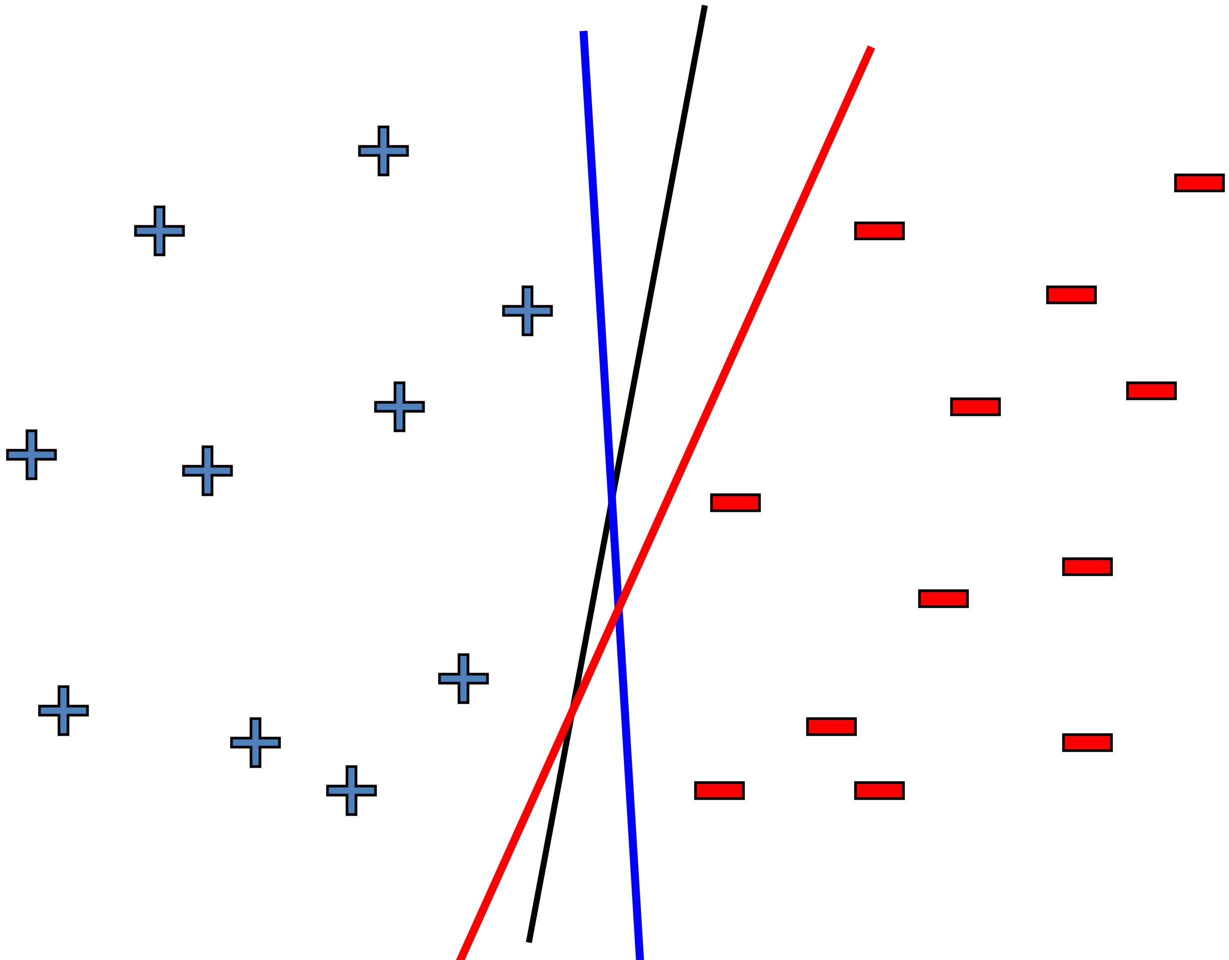
Optimal Classifier:

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

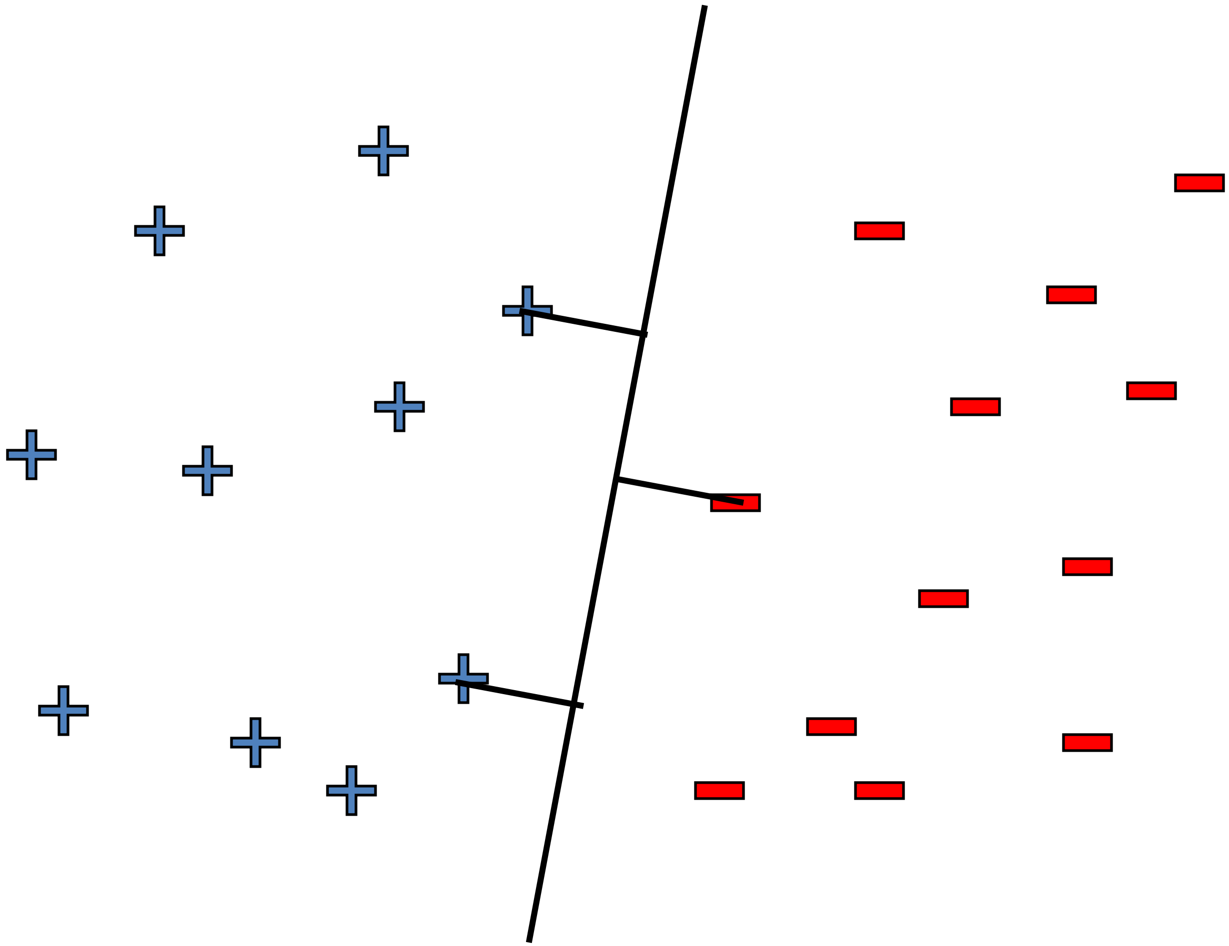
Why not learn P(Y|X) directly? Or better yet, why not learn the decision boundary directly?

- 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

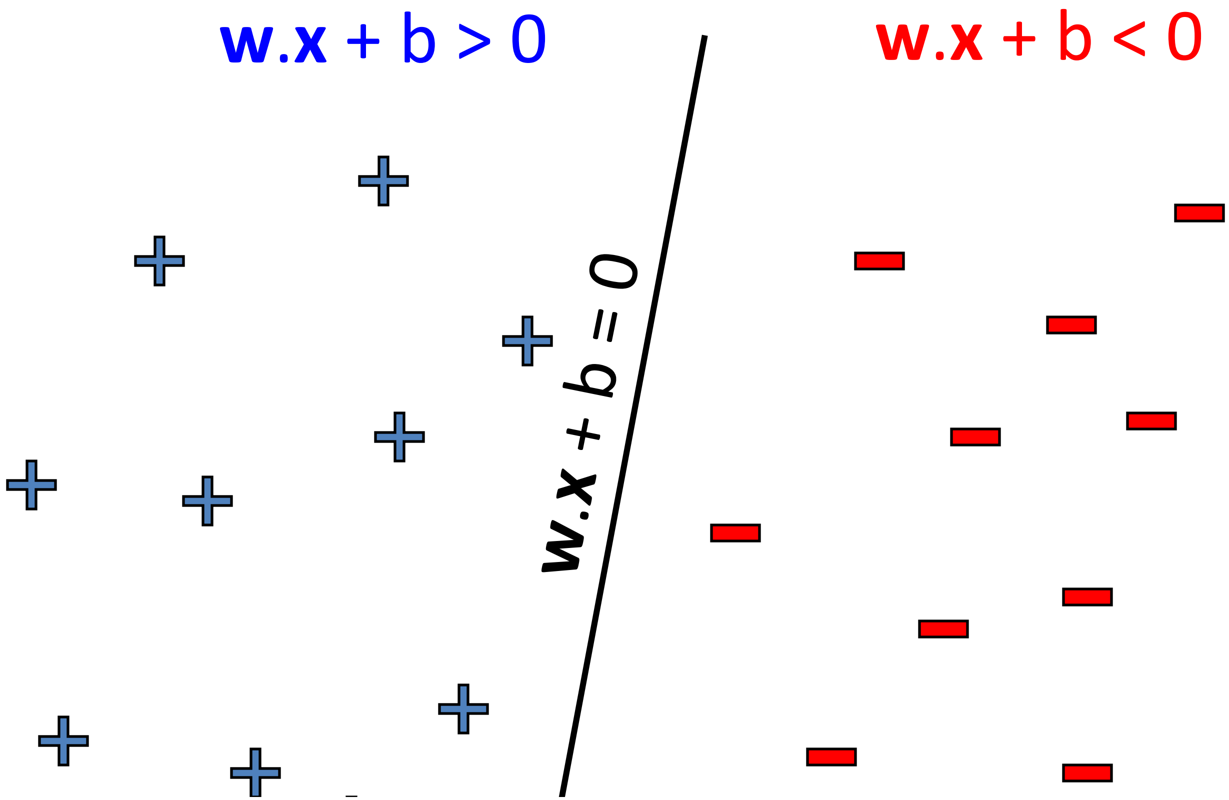
Linear classifiers – which line is better?



Pick the one with the largest margin!



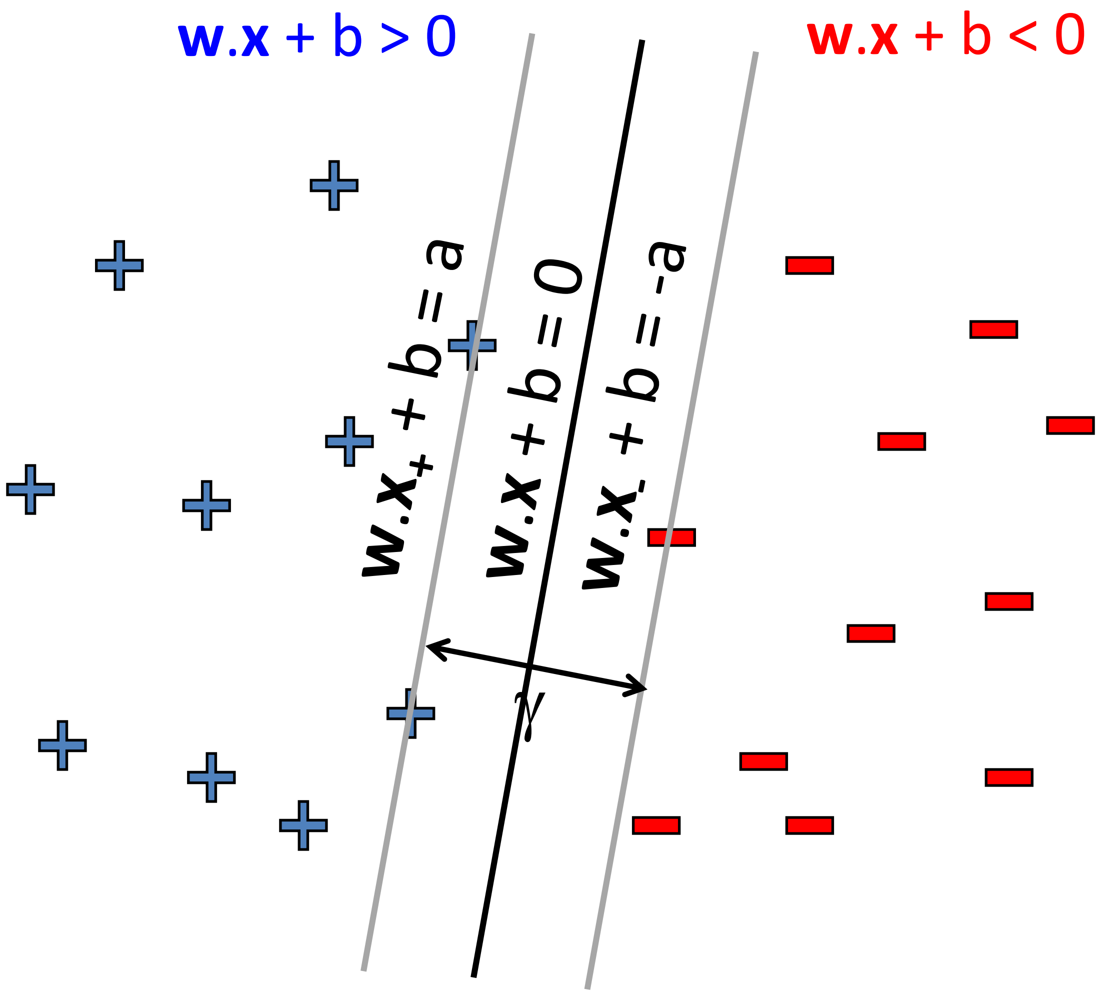
Parameterizing the decision boundary



$y_j \in \{-1, +1\}$ — class

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

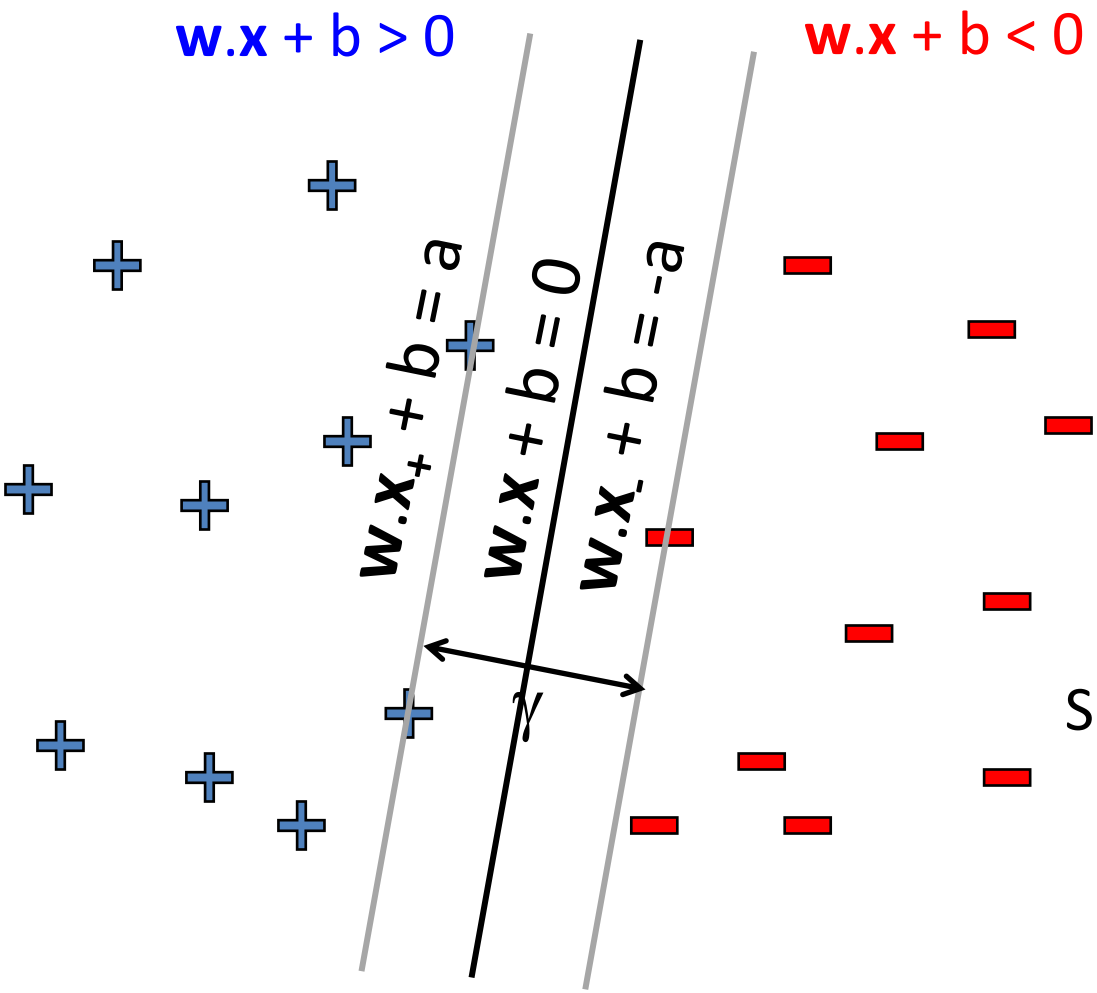
Maximizing the margin



Distance of closest examples from the line/hyperplane

margin = γ = 2a/‖w‖

Maximizing the margin

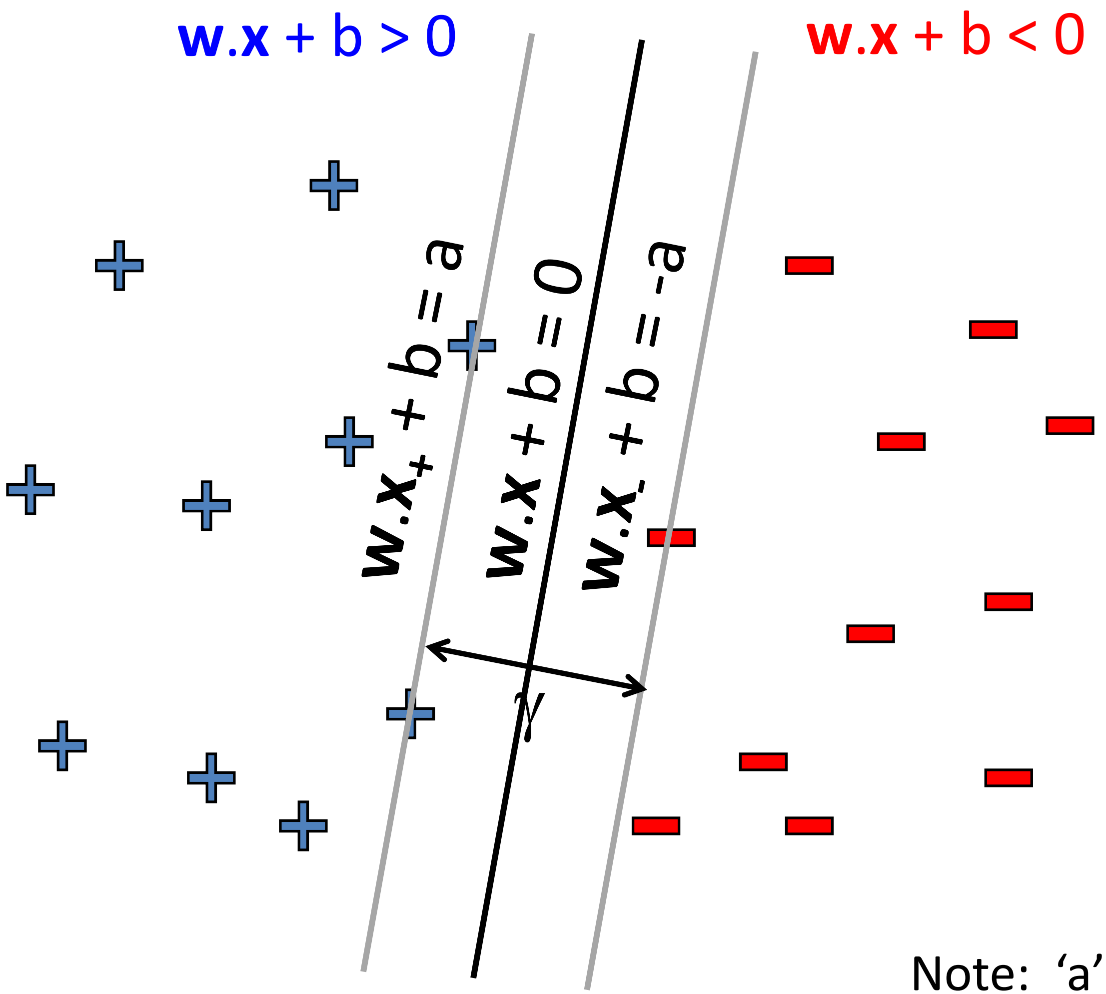


Distance of closest examples from the line/hyperplane

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

Smaller margin ⇔ larger $\|w\|$

Maximizing the margin



Distance of closest examples from the line/hyperplane

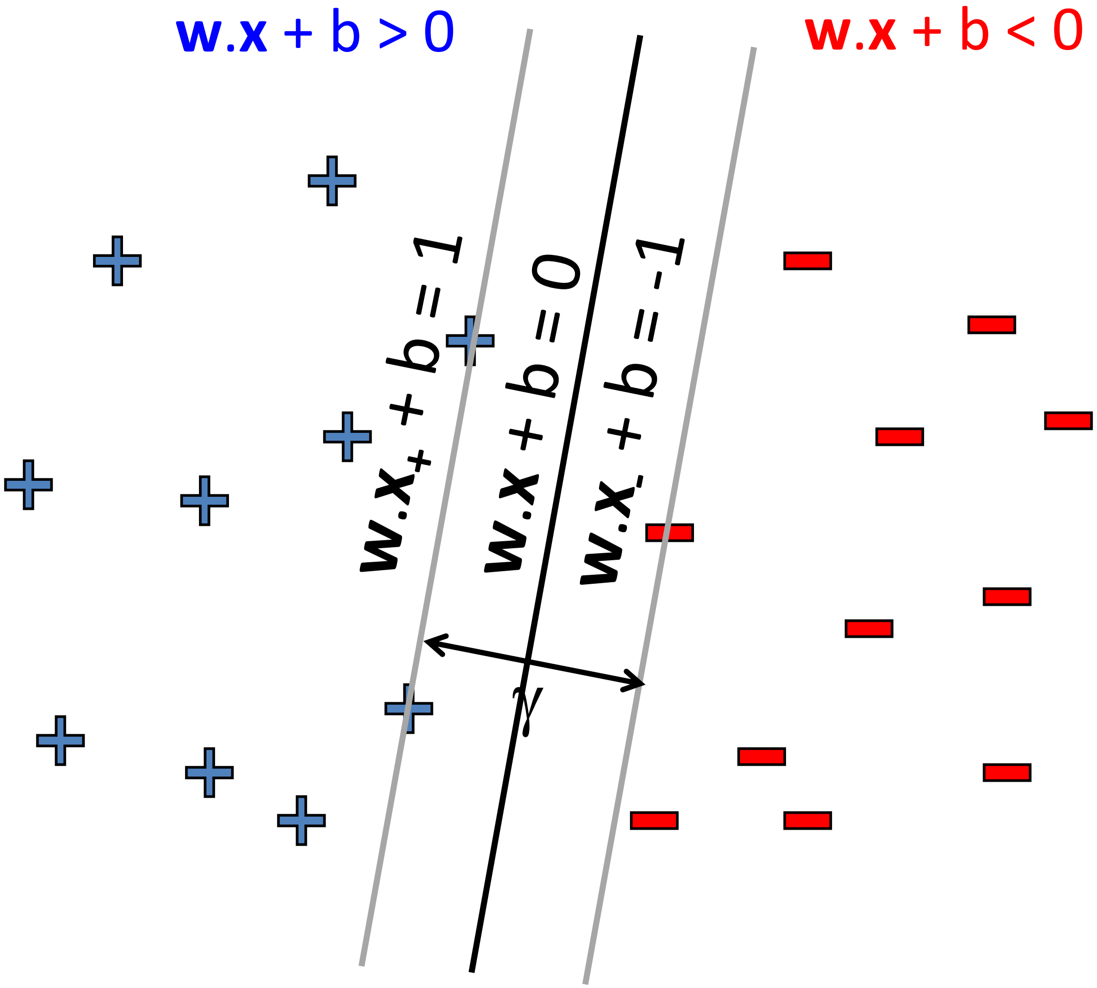
margin = $\gamma = 2a/\|w\|$

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

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

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

Support Vector Machines



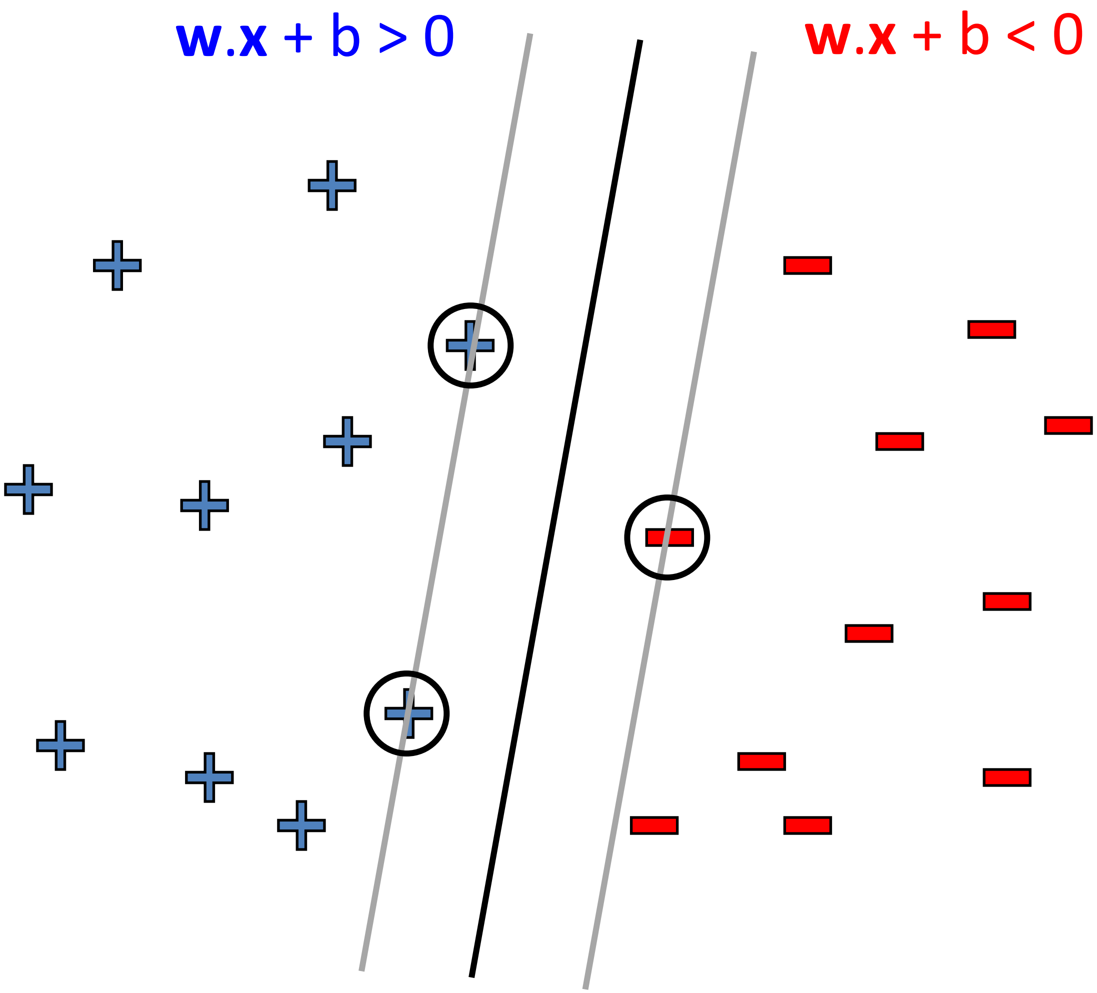
$$\min_{\mathbf{w},b} \ \mathbf{w}.\mathbf{w}$$

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

Solve efficiently by quadratic programming (QP)

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

Support Vectors



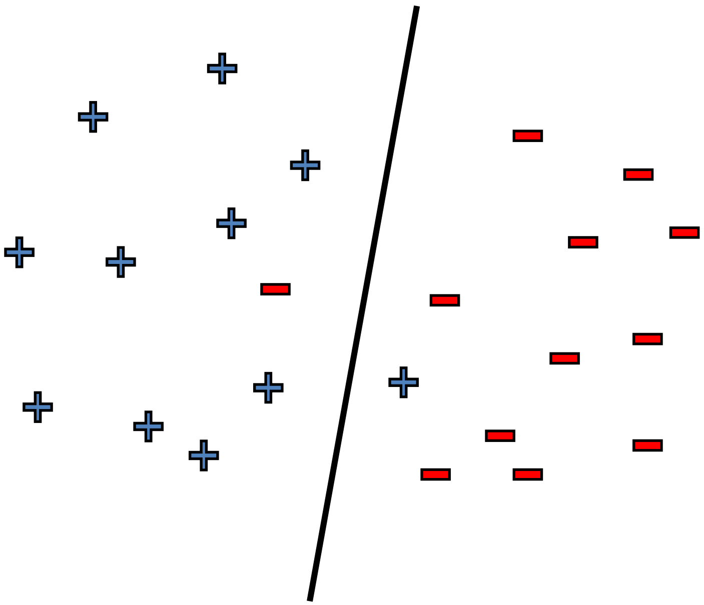
Linear hyperplane defined by "support vectors"

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?



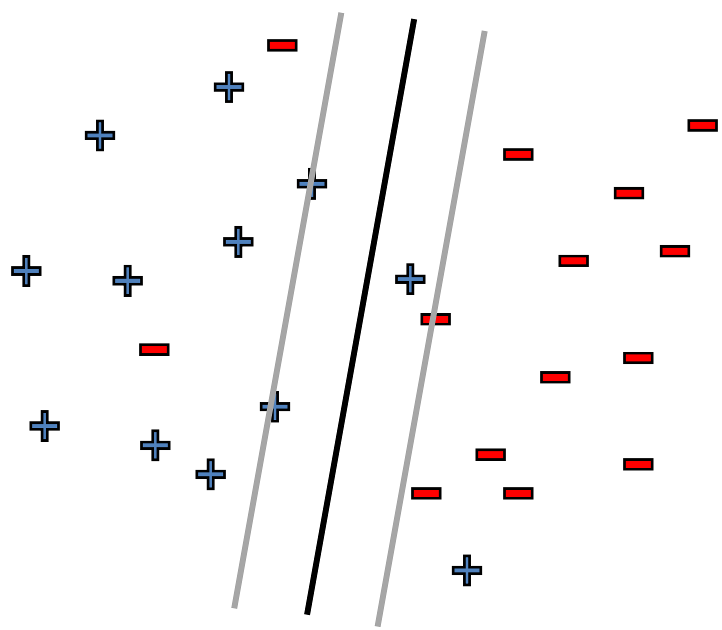
Use features of features of features of features....

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

But run risk of overfitting!

What if data is still not linearly separable?

Allow "error" in classification



Smaller margin ⇔ larger $\|w\|$

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

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

Maximize margin and minimize # mistakes on training data

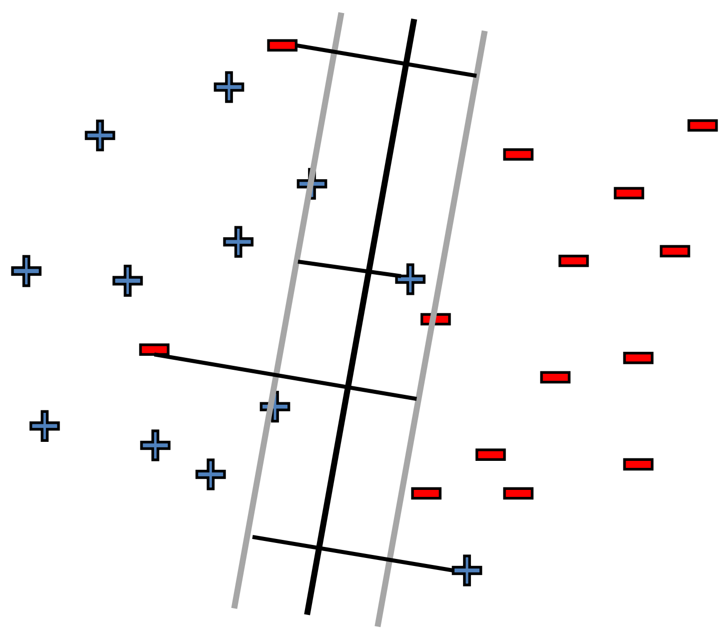
C - tradeoff parameter

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

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

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

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

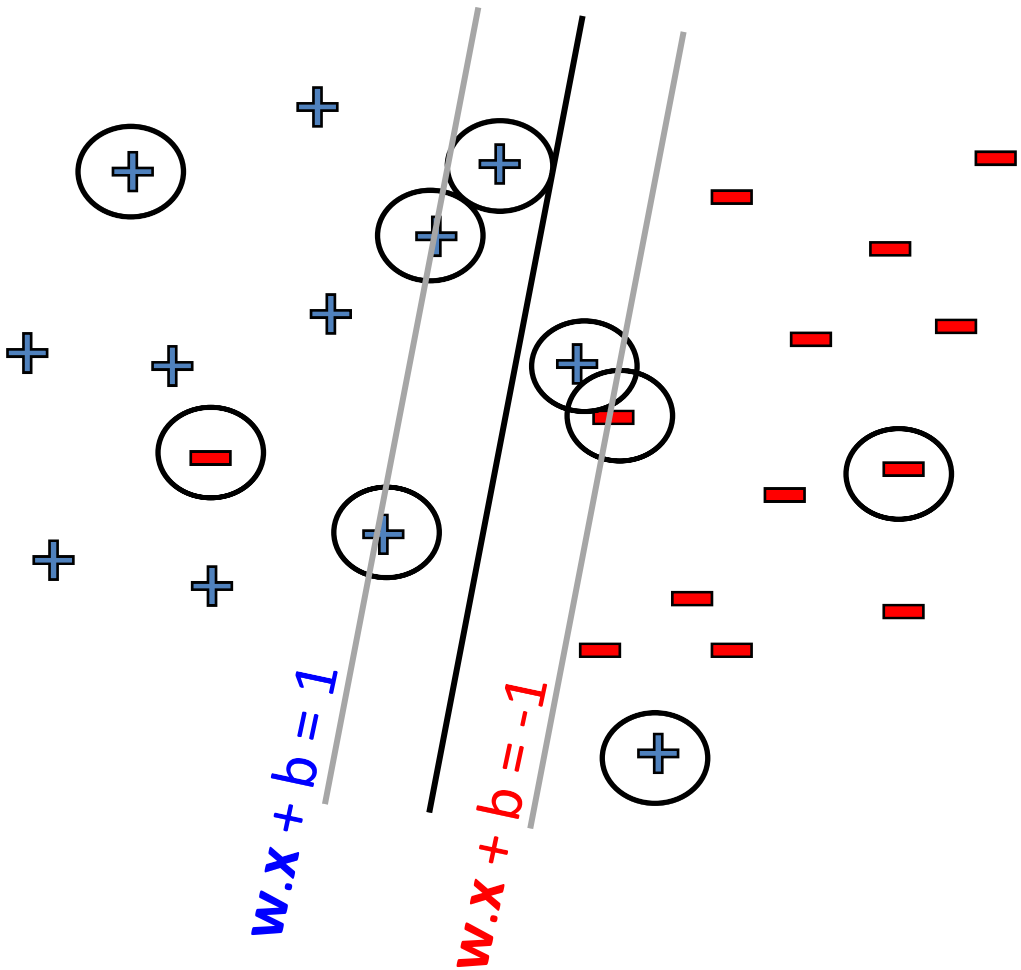
ξ_j - "slack" variables = (>1 if x_j misclassifed)

pay linear penalty if mistake

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

Still QP ☺

Slack variables – Hinge loss

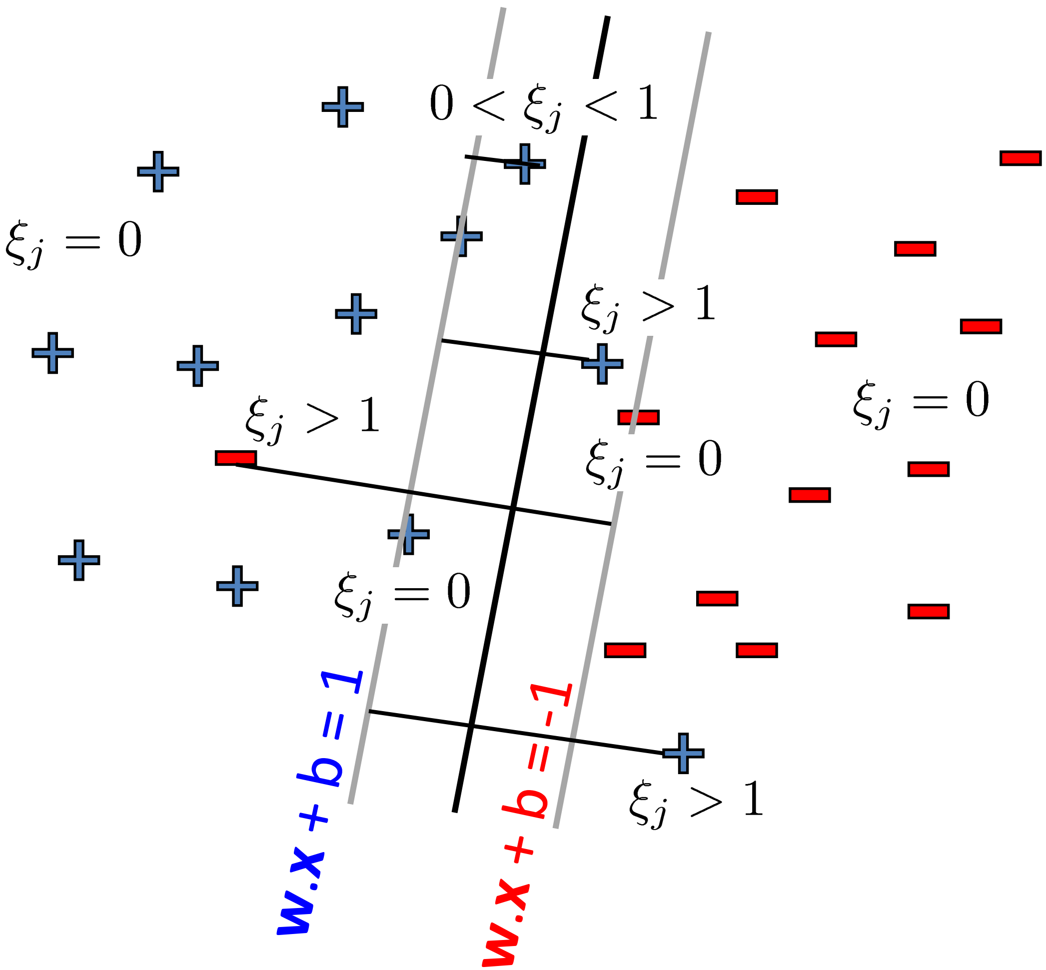


$(\mathbf{w}.\mathbf{x}_j+b)\ y_j \geq 1-\xi_j \quad \forall j$

What is the slack ξ_j for the following points?

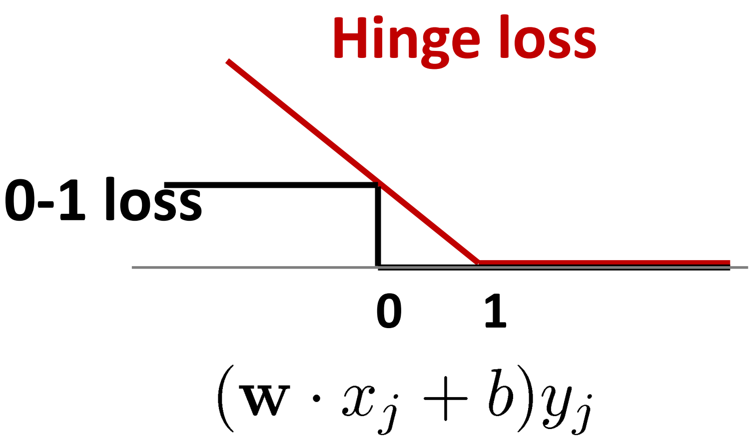
Confidence | Slack

Slack variables – Hinge loss



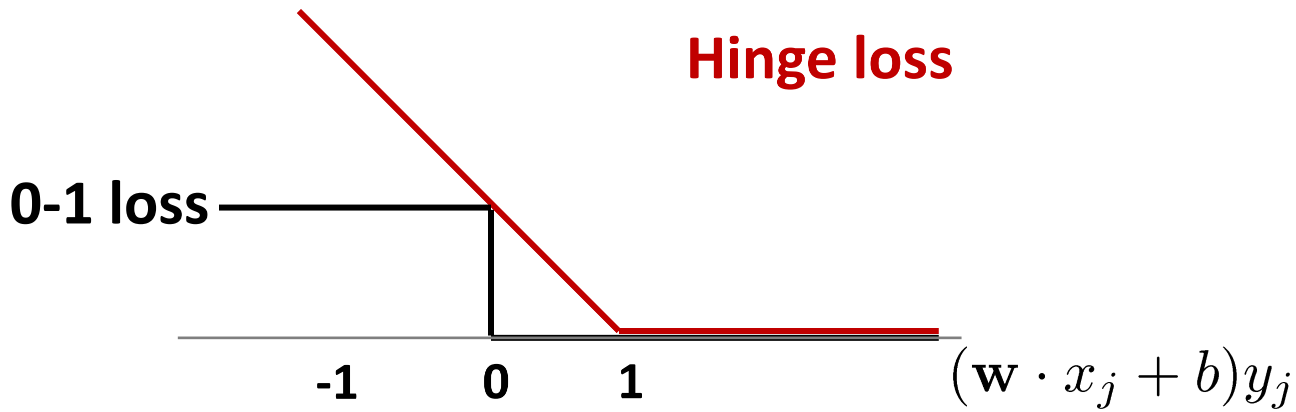
Notice that

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



Slack variables – Hinge loss

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



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

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

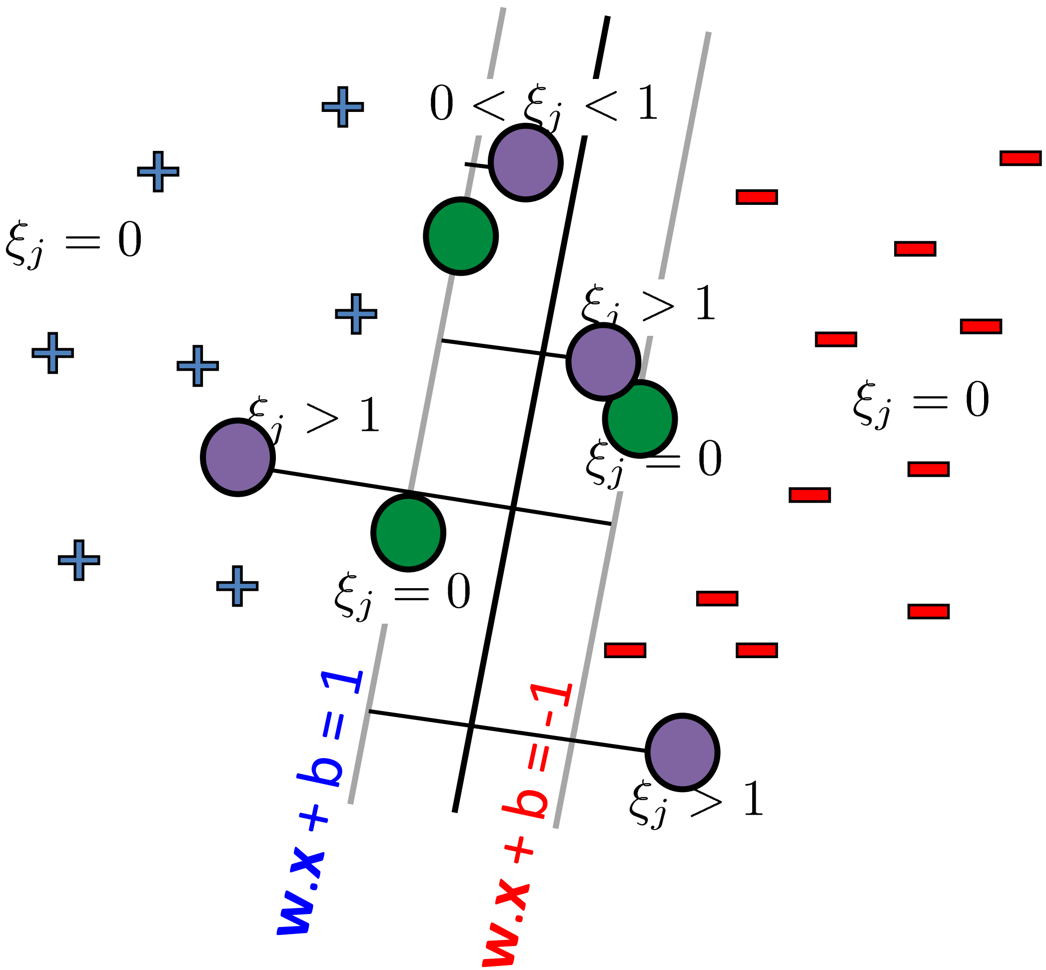
$$\xi_j \geq 0 \quad \forall j$$

⇔

Regularized hinge loss

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

Support Vectors



Margin support vectors

$\xi_j = 0$, $(\mathbf{w}.\mathbf{x}_j + b)\, y_j = 1$

(don't contribute to objective but enforce constraints on solution)

Correctly classified but on margin

Non-margin support vectors

$\xi_j > 0$

(contribute to both objective and constraints)

$1 > \xi_j > 0$ Correctly classified but inside margin

$\xi_j > 1$ Incorrectly classified