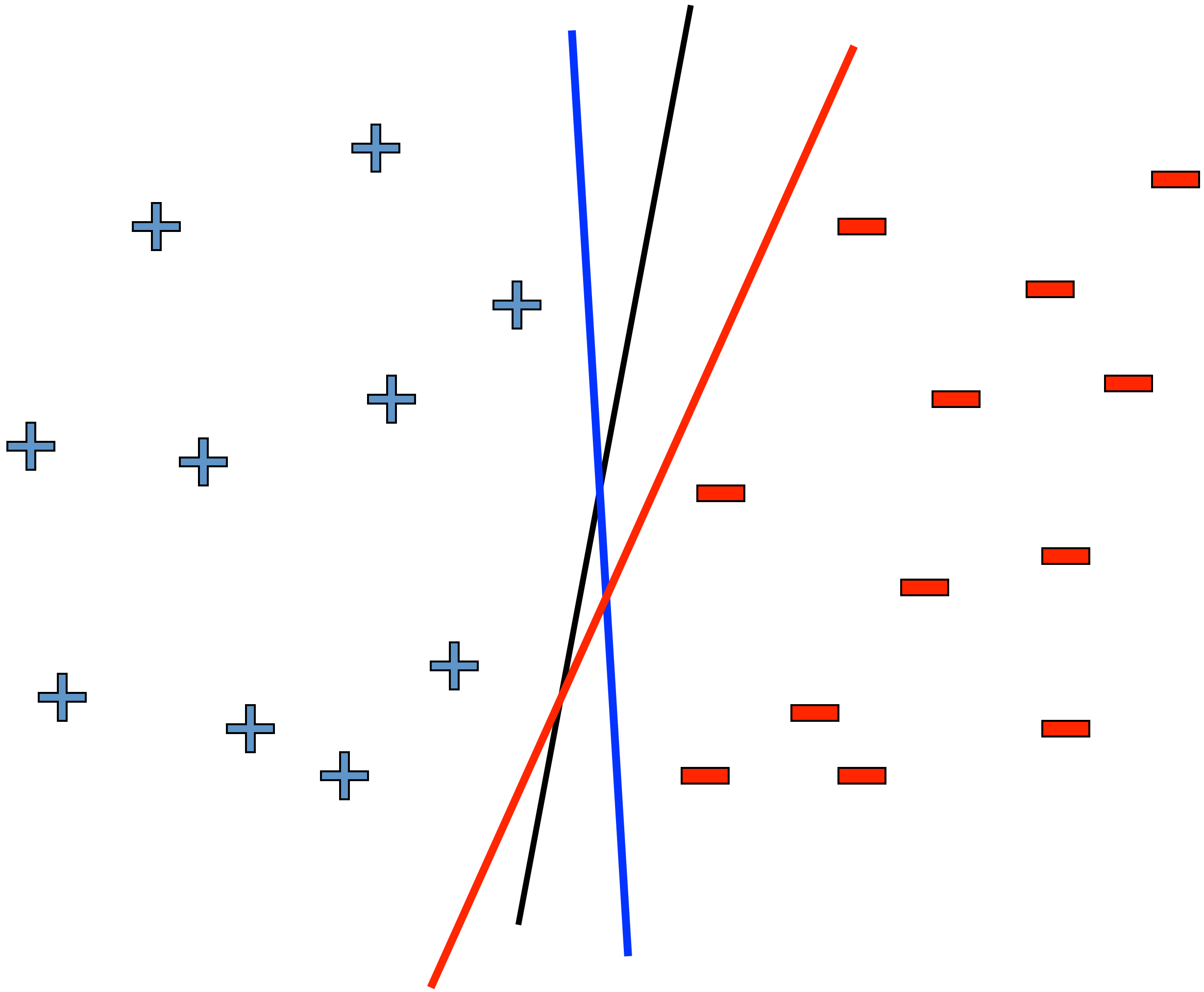
Support Vector Machines

Aarti Singh
Co-instructor: Pradeep Ravikumar

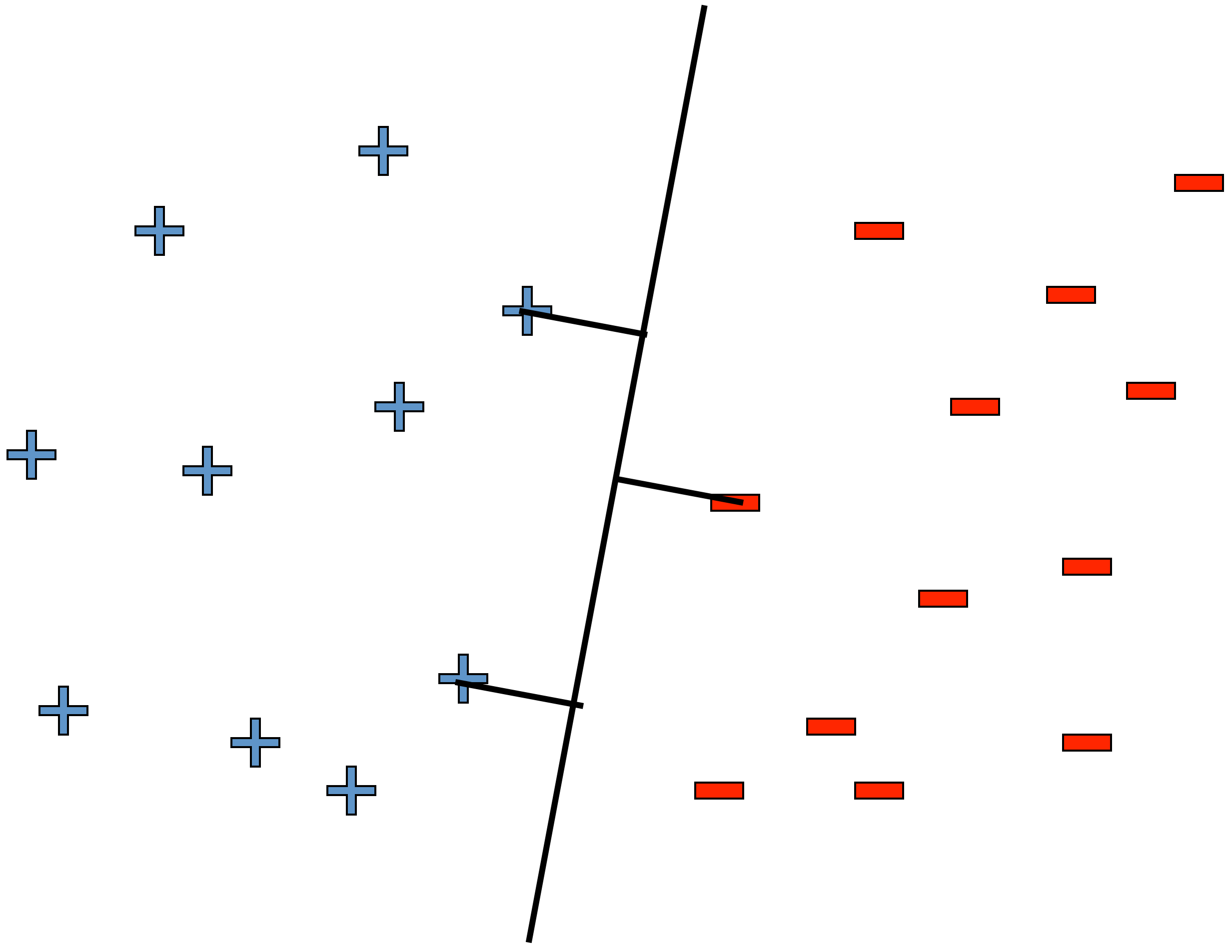
Machine Learning 10-701
Feb 13, 2017

At Pittsburgh G-20 summit …

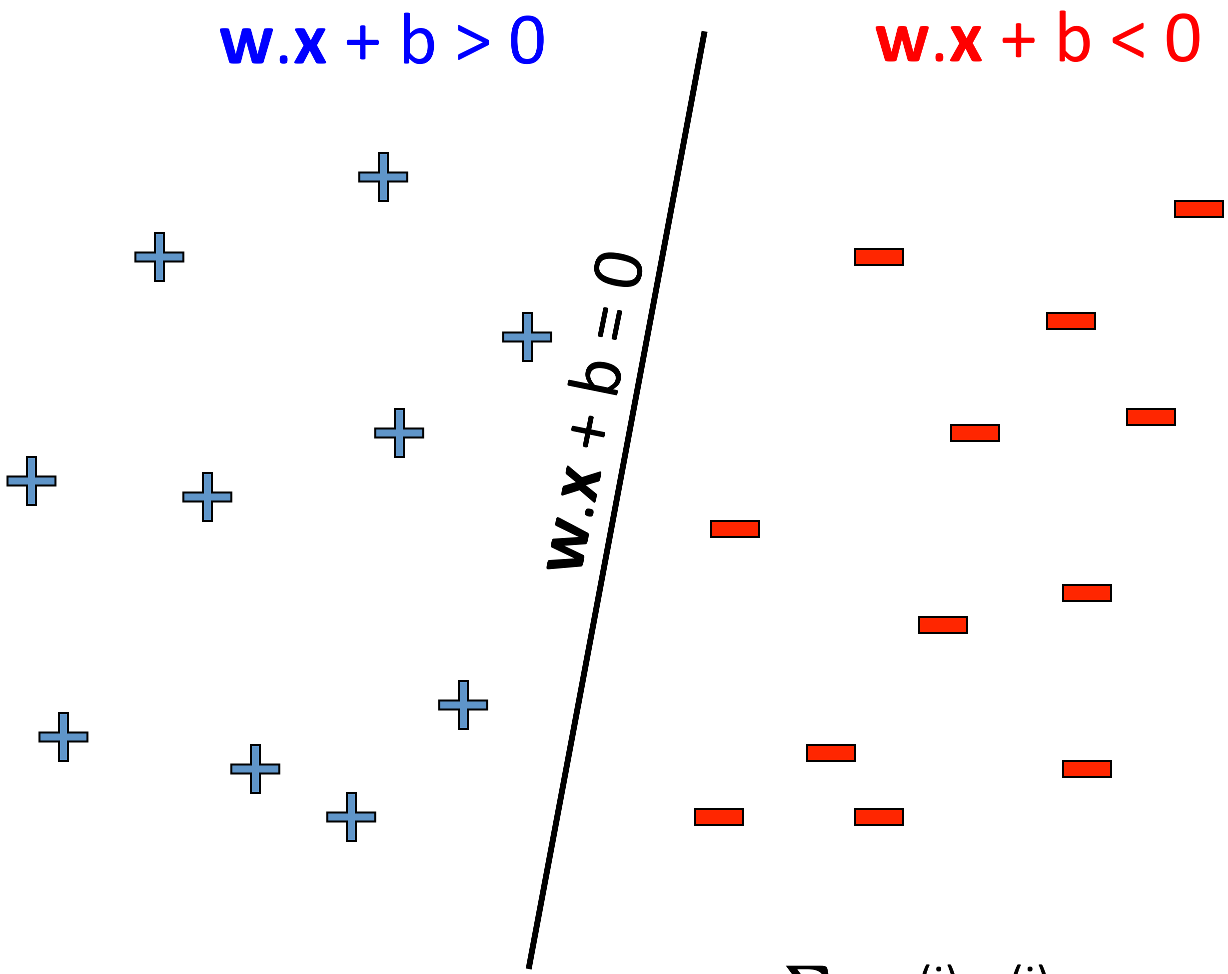
Linear classifiers – which line is better?



Pick the one with the largest margin!

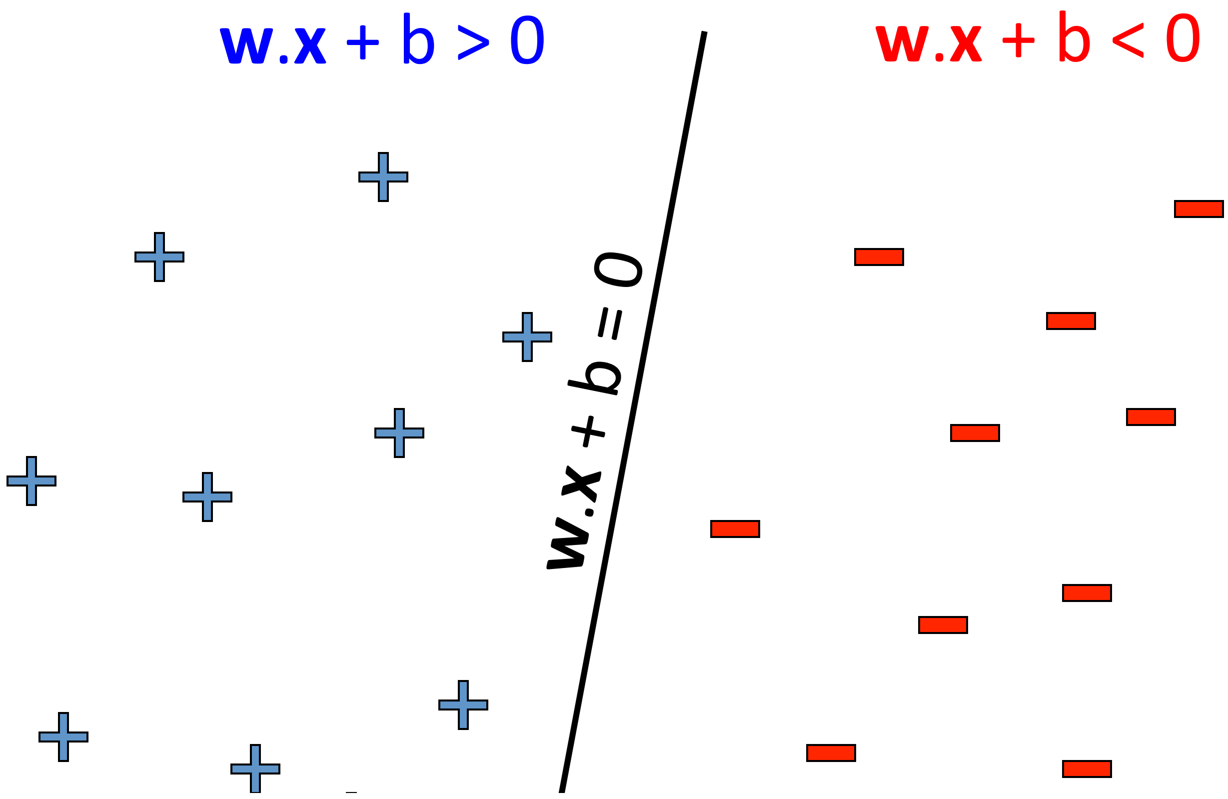


Parameterizing the decision boundary



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

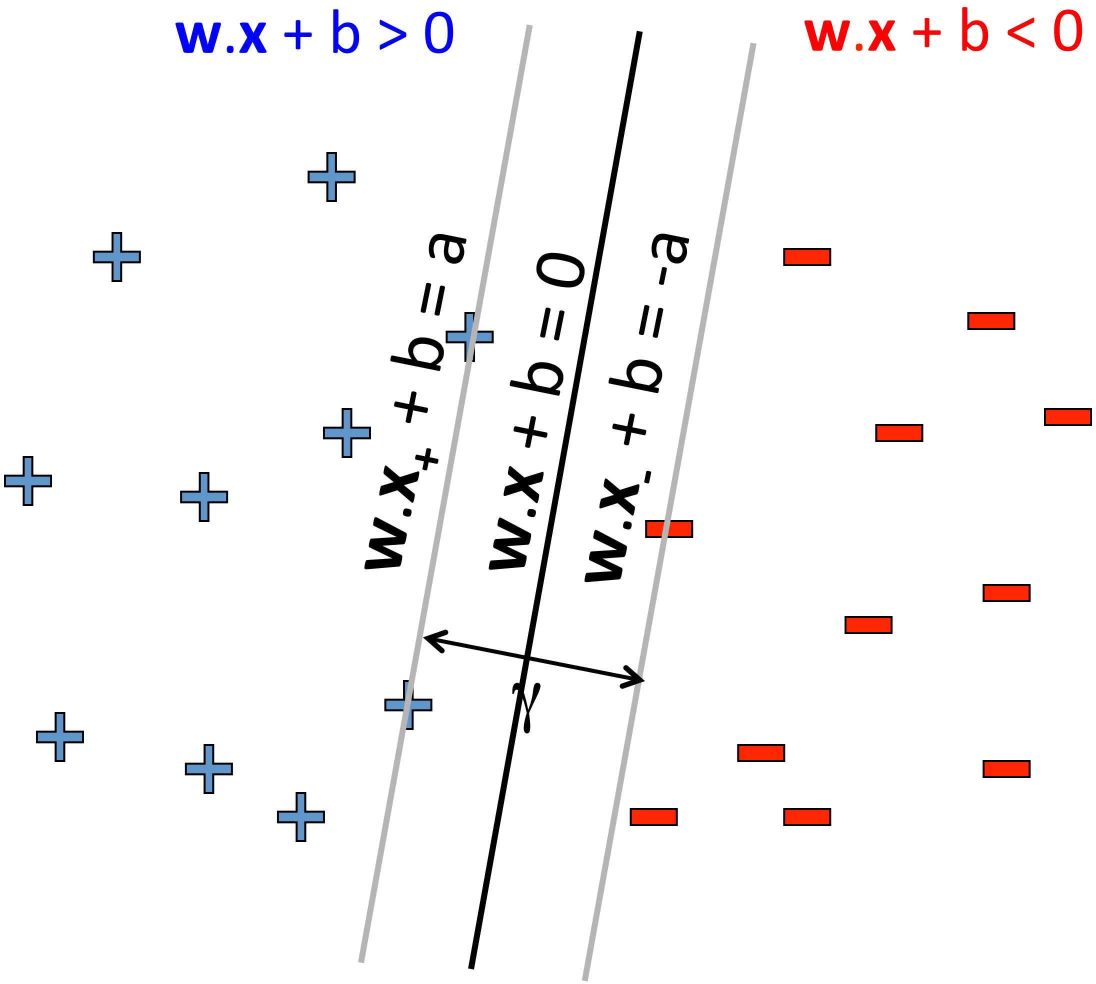
Parameterizing the decision boundary



$$y_j \in \{-1, +1\} \quad \text{— class}$$

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

Maximizing the margin



Distance of closest examples from the line/hyperplane

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

w is perpendicular to lines

$$\mathbf{w}.(\mathbf{x}_+ - \mathbf{x}_-) = 0$$

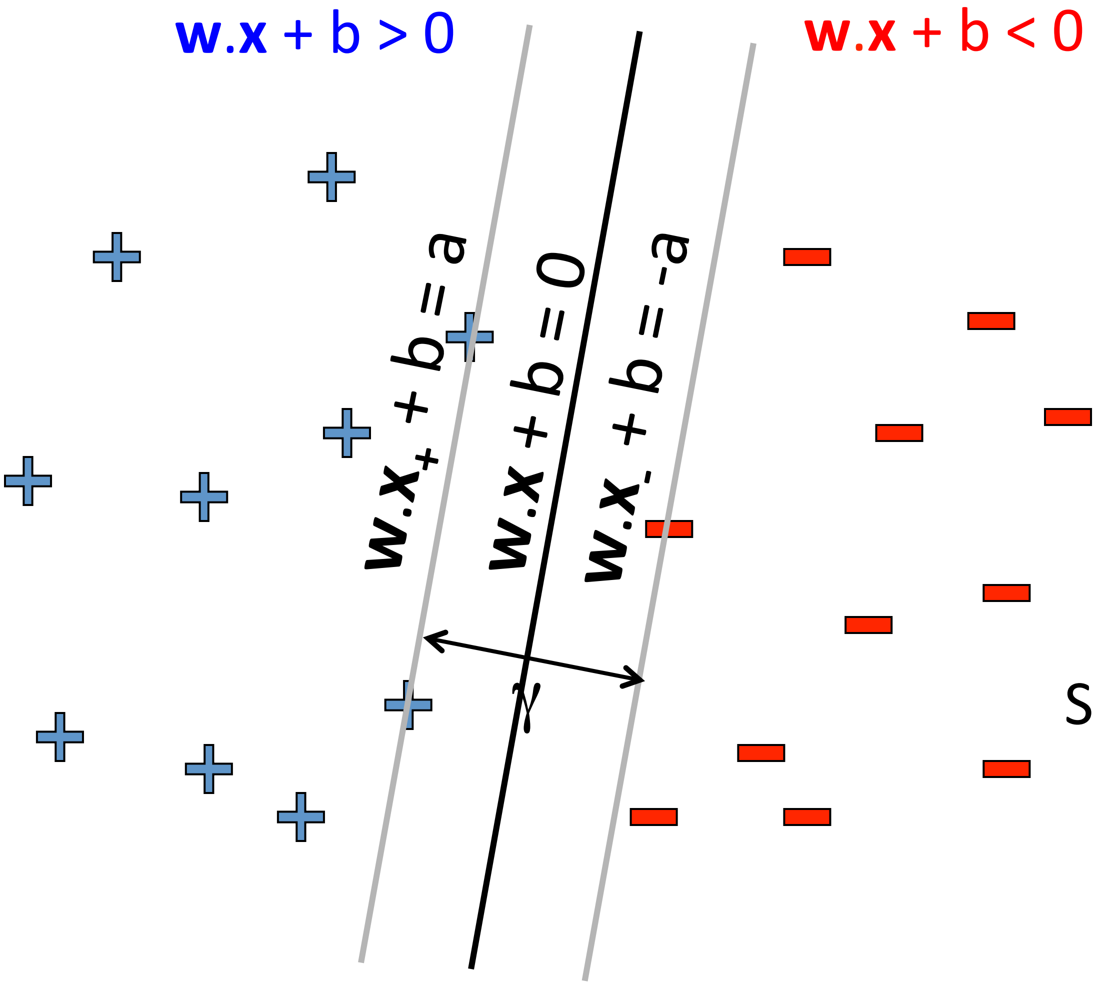
$$\mathbf{x}_+ = \mathbf{x}_- + \gamma \mathbf{w}/\|w\|$$

$$\mathbf{w}.\mathbf{x}_+ = \mathbf{w}.\mathbf{x}_- + \gamma\|w\|$$

$$a\text{-}b = -a\text{-}b + \gamma\|w\|$$

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

Maximizing the margin

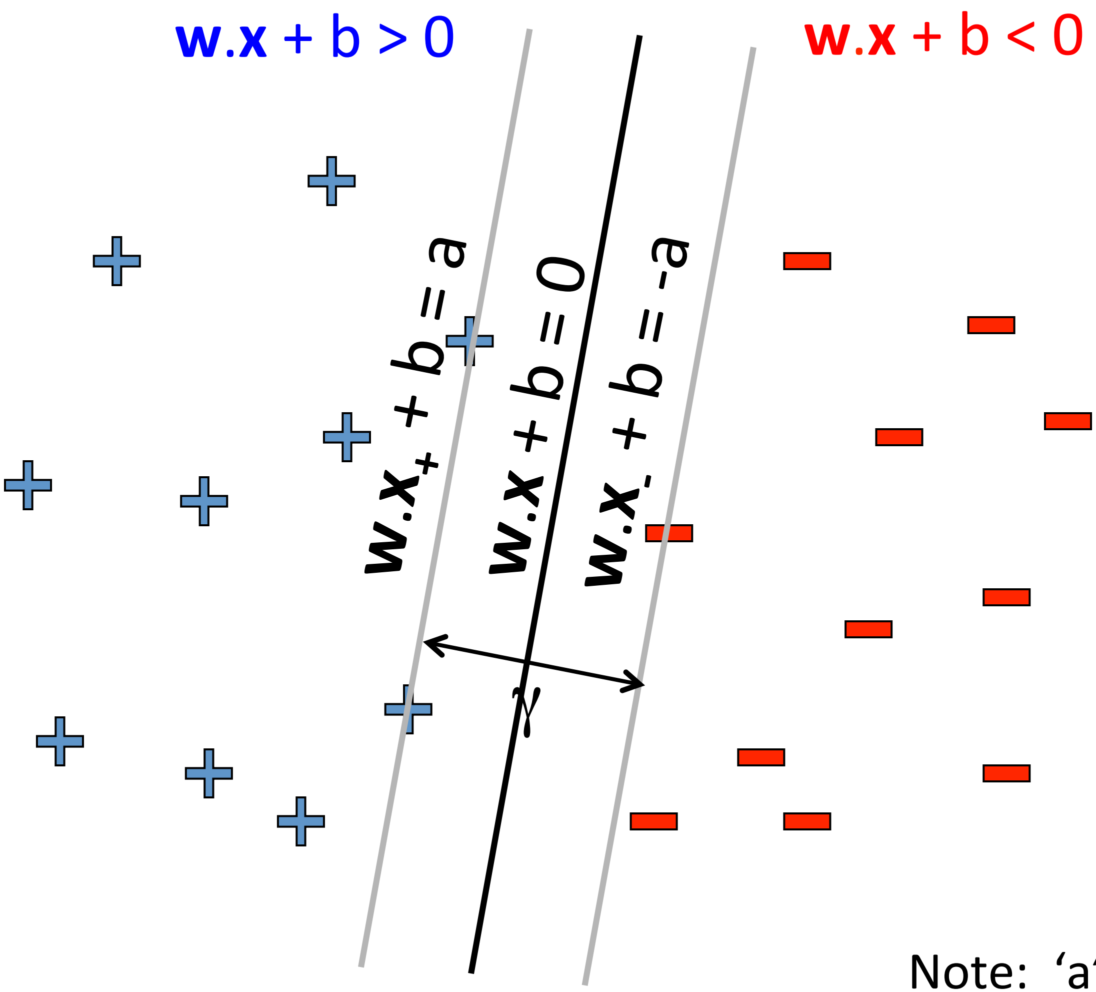


Distance of closest examples from the line/hyperplane

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

Smaller margin ⇔ larger $\|w\|$

Maximizing the margin



Distance of closest examples from the line/hyperplane

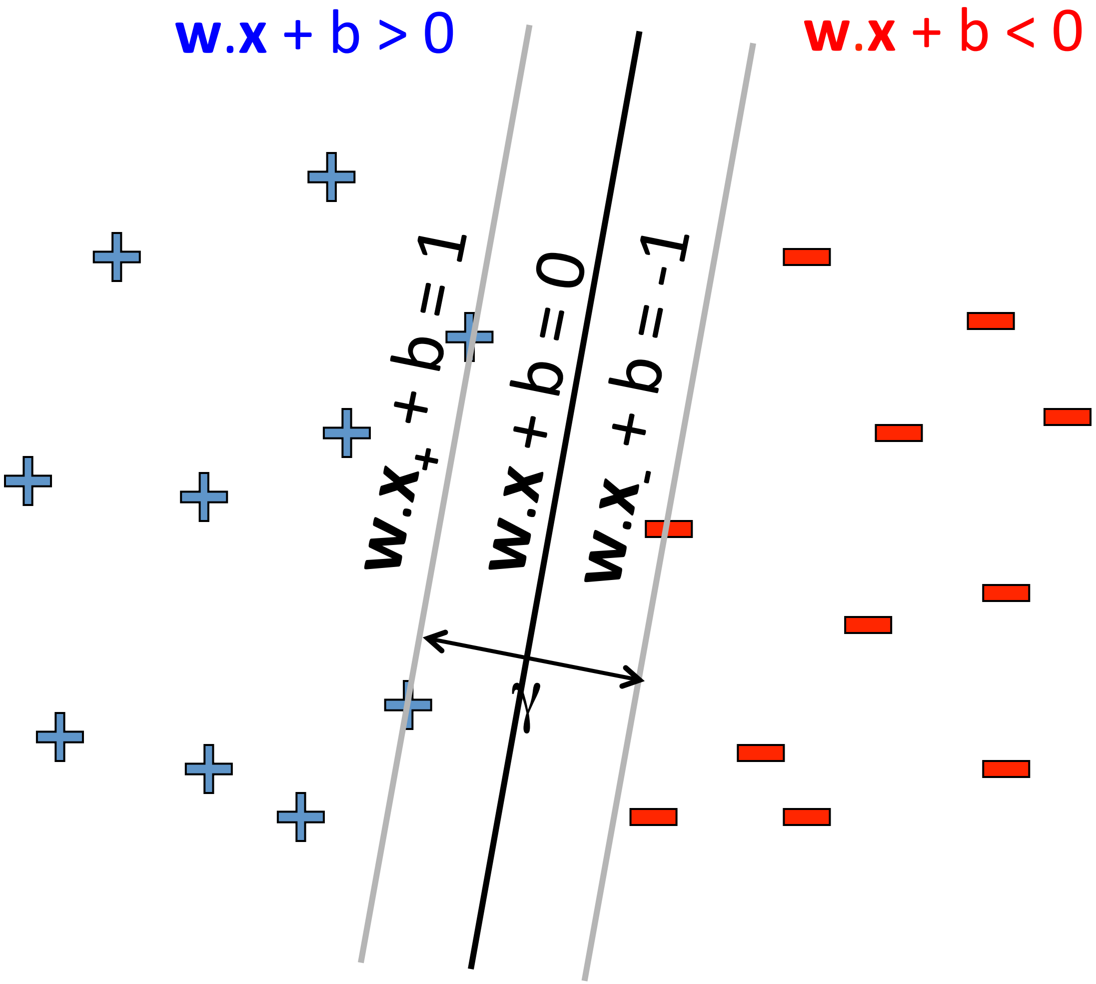
$$\boxed{\text{margin} = \gamma = 2a/\|w\|}$$

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

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

<u>Note:</u> 'a' is arbitrary (can normalize equations by a)

Support Vector Machines



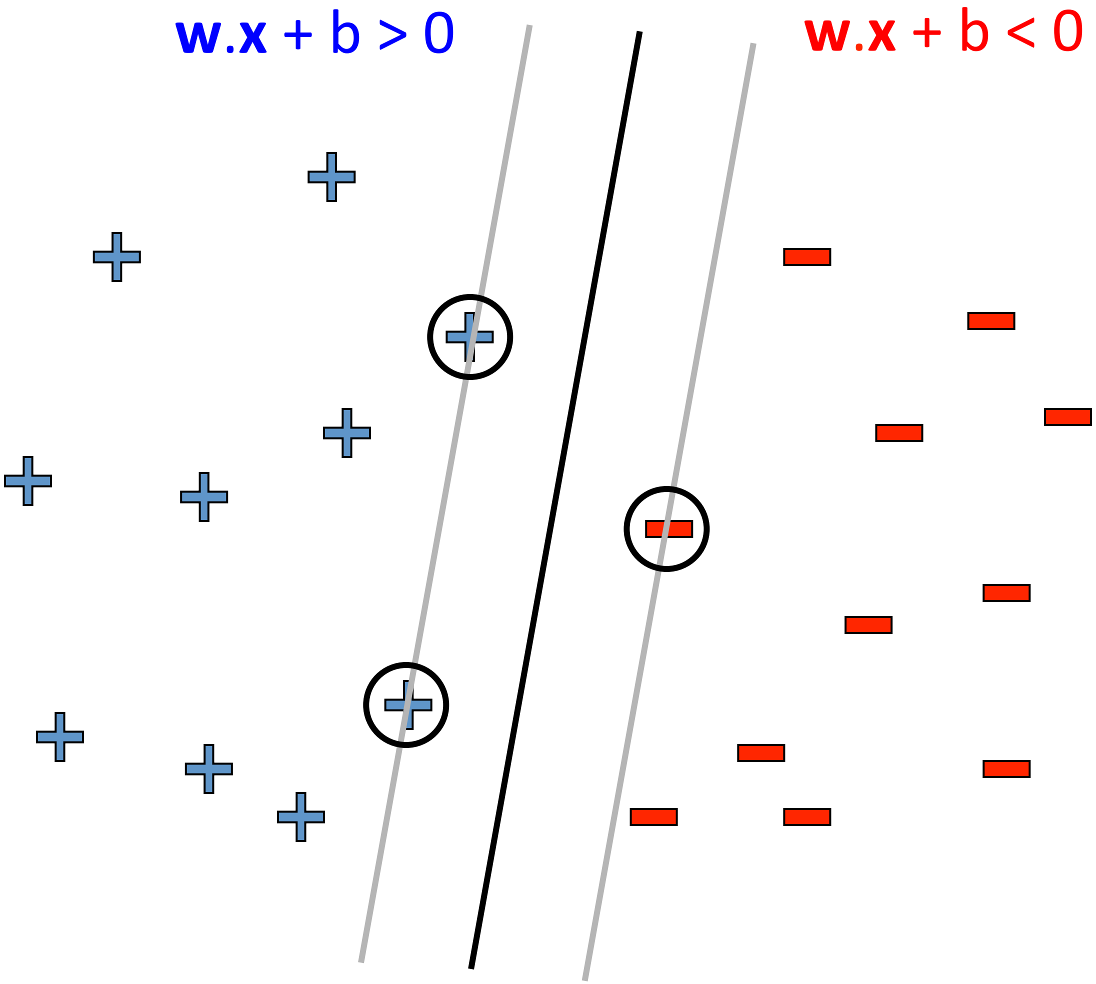
$$\min_{\mathbf{w},b} \ \mathbf{w}.\mathbf{w}$$

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

Solve efficiently by quadratic programming (QP)

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

Support Vectors



Linear hyperplane defined by "support vectors"

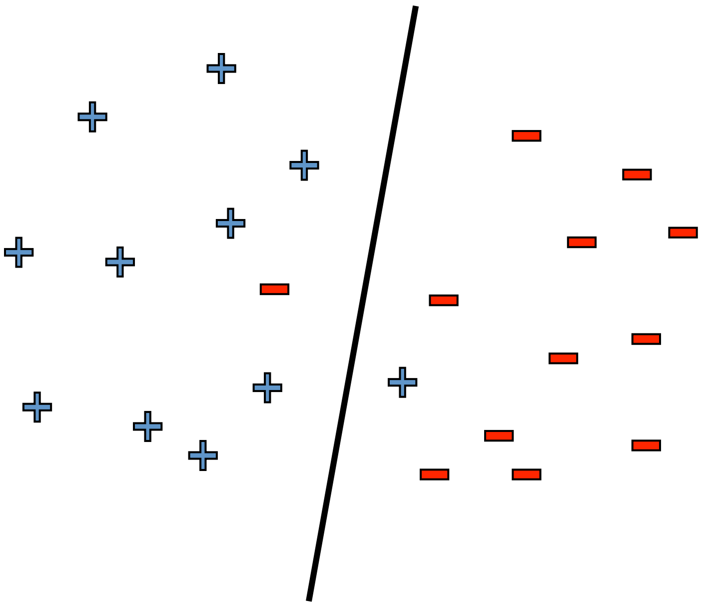
Moving other points a little doesn't effect the decision boundary

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?



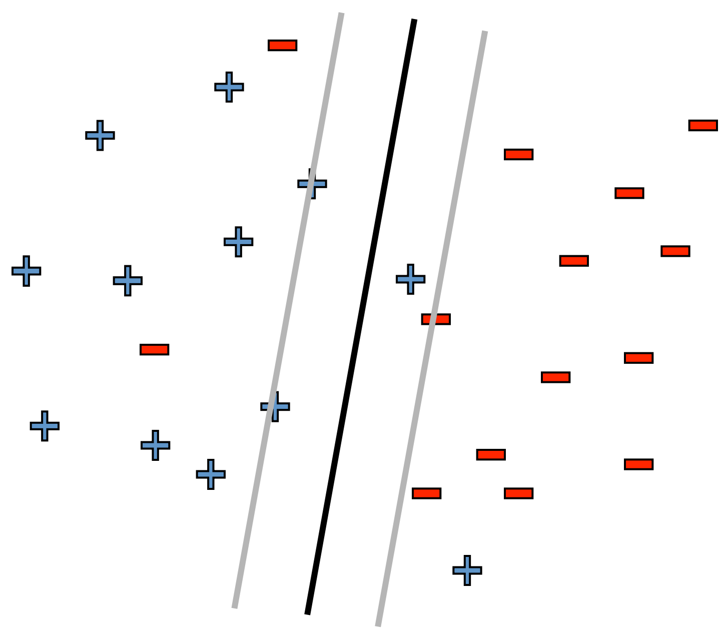
Use features of features of features of features....

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

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)

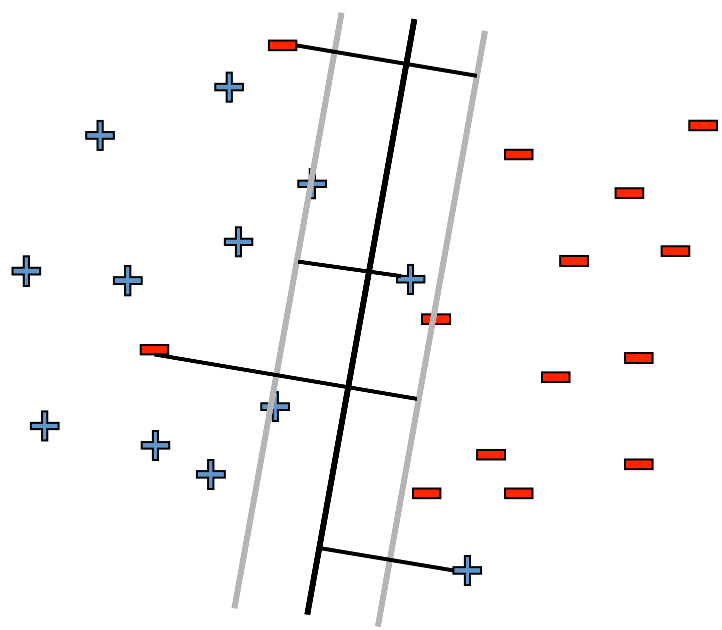
What if data is still not linearly separable?

Allow "error" in classification

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

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

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



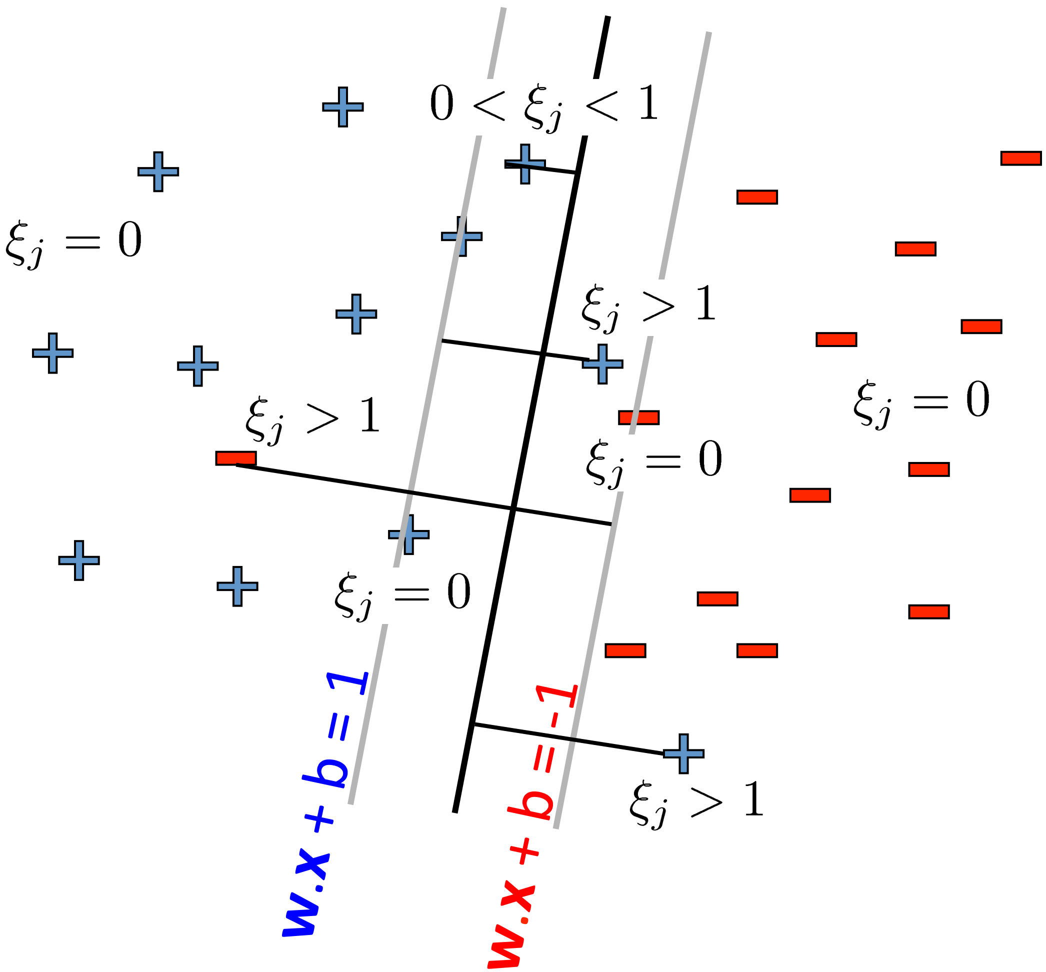
Soft margin approach

ξ_j - "slack" variables
= (>1 if x_j misclassifed)
pay linear penalty if mistake

C - tradeoff parameter (chosen by cross-validation)

Still QP ☺

Soft-margin SVM



Soften the constraints:

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

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

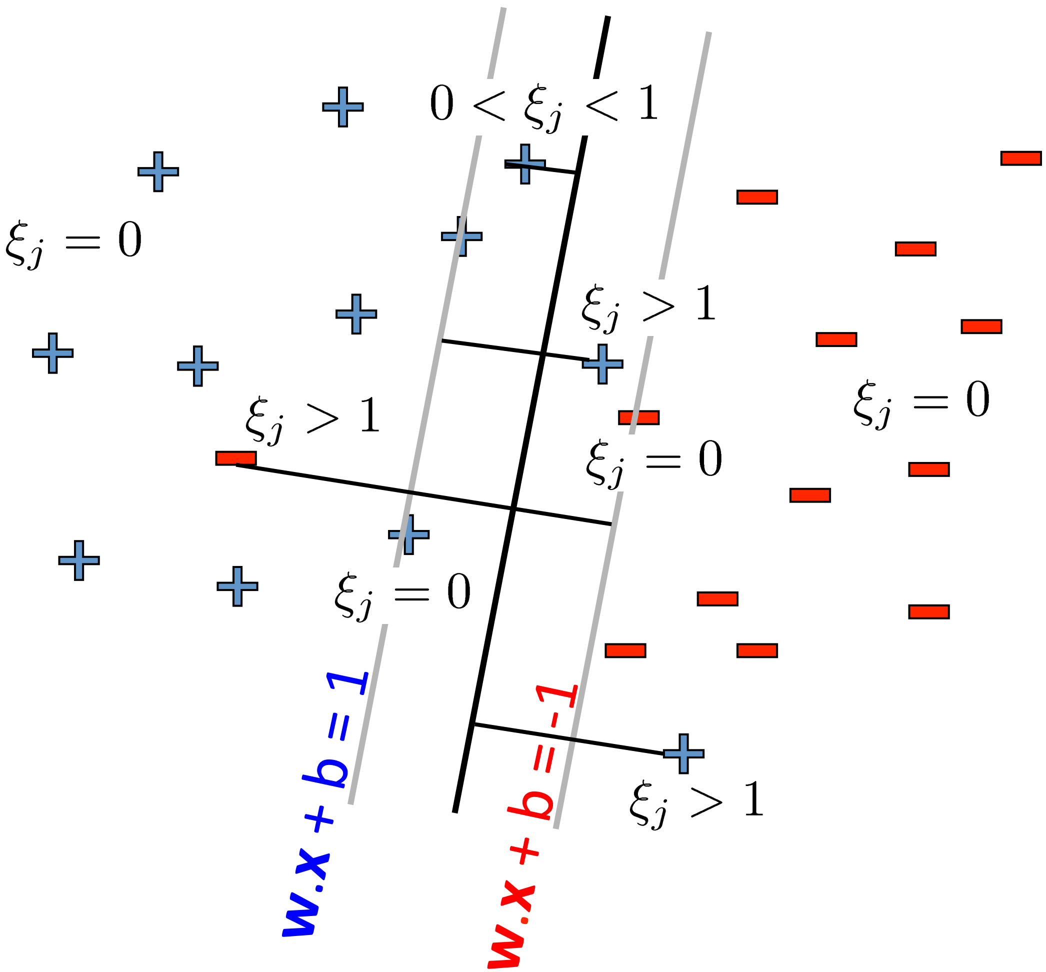
Penalty for misclassifying:

$$C\, \xi_j$$

How do we recover hard margin SVM?

Set $C = \infty$

Slack variables – Hinge loss

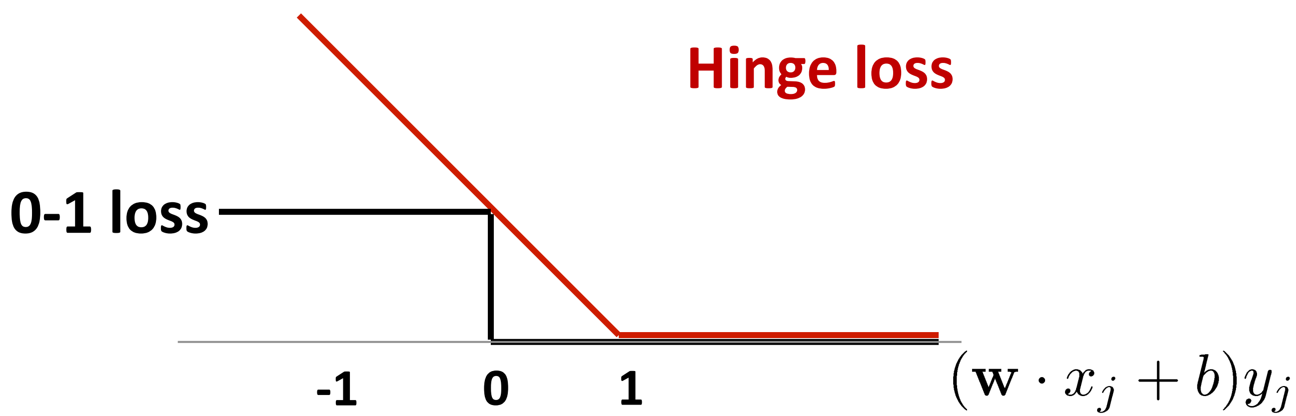


Notice that

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

Slack variables – Hinge loss

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



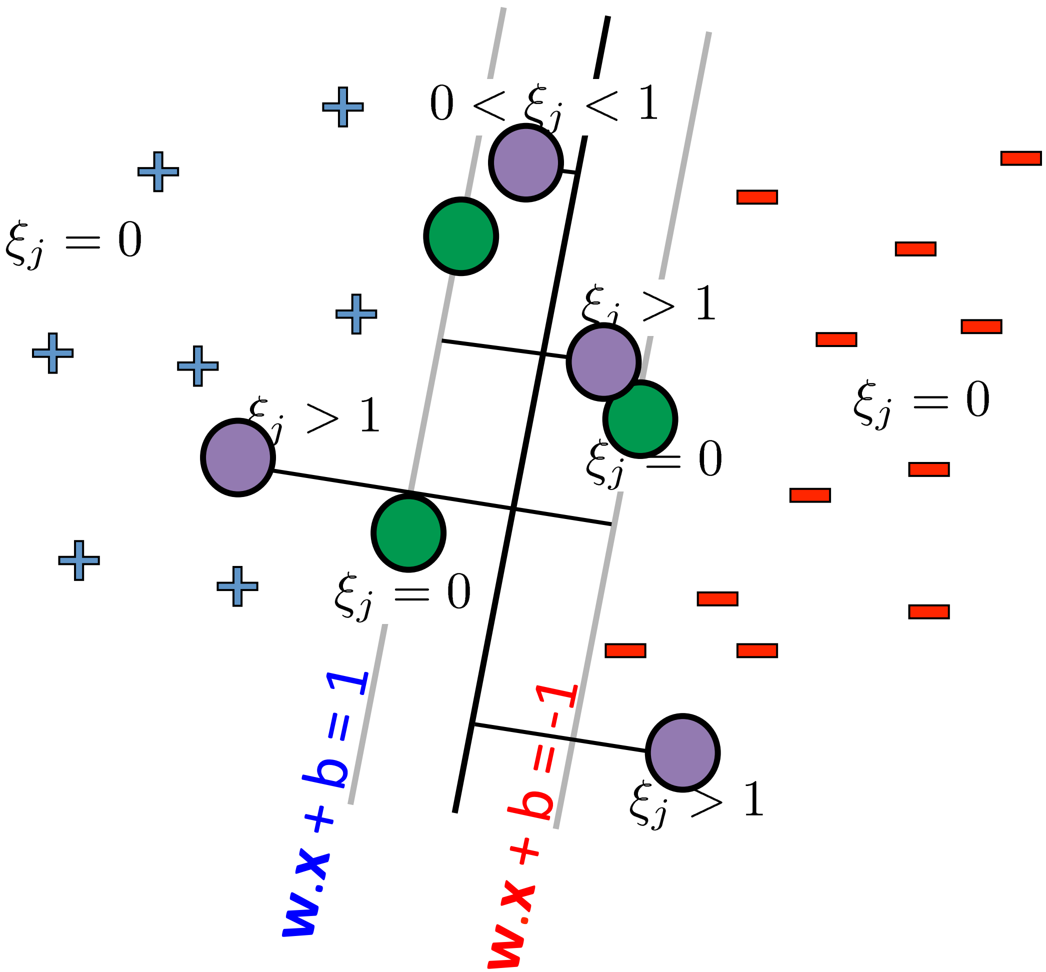
$$\min_{\mathbf{w},b} \ \mathbf{w}.\mathbf{w} + C \sum_j \xi_j$$
$$\text{s.t. } (\mathbf{w}.\mathbf{x}_j + b)\, y_j \geq 1 - \xi_j \quad \forall j$$
$$\xi_j \geq 0 \quad \forall j$$

$\Leftrightarrow$

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

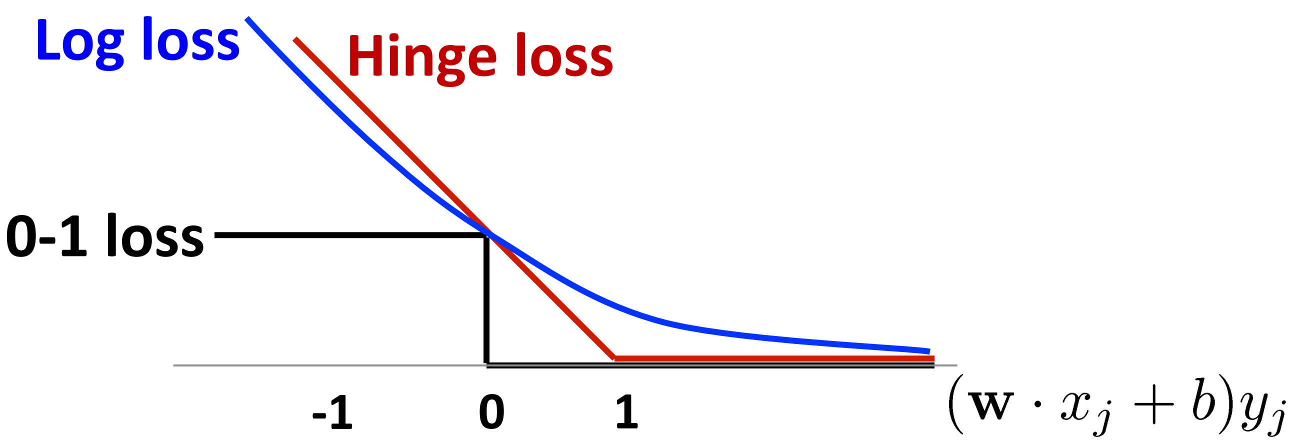
SVM vs. Logistic Regression

SVM : **Hinge loss**

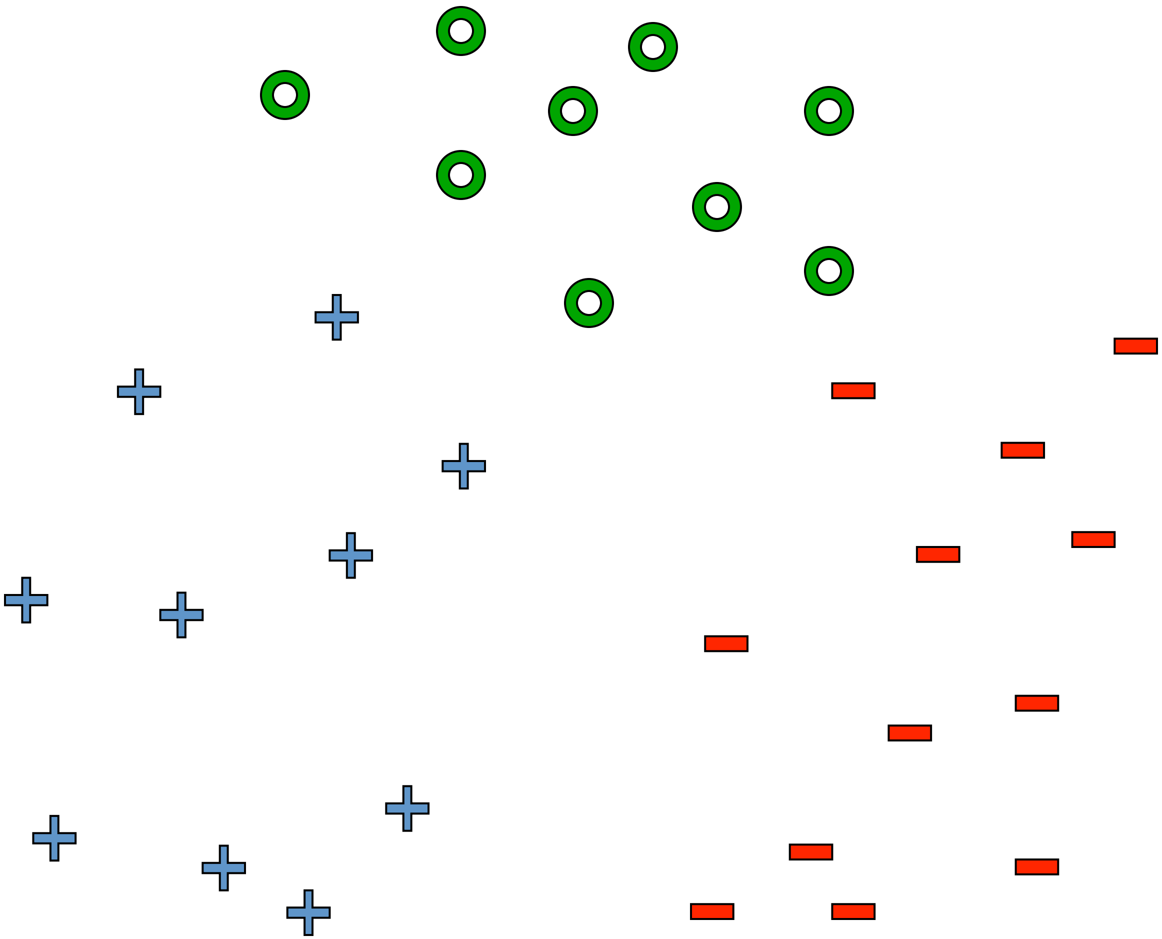
$$\mathrm{loss}(f(x_j), y_j) = (1 - (\mathbf{w} \cdot x_j + b)y_j))_+$$

Logistic Regression : **Log loss** (-ve log conditional likelihood)

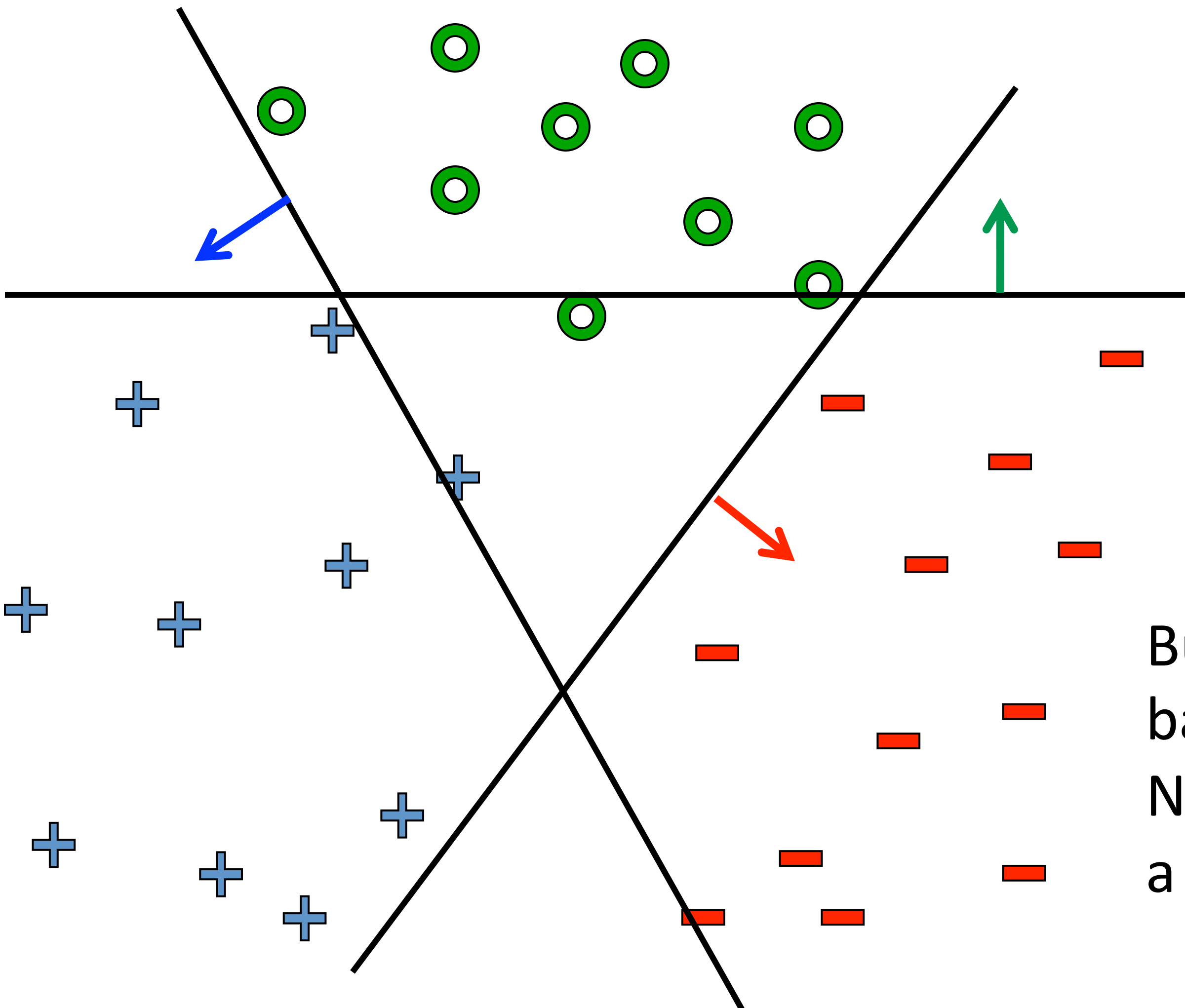
$$\mathrm{loss}(f(x_j), y_j) = -\log P(y_j \mid x_j, \mathbf{w}, b) = \log(1 + e^{-(\mathbf{w} \cdot x_j + b)y_j})$$



What about multiple classes?



One vs. rest



Learn 3 classifiers separately:
Class k vs. rest

$(\mathbf{w}_k, b_k)_{k=1,2,3}$

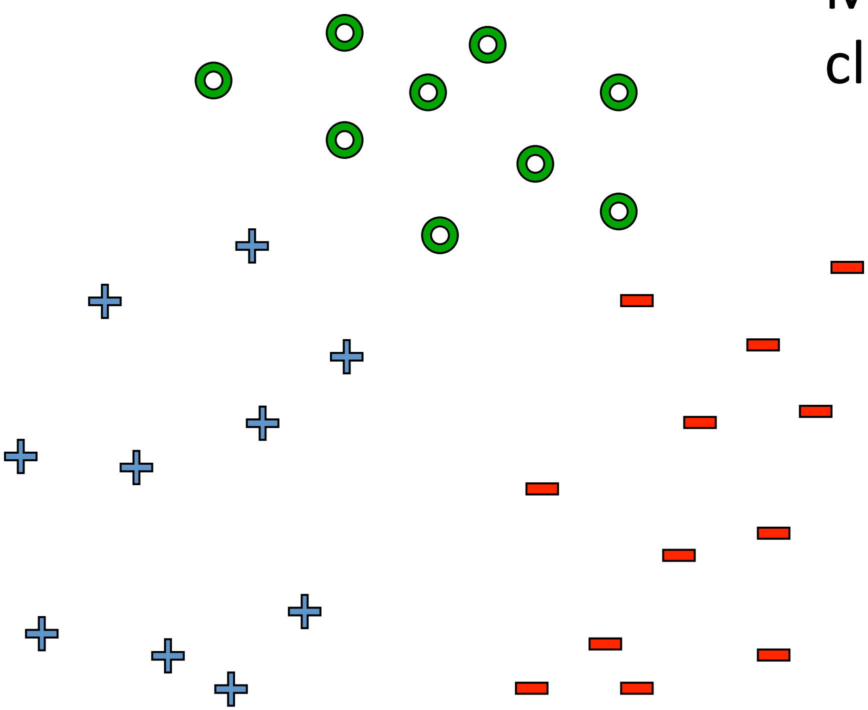
$$y = \arg\max_k \mathbf{w}_k . x + b_k$$

But $\mathbf{w}_k$s may not be based on the same scale.
Note: $(a\mathbf{w}).x + (ab)$ is also a solution

Learn 1 classifier: Multi-class SVM

Simultaneously learn 3 sets of weights

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



Margin - gap between correct class and nearest other class

$y = \arg\max_k \mathbf{w}^{(k)}.x + b^{(k)}$

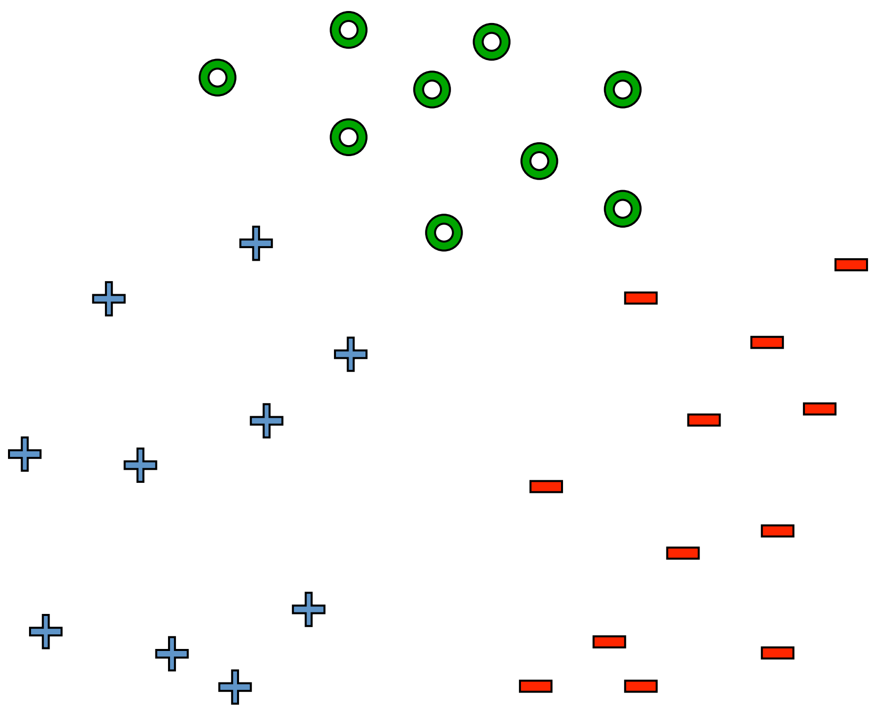
Learn 1 classifier: Multi-class SVM

Simultaneously learn 3 sets of weights

$$\text{minimize}_{\mathbf{w},b} \quad \sum_y \mathbf{w}^{(y)}.\mathbf{w}^{(y)} + C\sum_j \sum_{y \neq y_j} \xi_j^{(y)}$$

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

$$\xi_j^{(y)} \geq 0 \qquad , \ \forall y \neq y_j, \ \forall j$$



$y = \arg\max \mathbf{w}^{(k)}.x + b^{(k)}$

Joint optimization: $\mathbf{w}_k$s have the same scale.

SVM – linearly separable case

n training points $(x_1, \ldots, x_n)$

d features x_i is a d-dimensional vector

- Primal problem: $\text{minimize}_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}.\mathbf{w}$

  $$\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j \geq 1, \ \ \forall j$$

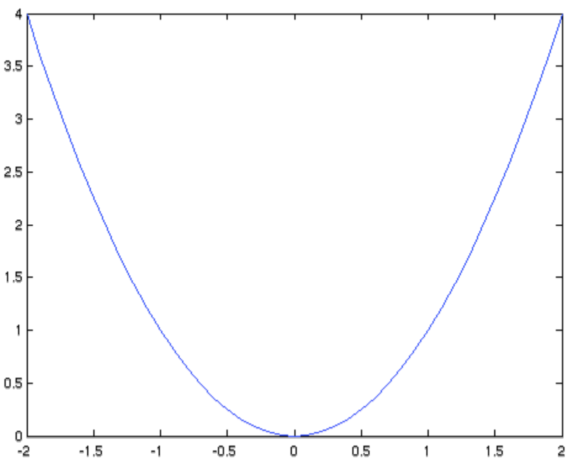
w – weights on features (d-dim problem)

- Convex quadratic program solution – quadratic objective, linear constraints
- But expensive to solve if d is very large
- Often solved in dual form (n-dim problem)

Constrained Optimization

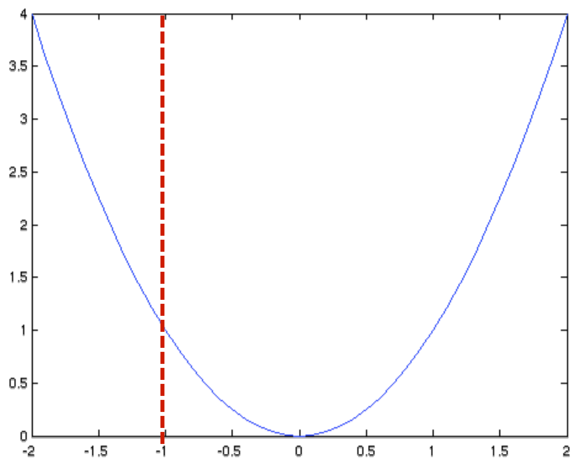
$$\min_x \; x^2$$
$$\text{s.t.} \quad x \geq b$$

$$\min_x \; x^2$$



$x^* = 0$

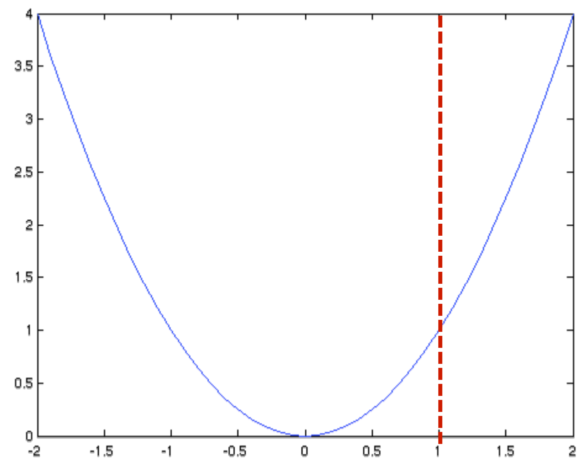
$$\min_x \; x^2$$
$$\text{s.t.} \quad x \geq -1$$



$x^* = 0$

Constraint inactive

$$\min_x \; x^2$$
$$\text{s.t.} \quad x \geq 1$$

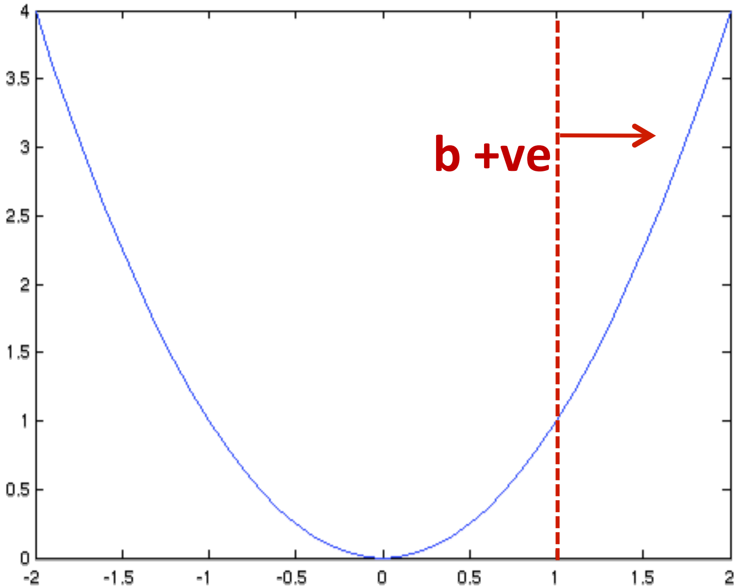


$x^* = 1$

Constraint active and tight

$$x^* = \max(b, 0)$$

Constrained Optimization – Dual Problem



$$x^* = b$$

Primal problem:

$$\min_x \; x^2$$
$$\text{s.t.} \quad x \geq b$$

Moving the constraint to objective function

Lagrangian:

$$L(x, \alpha) = x^2 - \alpha(x - b)$$
$$\text{s.t.} \quad \alpha \geq 0$$

Dual problem:

$$\max_\alpha \; d(\alpha) \rightarrow \min_x L(x, \alpha)$$
$$\text{s.t.} \quad \alpha \geq 0$$

α = 0 constraint is inactive

α > 0 constraint is active

Solving the dual

Solving:

$$\max_\alpha \min_x \underbrace{x^2 - \alpha(x-b)}_{L(x,\alpha)}$$
$$\text{s.t.} \quad \alpha \geq 0$$

Optimization over x is unconstrained.

$$\frac{\partial L}{\partial x} = 2x - \alpha = 0 \Rightarrow x^* = \frac{\alpha}{2}$$

$$L(x^*, \alpha) = \frac{\alpha^2}{4} - \alpha\left(\frac{\alpha}{2} - b\right)$$
$$= -\frac{\alpha^2}{4} + b\alpha$$

Now need to maximize $L(x^*,\alpha)$ over $\alpha \geq 0$

Solve unconstrained problem to get α' and then take $\max(\alpha',0)$

$$\frac{\partial}{\partial \alpha} L(x^*, \alpha) = -\frac{\alpha}{2} + b \quad \Rightarrow \alpha' = 2b$$

$$\Rightarrow \alpha^* = \max(2b, 0) \qquad \Rightarrow x^* = \frac{\alpha^*}{2} = \max(b, 0)$$

α = 0 constraint is inactive, α > 0 constraint is active and tight

Dual SVM – linearly separable case

n training points, d features $(x_1, \ldots, x_n)$ where x_i is a d-dimensional vector

- Primal problem: $\text{minimize}_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}.\mathbf{w}$
  $\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j \geq 1, \ \forall j$

w – weights on features (d-dim problem)

- Dual problem (derivation):

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}.\mathbf{w} - \sum_j \alpha_j \left[\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j - 1\right]$$
$$\alpha_j \geq 0, \ \forall j$$

α – weights on training pts (n-dim problem)

Dual SVM – linearly separable case

- Dual problem:

$$\max_\alpha \min_{\mathbf{w},b} L(\mathbf{w},b,\alpha) = \frac{1}{2}\mathbf{w}.\mathbf{w} - \sum_j \alpha_j \left[\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j - 1\right]$$

$$\alpha_j \geq 0, \ \forall j$$

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \qquad \Rightarrow \mathbf{w} = \sum_j \alpha_j y_j \mathbf{x}_j$$

$$\frac{\partial L}{\partial b} = 0 \qquad \Rightarrow \sum_j \alpha_j y_j = 0$$

If we can solve for αs (dual problem), then we have a solution for **w**,b (primal problem)

Dual SVM – linearly separable case

$$\text{maximize}_{\alpha} \quad \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i . \mathbf{x}_j$$

$$\sum_i \alpha_i y_i = 0$$
$$\alpha_i \geq 0$$

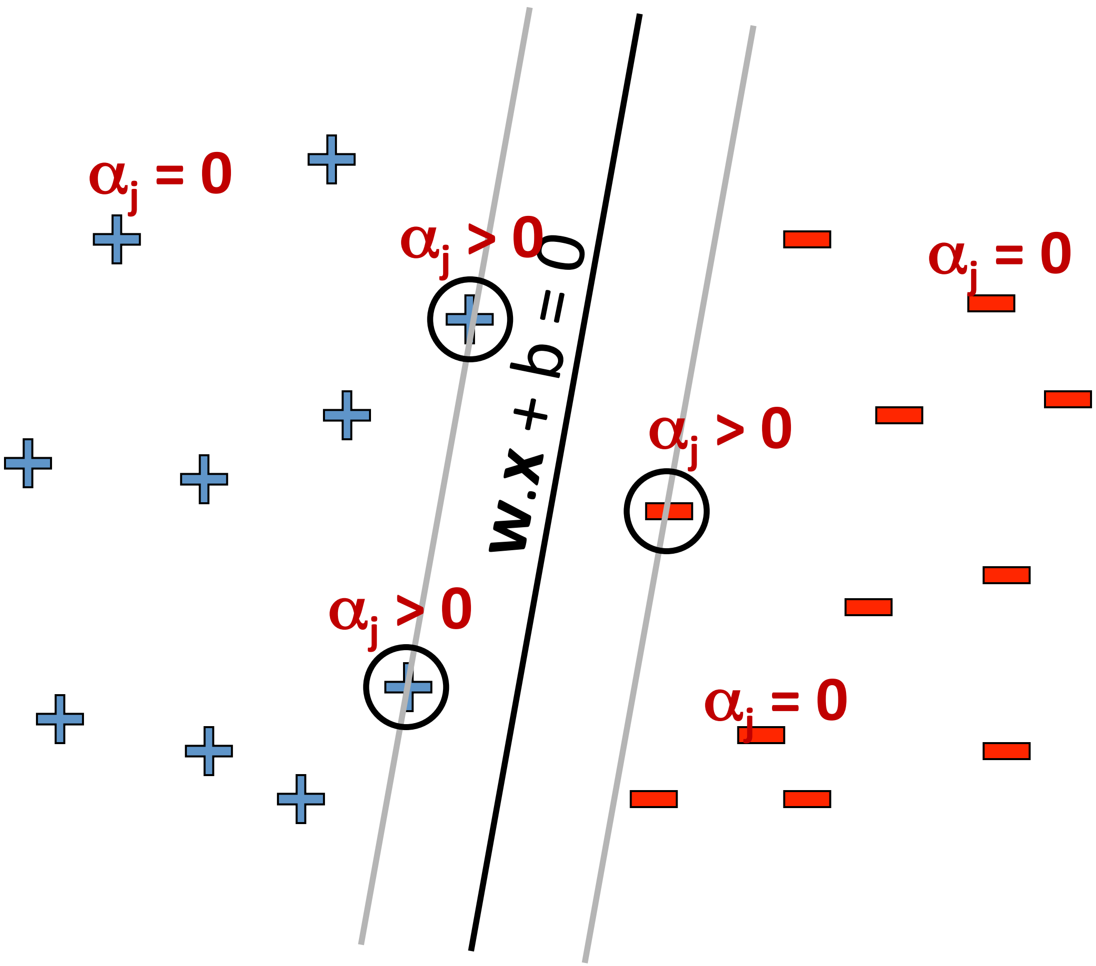
Dual problem is also QP

Solution gives α_js $\longrightarrow$

$$\boxed{\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i}$$

What about b?

Dual SVM: Sparsity of dual solution



$$\mathbf{w} = \sum_j \alpha_j y_j \mathbf{x}_j$$

Only few α_js can be non-zero : where constraint is active and tight

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

Support vectors – training points j whose α_js are non-zero

Dual SVM – linearly separable case

$$\text{maximize}_\alpha \quad \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \mathbf{x}_i.\mathbf{x}_j$$

$$\sum_i \alpha_i y_i = 0$$

$$\alpha_i \geq 0$$

Dual problem is also QP

Solution gives α_js $\longrightarrow$

$$\boxed{\begin{array}{l} \mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \\ b = y_k - \mathbf{w}.\mathbf{x}_k \\ \text{for any } k \text{ where } \alpha_k > 0 \end{array}}$$

Use support vectors to compute b since constraint is tight $(w.x_k + b)y_k = 1$

Dual SVM – non-separable case

- Primal problem:

$$\text{minimize}_{\mathbf{w},b} \quad \frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_j \xi_j$$
$$\left(\mathbf{w}.\mathbf{x}_j + b\right) y_j \geq 1 - \xi_j, \quad \forall j \qquad \boxed{\alpha_j}$$
$$\xi_j \geq 0, \quad \forall j \qquad \boxed{\mu_j}$$

Lagrange Multipliers

- Dual problem:

$$\max_{\alpha,\mu} \min_{\mathbf{w},b} L(\mathbf{w}, b, \alpha, \mu)$$
$$s.t. \alpha_j \geq 0 \quad \forall j$$
$$\mu_j \geq 0 \quad \forall j$$

Dual SVM – non-separable case

$$\text{maximize}_\alpha \quad \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j y_i y_j \mathbf{x}_i.\mathbf{x}_j$$

$$\sum_i \alpha_i y_i = 0$$

$$\boxed{C \geq} \alpha_i \geq 0$$

comes from $\frac{\partial L}{\partial \mu} = 0$

Intuition:

Earlier - If constraint violated, $\alpha_i \rightarrow \infty$

Now - If constraint violated, $\alpha_i \leq C$

Dual problem is also QP

Solution gives α_js $\longrightarrow$

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

$$b = y_k - \mathbf{w}.\mathbf{x}_k$$

for any k where $C > \alpha_k > 0$

What you need to know

- Maximizing margin
- Derivation of SVM formulation
- Slack variables and hinge loss
- Relationship between SVMs and logistic regression
  - 0/1 loss
  - Hinge loss
  - Log loss
- Tackling multiple class
  - One against All
  - Multiclass SVMs
- Dual SVM formulation (revisit again)
  - Easier to solve when dimension high $d > n$