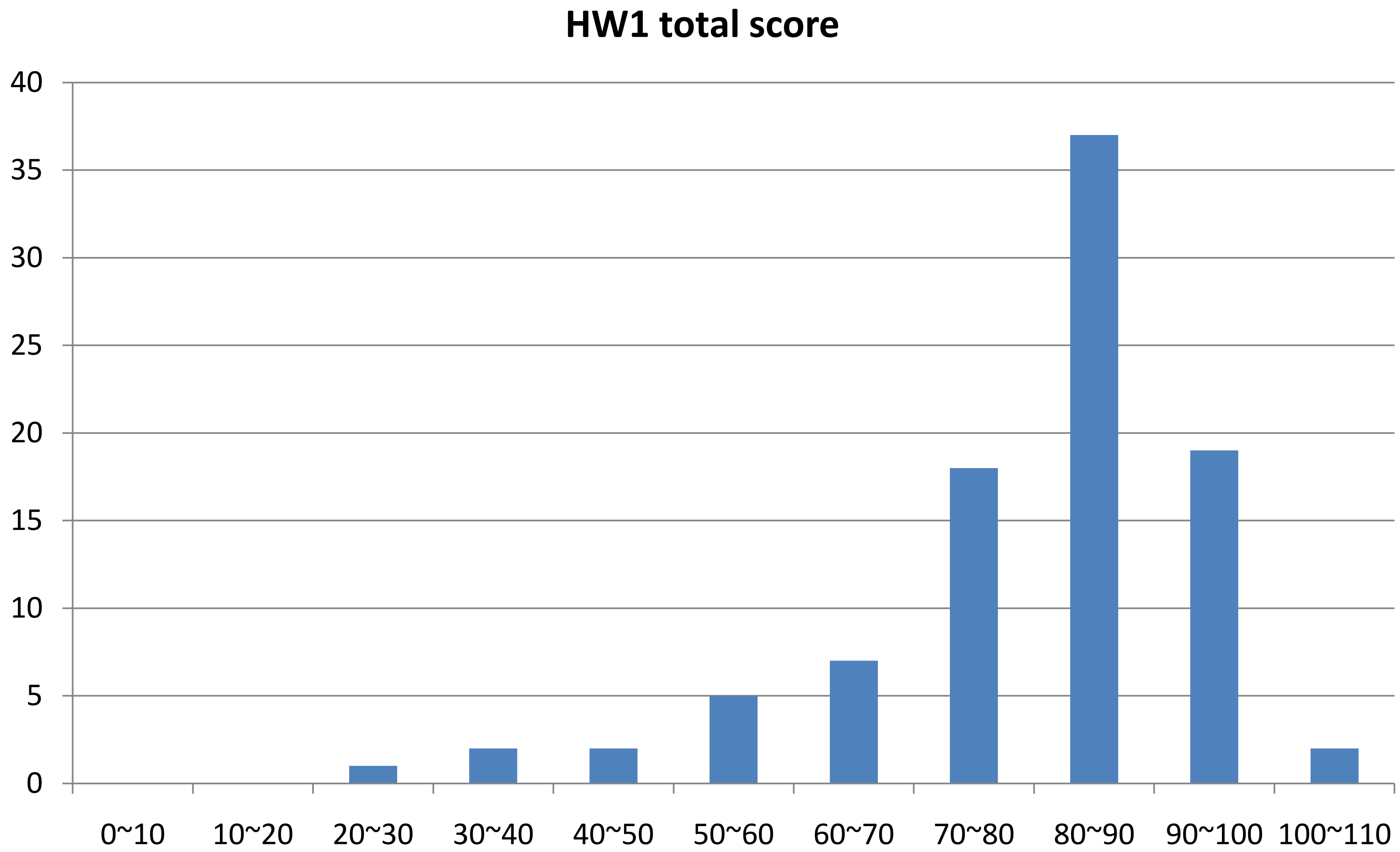
Announcements - Homework

- Homework 1 is graded, please collect at end of lecture
- Homework 2 due today
- Homework 3 out soon (watch email)
  - Ques 1 – midterm review

HW1 score distribution



Announcements - Midterm

- When: Wednesday, 10/20
- Where: In Class
- What: You, your pencil, your textbook, your notes, course slides, your calculator, your good mood :)
- What NOT: No computers, iphones, or anything else that has an internet connection.
- Material: Everything from the beginning of the semester, until, and including SVMs and the Kernel trick

Recitation Tomorrow!

- Boosting, SVM (convex optimization), **Midterm review**!
- Strongly recommended!!
- Place: NSH 3305 (**Note: change from last time**)
- Time: 5-6 pm

Rob

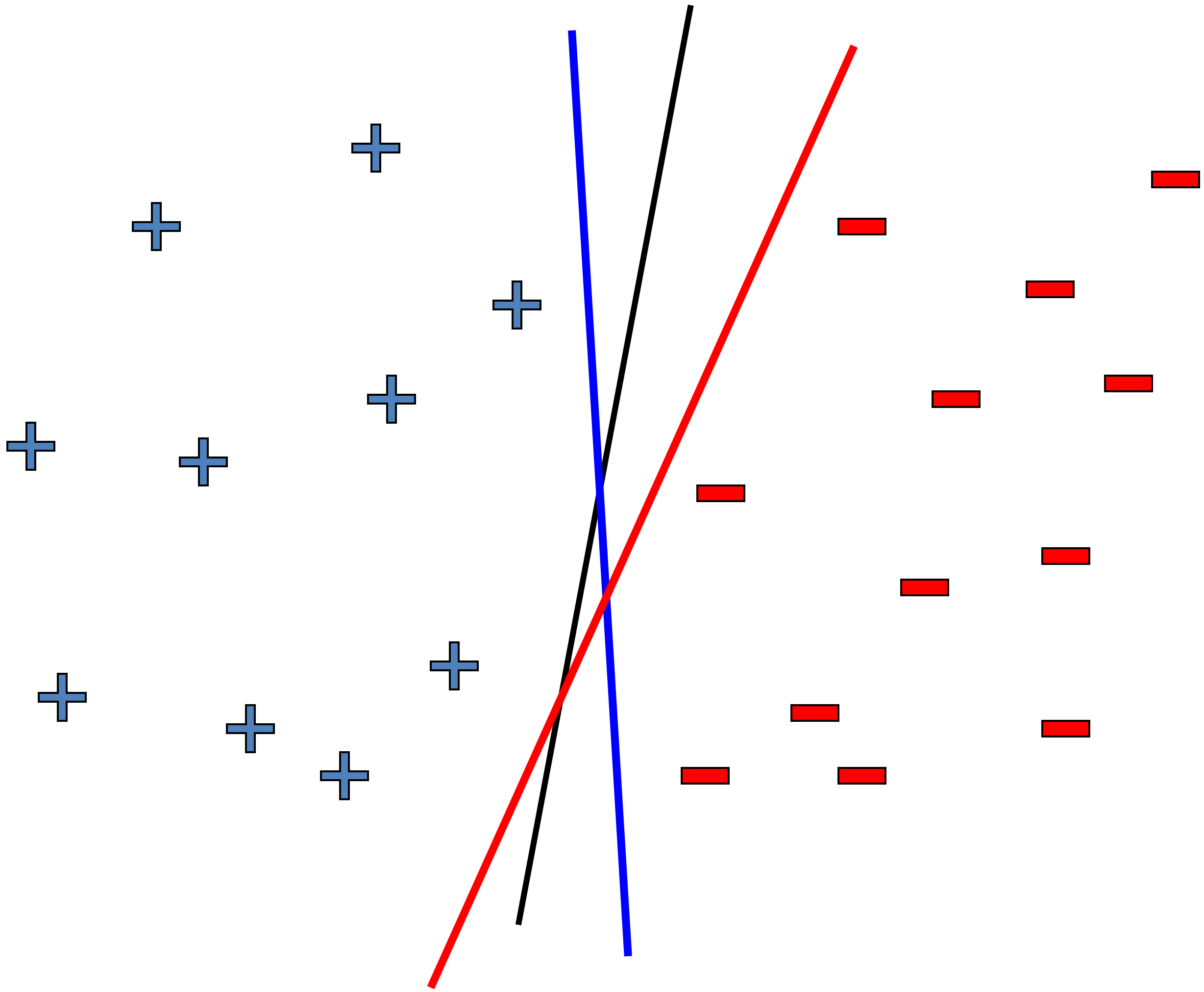
Support Vector Machines

Aarti Singh

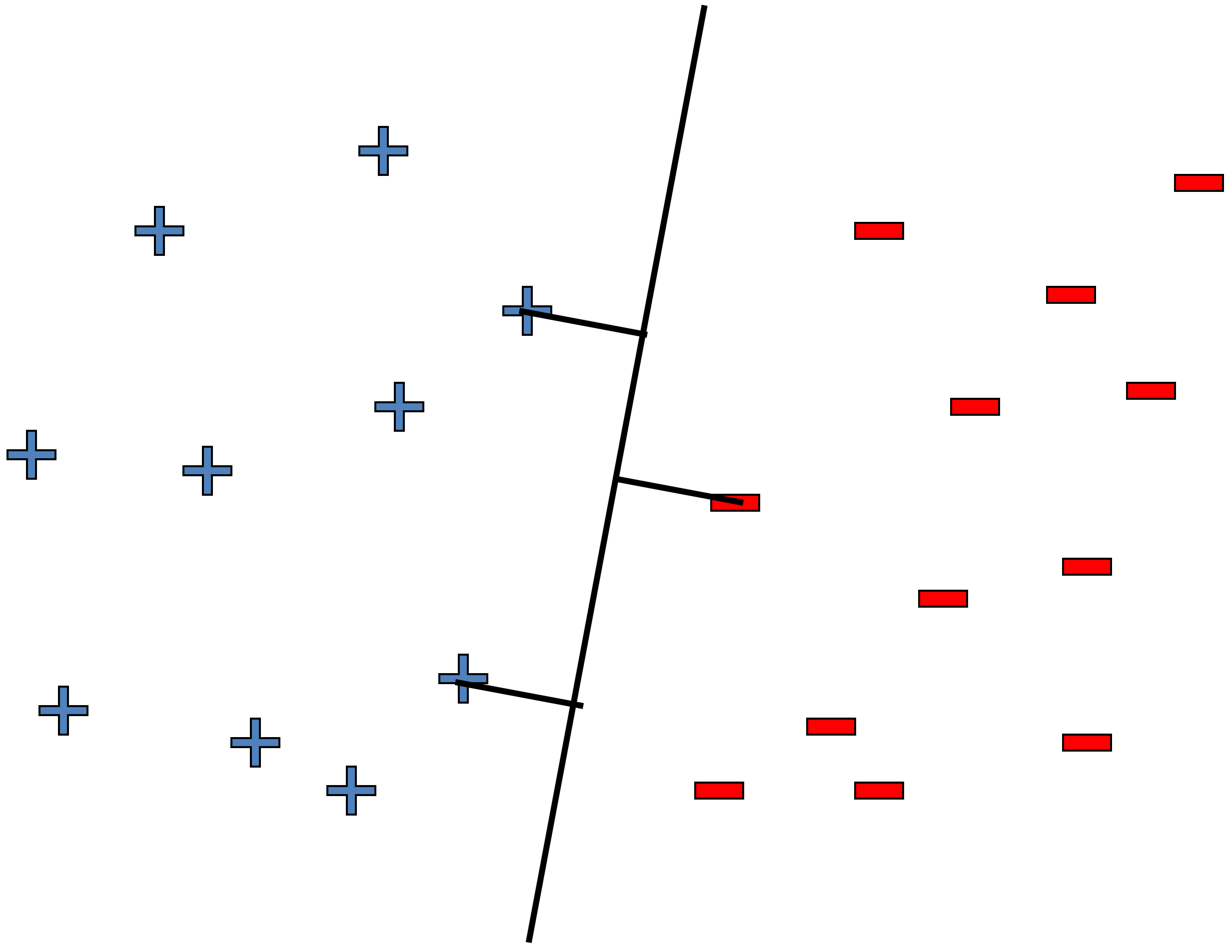
Machine Learning 10-701/15-781
Oct 13, 2010

At Pittsburgh G-20 summit …

Linear classifiers – which line is better?



Pick the one with the largest margin!

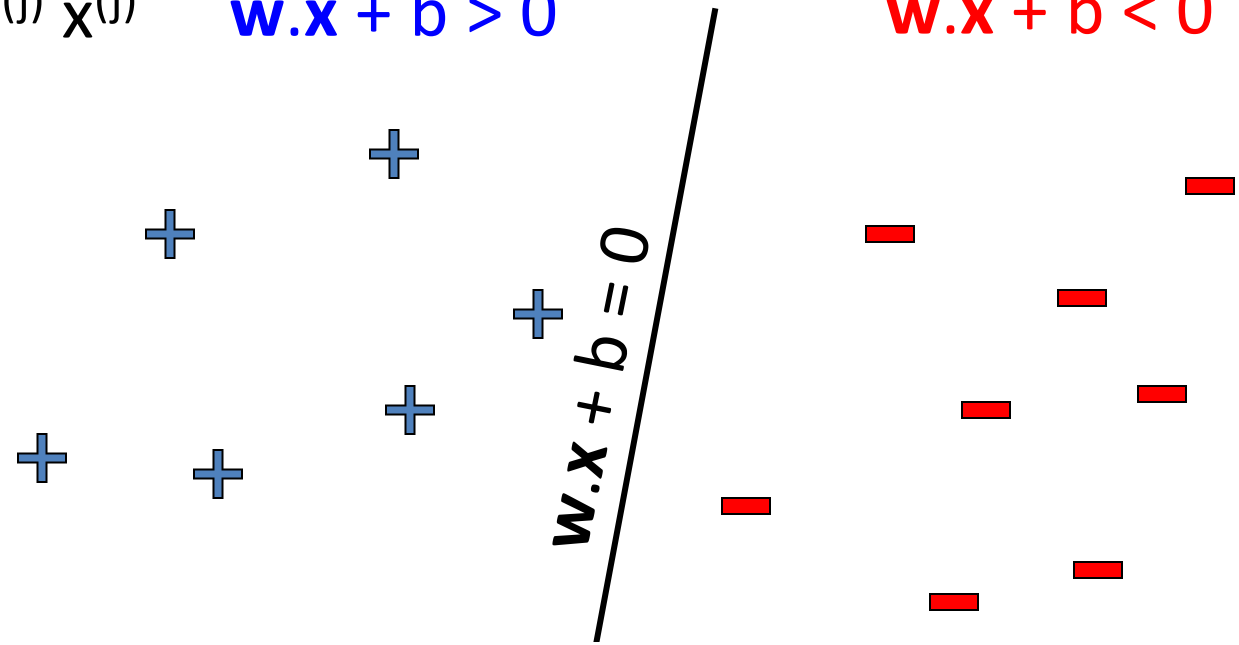


Parameterizing the decision boundary

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

$\mathbf{w}.\mathbf{x} + b > 0$

$\mathbf{w}.\mathbf{x} + b < 0$



Example i (= 1,2,…,n):

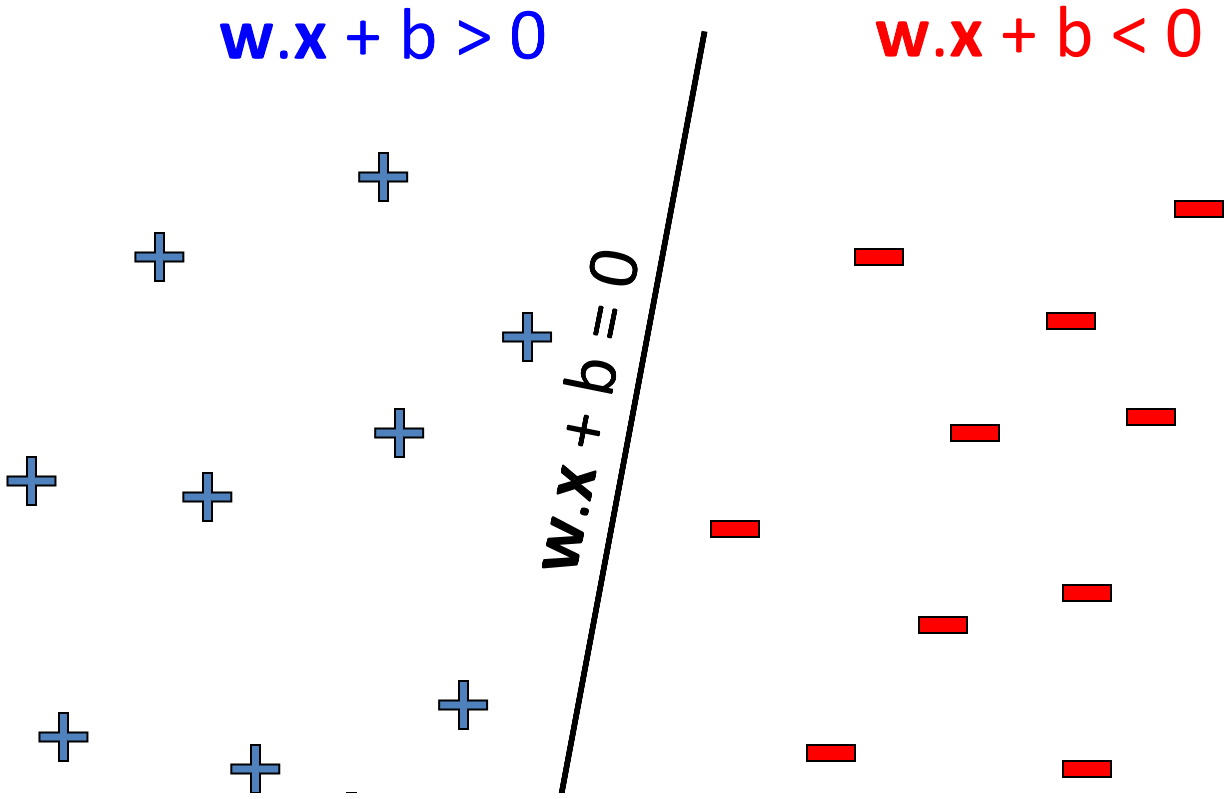
$$\left\langle x_i^{(1)}, \ldots, x_i^{(m)} \right\rangle \text{ — } m \text{ features}$$

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

Data:

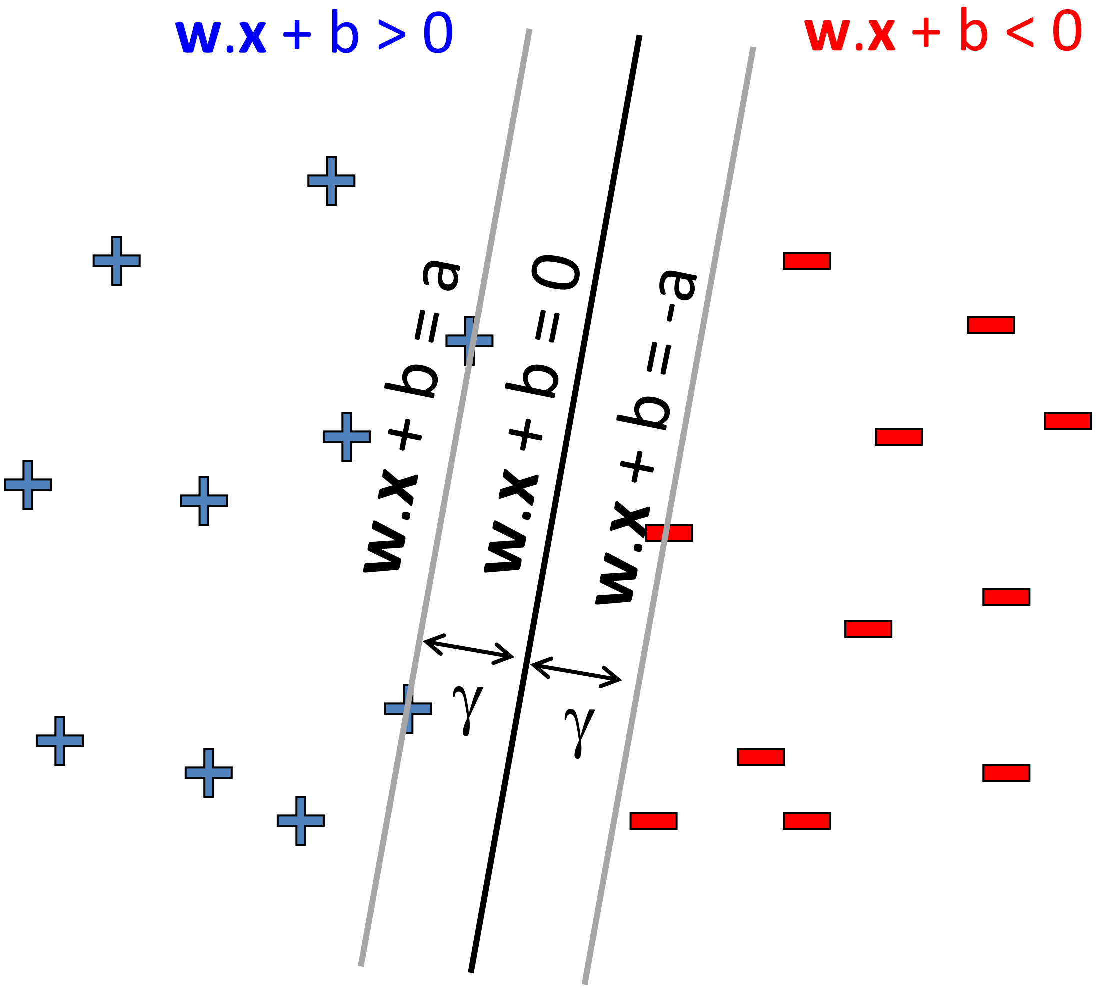
$$\left\langle x_1^{(1)}, \ldots, x_1^{(m)}, y_1 \right\rangle$$
$$\vdots$$
$$\left\langle x_n^{(1)}, \ldots, x_n^{(m)}, y_n \right\rangle$$

Parameterizing the decision boundary



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

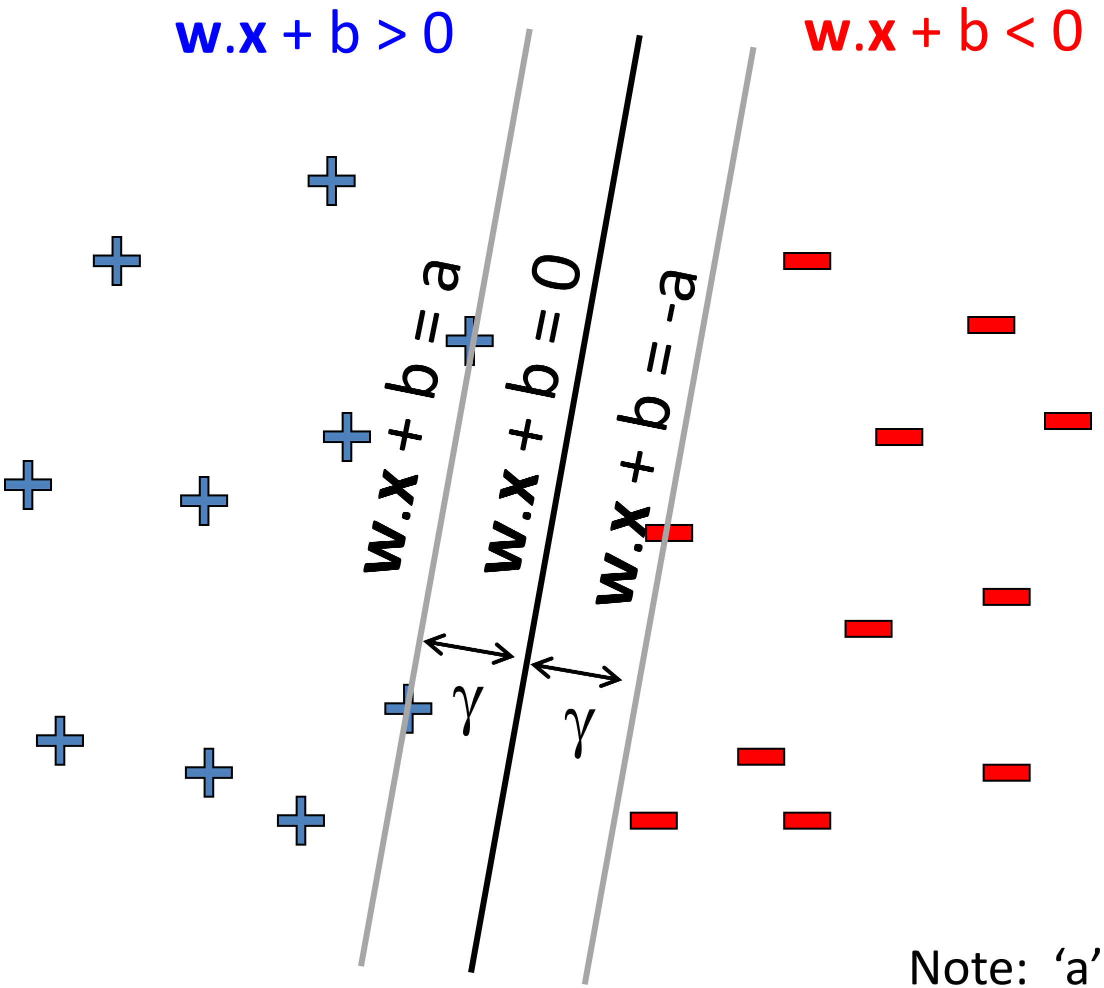
Maximizing the margin



Distance of closest examples from the line/hyperplane

margin = γ = 2a/‖w‖

Maximizing the margin



Distance of closest examples from the line/hyperplane

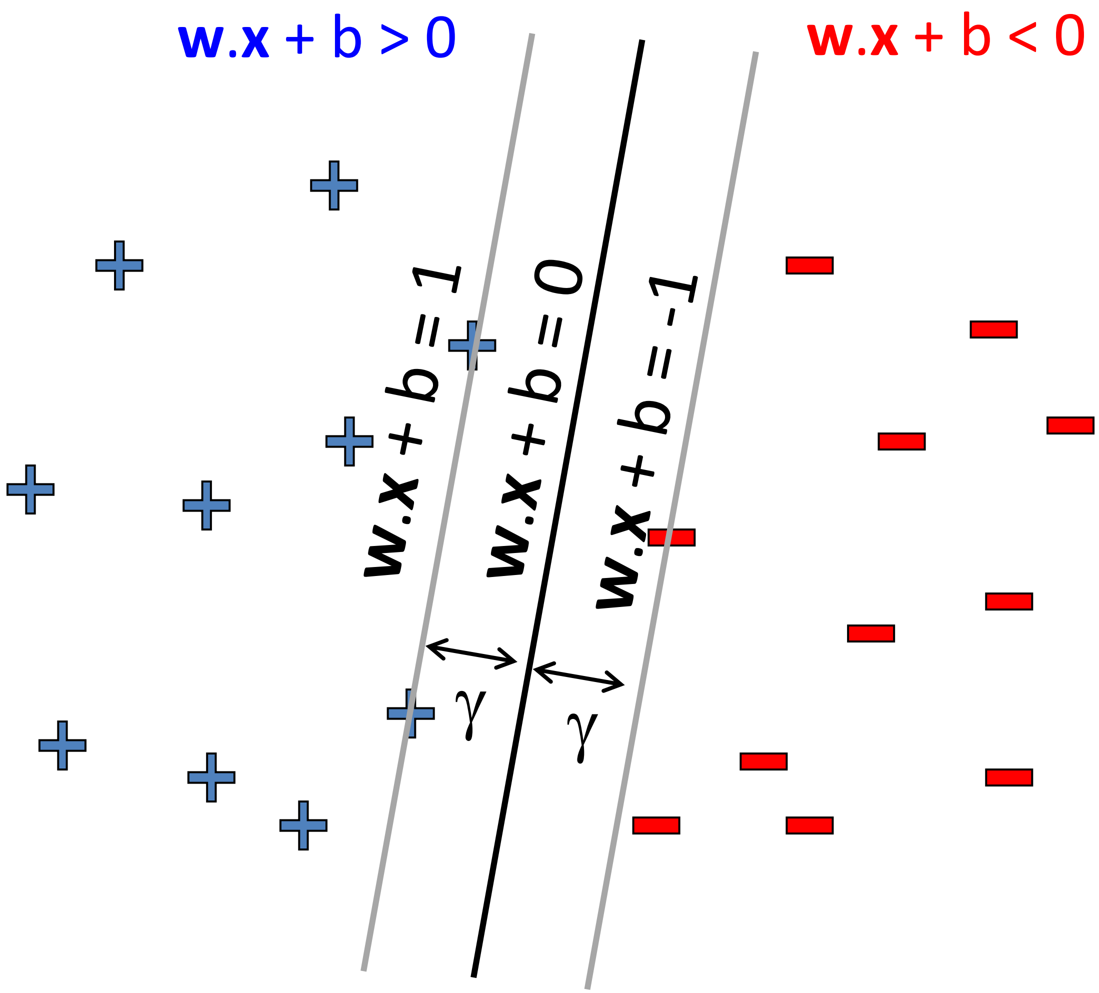
$$\text{margin} = \gamma = 2a/\|w\|$$

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

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

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

Support Vector Machines



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

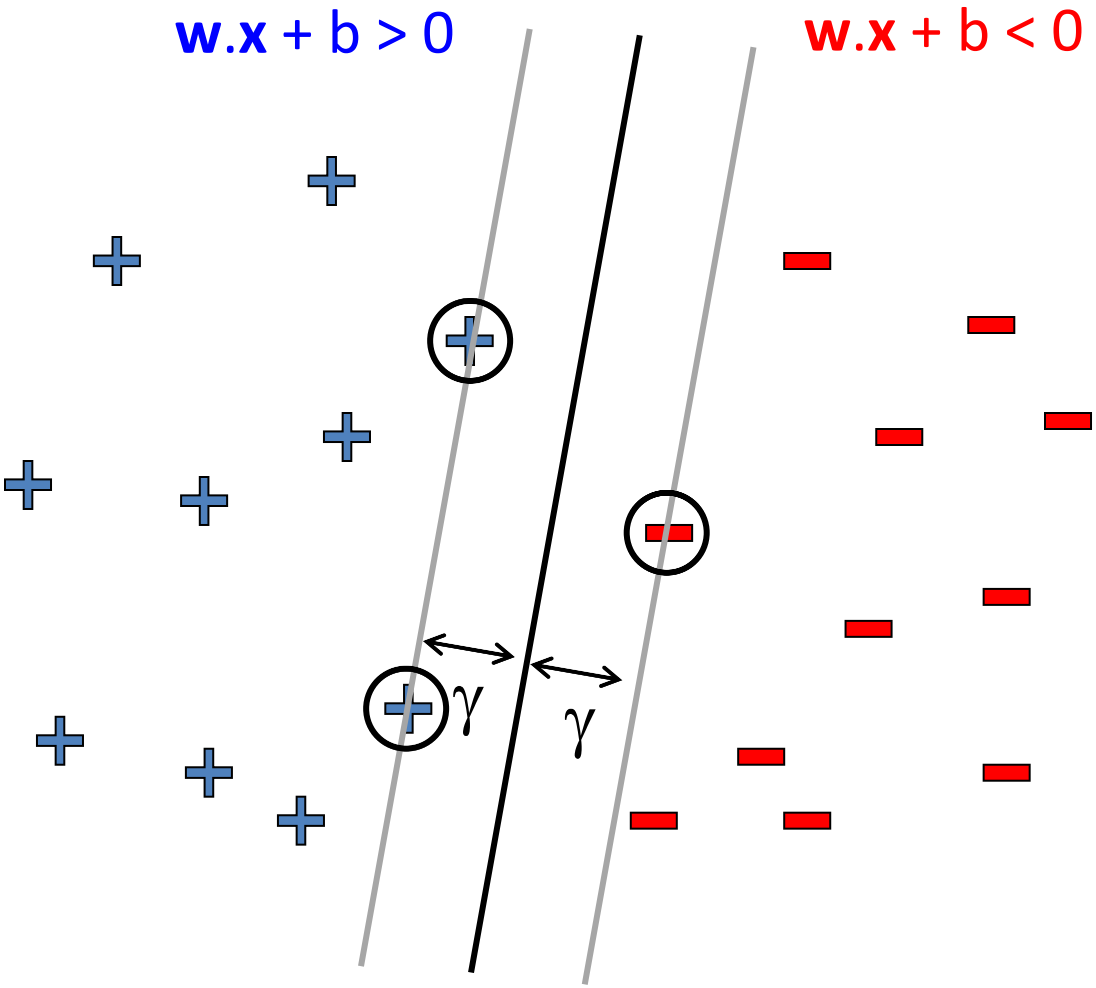
$$\text{s.t. } (\mathbf{w}.\mathbf{x}_j + b)\, y_j \geq 1 \quad \forall j$$

Solve efficiently by quadratic programming (QP)

- Well-studied solution algorithms

Linear hyperplane defined by "<u>support vectors</u>"

Support Vectors



Linear hyperplane defined by "<u>support vectors</u>"

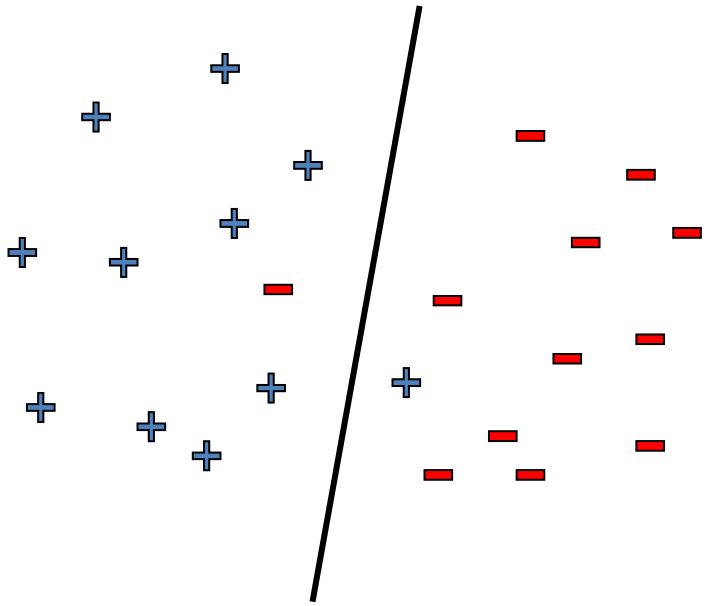
Moving other points a little doesn't effect the decision boundary

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

How many support vectors in linearly separable case?

$\leq$ m+1

What if data is not linearly separable?



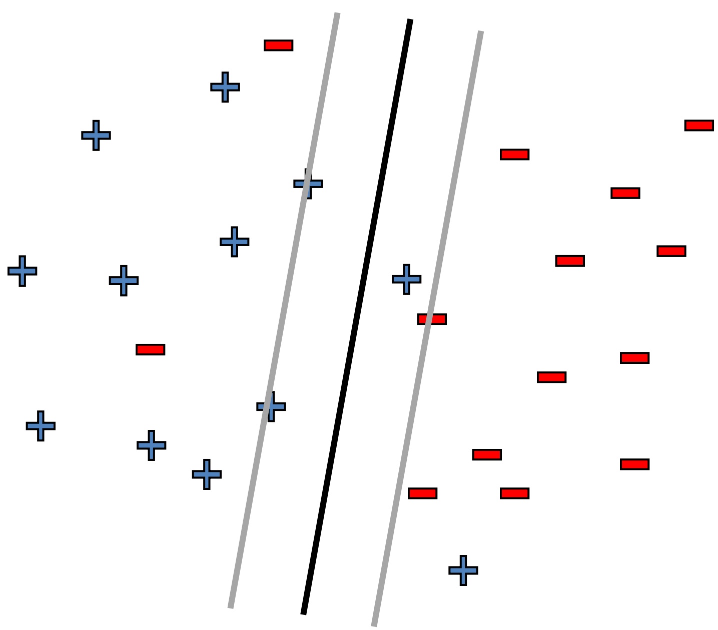
Use features of features of features of features....

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

But run risk of overfitting!

What if data is still not linearly separable?

Allow "error" in classification



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

Maximize margin and minimize # mistakes on training data

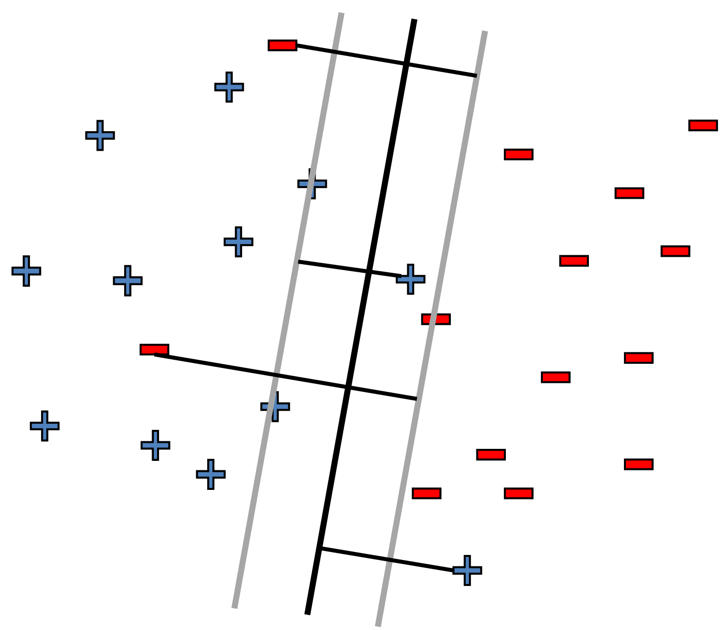
C - tradeoff parameter

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

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

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

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

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

C - tradeoff parameter (chosen by cross-validation)

Still QP ☺

Slack variables – Hinge loss

Complexity penalization

$$\xi_j = \text{loss}(f(x_j), y_j)$$

$$f(x_j) = \text{sgn}(\mathbf{w} \cdot \text{x}_\text{j} + \text{b})$$

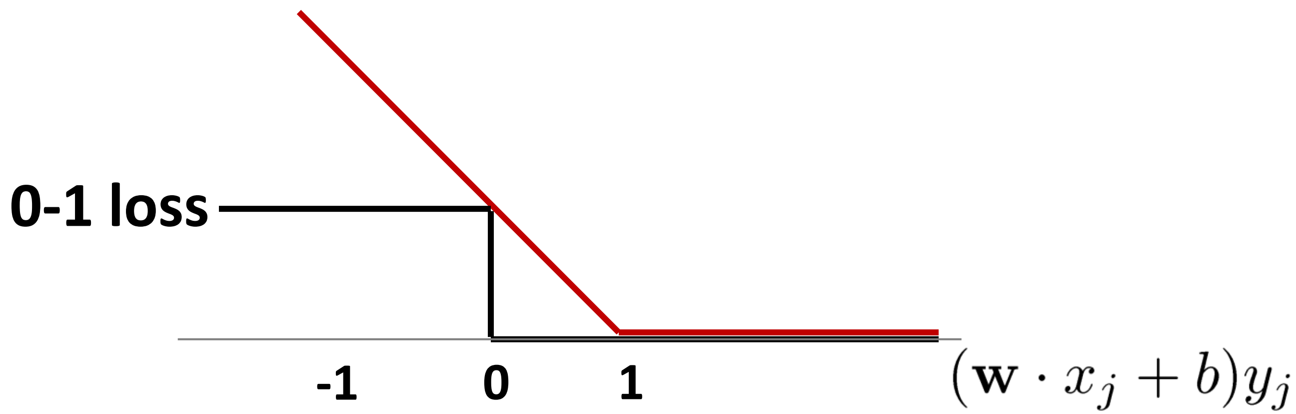
$$\min_{\mathbf{w},b} \ \mathbf{w}.\mathbf{w} + C \sum_j \xi_j$$

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

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

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

Hinge loss



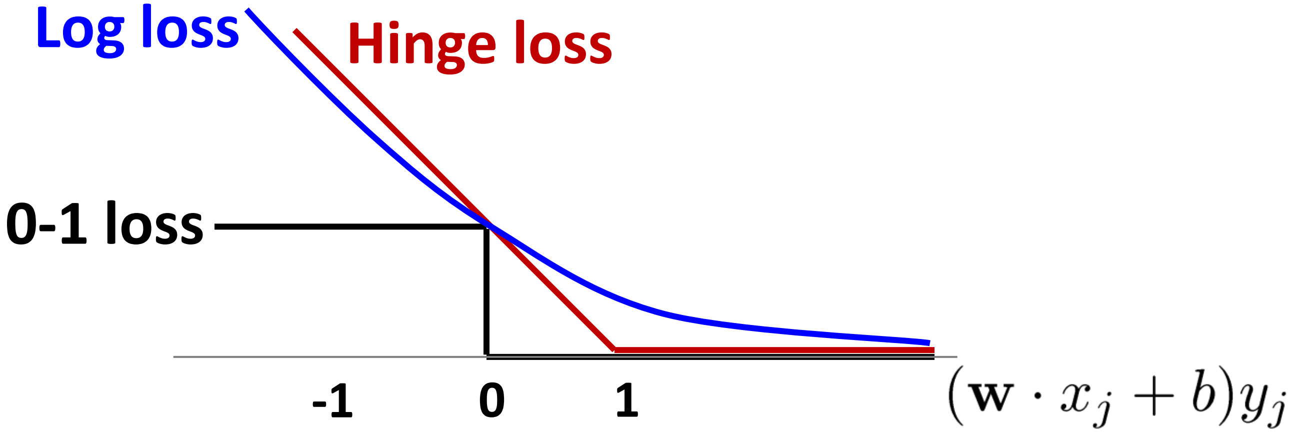
SVM vs. Logistic Regression

SVM : **Hinge loss**

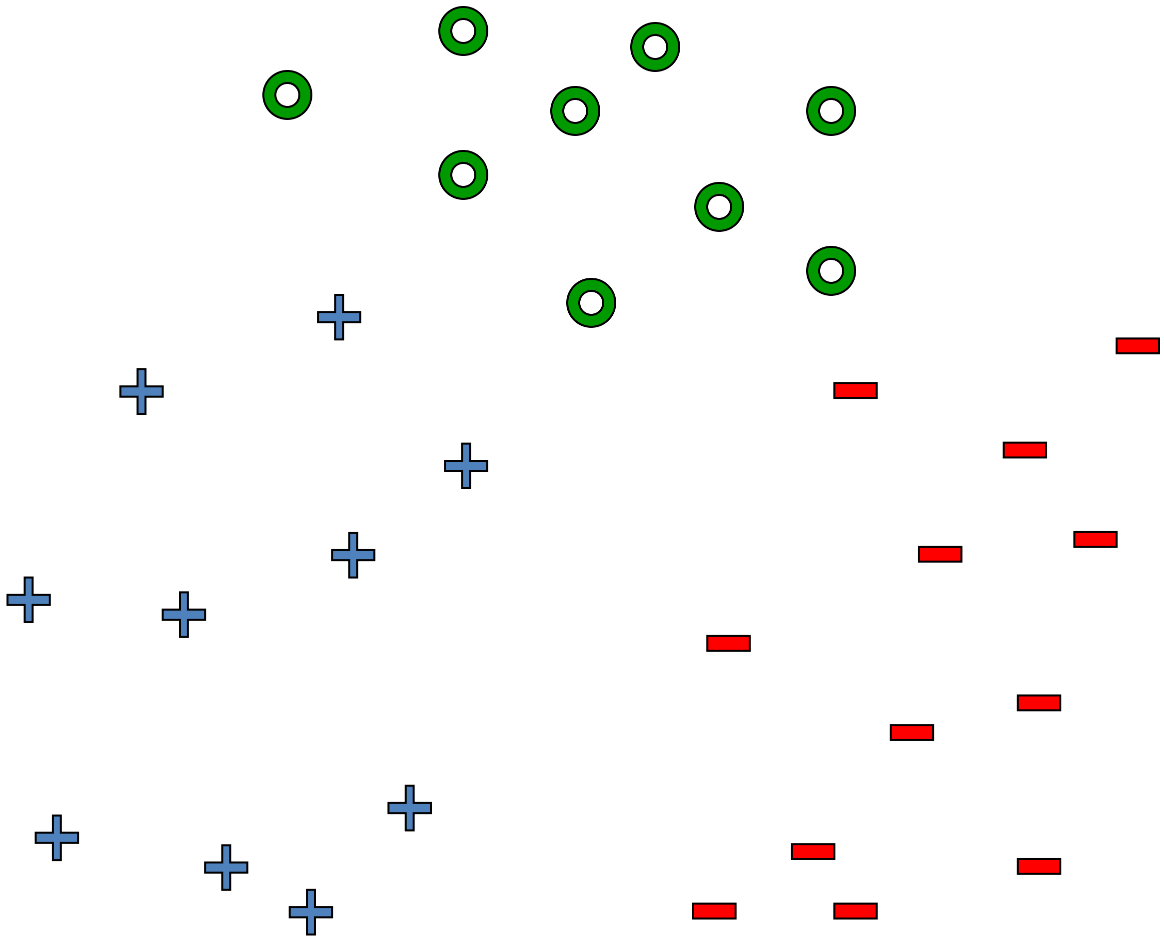
$$\text{loss}(f(x_j), y_j) = (1 - (\mathbf{w} \cdot x_j + b)y_j))_+$$

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

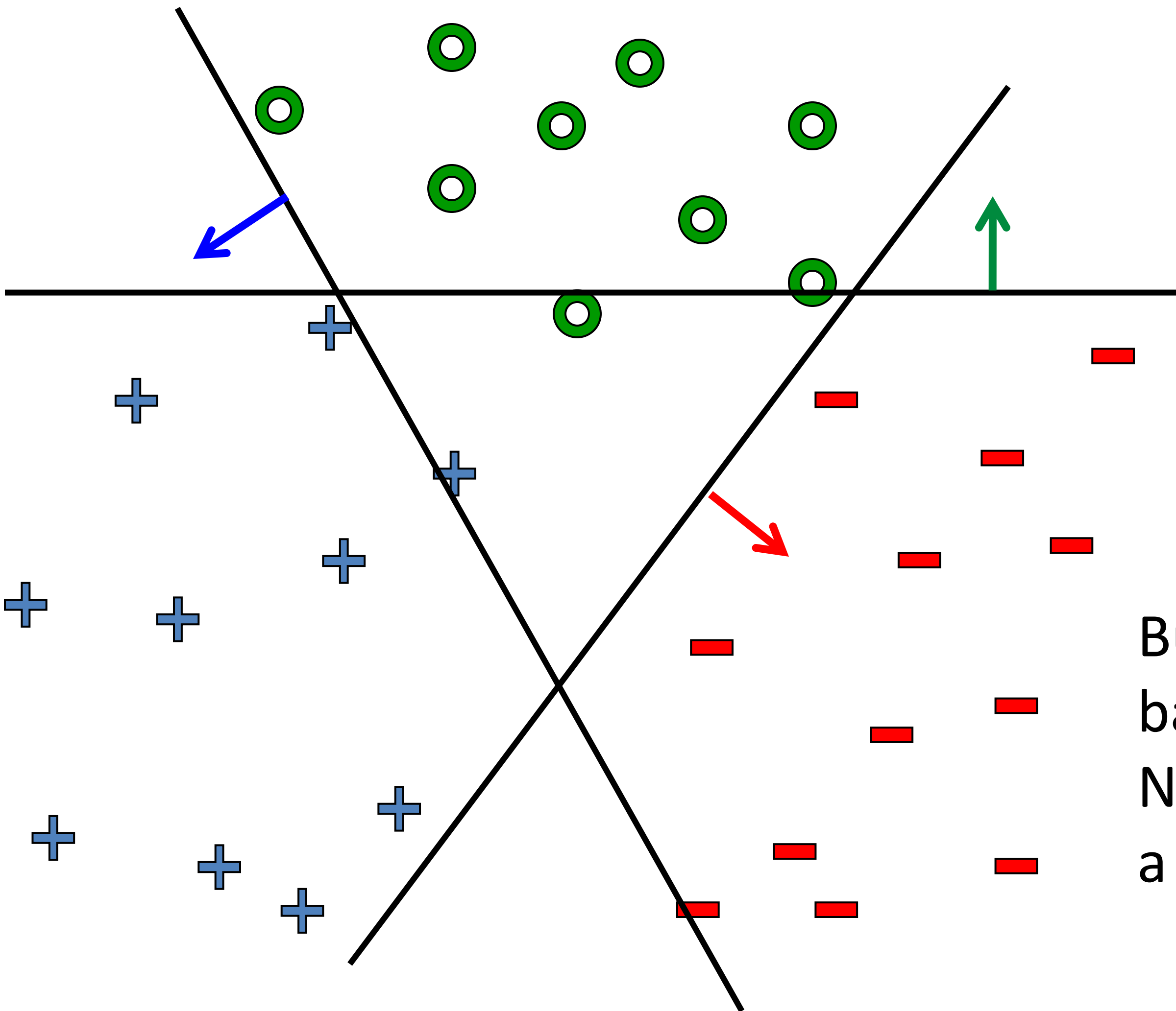
$$\text{loss}(f(x_j), y_j) = -\log P(y_j \mid x_j, \mathbf{w}, b) = \log(1 + e^{-(\mathbf{w} \cdot x_j + b)y_j})$$



What about multiple classes?



One against all



Learn 3 classifiers separately:

Class k vs. rest

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

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

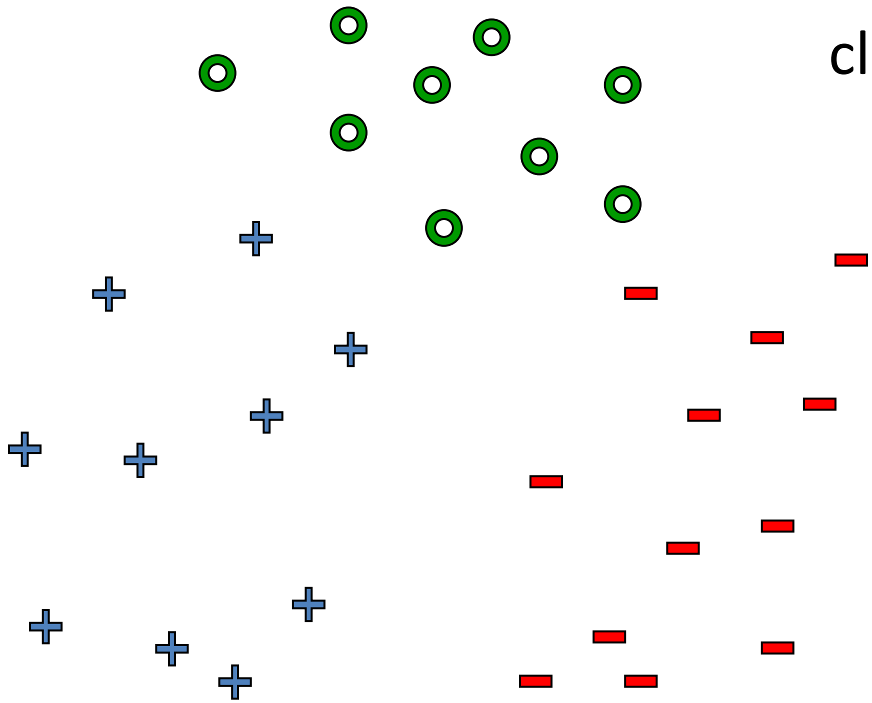
But $\mathbf{w}_k$s may not be based on the same scale.

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 $\mathbf{w}^{(k)}.x + b^{(k)}$

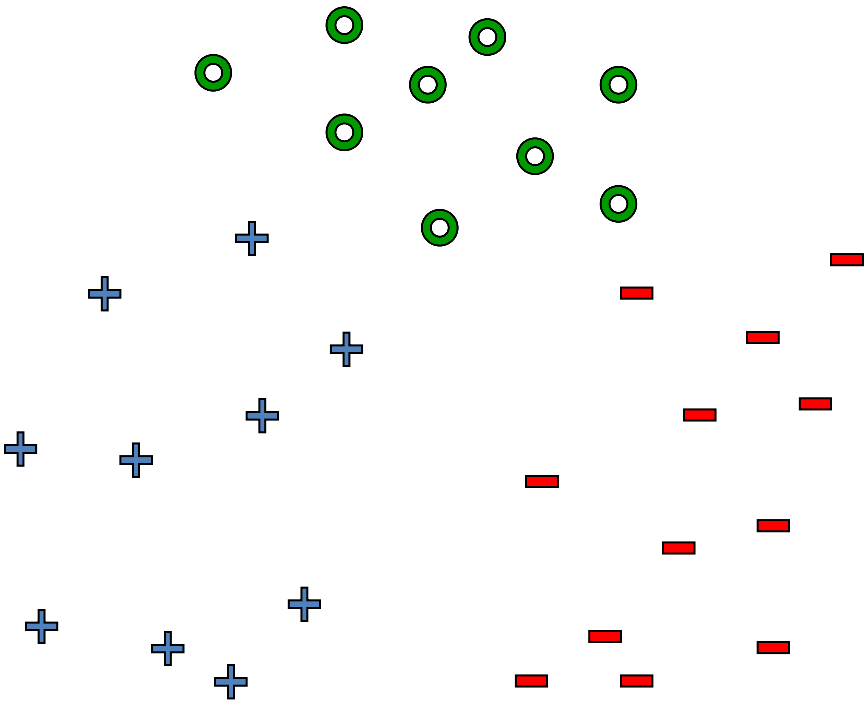
Learn 1 classifier: Multi-class SVM

Simultaneously learn 3 sets of weights

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

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

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



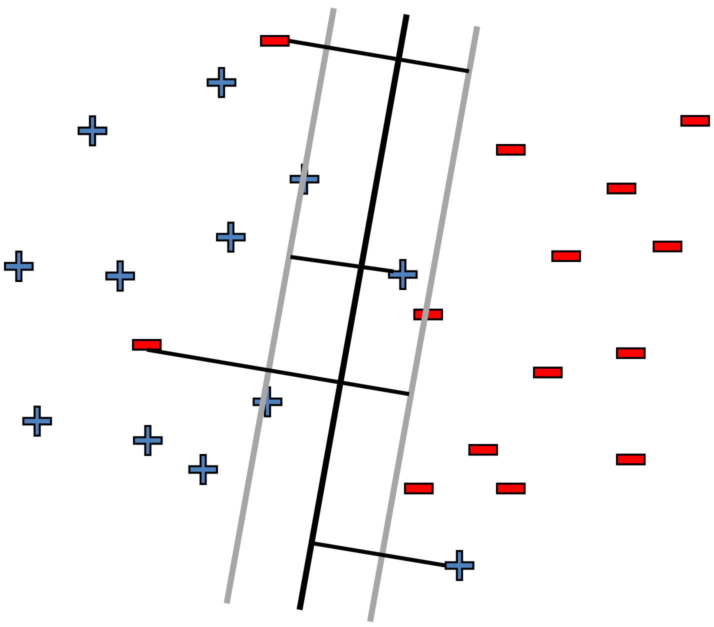
y = arg max $\mathbf{w}^{(k)}.x + b^{(k)}$

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

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

SVMs reminder



Soft margin approach

Regularization   Hinge loss

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

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

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

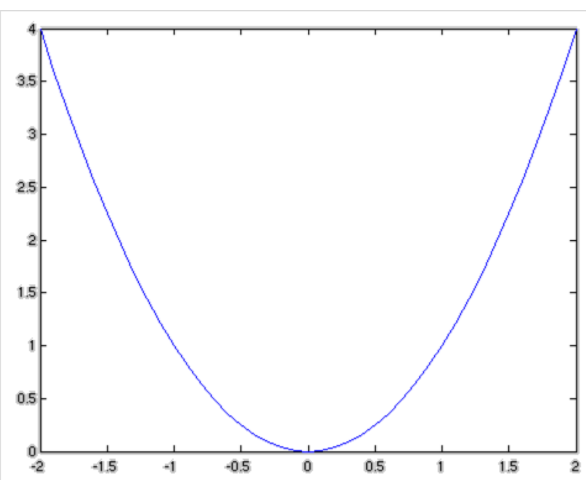
Today's Lecture

- Learn one of the most interesting and exciting recent advancements in machine learning
  - The "kernel trick"
  - High dimensional feature spaces at no extra cost!
- But first, a detour
  - Constrained optimization!

Constrained Optimization

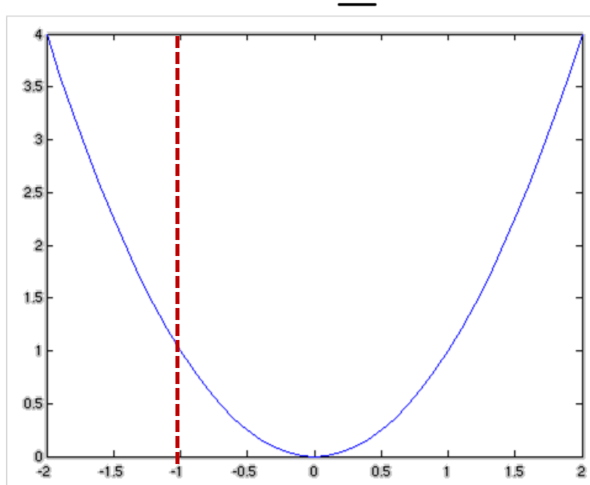
$$\min_x \; x^2$$
$$\text{s.t.} \quad x \geq b$$

$$\min_x \; x^2$$



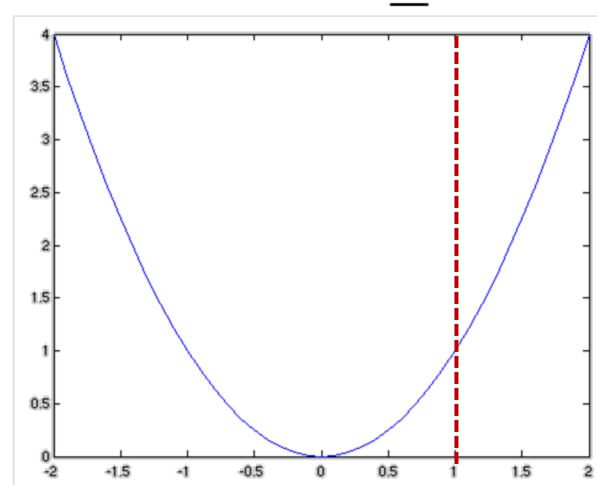
$$x^* = 0$$

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



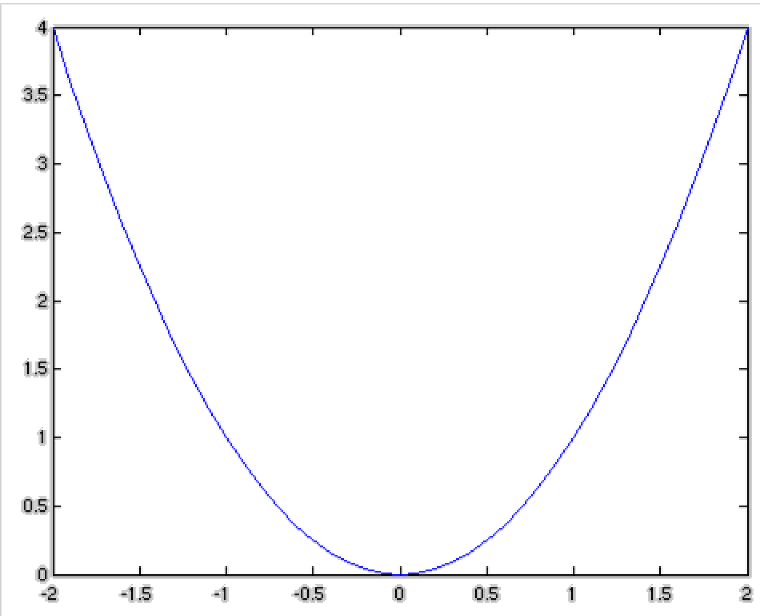
$$x^* = 0$$

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



$$x^* = 1$$

Lagrange Multiplier – Dual Variables



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

Moving the constraint to objective function

Lagrangian:

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

Solve:

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

Constraint is tight when $\alpha > 0$

Duality

Primal problem:

$$f^* = \min_x \underbrace{x^2}_{f(x)} \quad \text{s.t.} \quad x \geq b$$

Dual problem:

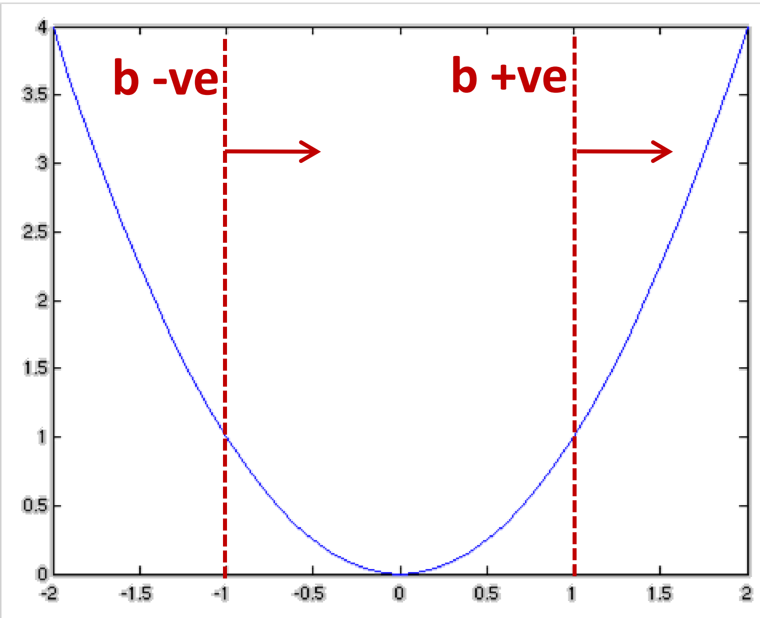
$$g^* = \min_x \underbrace{\max_\alpha \; x^2 - \alpha(x-b)}_{g(x)} \quad \text{s.t.} \quad \alpha \geq 0$$

Weak duality – $g^* \leq f^*$

For all feasible points $\tilde{x}$ $\quad g^* \leq g(\tilde{x}) \leq f(\tilde{x})$

Strong duality – $g^* = f^*$ (holds under KKT conditions)

Lagrange Multiplier – Dual Variables



$x^* = 0 \quad x^* = b$

Solving:

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

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

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

When $\alpha > 0$, constraint is tight