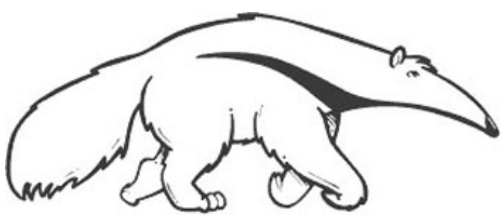
Machine Learning and Data Mining

Support Vector Machines

Prof. Alexander Ihler

Fall 2012



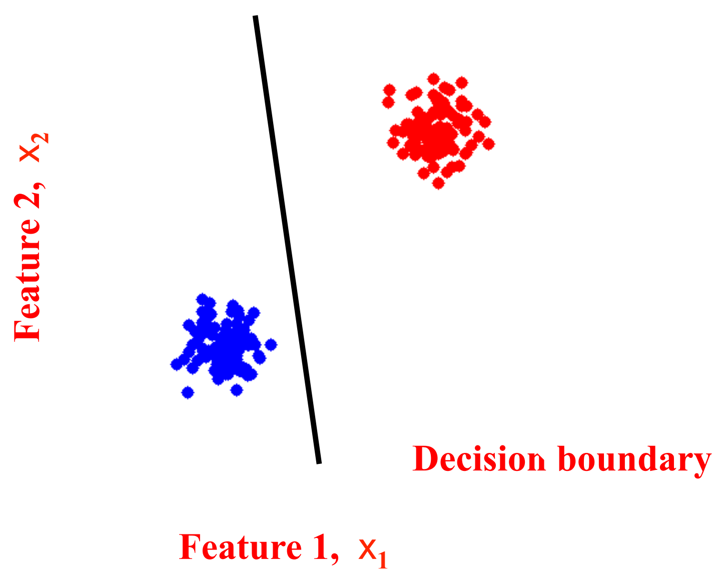
BREN:ICS
INFORMATION AND COMPUTER SCIENCES

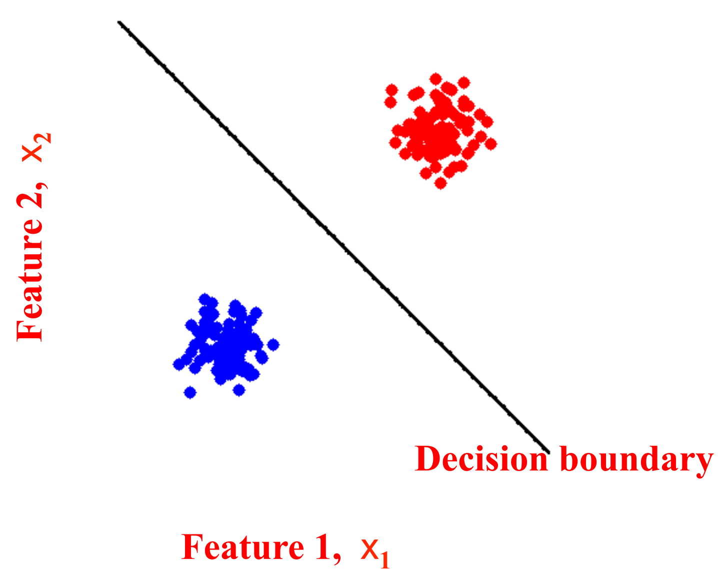
UNIVERSITY of CALIFORNIA IRVINE

Linear Classifiers

- Which decision boundary is "better"?
  - Both have zero training error (perfect training accuracy)
  - But, one of them seems intuitively better…

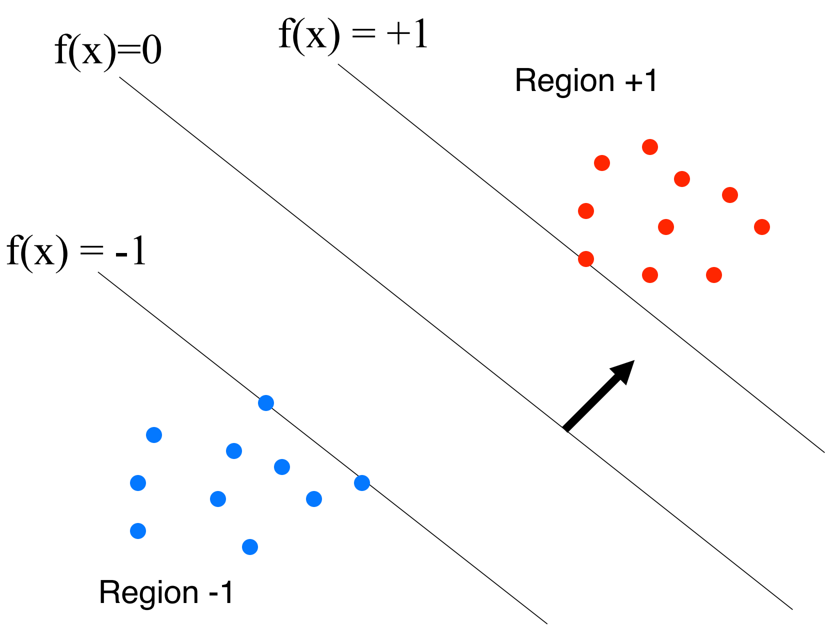
- How can we quantify "better",
  and learn the "best" parameter settings?





One possible answer…

- Maybe we want to maximize our “margin”
- Define class +1 in some region, class –1 in another
- Make those regions as far apart as possible



We could define such a function:

$f(x) = w*x' + b$

$f(x) > +1$ in region +1
$f(x) < -1$ in region –1

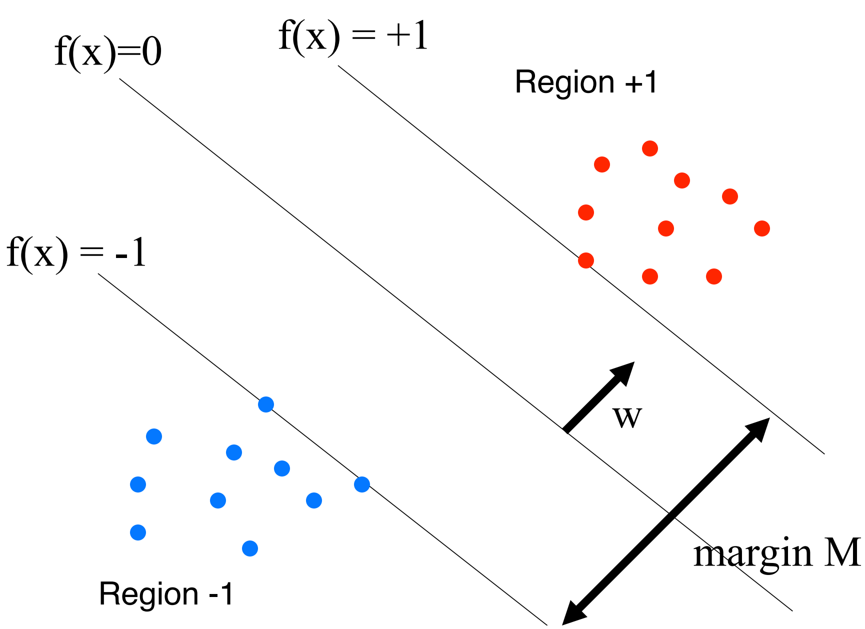
Passes through zero in center…

“Support vectors” – data points on margin

Computing the margin width

- Vector "w" is perpendicular to the boundaries (why?)
- Choose x_0 st $f(x_0) = -1$; let x_1 be the closest point with $f(x_1) = +1$
  - $x_1 = x_0 + r * w$ (why?)
- Closest two points on the margin also satisfy

$$w^T x_0 + b = -1 \qquad\qquad w^T x_1 + b = +1$$



$$\begin{aligned} & w^T(x_0 + rw) + b = +1 \\ \Rightarrow\ & r\|w\|^2 + w^T x_0 + b = +1 \\ \Rightarrow\ & r\|w\|^2 - 1 = +1 \\ \Rightarrow\ & r = \frac{2}{\|w\|^2} \end{aligned}$$

$$\begin{aligned} M &= \|x_1 - x_0\| = \|rw\| \\ &= \frac{2}{\|w\|^2}\|w\| = \frac{2}{\sqrt{w^T w}} \end{aligned}$$

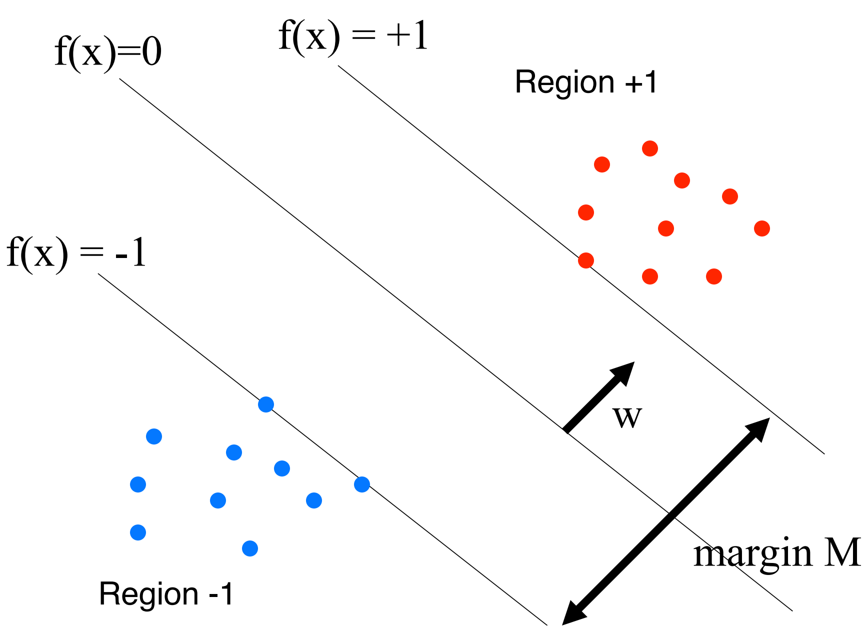
Maximum margin classifier

- Constrained optimization
  - Get all data points correct
  - Maximize the margin

$$w^* = \arg\max_w \frac{2}{\sqrt{w^T w}}$$

such that "all data on the correct side of the margin"

This is an example of a quadratic program: quadratic cost function, linear constraints



$$w^* = \arg\min_w \sum_j w_j^2$$

s.t.

$$y^{(i)} = +1 \Rightarrow \quad w^T x^{(i)} + b \geq +1$$

$$y^{(i)} = -1 \Rightarrow \quad w^T x^{(i)} + b \leq -1$$

(N constraints)

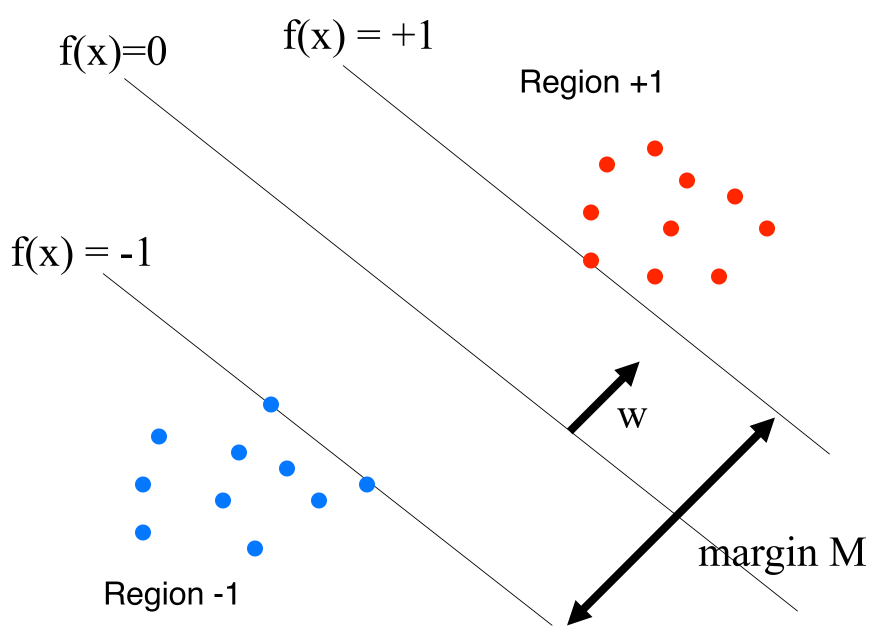
Maximum margin classifier

- Constrained optimization
  - Get all data points correct
  - Maximize the margin

$$w^* = \arg\max_w \frac{2}{\sqrt{w^T w}}$$

such that "all data on the correct side of the margin"

This is an example of a quadratic program: quadratic cost function, linear constraints



$$w^* = \arg\min_w \sum_j w_j^2$$

s.t.

$$y^{(i)}(\, w^T x^{(i)} + b\,) \geq +1$$

(N constraints)

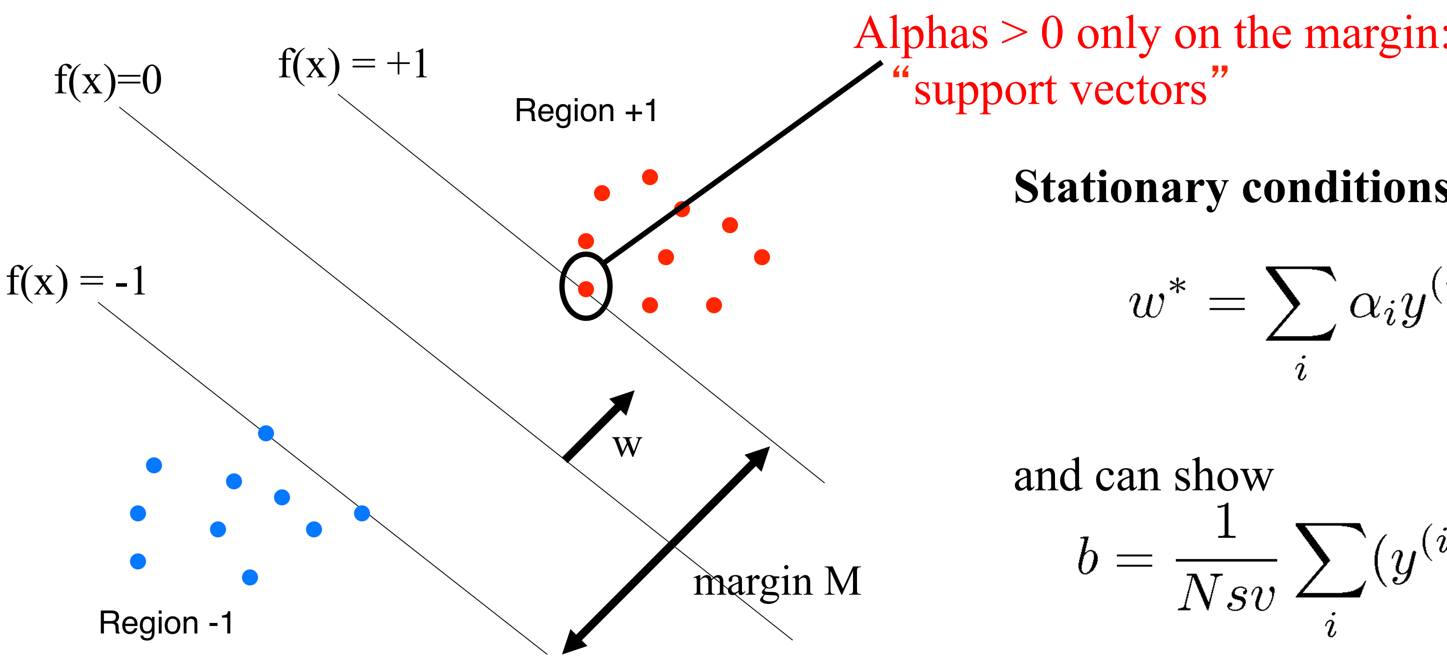
Dual form

- Use Lagrange multipliers
  - Enforce inequality constraints

$$w^* = \arg\min_w \sum_j w_j^2$$

$$s.t. \quad y^{(i)}(\, w^T x^{(i)} + b\,) \geq +1$$

$$w^* = \arg\min_w \max_{\alpha \geq 0} \frac{1}{2} \sum_j w_j^2 + \sum_i \alpha_i (\,1 - y^{(i)}(\,w^T x^{(i)} + b\,)\,)$$



Stationary conditions wrt w:

$$w^* = \sum_i \alpha_i y^{(i)} x^{(i)}$$

and can show

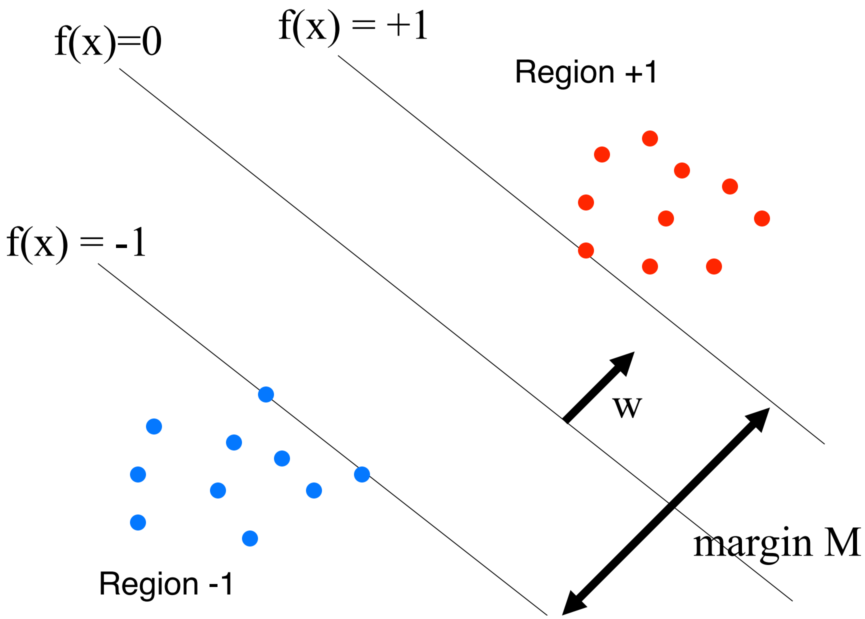
$$b = \frac{1}{Nsv} \sum_i (y^{(i)} - w^T x^{(i)})$$

Dual form

- Use Lagrange multipliers
  - Enforce inequality constraints
  - Write solely in terms of alphas:

$$\max_{\alpha \geq 0} \sum_i \alpha_i - \frac{1}{2} \sum_j \alpha_i \alpha_j y^{(i)} y^{(j)} {x^{(i)}}^T x^{(j)}$$

$$\text{s.t.} \ \sum_i \alpha_i y^{(i)} = 0$$

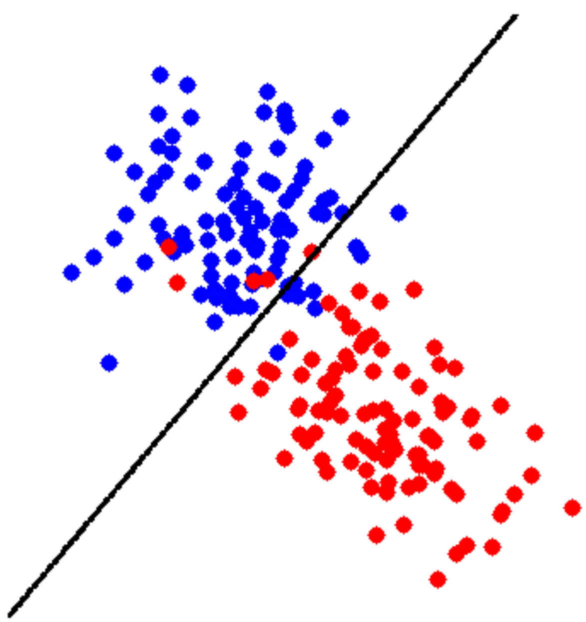


$$w^* = \sum_i \alpha_i y^{(i)} x^{(i)}$$

$$b = \frac{1}{Nsv} \sum_i (y^{(i)} - w^T x^{(i)})$$

Maximum margin classifier

- What if the data are not linearly separable?



$$\text{Margin:} \quad \min_w \sum_j w_j^2$$

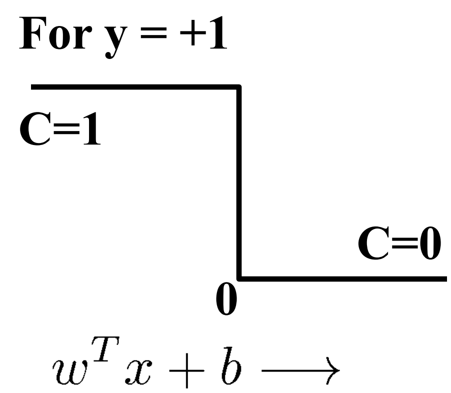
$$\text{Error:} \quad \min_w \sum_i C(y^{(i)}, w^T x^{(i)} + b)$$

$$w^* = \arg\min_w \sum_j w_j^2 + R \sum_i C(y^{(i)}, w^T x^{(i)} + b)$$

Might remind you of regularization, 1/R…

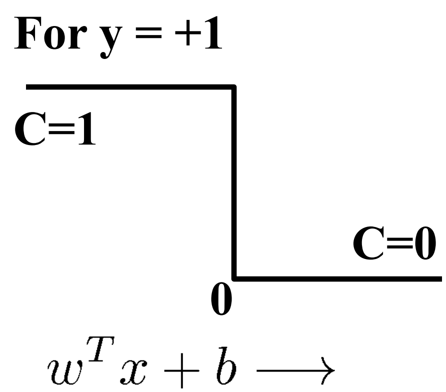
Maximum margin classifier

- Cost function C(.) ?
- C = # of misclassified data?
  - Not smooth = hard to train



Maximum margin classifier

- Cost function C(.) ?
- C = # of misclassified data?
  - Not smooth = hard to train



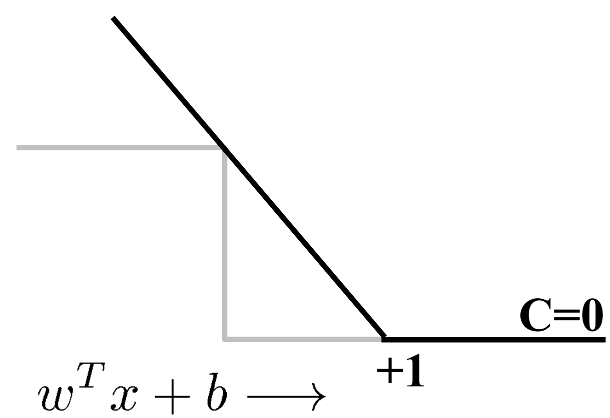
- C = distance from the "correct" place

$$w^* = \arg\min_{w,\epsilon} \sum_j w_j^2 + R \sum_i \epsilon^{(i)}$$

$s.t.$

$$y^{(i)}(\,w^T x^{(i)} + b\,) \geq +1 - \epsilon^{(i)}$$

$$\epsilon^{(i)} \geq 0$$

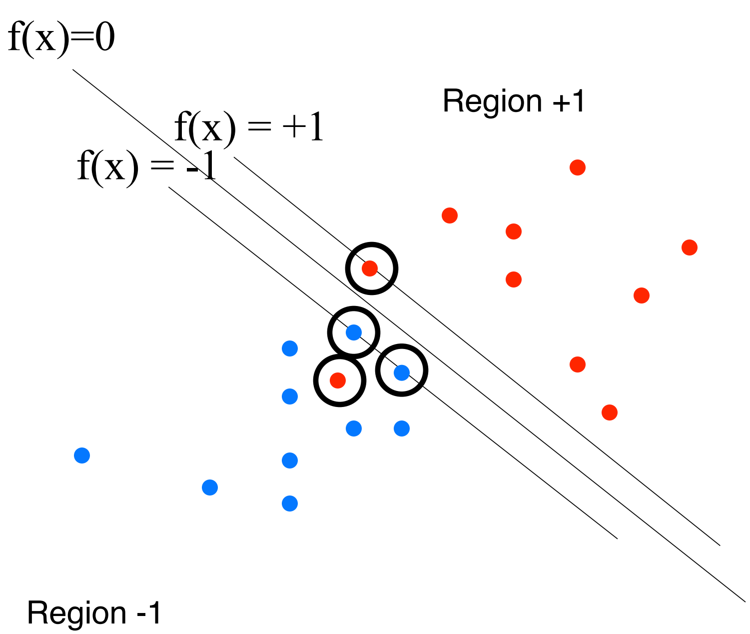


Dual form

- Equivalent form:

$$\max_{0 \leq \alpha \leq R} \sum_i \alpha_i - \frac{1}{2} \sum_j \alpha_i \alpha_j \underbrace{y^{(i)} y^{(j)} x^{(i)^T} x^{(j)}}_{K_{ij}}$$

$$\text{s.t.} \quad \sum_i \alpha_i y^{(i)} = 0$$



Support vectors now data on or past margin…

$$w^* = \sum_i \alpha_i y^{(i)} x^{(i)}$$

$$b = \frac{1}{Nsv} \sum_i (y^{(i)} - w^T x^{(i)})$$

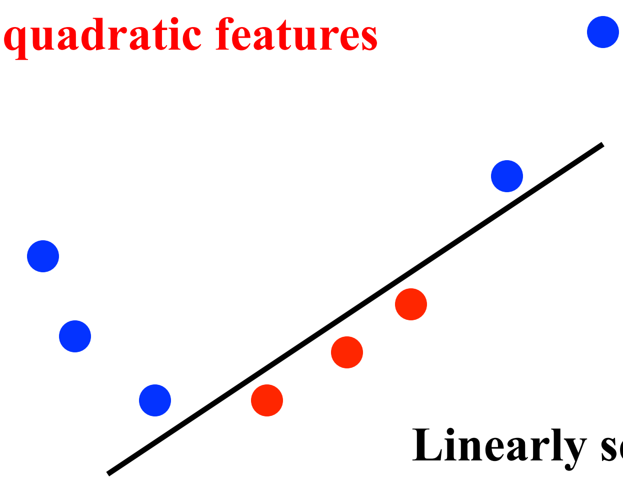
Adding features

- Linear classifier can't learn some functions

1D example:

Not linearly separable

Add quadratic features



Linearly separable in new features…

Adding features

- Feature function Phi

$$\max_{0 \leq \alpha \leq R} \sum_i \alpha_i - \frac{1}{2} \sum_j \alpha_i \alpha_j y^{(i)} y^{(j)} \Phi(x^{(i)})^T \Phi(x^{(j)}) \quad \text{s.t.} \ \sum_i \alpha_i y^{(i)} = 0$$

For example, polynomial features:

$$\Phi(x) = \begin{pmatrix} 1 & \sqrt{2}x_1 & \sqrt{2}x_2 & \cdots & x_1^2 & x_2^2 & \cdots & \sqrt{2}x_1x_2 & \sqrt{2}x_1x_3 & \cdots \end{pmatrix}$$

Implicit features

- Need $\Phi(x^{(i)})^T \Phi(x^{(j)})$

$$\Phi(x) = \begin{pmatrix} 1 & \sqrt{2}x_1 & \sqrt{2}x_2 & \cdots & x_1^2 & x_2^2 & \cdots & \sqrt{2}x_1x_2 & \sqrt{2}x_1x_3 & \cdots \end{pmatrix}$$

$$\Phi(a) = \begin{pmatrix} 1 & \sqrt{2}a_1 & \sqrt{2}a_2 & \cdots & a_1^2 & a_2^2 & \cdots & \sqrt{2}a_1a_2 & \sqrt{2}a_1a_3 & \cdots \end{pmatrix}$$

$$\Phi(b) = \begin{pmatrix} 1 & \sqrt{2}b_1 & \sqrt{2}b_2 & \cdots & b_1^2 & b_2^2 & \cdots & \sqrt{2}b_1b_2 & \sqrt{2}b_1b_3 & \cdots \end{pmatrix}$$

$$\Phi(a)^T \Phi(b) = 1 + \sum_j 2a_jb_j + \sum_j a_j^2 b_j^2 + \sum_j \sum_{k>j} 2a_ja_kb_jb_k + \ldots$$

$$= (1 + \sum_j a_jb_j)^2$$

$$= K(a,b)$$

Common kernel functions

- Polynomial

$$K(a,b) = (1 + \sum_j a_j b_j)^2$$

- Radial-basis functions

$$K(a,b) = \exp(-(a-b)^2/2\sigma^2)$$

- Neural-net style

$$K(a,b) = \tanh(ca^T b + h)$$