+

Machine Learning and Data Mining

Support Vector Machines

Prof. Alexander Ihler

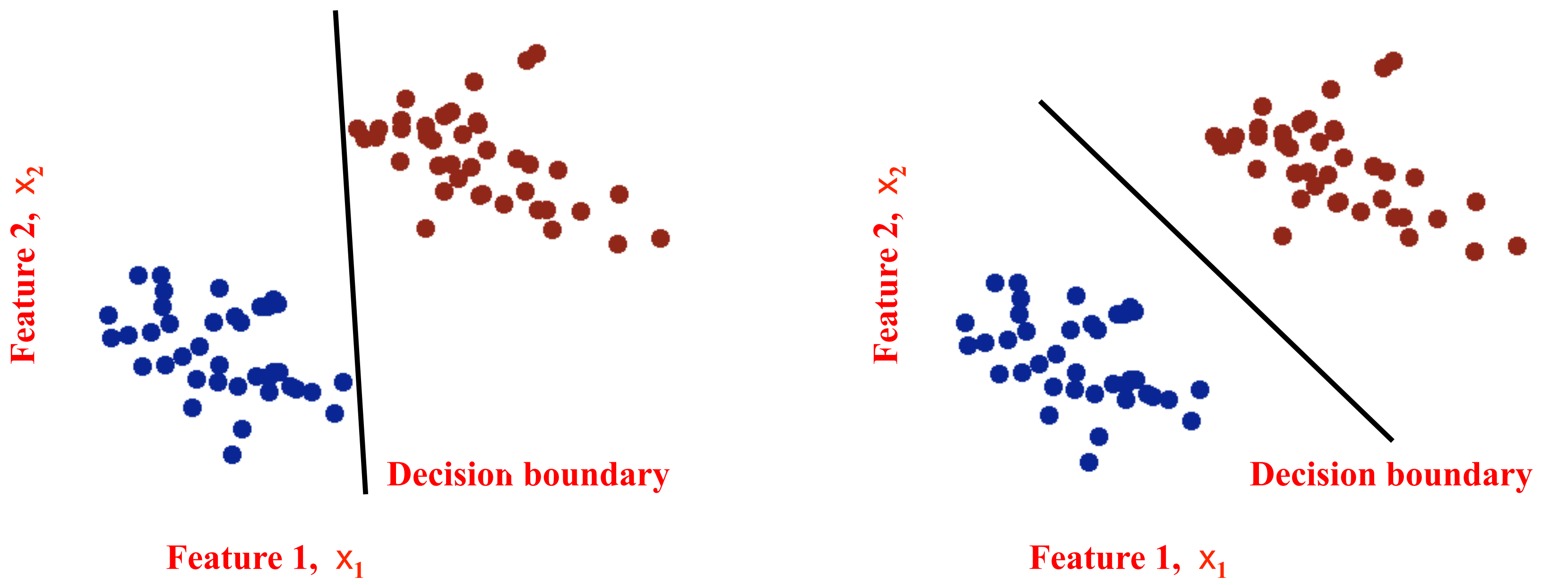
BREN:ICS INFORMATION AND COMPUTER SCIENCES

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…

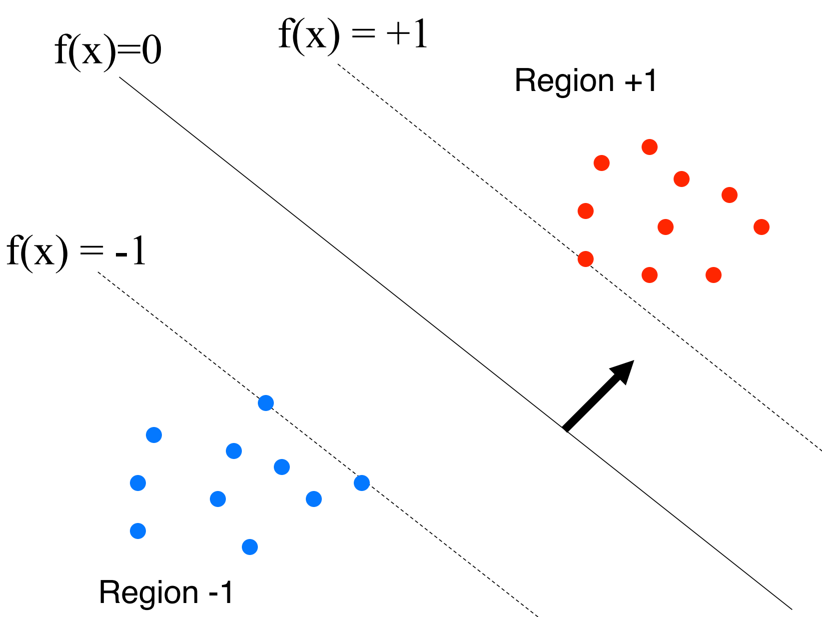
- How can we quantify "better",
  and learn the "best" parameter settings?



One possible answer…

- Maybe we want to maximize our "margin"
- To optimize, relate to model parameters
- Remove "scale invariance"
  - Define class +1 in some region, class –1 in another
  - Make those regions as far apart as possible

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots$$
$$\Downarrow$$
$$b + w_1 x_1 + w_2 x_2 + \ldots$$



We could define such a function:

f(x) = w*x' + b

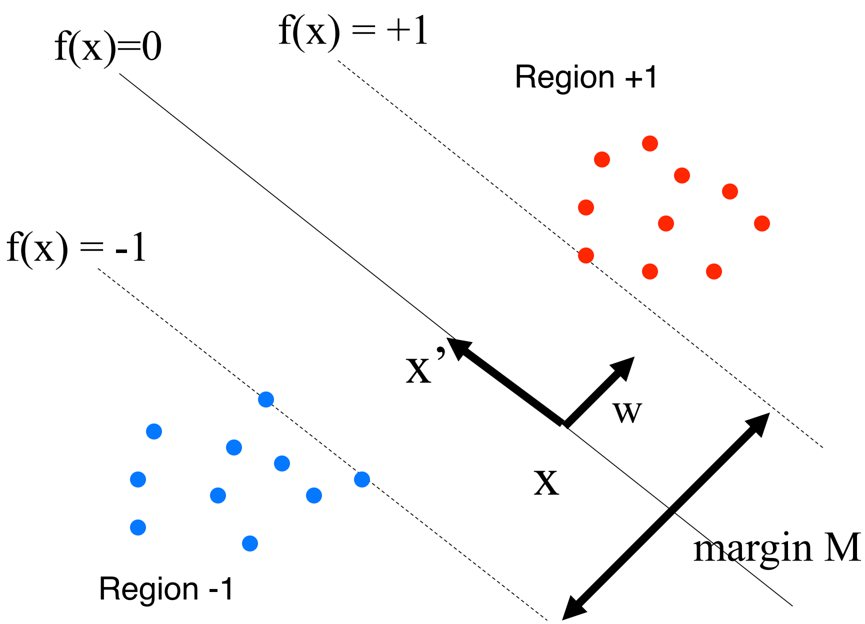
f(x) > +1 in region +1
f(x) < –1 in region –1

Passes through zero in center…

"Support vectors" – data points on margin

Computing the margin width

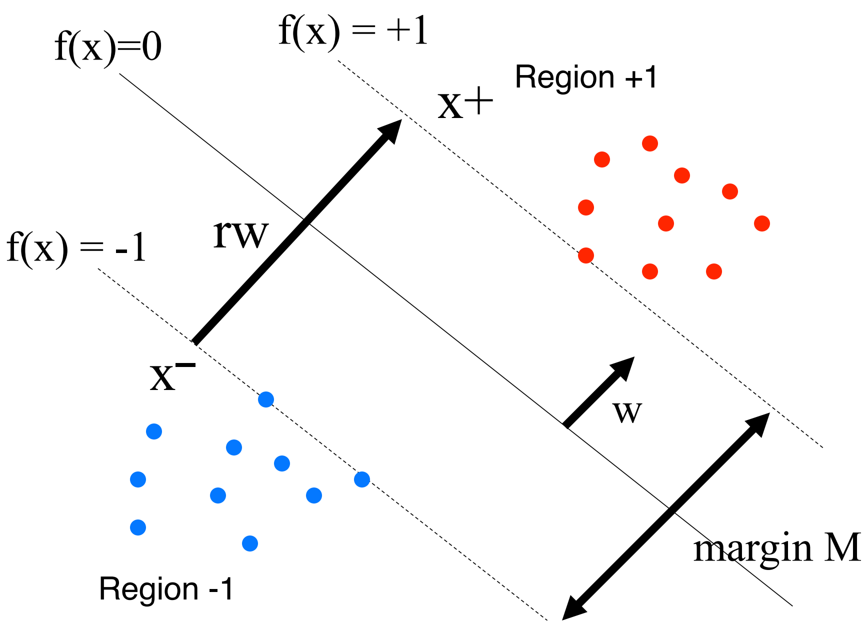
- Vector $\underline{w}=[w_1\ w_2\ \ldots]$ is perpendicular to the boundaries (why?)
- $w\,x + b = 0$ & $w\,x' + b = 0$ => $w\,(x'-x) = 0$: orthogonal



Computing the margin width

- Vector $\underline{w}=[w_1\ w_2\ \ldots]$ is perpendicular to the boundaries
- Choose $\underline{x}^-$ st $f(\underline{x}^-) = -1$; let $\underline{x}^+$ be the closest point with $f(\underline{x}^+) = +1$
  - $\underline{x}^+ = \underline{x}^- + r * \underline{w}$ (why?)
- Closest two points on the margin also satisfy

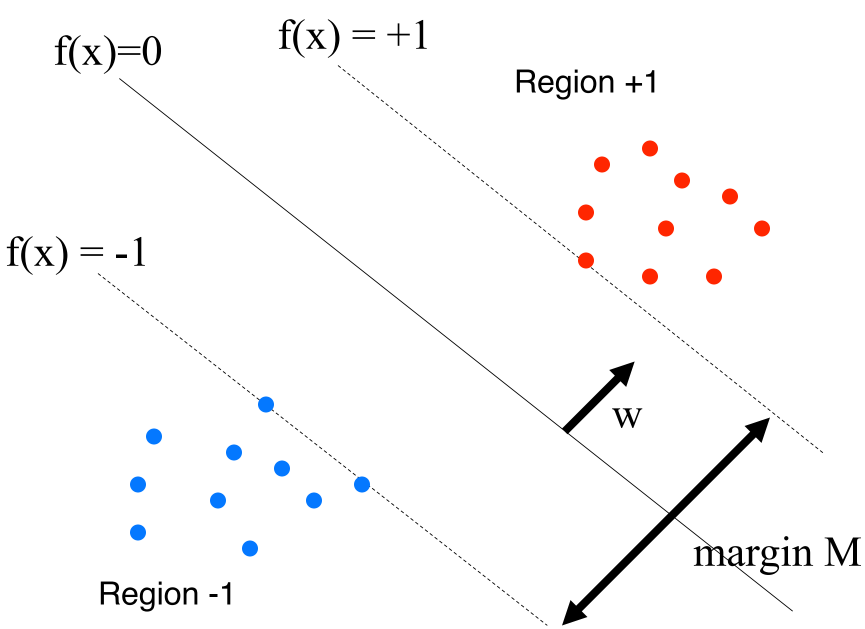
$$w \cdot x^- + b = -1 \qquad\qquad w \cdot x^+ + b = +1$$



Computing the margin width

- Vector $\underline{w}=[w_1\ w_2\ \ldots]$ is perpendicular to the boundaries
- Choose $\underline{x}^-$ st $f(\underline{x}^-) = -1$; let $\underline{x}^+$ be the closest point with $f(\underline{x}^+) = +1$
  - $\underline{x}^+ = \underline{x}^- + r * \underline{w}$
- Closest two points on the margin also satisfy

$$w \cdot x^- + b = -1 \qquad\qquad w \cdot x^+ + b = +1$$



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

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

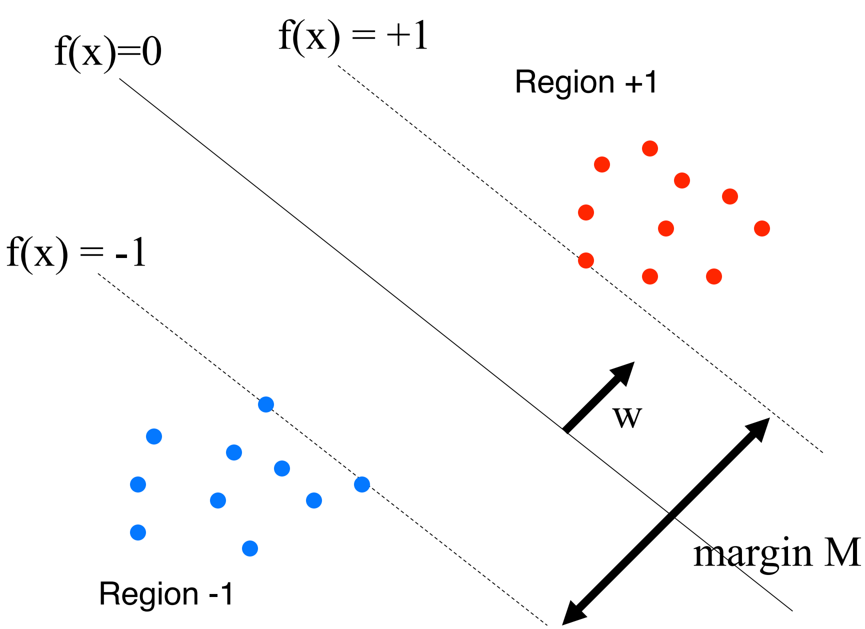
Maximum margin classifier

- 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



Primal problem:

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

s.t.

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

(m constraints)

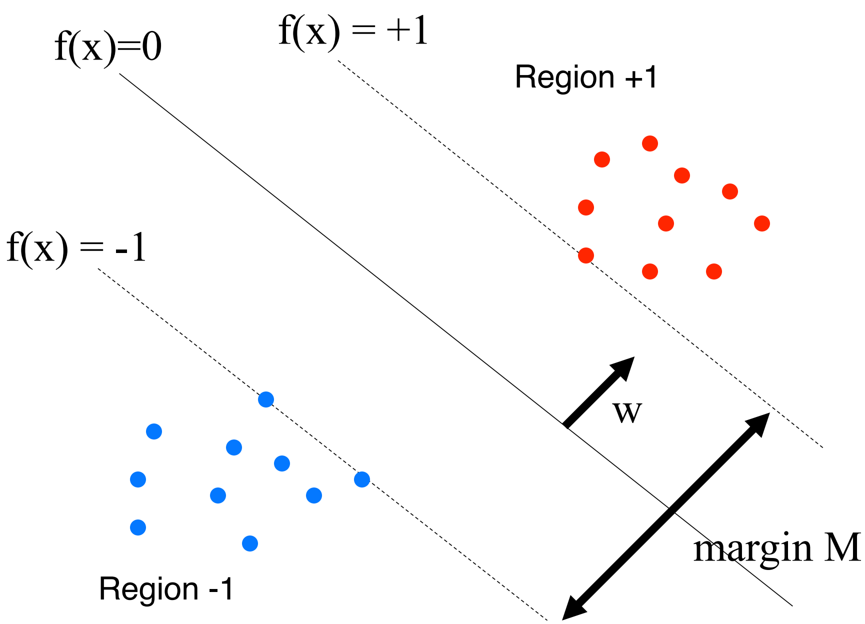
Maximum margin classifier

- 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



Primal problem:

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

s.t.

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

(m constraints)

Lagrangian optimization

- Want to optimize constrained system: $\theta = (\text{w,b})$

$$\underbrace{w^* = \arg\min_{w,b} \sum_j w_j^2}_{f(\theta)} \qquad s.t. \qquad \underbrace{1 - y^{(i)}(\,w \cdot x^{(i)} + b\,) \leq 0}_{g_i(\theta) \leq 0}$$

- Introduce Lagrange mutipliers α (one per constraint)

$$\theta^* = \arg\min_{\theta} \max_{\alpha \geq 0} f(\theta) \; + \; \sum_i \alpha_i \, g_i(\theta)$$

  - Can optimize θ, α jointly, with a simple constraint set
  - Then: $g_i(\theta) \leq 0 \;:\; \alpha_i = 0$

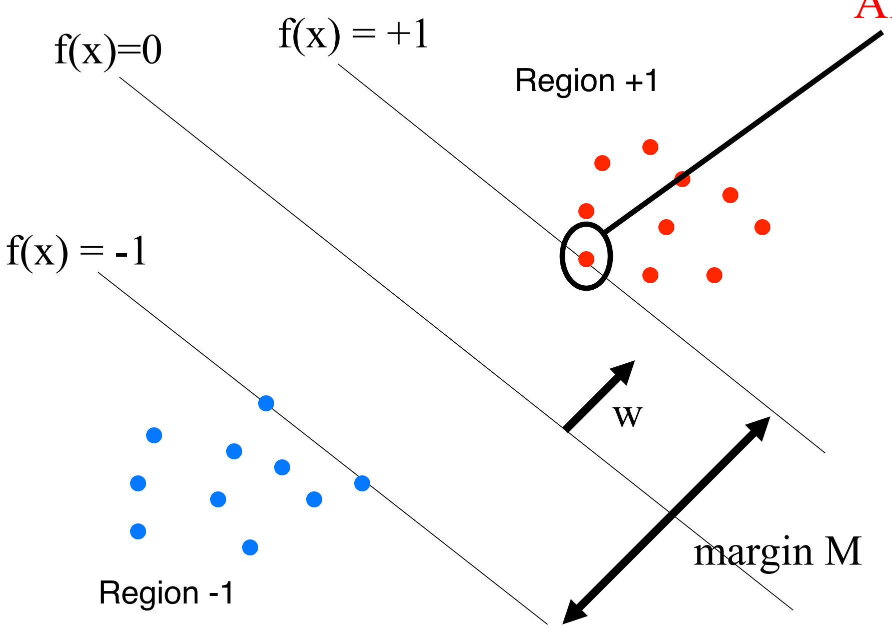
    $g_i(\theta) > 0 \;:\; \alpha_i \rightarrow +\infty$

  - Any optimum of the original problem is a saddle point of the new
  - KKT complementary slackness: $\alpha_i > 0 \;\Rightarrow\; g_i(\theta) = 0$

Optimization

- Use Lagrange multipliers
  - Enforce inequality constraints

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



Alphas > 0 only on the margin: "support vectors"

Stationary conditions wrt w:

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

and since any support vector has y = wx + b,

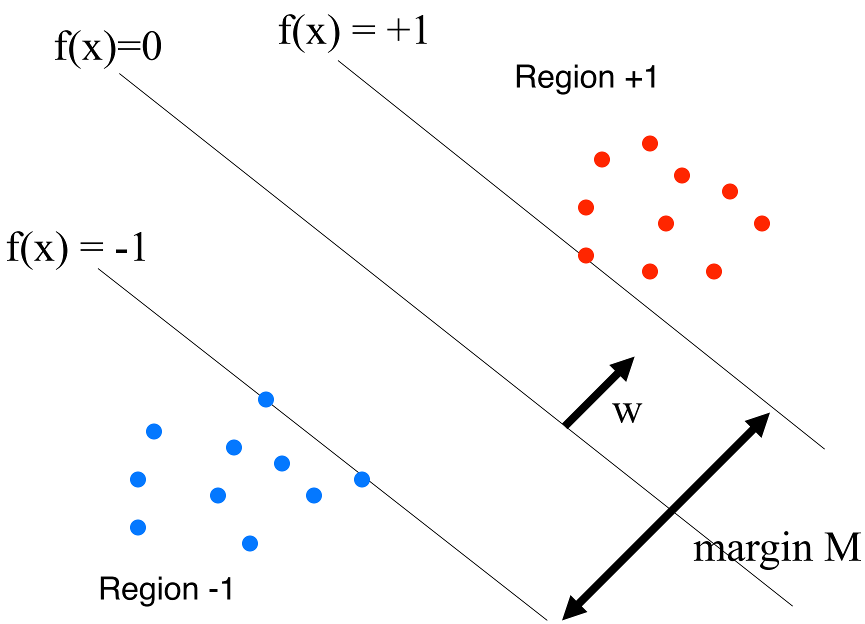
$$b = \frac{1}{Nsv} \sum_{i \in SV} (y^{(i)} - w \cdot x^{(i)})$$

Dual form

- Use Lagrange multipliers
  - Enforce inequality constraints
  - Use solution w* to write solely in terms of alphas:

$$\max_{\alpha \geq 0} \sum_i \left[\alpha_i - \frac{1}{2} \sum_j \alpha_i \alpha_j \, y^{(i)} y^{(j)} \, (x^{(i)} \cdot x^{(j)}) \, \right]$$

$$\text{s.t.} \sum_i \alpha_i y^{(i)} = 0 \qquad \text{(since derivative wrt b = 0)}$$



Another quadratic program:
optimize m vars with 1+m (simple) constraints
cost function has m^2 dot products

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

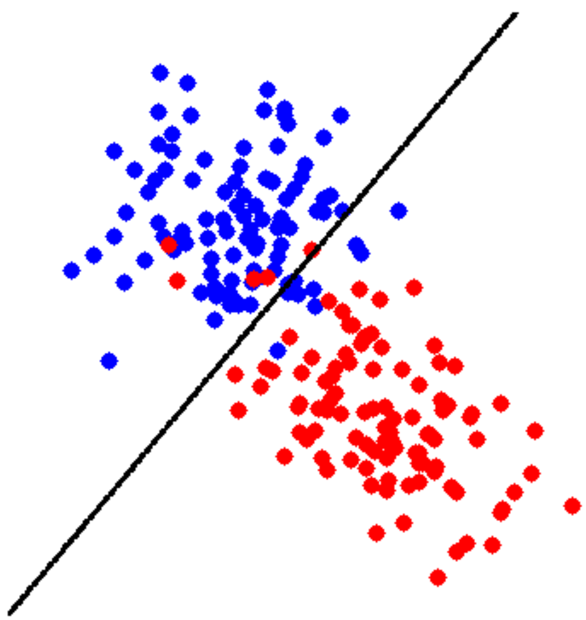
$$b = \frac{1}{Nsv} \sum_{i \in SV} (y^{(i)} - w \cdot x^{(i)})$$

Maximum margin classifier

- What if the data are not linearly separable?
  - Want a large "margin": Want low error:

$$\min_w \sum_j w_j^2 \qquad\qquad \min_w \sum_i J(y^{(i)}\ ,\ w \cdot x^{(i)} + b)$$

  - "Soft margin" : introduce slack variables for violated constraints



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

$$s.t.$$

$$y^{(i)}(\, w^T x^{(i)} + b\,) \geq +1 - \epsilon^{(i)} \quad \text{(violate margin by } \epsilon\text{)}$$

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

Assigns "cost" R proportional to distance from margin
Another quadratic program!

Maximum margin classifier

- Soft margin optimization:

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

  - For *any* weights w,

  we can choose ϵ to satisfy constraints $y^{(i)}(\,w^T x^{(i)} + b\,) \geq +1 - \epsilon^{(i)}$

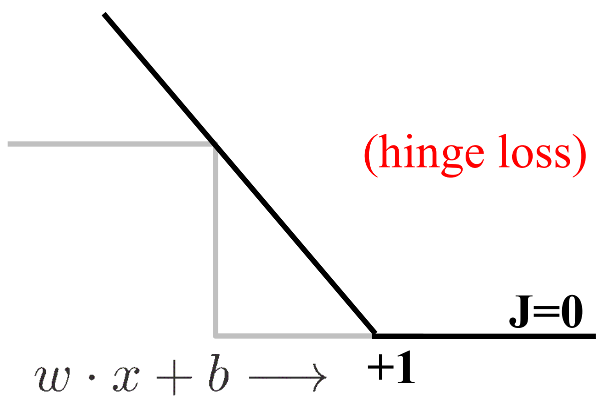
  - Write ϵ* as a function of w (call this J) and optimize directly

- J = distance from the "correct" place

$$J_i = \max[\,0\,,\,1 - y^{(i)}(\,w \cdot x^{(i)} + b\,)\,]$$

$$w^* = \arg\min_{w} \frac{1}{R} \sum_j w_j^2 + \sum_i J_i(y^{(i)}\ ,\ w \cdot x^{(i)} + b)$$

(L2 regularization on the weights)



Dual form

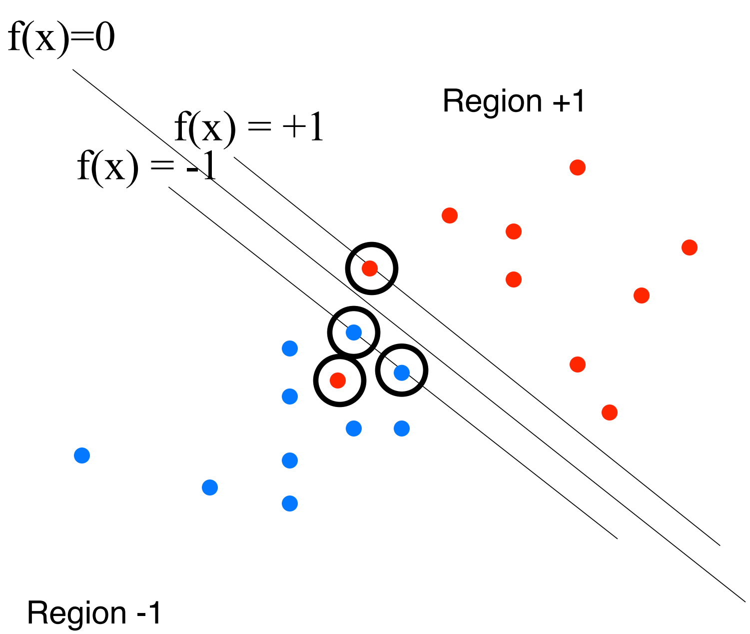
- Equivalent form:

K_{ij}

$$\max_{0 \le \alpha \le R} \sum_i \alpha_i - \frac{1}{2} \sum_j \alpha_i \alpha_j \, y^{(i)} y^{(j)} \, (x^{(i)} \cdot x^{(j)})$$

K_{ij} measures "similarity" of x_i and x_j (their dot product)

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



Support vectors now data on or past margin…

Prediction:

$$\hat{y} = w^* \cdot x + b = \sum_i \alpha_i y^{(i)} x^{(i)} \cdot x + b$$

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

$$b = \dots$$

More complicated; can solve e.g. using any $\alpha \in$ (0,R)

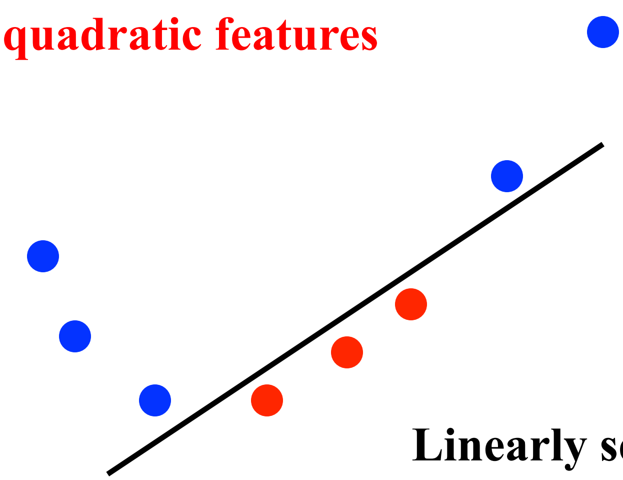
Adding features

- 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

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

For example, polynomial features:

$$\Phi(x) = \left(1 \; \sqrt{2}x_1 \; \sqrt{2}x_2 \; \cdots \; x_1^2 \; x_2^2 \; \cdots \; \sqrt{2}x_1 x_2 \; \sqrt{2}x_1 x_3 \; \cdots\right)$$

Implicit features

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

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

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

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

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

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

$$= K(a, b)$$

Common kernel functions

- Polynomial

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

- Radial-basis functions

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

- Neural-net style

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

- Others for specialized data (text, genetics, etc.)

Summary

- Support vector machines
- "Large margin" for separable data
  - Primal QP: maximize margin subject to linear constraints
  - Lagrangian optimization simplifies constraints
  - Dual QP: m variables; involves m^2 dot product
- "Soft margin" for non-separable data
  - Primal form: regularized hinge loss
  - Dual form: m-dimensional QP
- Kernels
  - Dual form involves only pairwise similarity
  - Mercer kernels: dot products in implicit high-dimensional space