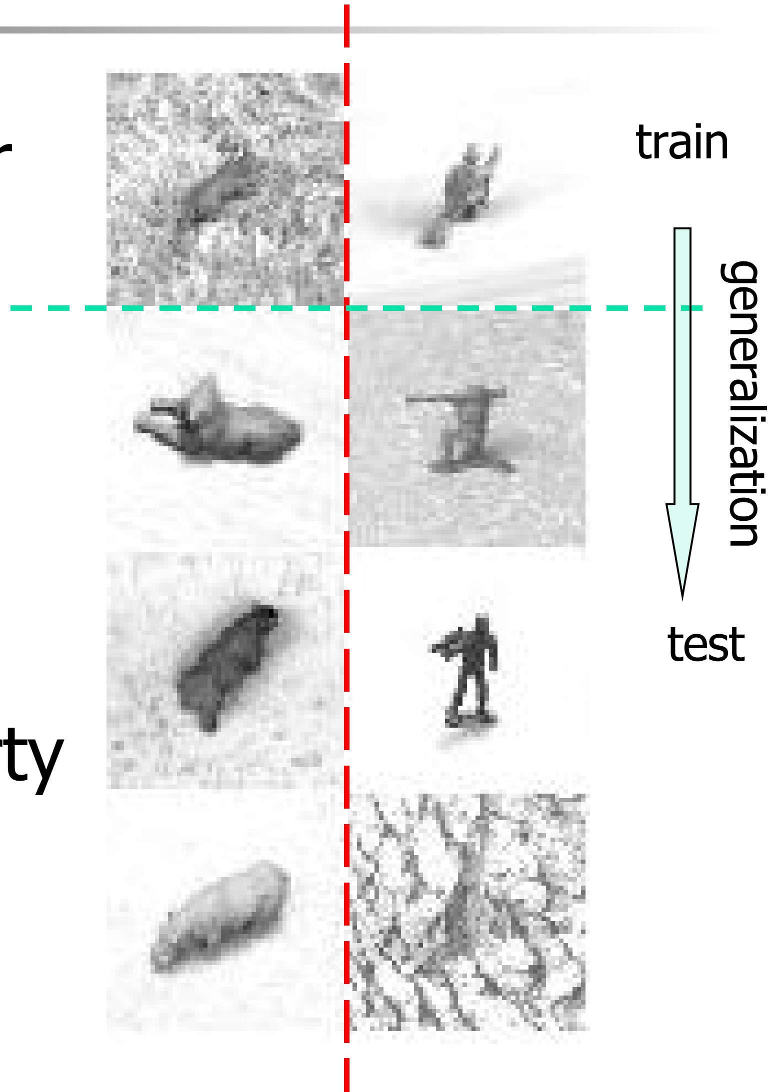
Support Vector Machines

Fu Jie Huang

Dec 5, 2006

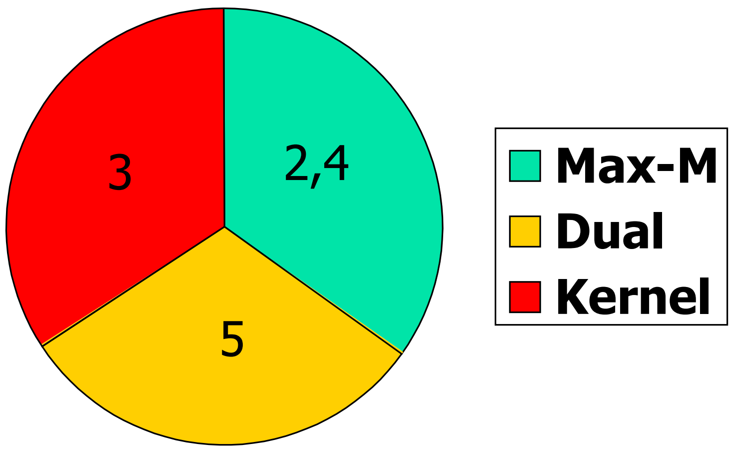
What’s SVM

- A (mostly) binary classifier
- A linear classifier
- Supervised training
- Nice generalization property



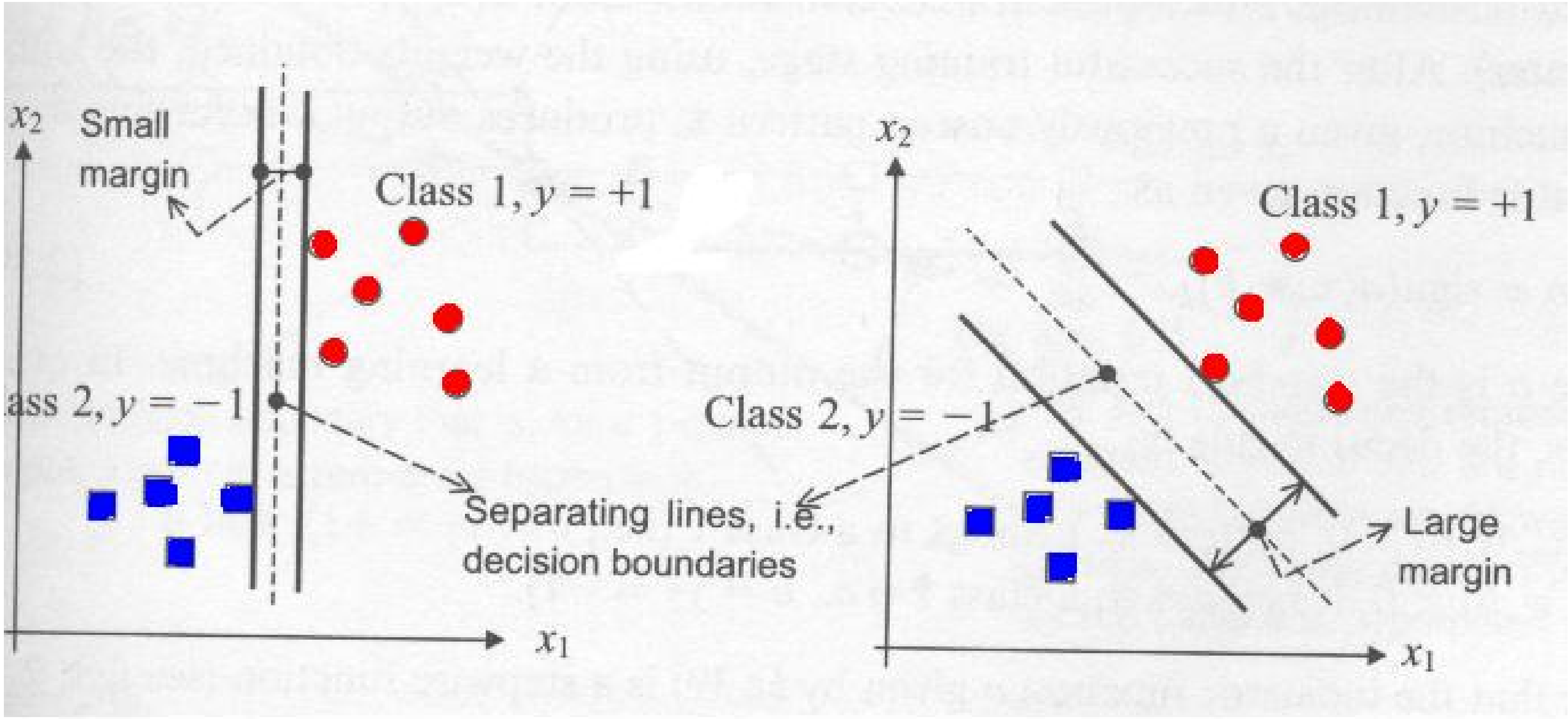
Main Ideas

- Maximal margin
- Dual space
- Kernel trick



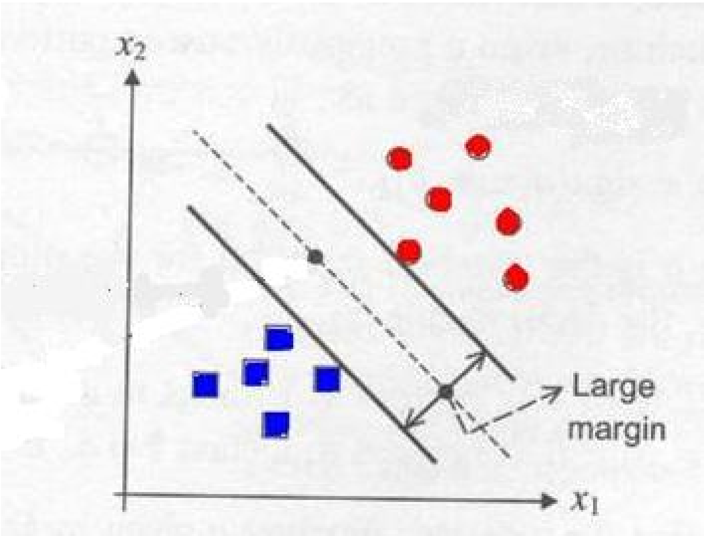
Maximal Margin

- Separating hyperplane is not unique
- Choose the one with… maximal margin



Motivation

- Generalization Error
  - Small
  - Predictable

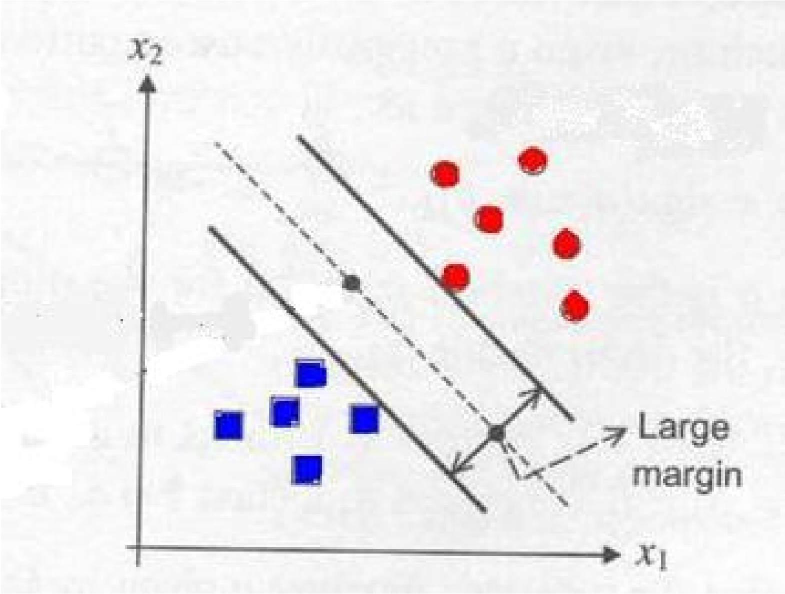


generalization error

$$E_{test} = E_{train} + E_{generalization}$$

Definition

- Maximize minimum distance
  - Data points to hyperplane
- Normalization needed



$$\max_{w,b} \min_{i} \frac{|\mathbf{w}^T \mathbf{x}_i + b|}{\|\mathbf{w}\|}$$

Canonical Form

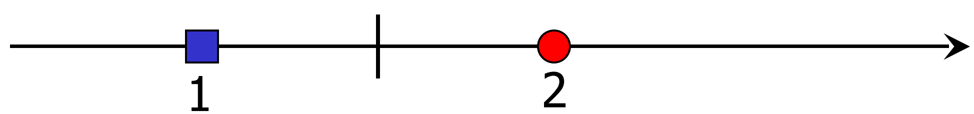
- Normalize by scaling $f(w, b) = k \cdot \boldsymbol{w}^T \boldsymbol{x} + k \cdot b = 0$

$$\min_i |\boldsymbol{w}^T \boldsymbol{x}_i + b| = 1$$

$$x - 1.5 = 0$$

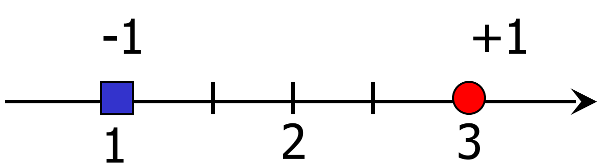
$$2x - 3 = 0$$

Canonical form (w=2)
|2*2-3|=1



Quadratic Programming

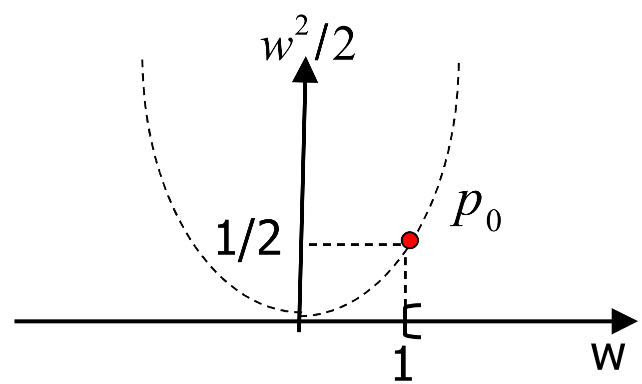
$$\max_{\boldsymbol{w},b}\ \min_{i} \frac{|\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}}+b|}{\|\boldsymbol{w}\|} \quad \xrightarrow{\ \min_i |\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{i}}+b|=1\ } \quad \begin{array}{c} \min_{w,b} \frac{1}{2}\langle \boldsymbol{w}^{\boldsymbol{T}}\cdot\boldsymbol{w}\rangle \\ y_i(\langle \boldsymbol{w}^{\boldsymbol{T}}\cdot\boldsymbol{x}_{\boldsymbol{i}}\rangle+b)\geq 1 \end{array}$$



$$\min_{w} \frac{w^2}{2}$$

$$(+1)(w\cdot 3+b)\geqslant 1$$

$$(-1)(w\cdot 1+b)\geqslant 1$$



An Alternative Way

$$\min f_0(x)$$
$$f_i(x) \geqslant 0$$

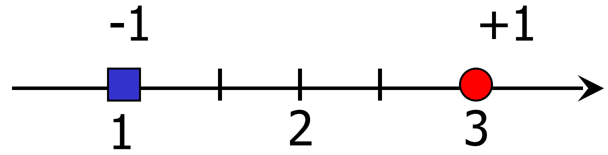
- Step 1-2-3:
  - Combine target and constraints
  - Minimize over primal
  - Maximize over dual

$$L(x, \boldsymbol{\lambda}) = f_0(x) - \sum \lambda_i f_i(x)$$

$$Q(\boldsymbol{\lambda}) = \min_x L(x, \boldsymbol{\lambda})$$

$$\max_\lambda Q(\boldsymbol{\lambda}), \lambda > 0$$

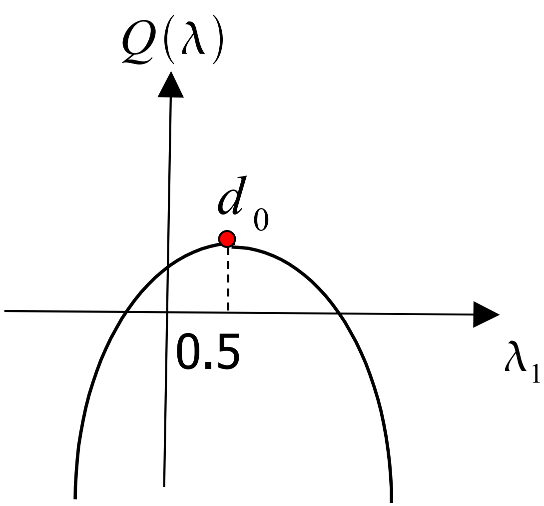
Dual Space



$$\min_{w} \frac{w^2}{2}$$

$$(+1)(w \cdot 3 + b) \geqslant 1$$

$$(-1)(w \cdot 1 + b) \geqslant 1$$



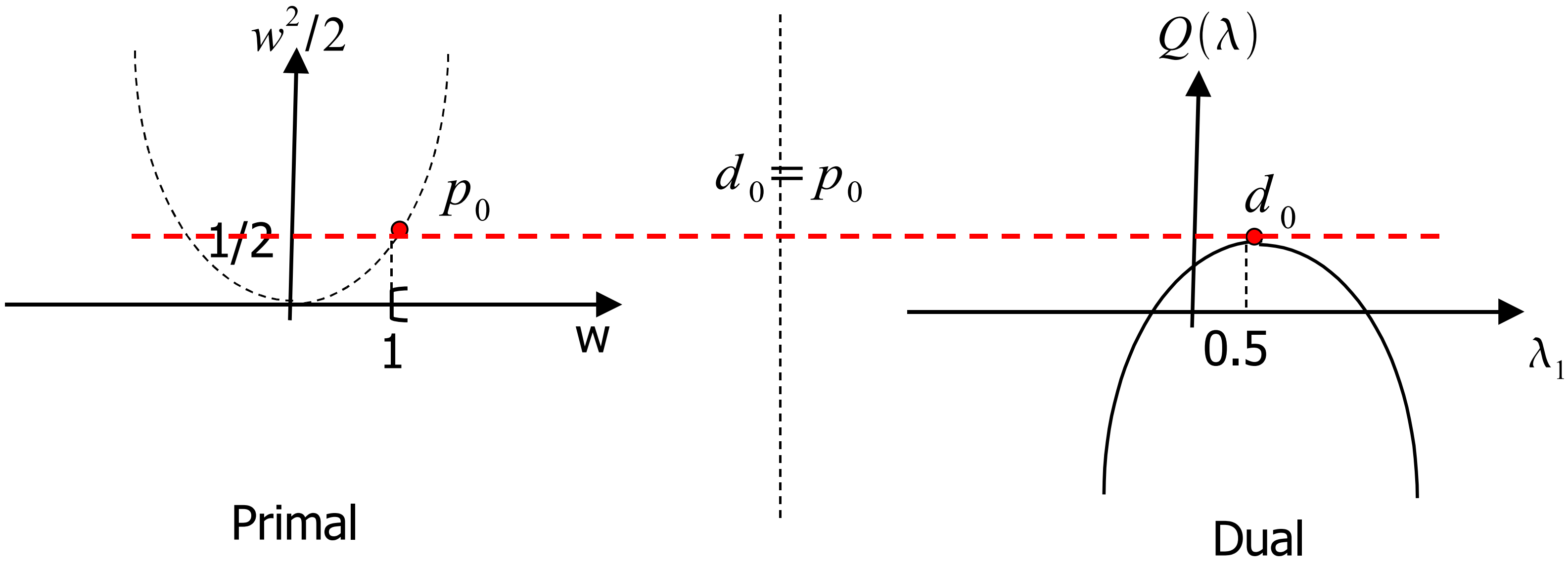
$$L(w, b, \boldsymbol{\lambda}) = w^2/2 - \lambda_1(3\mathrm{w} + b - 1) - \lambda_2(-w - b - 1)$$

$$\min_{w,b} L(w, b, \lambda) \Rightarrow \begin{cases} \lambda_1 = \lambda_2 \\ w = 3\lambda_1 - \lambda_2 = 2\lambda_1 \\ Q(\boldsymbol{\lambda}) = Q(\lambda_1) = -2\lambda_1^2 + 2\lambda_1 \end{cases}$$

$$\max_{\lambda} Q(\lambda) \Rightarrow \lambda_1 = \lambda_2 = 1/2, w = 1, b = 2$$

Strong Duality

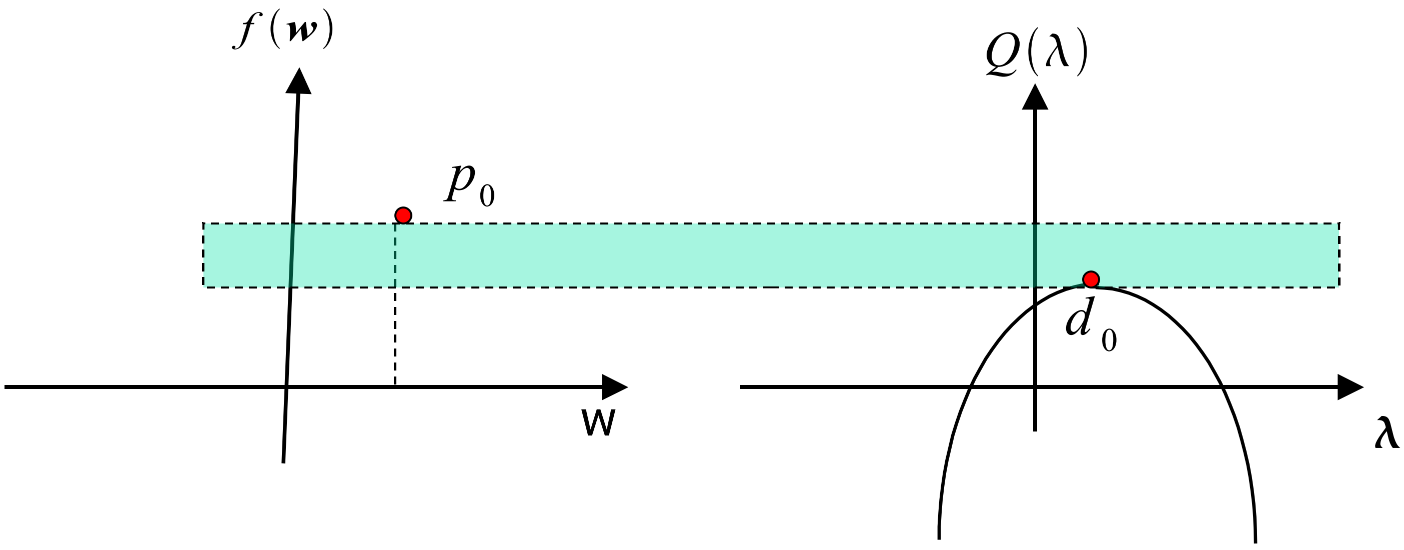
- Primal and dual space optimization:
  - Same result!



Duality Gap

$$d_0 < p_0$$

- In a general case
  - Strong duality is not true
  - "Step 1-2-3" a lower bound, not a solution



Convexity Saves the Day

- Convex function
  - Quadratic programming
- Convex set
  - Linear constraints
- No duality gap

$$\min_{w,b} \frac{1}{2}\langle \boldsymbol{w}^T \cdot \boldsymbol{w} \rangle$$

$$y_i(\langle \boldsymbol{w}^T \cdot \boldsymbol{x_i} \rangle + b) \geq 1$$

$$d_0 = p_0$$

Dual Form

- Formalize "step 1-2-3"
- H
  - Hessian matrix
  - Gram matrix
- Lambda
  - Support vector
  - Sparse

$$\max_{\lambda} Q(\boldsymbol{\lambda}) = -0.5\boldsymbol{\lambda}^{\boldsymbol{T}} \boldsymbol{H} \boldsymbol{\lambda} + \boldsymbol{f}^{\boldsymbol{T}} \boldsymbol{\lambda}$$

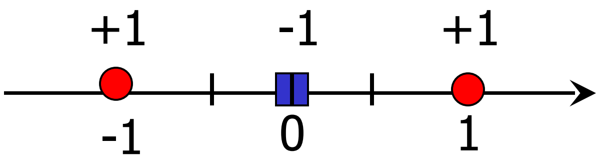
$$\boldsymbol{y}^{\boldsymbol{T}} \boldsymbol{\lambda} = 0$$

$$\boldsymbol{\lambda} \geq 0$$

$$where,\ H_{ij} = y_i y_j \langle \boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{T}} \cdot \boldsymbol{x}_{\boldsymbol{j}} \rangle$$

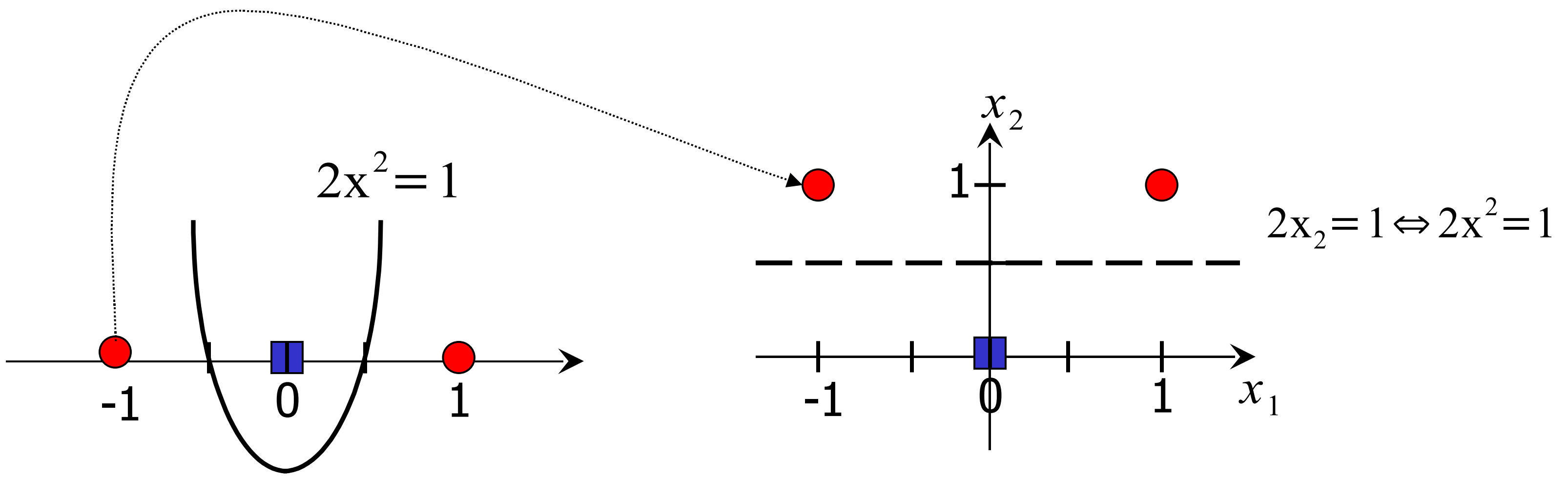
$$\boldsymbol{f}\ is\ a\ unit\ vector$$

Non-separable Data



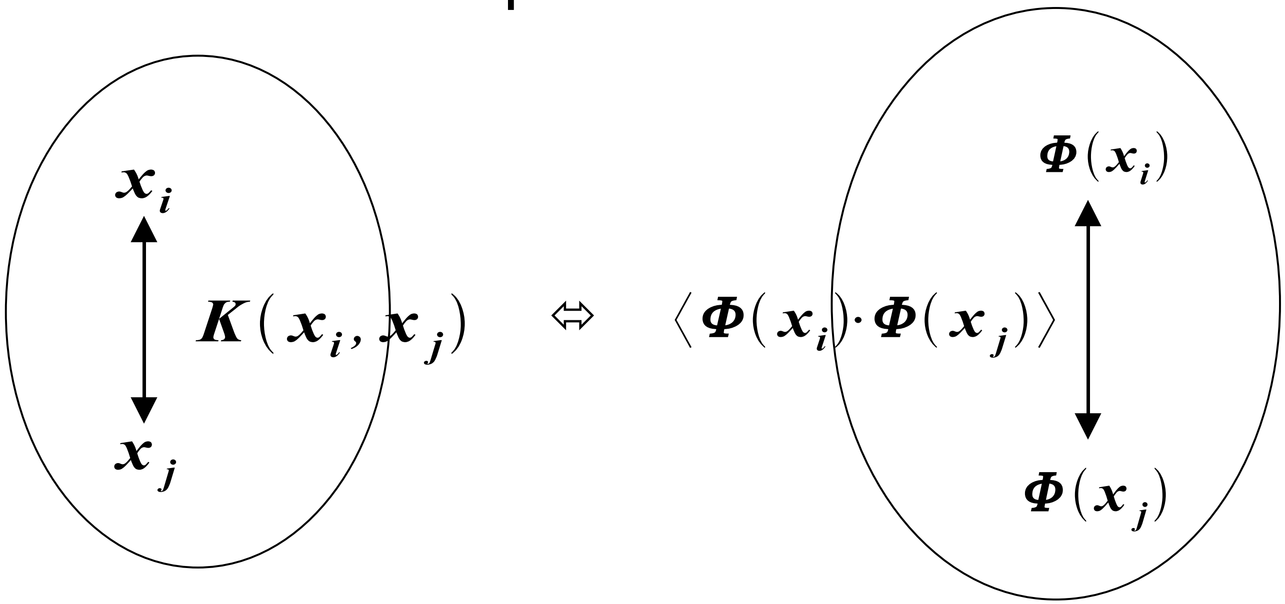
- Solutions:
  - Nonlinear classifier
  - Increase the dimension

$$[x] \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x \\ x^2 \end{bmatrix}$$

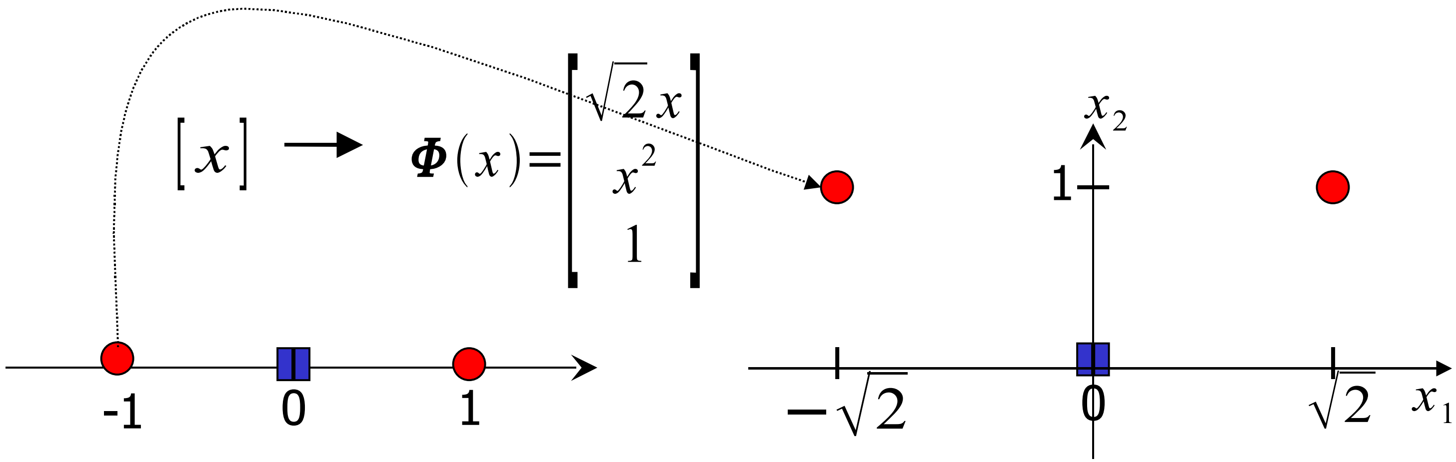


Kernel Trick

- Kernel function
  - in the original space
- Inner product
  - In the feature space with increased dimension



Example



$$\boldsymbol{K}(x_i, x_j) = (x_i x_j + 1)^2$$

$$\langle \boldsymbol{\Phi}(x_i) \cdot \boldsymbol{\Phi}(x_j) \rangle = 2\mathrm{x}_i x_j + x_i^2 x_j^2 + 1 = (x_i x_j + 1)^2 = \boldsymbol{K}(x_i, x_j)$$

Curse of Dimensionality

- Primal space
  - Makes optimization much harder
- Dual space
  - Can be avoided

$$\min_{w,b} \frac{1}{2}\langle \boldsymbol{\Phi}^{\boldsymbol{T}}(\boldsymbol{w})\cdot\boldsymbol{\Phi}(\boldsymbol{w})\rangle$$

$$y_i(\langle \boldsymbol{\Phi}^{\boldsymbol{T}}(\boldsymbol{w})\cdot\boldsymbol{\Phi}(\boldsymbol{x_i})\rangle + b) \geq 1$$

$$\max_{\lambda} Q(\lambda) = -0.5\lambda^{\boldsymbol{T}}\boldsymbol{H}\lambda + \boldsymbol{f}^{\boldsymbol{T}}\lambda$$

$$\boldsymbol{y}^{\boldsymbol{T}}\lambda = 0$$

$$\lambda \geq 0$$

$$where,\ H_{ij} = y_i y_j \boldsymbol{K}(\boldsymbol{x_i}, \boldsymbol{x_j})$$

$$\boldsymbol{f}\ is\ a\ unit\ vector$$

Mercer Condition

- Dual form is convex
  - H is P.S.D.
  - Kernel must be P.S.D.

$$Q(\lambda) = -0.5\lambda^T \boldsymbol{H} \lambda + \boldsymbol{f}^T \lambda$$
$$where,\ H_{ij} = y_i y_j K(\boldsymbol{x_i}, \boldsymbol{x_j})$$

- Mercer kernels
  - Polynomial
  - Gaussian

$$K(\boldsymbol{x}, \boldsymbol{y}) = [\langle \boldsymbol{x}^T \boldsymbol{y} \rangle + 1]^p$$

$$K(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{e}^{-(\boldsymbol{x}-\boldsymbol{y})^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{y})/2}$$

Why Is SVM So Popular?

- Works very well
- Fool-proof
  - Only 2 kernels to choose from
  - Very few hand-crafted parameters
- Error bound easy to get $\frac{|SV|}{N}$

$$E_{test} = E_{train} + E_{generalization}$$

Disadvantages

- Slow
  - Training: quadratic programming
    - SMO, etc.
    - Hardware
  - Testing: depend on number of SV
- Big
  - Curse of sample size: H matrix
    - Online SVM