컴퓨터비전 및 패턴인식 연구회
2009.2.12

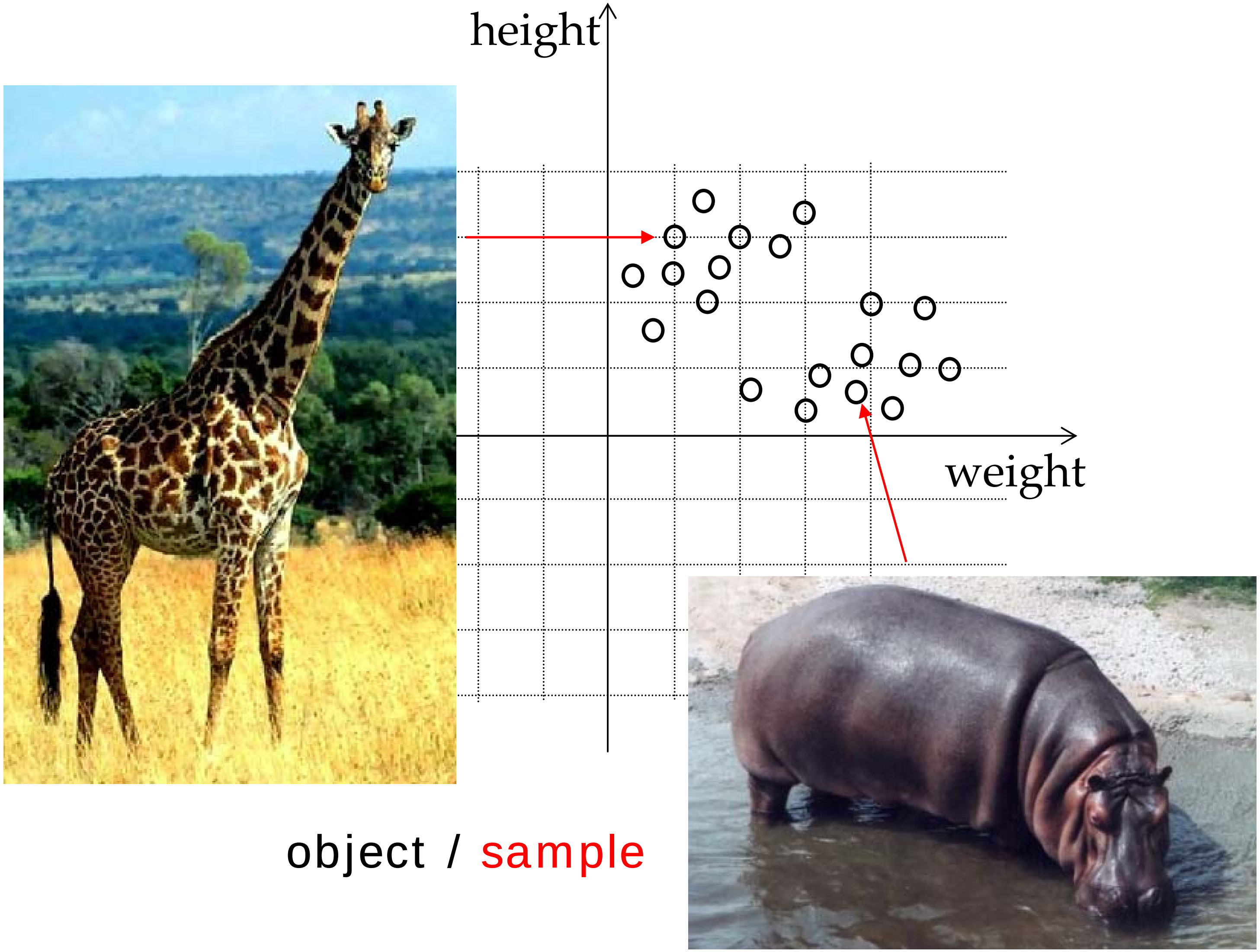
Support Vector Machines

http://cespc1.kumoh.ac.kr/~nonezero/svm-ws-cvpr.pdf

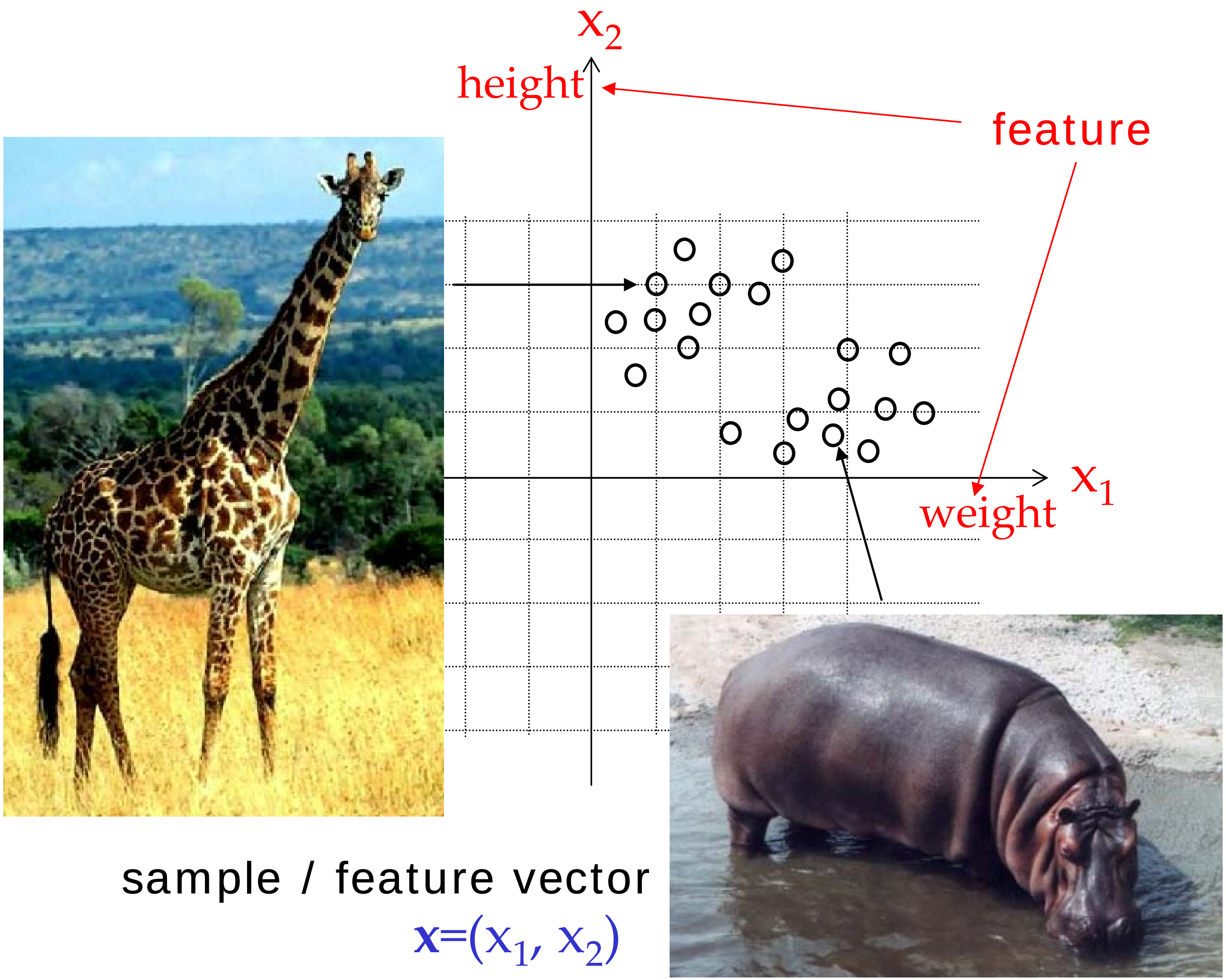
금오공과대학교
컴퓨터공학부
고 재 필

Contents

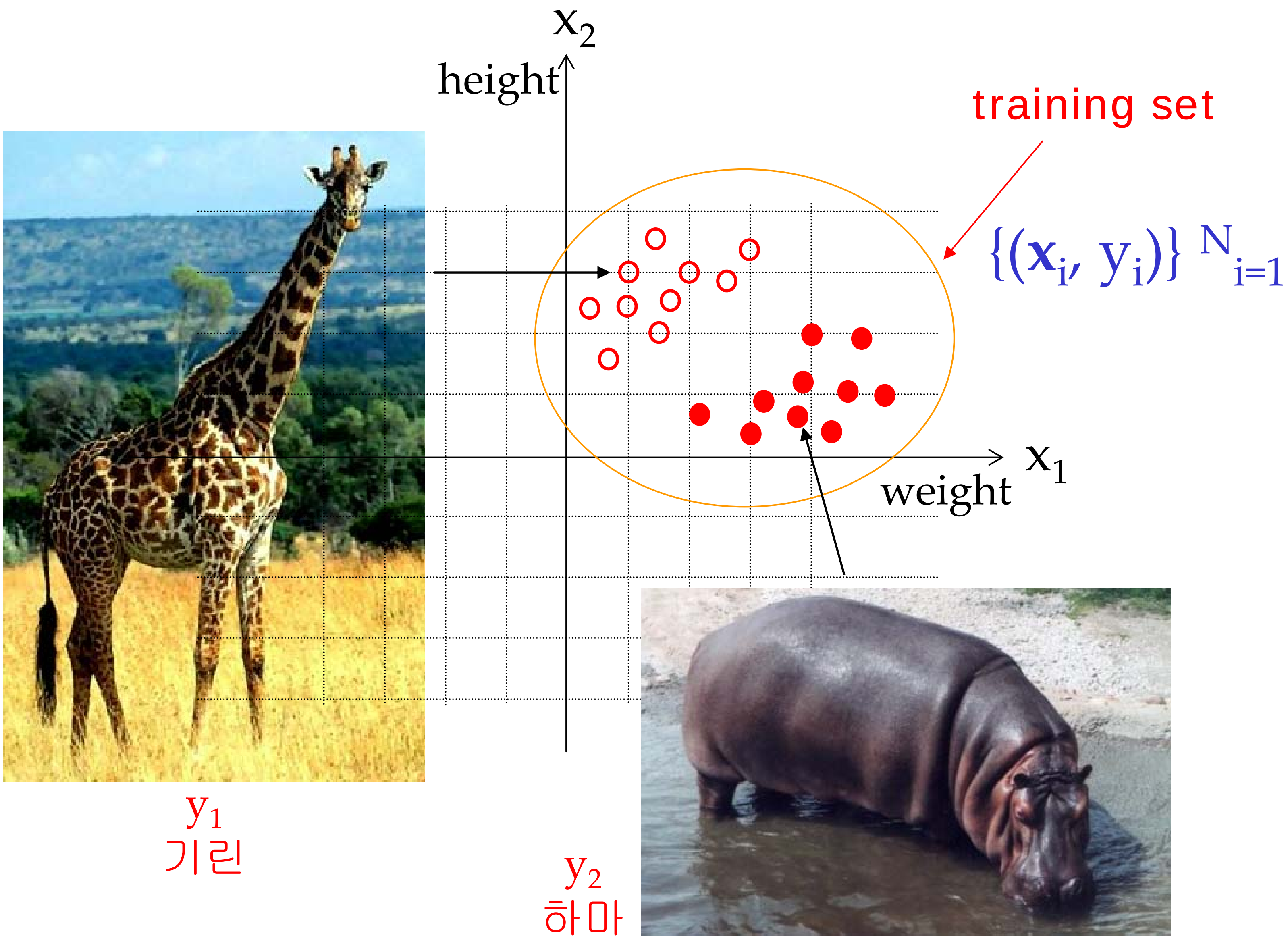
Sample



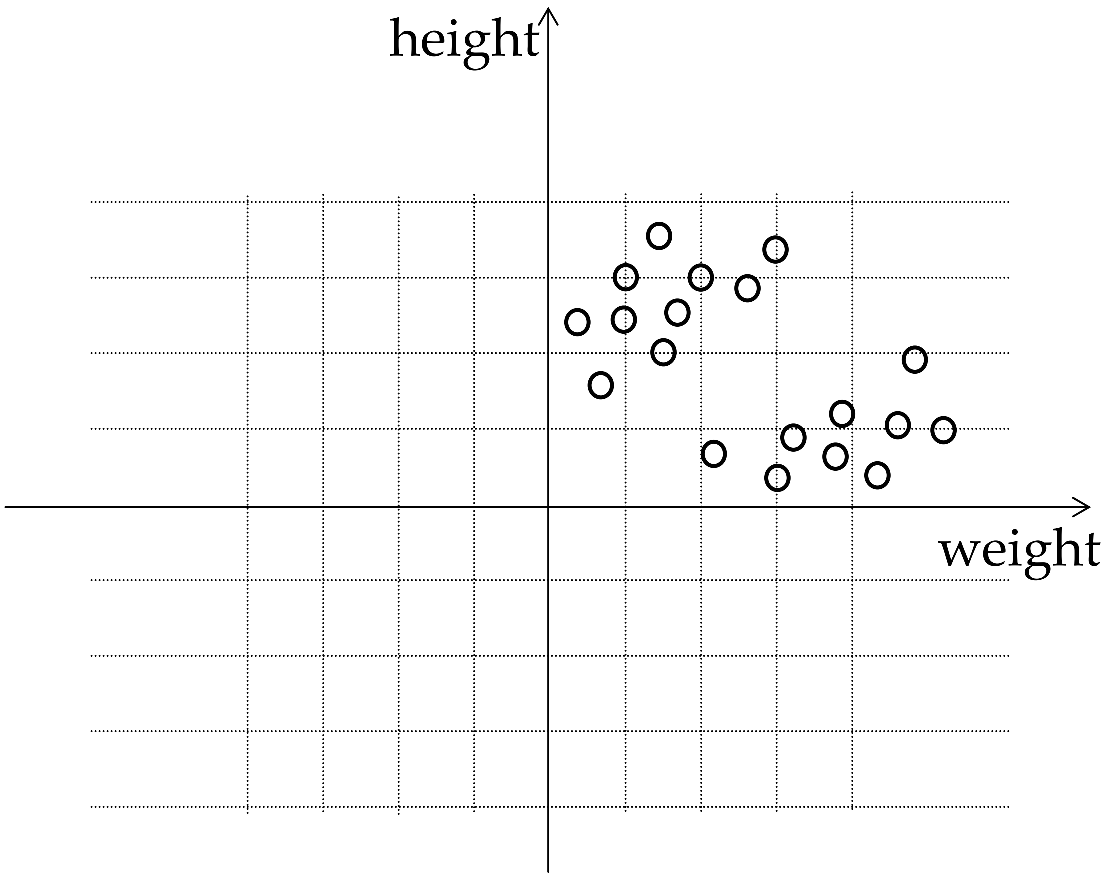
Feature



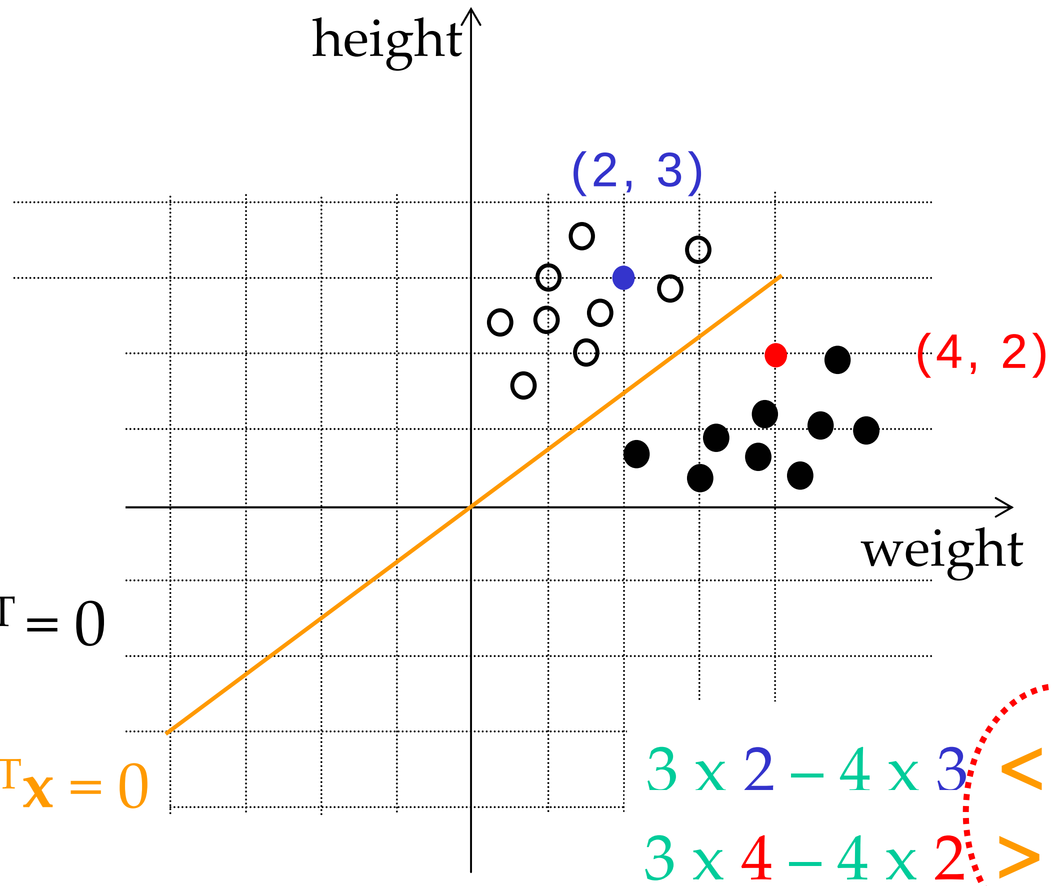
Training Set



How to Classify Them Using Computer ?



How to Classify Them Using Computer ?



$x_2 = 3/4x_1$

$4x_2 = 3x_1$

$3x_1 - 4x_2 = 0$

$(3\ {-4})\ (x_1\ x_2)^T = 0$

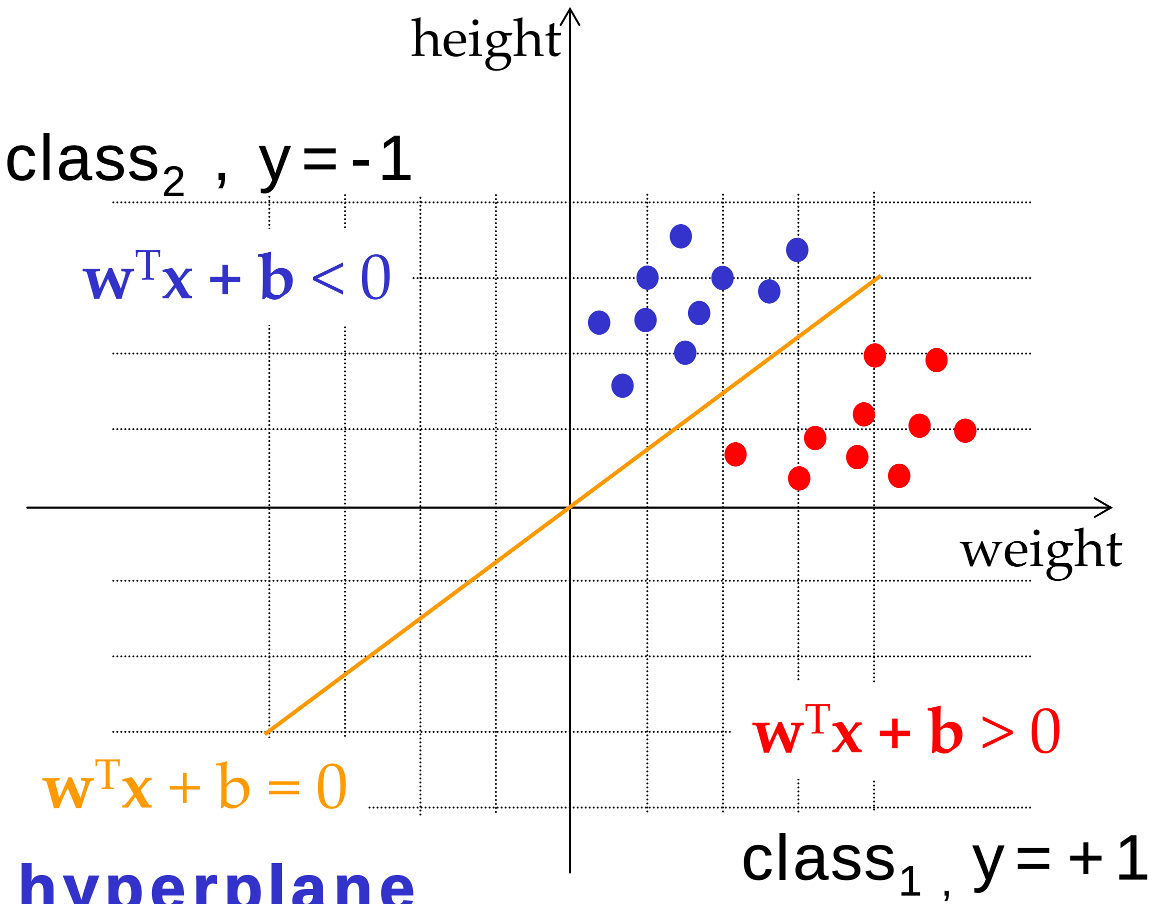
$\mathbf{w} = (3\ {-4})^T$

$\mathbf{w}^T\mathbf{x} = 0$

$3 \times 2 - 4 \times 3 < 0$

$3 \times 4 - 4 \times 2 > 0$

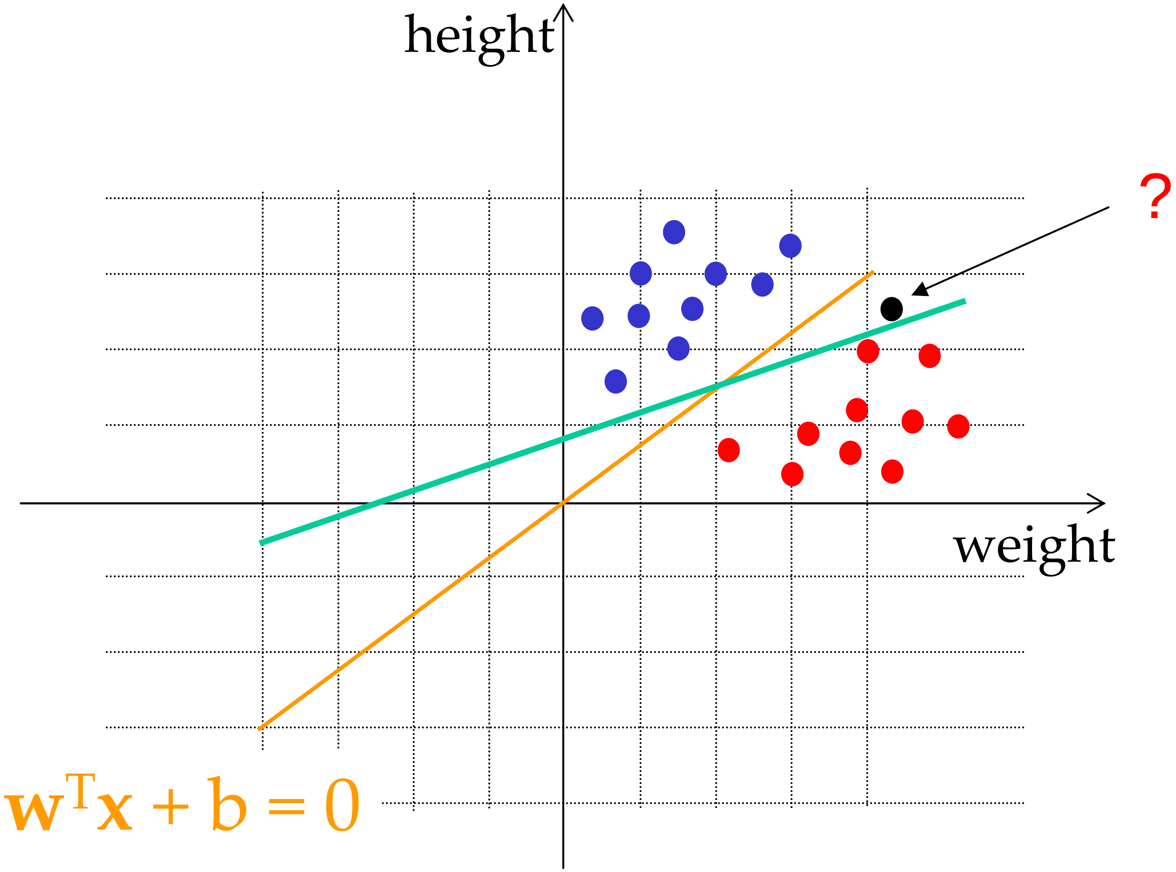
Linear Classification



hyperplane

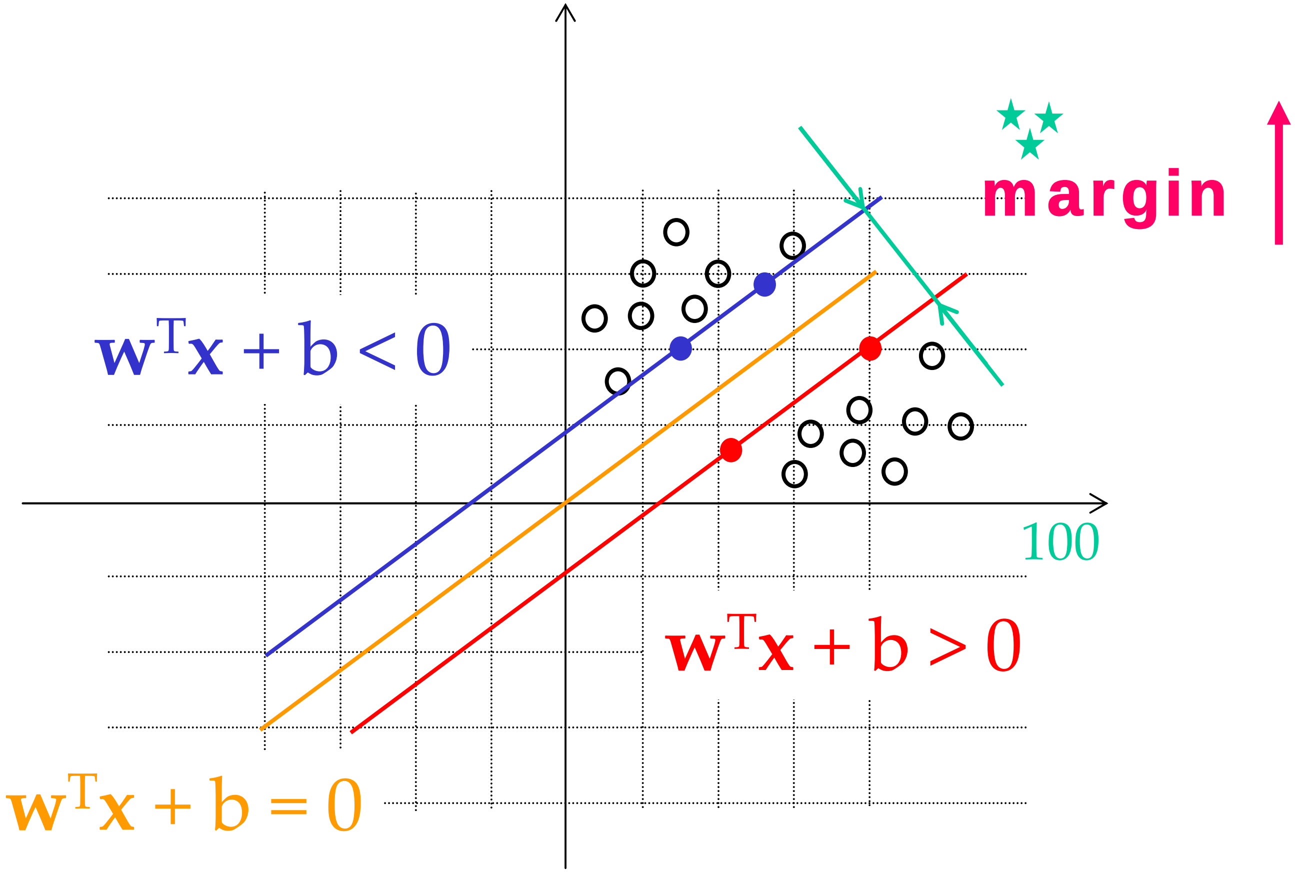
decision surface

Optimal Hyperplane



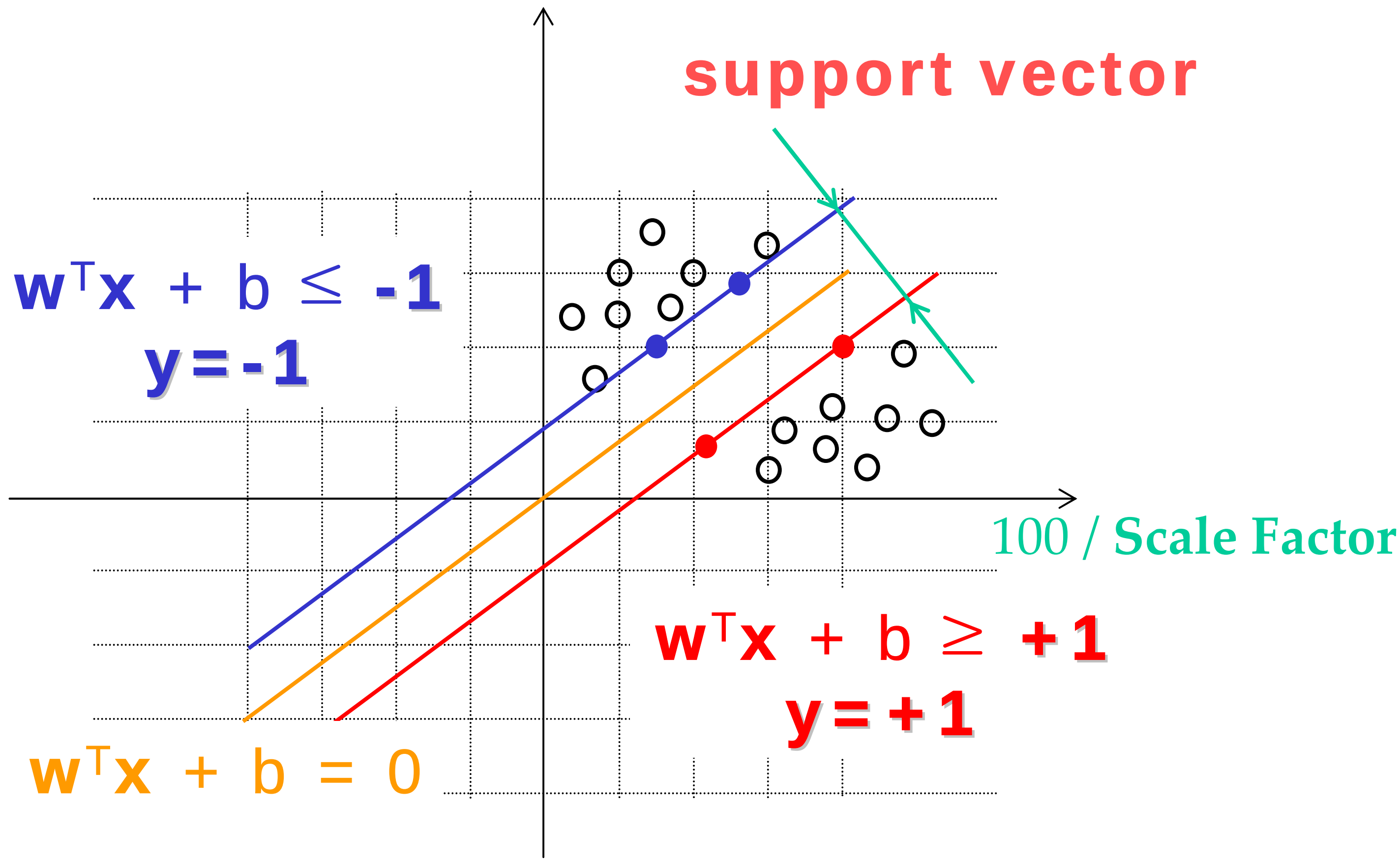
$\mathbf{w}^T\mathbf{x} + b = 0$

Optimal Hyperplane



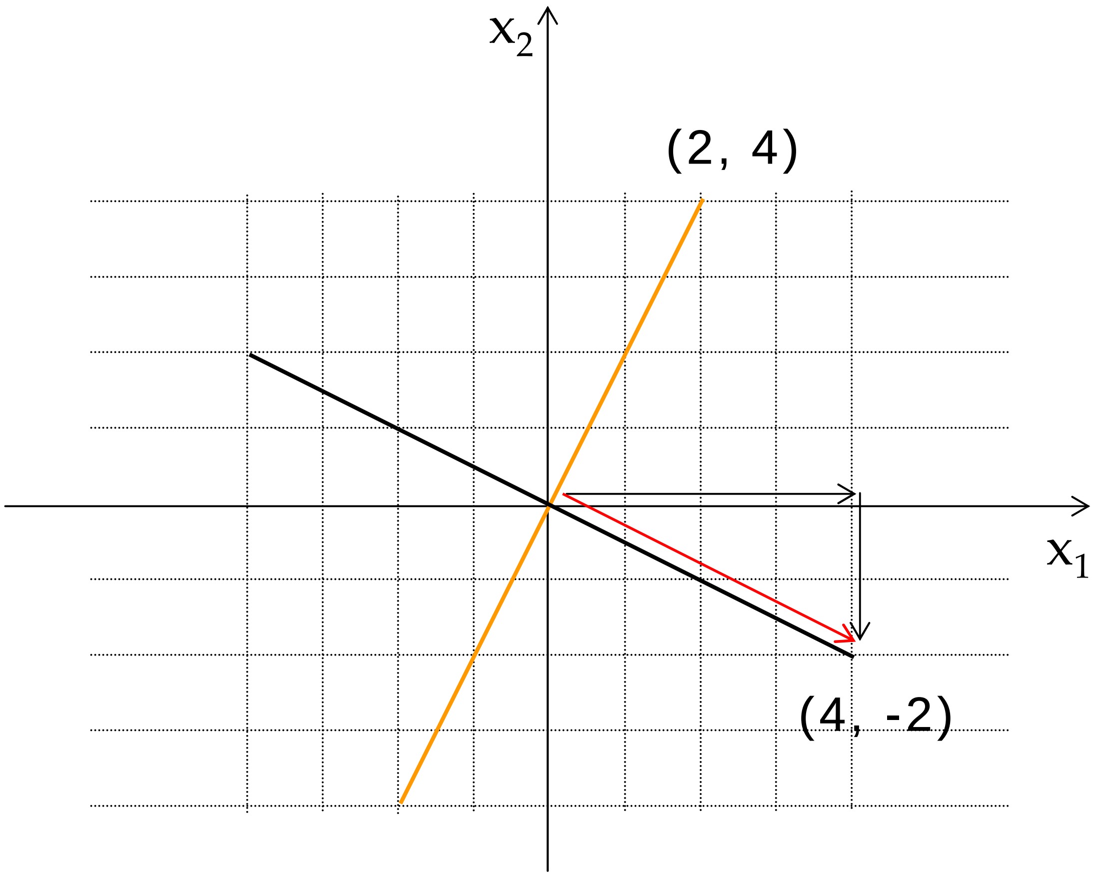
optimal hyperplane

Canonical Hyperplane



optimal hyperplane

Normal Vector



$x_2 = 4/2x_1$

$2x_2 = 4x_1$

$4x_1 - 2x_2 = 0$

$(4\ {-2})\,(x_1\ x_2)^T = 0$

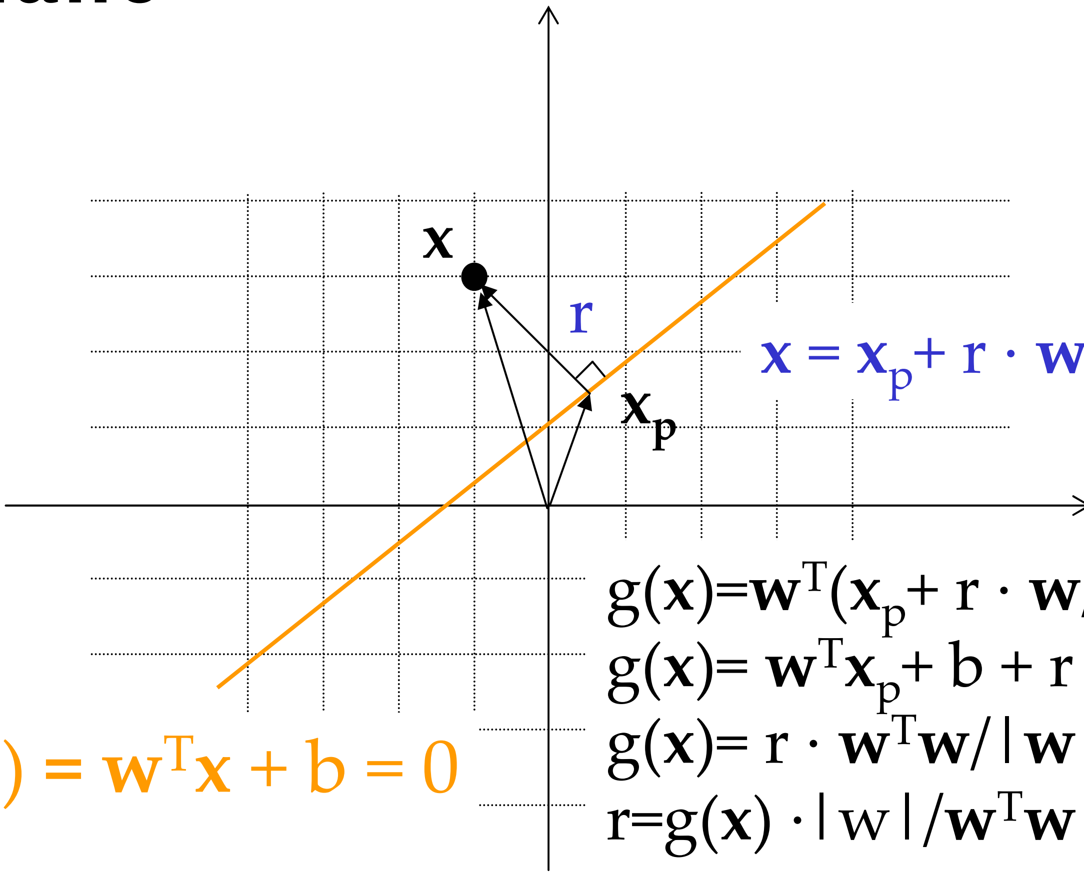
$\mathbf{w}^T\mathbf{x} = 0$

$\mathbf{w} = (4\ {-2})^T$

$(2\ 4)(4\ {-2})^T = 8 - 8 = 0$

$\mathbf{x}\cdot\mathbf{y} = \langle \mathbf{x}\ \mathbf{y} \rangle = \mathbf{x}^T\mathbf{y} = x_1y_1 + x_2y_2 = |\mathbf{x}||\mathbf{y}|\cos\Theta$

Distance from a Sample to the Optimal Hyperplane



$$\mathbf{x} = \mathbf{x}_p + r \cdot \mathbf{w}/|\mathbf{w}|$$

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = 0$$

$$g(\mathbf{x}) = \mathbf{w}^T(\mathbf{x}_p + r \cdot \mathbf{w}/|\mathbf{w}|) + b$$

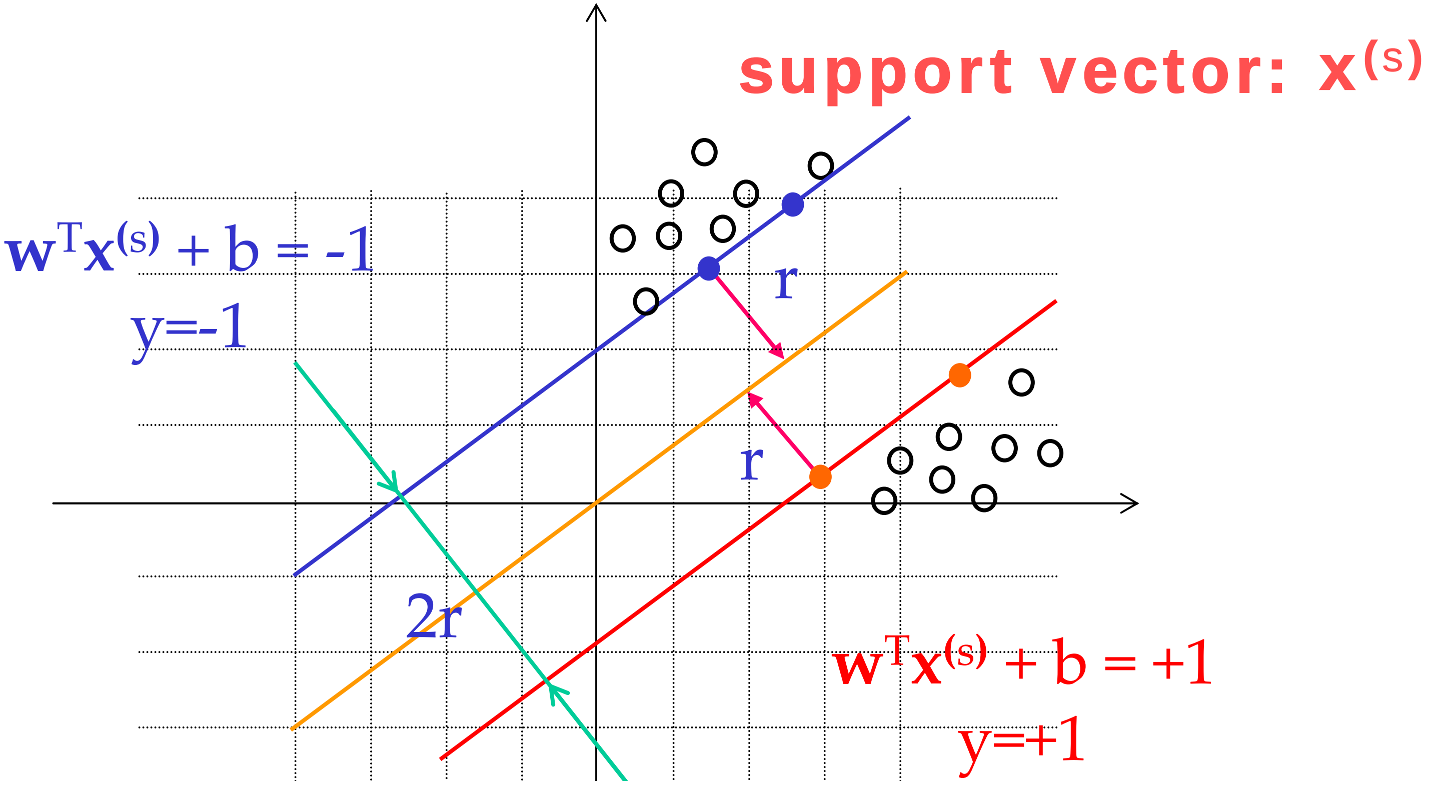
$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x}_p + b + r \cdot \mathbf{w}^T\mathbf{w}/|\mathbf{w}|$$

$$g(\mathbf{x}) = r \cdot \mathbf{w}^T\mathbf{w}/|\mathbf{w}|$$

$$r = g(\mathbf{x}) \cdot |w|/\mathbf{w}^T\mathbf{w}$$

$$r = g(\mathbf{x}) / |\mathbf{w}|$$

Margin of Separation



$r = g(\mathbf{x}^{(s)})/|\mathbf{w}|$

where, $g(\mathbf{x}^{(s)}) = \mathbf{w}^T\mathbf{x}^{(s)} + b = \pm 1$ for $y^{(s)} = \pm 1$

margin: $2r = 2 / |\mathbf{w}|$ $\quad \left| \frac{1}{\|\mathbf{w}\|} - \frac{-1}{\|\mathbf{w}\|} \right| = \frac{2}{\|\mathbf{w}\|}$

Finding the Optimal Hyperplane

margin: $\left| \frac{1}{\|\mathbf{w}\|} - \frac{-1}{\|\mathbf{w}\|} \right| = \frac{2}{\|\mathbf{w}\|}$

$\mathbf{w}^T\mathbf{x} + b \leq -1, \quad y=-1$

$\mathbf{w}^T\mathbf{x} + b \geq +1, \quad y=+1$

Constrained Optimization Problem

(convex) objective/cost function

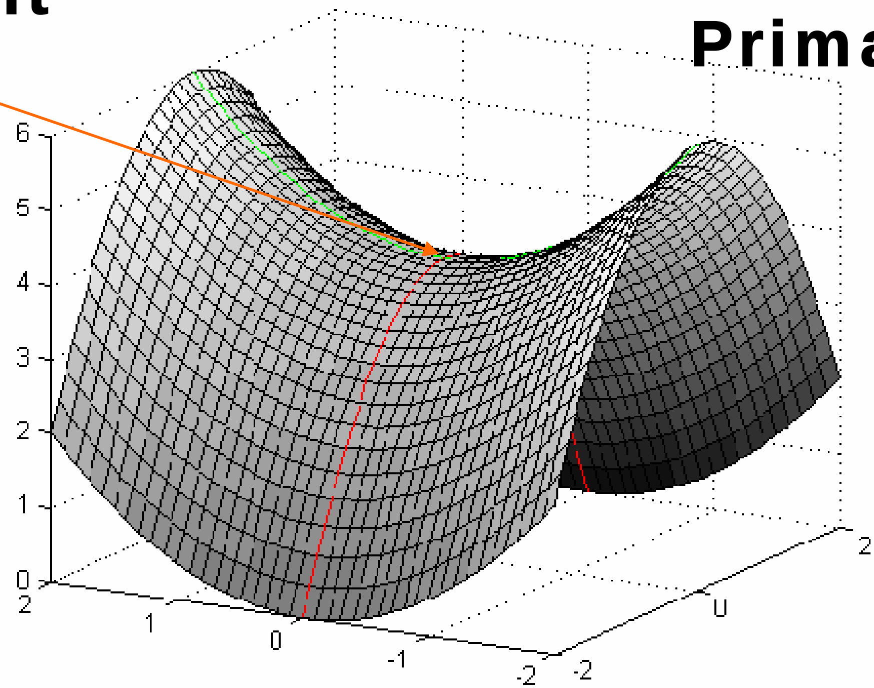
$$\begin{aligned} &\text{minimize} && \frac{1}{2}\mathbf{w}^\top \mathbf{w} \\ &\text{subject to} && y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \qquad i = 1, 2, \ldots, n \end{aligned}$$

(linear) constraints

Convex Quadratic Problem

Saddle Point

minimize
Primal problem



maximize
Dual problem

Lagrange Multipliers Method

cost function: $f(\mathbf{x})$

constraint function: $g(\mathbf{x})=0$

$$\nabla f(\mathbf{x}) = \lambda \nabla g(\mathbf{x})$$

tangent (line) ~ gradient = normal

convex / concave

$$\nabla f(\mathbf{x}) = 0, \; g(\mathbf{x})=0$$

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x})$$

Lagrange function   Lagrange multiplier

A Brief Overview of Optimization Theory

(2001년 추계 CVPR 튜토리얼)

- **Theorem:** $f \in C^1$ has a min. at x^* $\Rightarrow$ $\dfrac{\partial f}{\partial x}(x^*) = 0$.

  This condition, together with convexity of f, is also a sufficient condition.

- Example 1: $\min.\ f(x) = \dfrac{1}{2}(x_1^2 + x_2^2)$

  Solution:

$$\frac{\partial f}{\partial x} = 0 \Rightarrow \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2}\right] = [x_1 \ \ x_2] = 0 \qquad \therefore \ x^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

- In a constrained min. problem,

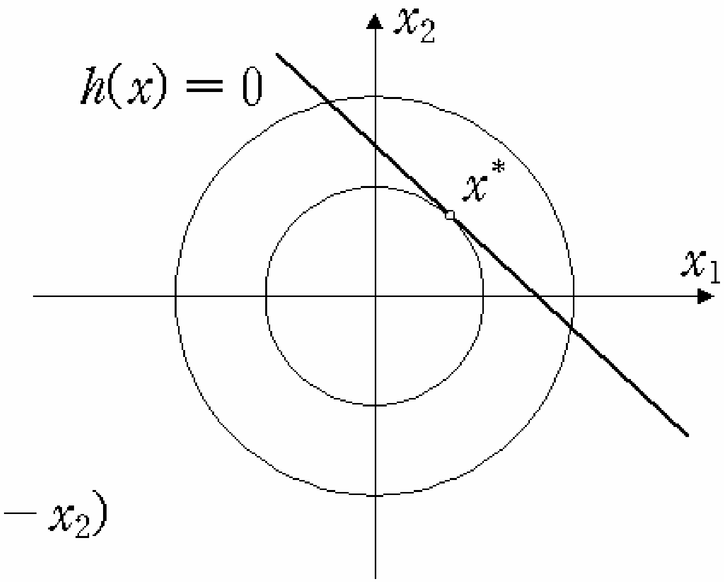
$$f \in C^1 \text{ has a min. at } x^* \not\Rightarrow \frac{\partial f}{\partial x}(x^*) = 0$$

A Brief Overview of Optimization Theory

- Example 2:

$$\min.\ f(x) = \frac{1}{2}(x_1^2 + x_2^2)$$

$$\text{s.t.}\quad h(x) = 1 - x_1 - x_2 = 0$$



Solution: Define the Lagrange function

$$L(x,\lambda) = f(x) + \lambda h(x)$$
$$= \frac{1}{2}(x_1^2 + x_2^2) + \lambda(1 - x_1 - x_2)$$

$$\frac{\partial L}{\partial x} = 0 \Rightarrow [x_1 - \lambda \ \ x_2 - \lambda] = 0.\ \therefore\ x_1 = x_2 = \lambda.$$

$$\frac{\partial L}{\partial \lambda} = 0 \Rightarrow 1 - x_1 - x_2 = 0.\ \therefore\ 1 - 2\lambda = 0.\ \therefore\ \lambda = \frac{1}{2}.$$

$$\therefore\ x^* = \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}$$

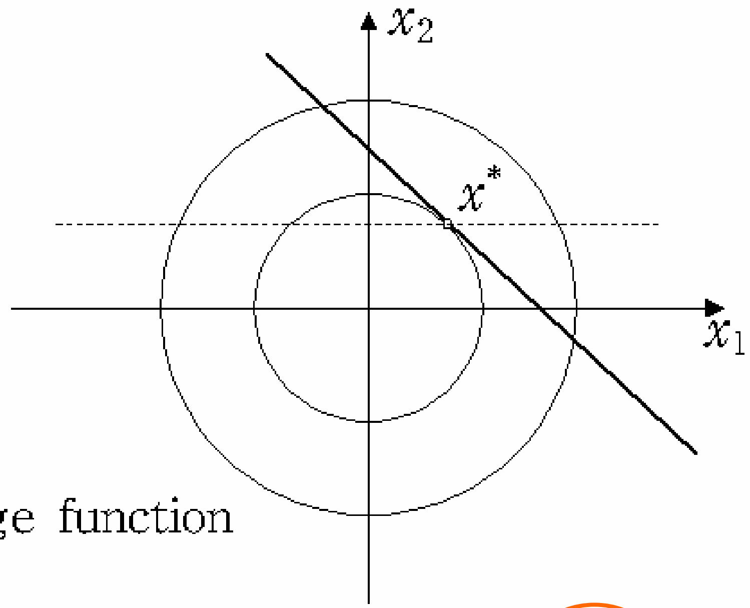
A Brief Overview of Optimization Theory

- Example 3:

$$\min.\ f(x) = \frac{1}{2}(x_1^2 + x_2^2)$$

$$\text{s.t.}\quad h(x) = 1 - x_1 - x_2 = 0\,,$$

$$g(x) = \frac{3}{4} - x_2 \le 0$$



Solution: Define the generalized Lagrange function

$$L(x, \lambda, \alpha) \triangleq f + \lambda h + \alpha g$$

$$= \frac{1}{2}(x_1^2 + x_2^2) + \lambda(1 - x_1 - x_2) + \alpha(\frac{3}{4} - x_2), \quad \alpha \ge 0$$

$$\frac{\partial L}{\partial x} = 0 \Rightarrow [x_1 - \lambda,\ x_2 - \lambda - \alpha] = 0.\ \therefore\ x_1 = \lambda\,,\ \ x_2 = \lambda + \alpha$$

$$\frac{\partial L}{\partial \lambda} = 0 \Rightarrow 1 - x_1 - x_2 = 0.\ \therefore\ 2\lambda + \alpha = 1$$

Also, $\alpha \geq 0$ and $\frac{3}{4} - x_2 \leq 0$.

One more condition is needed to solve the problem.

Karush

→ The Kuhn-Tucker complementarity condition

$$\alpha\left(\frac{3}{4} - x_2\right) = 0 \quad \text{i.e. } \alpha = 0 \text{ or } x_2 = \frac{3}{4}$$

① If $\alpha = 0$, then $\lambda = \frac{1}{2}$; thus $x_1 = x_2 = \frac{1}{2}$ ╳ $(\because x_2 \geq \frac{3}{4})$

② If $x_2 = \frac{3}{4}$, then $\begin{cases} \lambda + \alpha = \frac{3}{4} \\ 2\lambda + \alpha = 1 \end{cases} \therefore \begin{cases} \lambda = \frac{1}{4}, \ \alpha = \frac{1}{2} \\ x_1 = \frac{1}{4}, \ x_2 = \frac{3}{4} \end{cases}$

$$\therefore x^* = \begin{bmatrix} 1/4 \\ 3/4 \end{bmatrix}$$

A Brief Overview of Optimization Theory

<u>**Theorem**</u> (Kuhn-Tucker Theorem)

Given an opt. prob. with convex domain $\Omega \subseteq R^n$

$$\min \ f(x), \ x \in \Omega \ (x \text{ is primal variable})$$

$$\text{s.t.} \ g_i(x) \le 0, \ i = 1, \cdots, k \qquad h_j(x) = 0, \ j = 1, \cdots, m$$

primal opt. prob.

with $f \in C^1$ convex, and g_i, h_j affine, the following are necessary and sufficient conditions for a point $x^* \in \Omega$ to be an opt.:

$$\text{For } L(x, \alpha, \lambda) \triangleq f(x) + \sum_{i=1}^{k} \alpha_i g_i(x) + \sum_{j=1}^{m} \lambda_j h_j(x) = f + \alpha^T g + \lambda^T h,$$

$$\exists \ \alpha^* \text{ and } \lambda^* \text{ s.t. } \frac{\partial L}{\partial x}(x^*, \alpha^*, \lambda^*) = 0, \ \frac{\partial L}{\partial \lambda}(x^*, \alpha^*, \lambda^*) = 0$$

$$g_i(x^*) \le 0, \ \alpha_i^* \ge 0 \ \text{for } i = 1, \cdots, k,$$

$$\text{and } \underline{\alpha_i^* g_i(x^*) = 0, \ i = 1, \cdots, k}$$

Lagrange Function for the Optimal Hyperplane (Primal Problem)

$$\underset{w,b}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to} \quad y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1, \quad i = 1, \ldots, n.$$

$$L_P(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1]$$

Solution:

$$\frac{dLp}{d\mathbf{w}} = 0 \qquad \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

$$\frac{dLp}{db} = 0 \qquad 0 = \sum_{i=1}^{n} \alpha_i y_i \qquad (\alpha_i \geq 0)$$

KKT condition: $\alpha_i = 0 \text{ unless } y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$

Remark on Support Vector

KKT Condition $\alpha_i[1 - y_i(w^T x_i + b)] = 0$

If $\alpha_i \neq 0$ then $y_i(w^T x_i + b) = 1$

→ x_i is **support vector** All $\mathbf{x}_i$ for which $\alpha_i > 0$

$y_i(w^T x_i + b) \neq 1$

→ $\alpha_i = 0$ x_i is **not support vector**

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

w is only related to support vectors, $\mathbf{x}_i$

Lagrange Function for the Optimal Hyperplane (Dual Problem)

$$\max. \quad L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \quad ; \quad \alpha_i \geq 0$$

$$-\frac{1}{2} \alpha^T Q \alpha + \alpha^T \underline{1}$$

Optimizing L_D only depends on the input patterns in the form of a set of dot product $x^T x$

(+) Not depend on the dimension of the input pattern

(+) Can replace dot product with Kernel

$$L_P(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1]$$

$$= \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{w}^T \boldsymbol{x}_i - b\sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \alpha_i$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i^T x_j}$$

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i^T x_j} \qquad \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i^T x_j} = -\frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Q} \boldsymbol{\alpha} + \boldsymbol{\alpha}^T 1$$

n × n (> 1000)

Large Scale Quadratic Problem

Support Vector Machine Classifier for Linearly Separable Case

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i^\top \mathbf{x} + b^*\right)$$

optimal weight

$$\boldsymbol{w}^* = \sum_{i=1}^{n} \alpha_i^* y_i \boldsymbol{x}_i$$

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$$

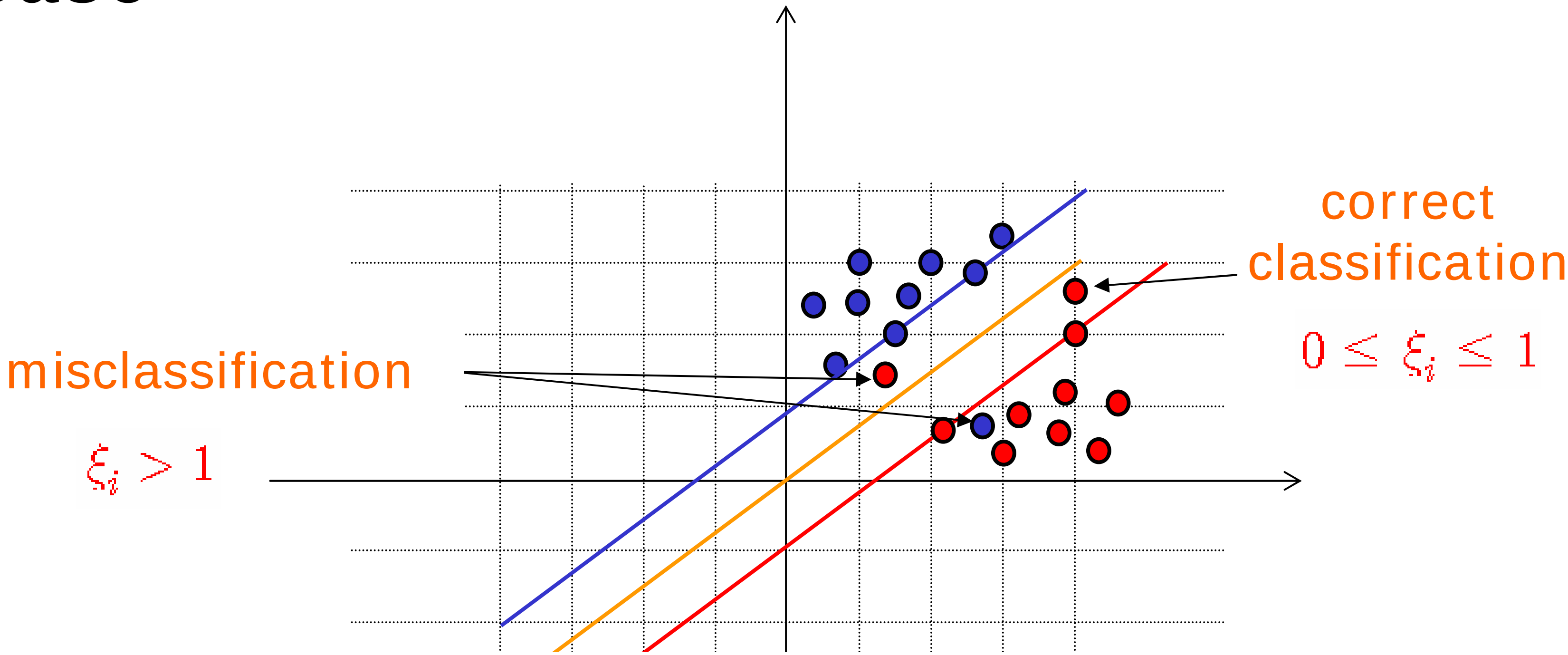
$$b_{y=+1} = 1 - \boldsymbol{w}^{*T} \boldsymbol{x}^{(s)}$$

$$b_{y=-1} = -1 - \boldsymbol{w}^{*T} \boldsymbol{x}^{(s)}$$

optimal bias

$$b^* = \frac{1}{2}(b_{y=+1} + b_{y=-1})$$

Optimal Hyperplane for Non-Separable Case



impossible to find a separating hyperplane

give them up as errors while minimizing the probabilities of classification error averaged over the training set

Soft Marin Technique

Problem: Can't satisfy $y_i[\mathbf{w}^\top \mathbf{x}_i + b] \geq 1$ for all i

Adopting Slack Variable

$$y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\begin{aligned} \mathbf{w}^\top \mathbf{x}_i + b &\geq +1 - \xi_i && \text{for } y_i = +1, \\ \mathbf{w}^\top \mathbf{x}_i + b &\leq -1 + \xi_i && \text{for } y_i = -1, \\ \xi_i &\geq 0 && k = 1, 2, \ldots, n \end{aligned}$$

Minimizing Errors

$\xi_i > 1$ $\quad \sum_{i=1}^{n} I(\xi_i > 1) = \#\ \text{errors}$ $\quad$ min. $\sum_{i=1}^{n} I(\xi_i > 1)$

For QP, replace $I(\xi_i > 1)$ by ξ_i

Lagrange Function for Soft Margin Tech.

$$\underset{w,b,\xi}{\text{minimize}} \quad L_P(\mathbf{w}, \xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$

$$\text{subject to} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

penalize errors

C

penalize complexity

$$L(\mathbf{w}, b, \alpha, \xi) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left[y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) - 1 + \xi_i\right] - \sum_{i=1}^{m} r_i\xi_i$$

★ $\alpha_i \geq 0$ and $r_i \geq 0$

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i = 0$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{m}\alpha_i y_i = 0$$

★ $$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - r_i = 0$$

Dual Problem for Soft Margin Tech.

$$\text{max.} \quad L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \quad ; \quad \boxed{0 \le \alpha_i \le C} \ \star$$

$$\alpha_i \ge 0 \text{ and } r_i \ge 0$$

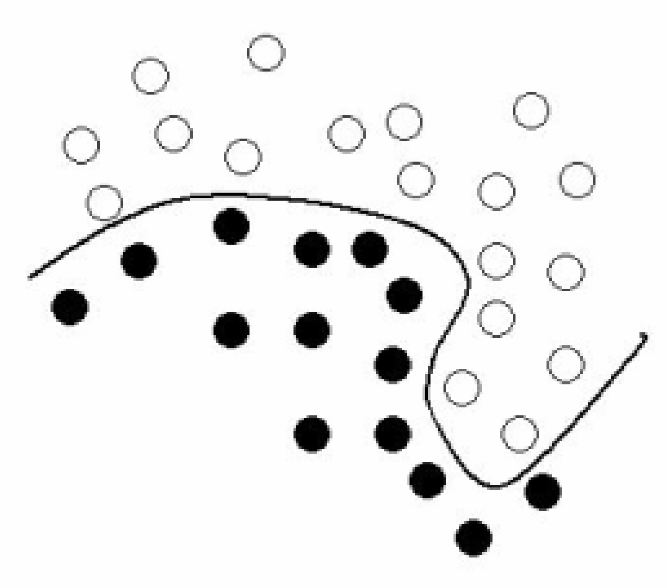
$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - r_i = 0$$

$0 < \alpha_i < C$ ***non-bound*** pattern

$\alpha_i = 0 \text{ or } \alpha_i = C$ ***bound*** pattern

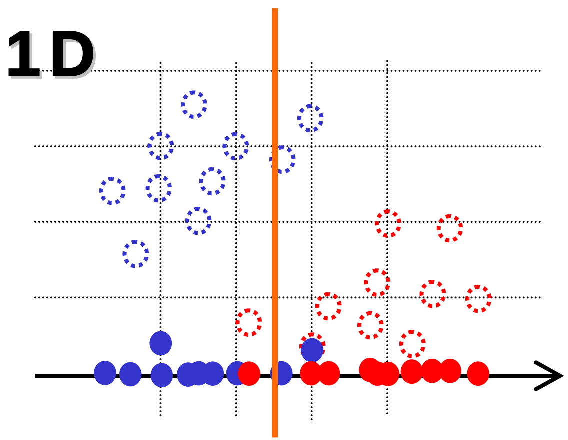
The optimal value of *C* is determined experimentally, it cannot be readily related to the characteristics of the dataset or model

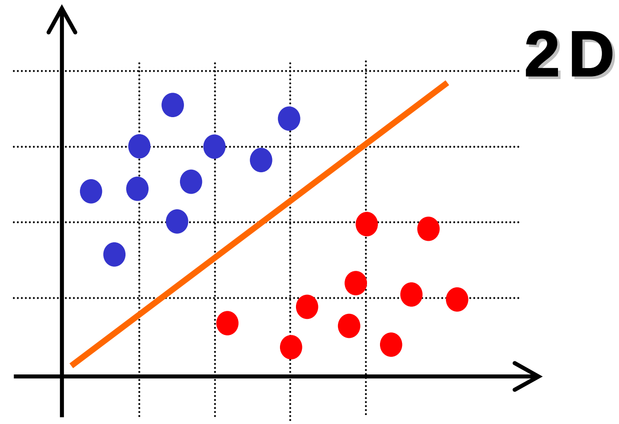
Nonlinear Support Vector Machines



What if the problem is not linear ?

Feature Space





linearly non-separable on 1D (**low-dimensional input space**)

linearly separable on 2D (**high-dimensional feature space**)

$$\mathbf{x}_i \rightarrow \phi(\mathbf{x}_i)$$

$$\mathbf{x}_i \cdot \mathbf{x}_j \rightarrow \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$$

Dual Problem and Classifier in the Feature Space (not feasible)

$$\text{max.} \quad L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$$

$\leftarrow \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \quad ; \quad 0 \leq \alpha_i \leq C$$

$\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x})$

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i^\top \mathbf{x} + b^* \right)$$

- The feature space $\phi(\mathbf{x}_i)$ can be huge or infinite !
- The phi feature has the form of **inner product**

Kernel

Kernel: a function k that takes 2 variables and computes a scalar value (a kind of **similarity**)

$$k(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}))$$

Kernel Matrix: m ×m matrix **K** with elements $K_{ij} = k(x_i, x_j)$.

Standard Kernels

Polynomial Kernel	$k(x, x_i) = (x^T x_i + 1)^d$
Radial Basis Function Kernel	$k(x, x_i) = \exp(-\parallel x - x_i \parallel^2 / 2\sigma^2)$
Sigmoid Kernel	$k(x, x_i) = \tanh(\beta_0 x^T x_i + \beta_1)$

Mercer's Condition

Valid Kernel functions should satisfy Mercer's Condition

For any g(x) for which: $\int g(x)^2 dx < \infty$

It must be the case that: $\int K(x, x')g(x)g(x')dxdx' \geq 0$

A criteria is that the kernel should be positive semi-definite

Theorem: If a kernel is positive semi-definite i.e.:

$$\sum_{i,j} K(x_i, x_j)c_i c_j \geq 0$$

$\{c_1, \ldots, c_n\}$ are real numbers

Then, there exists a function $\phi(x)$ defining an inner product of possibly higher dimension i.e.:

$$K(x, y) = \phi(x) \cdot \phi(y)$$

Dual Problem and Classifier with Kernel (Generalized Inner Product SVM)

$$\text{max.} \quad L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \leftarrow k(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \quad ; \quad 0 \le \alpha_i \le C$$

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{n} \alpha_i^* y_i \mathbf{x}_i^\top \mathbf{x} + b^* \right) \quad \leftarrow k(\mathbf{x}, \mathbf{x}_i)$$

- **Kernel Trick works without the mapping**

$$\mathbf{x}_i \rightarrow \phi(\mathbf{x}_i)$$

Example: XOR Problem (small scale QP problem)

- Training set: $\{ (\underset{\mathbf{x}_1}{-1\ -1}; -1), (\underset{\mathbf{x}_2}{-1\ +1}; +1), (\underset{\mathbf{x}_3}{+1\ -1}; +1), (\underset{\mathbf{x}_4}{+1\ +1}; -1)\}$
- Let kernel $k(\mathbf{x}, \mathbf{x}_i)=(1+\mathbf{x}^T\mathbf{x}_i)^2$, $\mathbf{x}=(x_1, x_2)^T$, $\mathbf{x}_i=(x_{i1}, x_{i2})^T$
- Then $k(\mathbf{x}, \mathbf{x}_i) = (1+\mathbf{x}^T\mathbf{x}_i)^2 = (1+x_1x_{i1}+x_2x_{i2})^2$

$$= 1 + x^2{}_1x^2{}_{i1} + 2x_1x_2x_{i1}x_{i2} + x^2{}_2x^2{}_{i2} + 2x_1x_{i1} + 2_{x2}x_{i2}$$

- A Mapping: $\varphi(\mathbf{x})=(1,\ x^2{}_1, x^2{}_2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2)^T$
- Kernel Matrix:

$$\Phi = \begin{bmatrix} 1 & 1 & 1 & -\sqrt{2} & -\sqrt{2} & \sqrt{2} \\ 1 & 1 & 1 & -\sqrt{2} & \sqrt{2} & -\sqrt{2} \\ 1 & 1 & 0 & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 1 & 1 & 1 & \sqrt{2} & \sqrt{2} & \sqrt{2} \end{bmatrix}$$

$$K = \begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}$$

4 x 4

Example: XOR Problem

- Dual Problem:

  $L_D(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - (9\alpha_1^2 - 2\alpha_1\alpha_2 - 2\alpha_1\alpha_3 + 2\alpha_1\alpha_4 + 9\alpha_2^2 + 2\alpha_2\alpha_3 - 2\alpha_2\alpha_4 + 9\alpha_3^2 - 2\alpha_3\alpha_4 + 9\alpha_4^2)/2$

- Optimizing L_D: $9\alpha_1 - 2\alpha_2 - \alpha_3 + \alpha_4 = 1$, $-\alpha_1 + 9\alpha_2 + \alpha_3 - \alpha_4 = 1$
  $-\alpha_1 + \alpha_2 + 9\alpha_3 - \alpha_4 = 1$, $\alpha_1 - \alpha_2 - \alpha_3 + 9\alpha_4 = 1$

- Optimal Lagrange multipliers: $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 1/8 > 0$

  All the samples are support vectors

- Optimal **w**:

  $\mathbf{w} = \sum \alpha_i y_i \varphi(\mathbf{x}_i) = (1/8)(-1)\varphi(\mathbf{x}_1) + (1/8)(+1)\varphi(\mathbf{x}_2) + (1/8)(+1)\varphi(\mathbf{x}_3) + (1/8)(-1)\varphi(\mathbf{x}_4) = (0\ \ 0\ \ 0\ \ 0\ \ 0\ \ -1\sqrt{2})^T$

- Optimal Hyperplane:

$$f(\mathbf{x}) = \mathrm{sgn}(\mathbf{w}^T\varphi(\mathbf{x})) = \mathrm{sgn}(-x_1x_2)$$

Sequential Minimal Optimization (Platt '98)

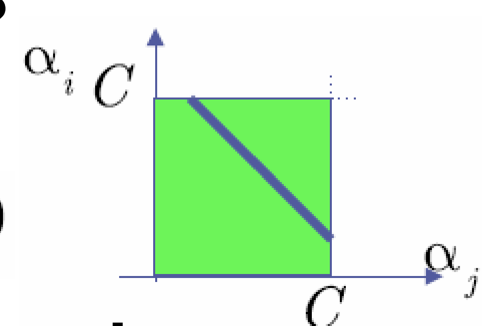
$$\max_{\alpha} \sum_i \alpha_i - \tfrac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j k\left(x_i, x_j\right) = -\frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Q}\boldsymbol{\alpha} + \boldsymbol{\alpha}^T 1$$

$$\text{subject to } \sum_i \alpha_i y_i = 0 \;\; \& \;\; \alpha_i \in \left[0, C\right]$$

In case of large scale QP problem

- **divide a large QP into a series of smaller QP sub-problems and optimize them sequentially**
- **What is the smallest working set ?**
- **Update just 2 Lagrange multipliers at a time**

$$\sum_i \alpha_i y_i = 0$$



- **Updating subset of variables while others fixed will also converge globally**

Sequential Minimal Optimization

- **Write dual prob. as a function of just 2 alphas**

$$L_D \propto \alpha_i + \alpha_j - \tfrac{1}{2}\left(K_{ii}\alpha_i^2 + 2K_{ij}\alpha_i\alpha_j + K_{jj}\alpha_j^2\right) - h_i\alpha_i - h_j\alpha_j$$

$$subject\,to:\ y_i\alpha_i + y_j\alpha_j + \sum_{t\neq i,j} y_t\alpha_t = 0 \quad and\ \alpha_i, \alpha_j \in [0, C]$$

- **Update Rules**

- no numerical part
- memory for buffering E

$$S = y_i y_j$$

$$L = \max\left(0, \alpha_j + S\alpha_i - \tfrac{1}{2}(S+1)C\right) \qquad E1 = \sum_t \alpha_t y_t k(x_i, x_t) + b - y_i$$

$$H = \min\left(C, \alpha_j + S\alpha_i - \tfrac{1}{2}(S-1)C\right) \qquad E2 = \sum_t \alpha_t y_t k(x_j, x_t) + b - y_j$$

$$\alpha_j^{NEW} = \alpha_j + \frac{y_j(E1 - E2)}{k(x_i, x_i) + k(x_j, x_j) - 2k(x_i, x_j)} \quad clipped\ inside\ [L, H]$$

$$\alpha_i^{NEW} = \alpha_i + S\left(\alpha_j - \alpha_j^{NEW}\right)$$

Link to Statistical Learning Theory (Vapnik '95)

Learn *f* from training set is to minimize the following:

$$R[f] = \int \frac{1}{2}|f(\mathbf{x}) - y|\,dP(\mathbf{x}, y) \quad \text{Expected Risk}$$

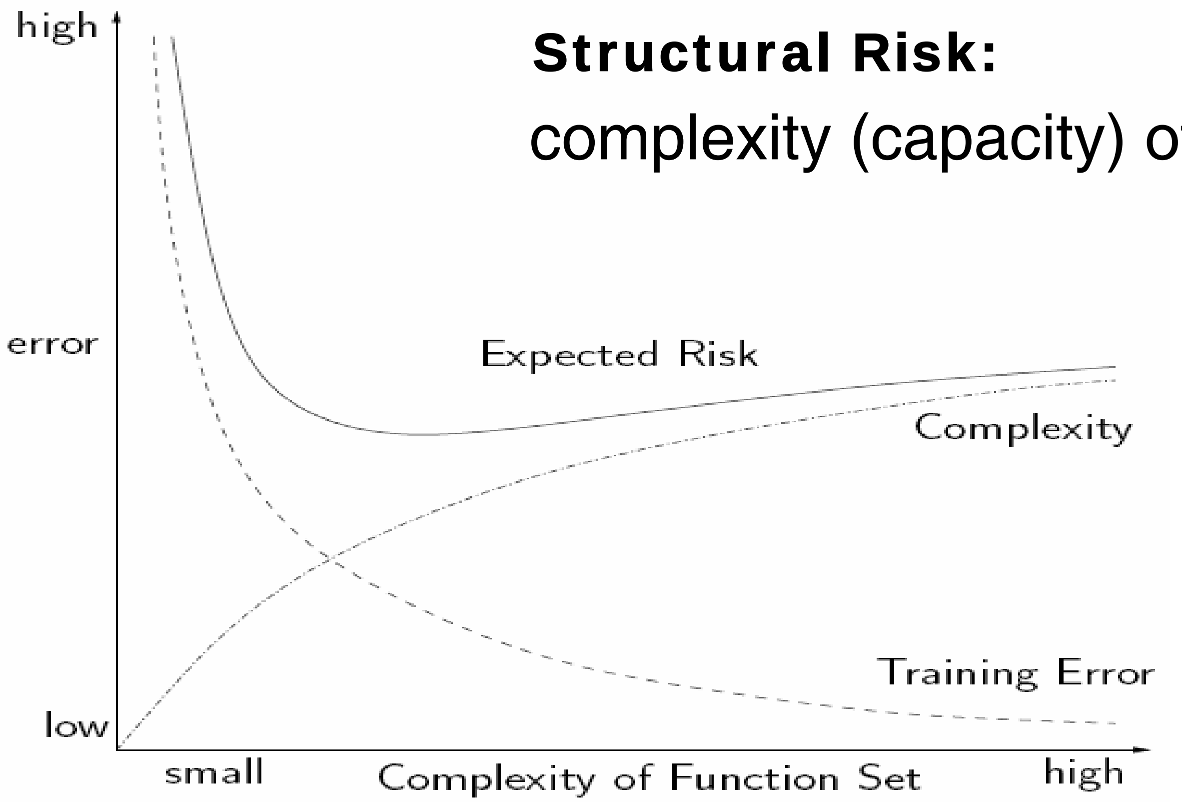
$y \in \{\pm 1\}$; $P(\mathbf{x}, y)$: Unknown

Minimize instead the average risk over the training set:

$$R_{\mathrm{emp}}[f] = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{2}|f(\mathbf{x}_i) - y_i| \quad \text{Empirical Risk}$$

A Risk Bound

$$R[f] \leq \underbrace{R_{\text{emp}}[f]}_{\text{minimize}} + \underbrace{\sqrt{\frac{h(\ln \frac{2m}{h} + 1) - \ln(\eta/4)}{m}}}_{\text{minimize}}$$

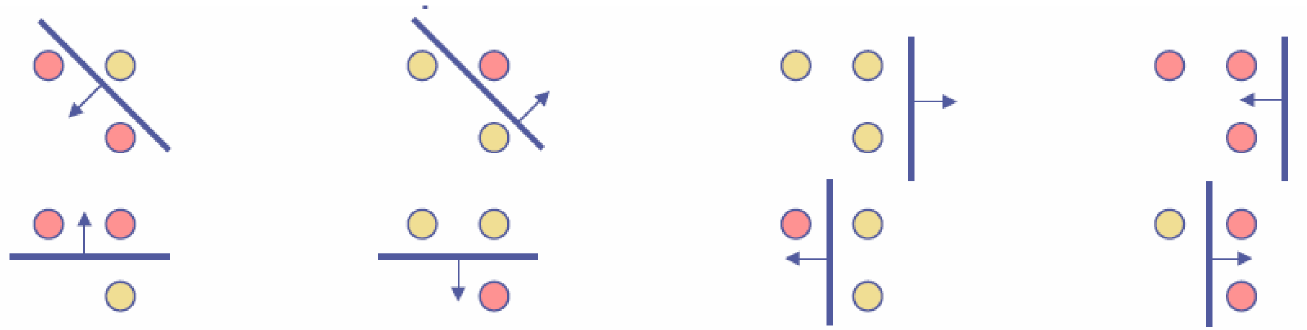


Structural Risk:

complexity (capacity) of func. set

Vapnik-Chervonenkis (VC) Dimension

- **Maximum number of points that can be labeled in all possible way**
- **VC dimension of linear classifiers in N-dimensions is $h=N+1$**



Lines(dichotomies) can shatter 3 points in 2d

- **Measure of Complexity of Function Set**

Minimizing VC dim. → Minimizing Complexity

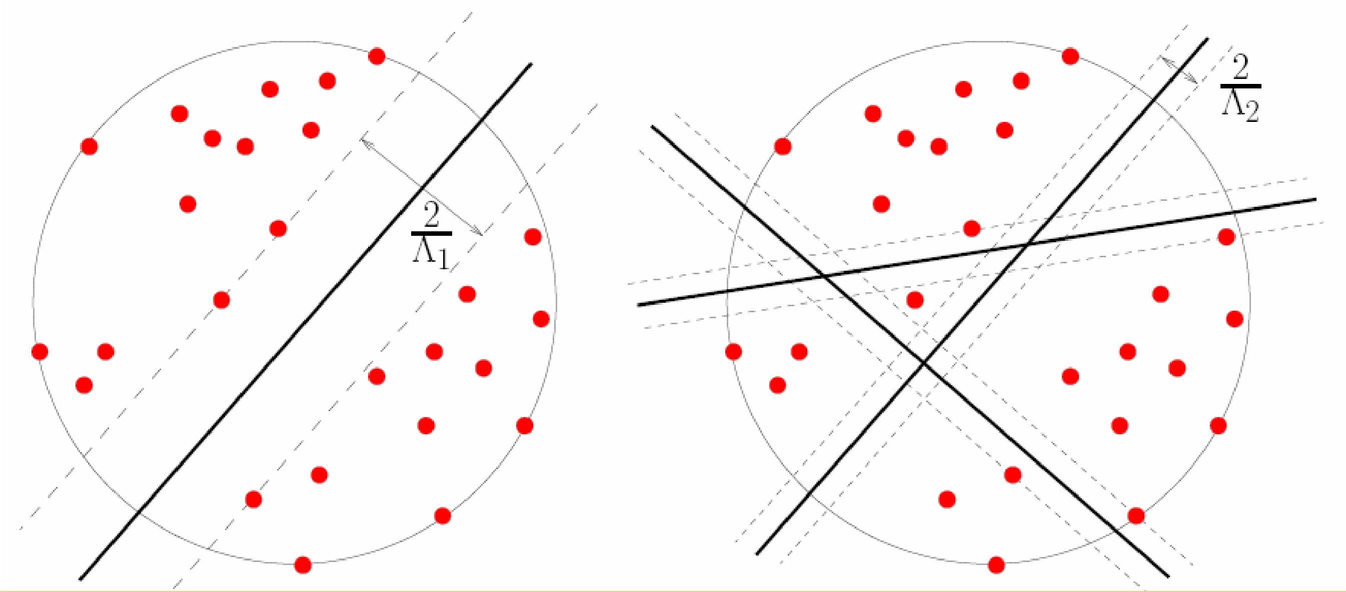
VC Dimension of Marin Hyperplanes

- **VC dimension satisfies the following:**

$$h \leq \min(R^2\Lambda^2 + 1, N + 1)$$

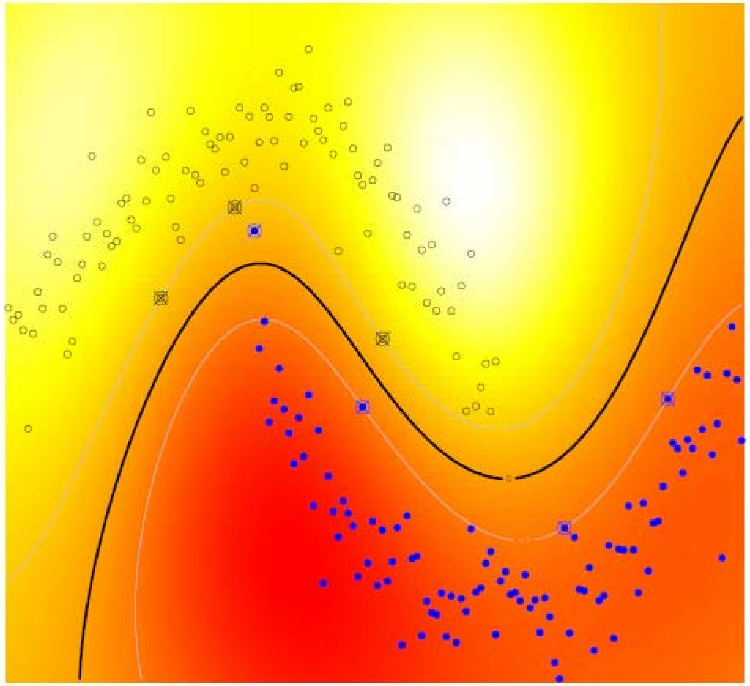
R is the smallest sphere containing a set of points

$$\|\mathbf{w}\| \leq \Lambda$$

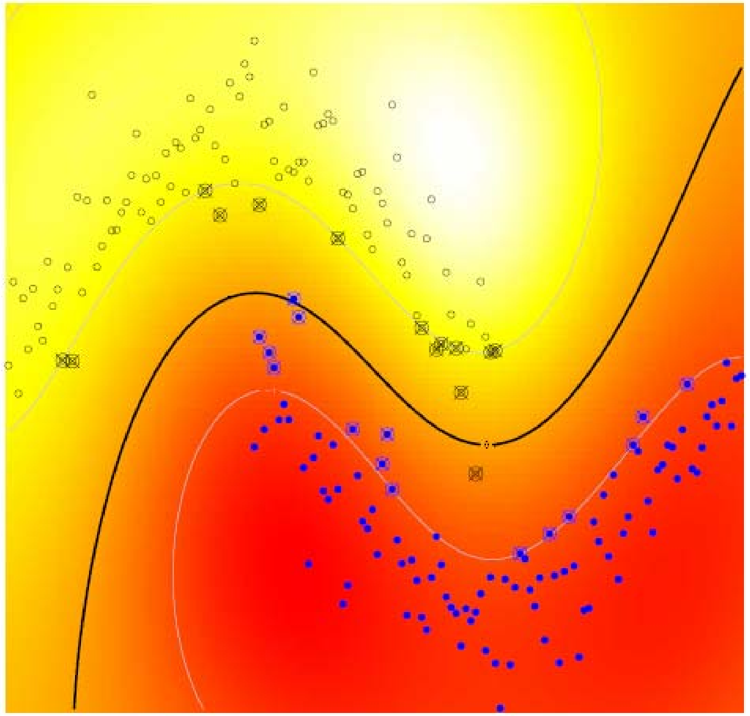


Maximizing Margin → Minimizing VC dim.

Results for Gaussian Kernel

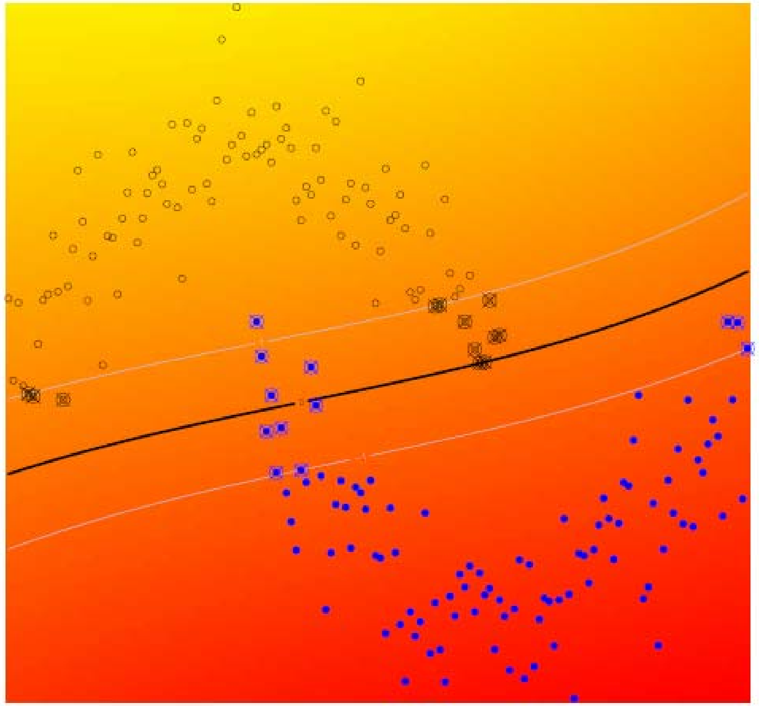


$\sigma = 0.5,\ C = 50$

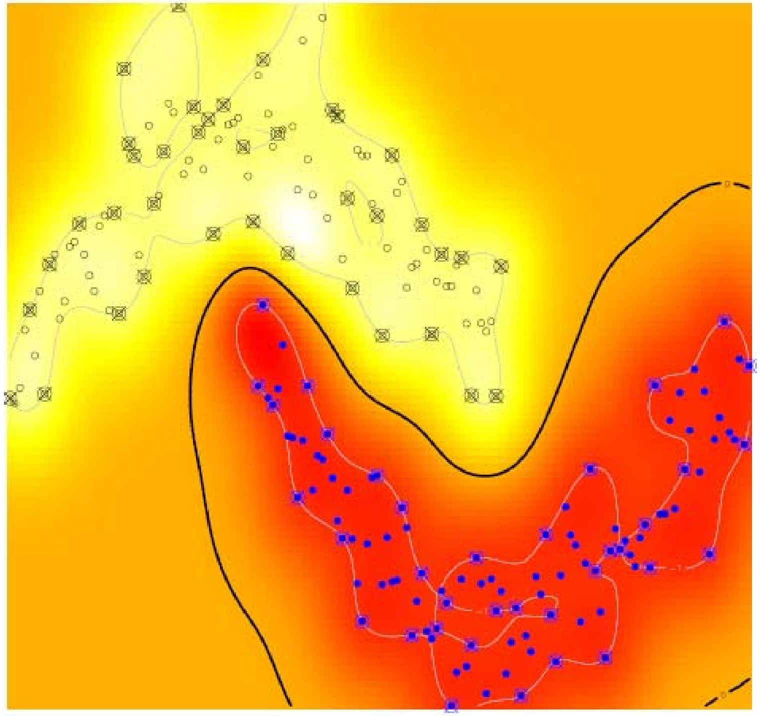


$\sigma = 0.5,\ C = 1$

Results for Gaussian Kernel



$\sigma = 0.02,\ C = 50$



$\sigma = 10,\ C = 50$

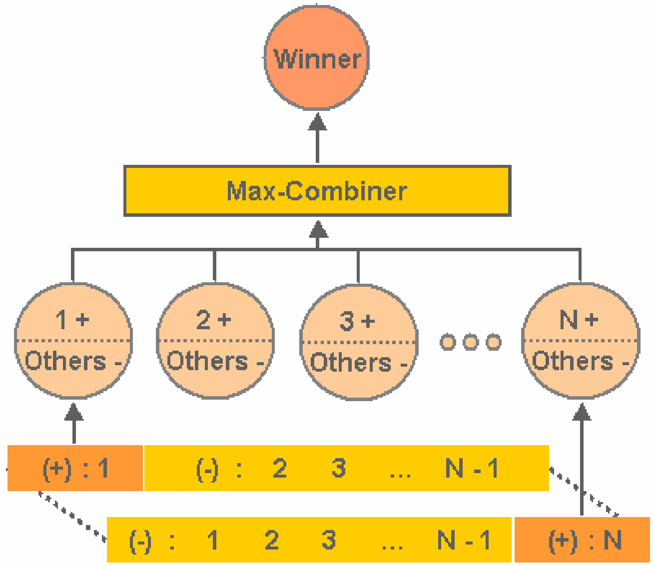
Summary

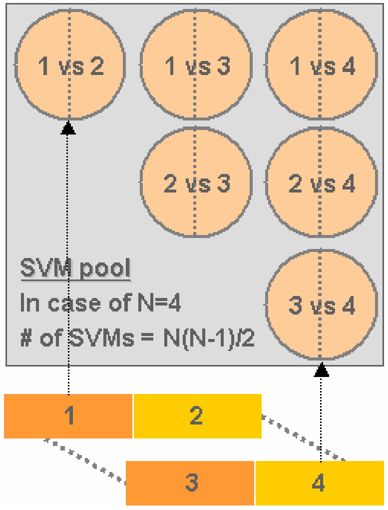
- Optimal Separating Hyperplane (Margin)
- Global Minimum Solution (Convexity)
- Only SVs are Relevant (Sparseness)
- Automatically selects SVs; # of SVs can be considered as # of hidden units of MLP

- Model selection problem; kernel selection
- Training speed and method for a large training set
- Binary classifier

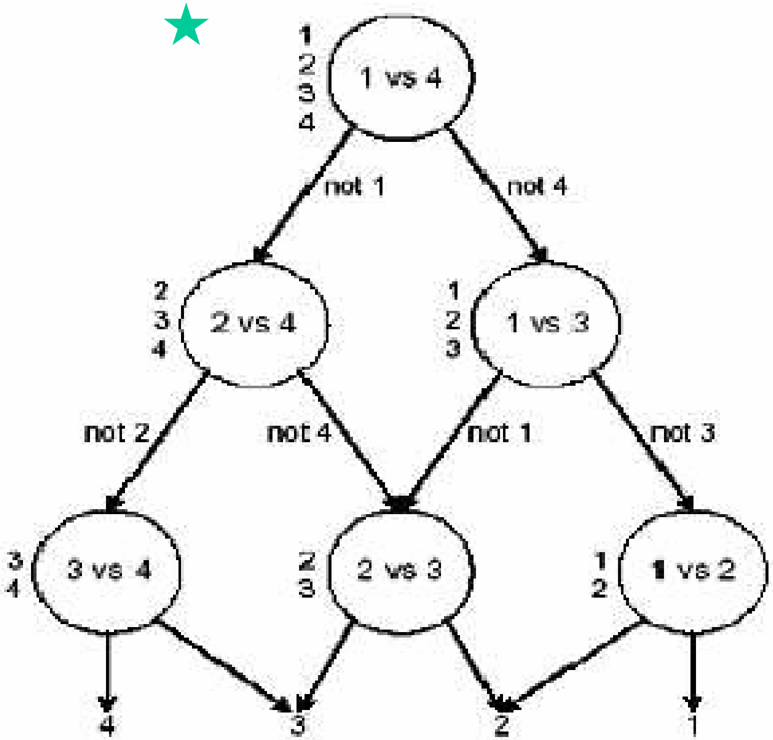
Multiclass Support Vector Machines (Multiple Classifiers System)

- Ensemble of binary support vector machines
- Categorized by Coding / Decoding
- OPC (one-per-class), PWC(pair-wise coupling), ECOC (error correcting output coding)

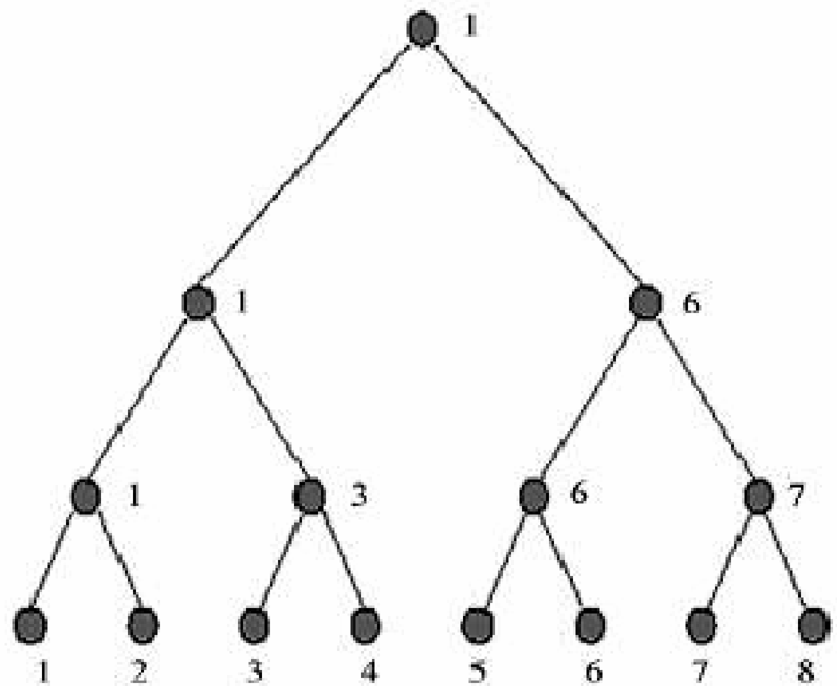




Tree-based Methods



(a) Example of top-down tree structure (DDAG)

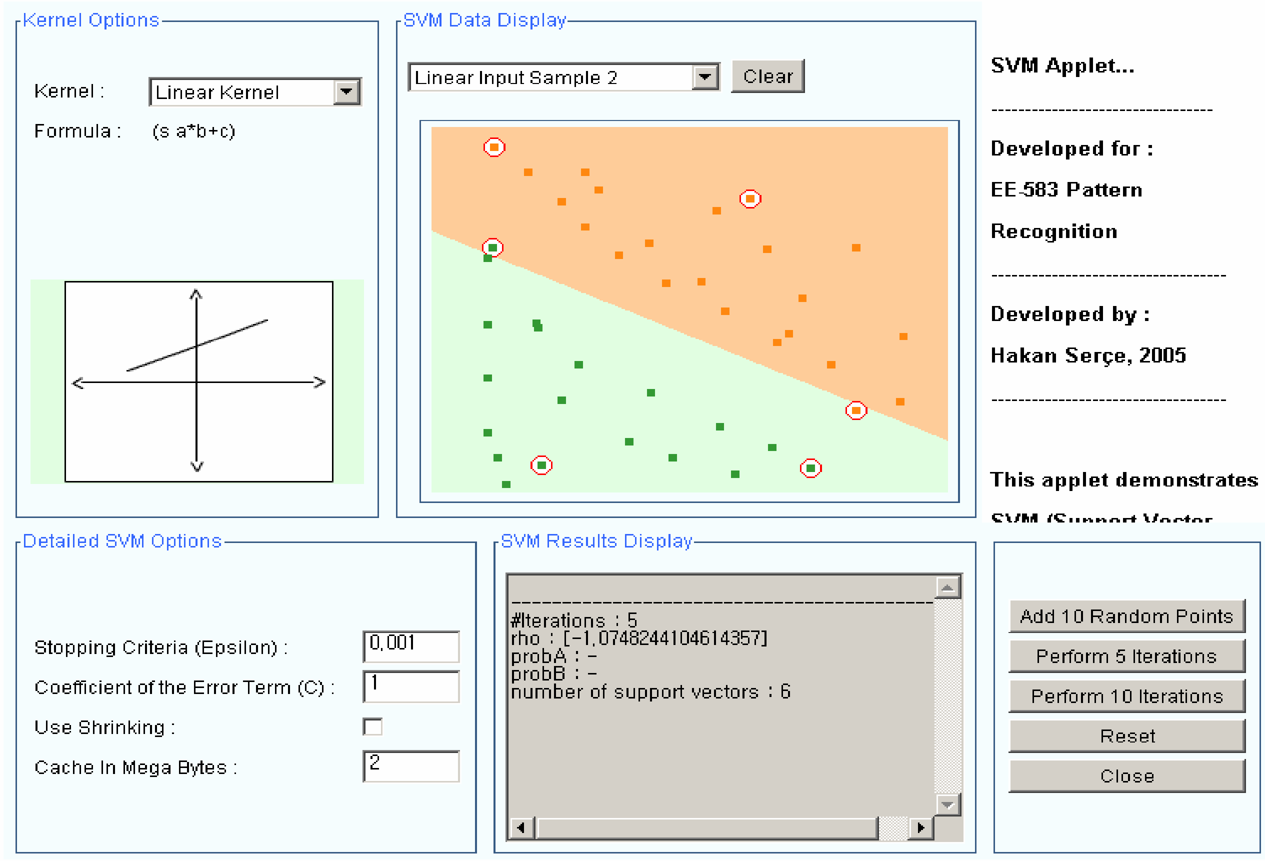


(b) Example of bottom-up tree structure (Pairwise SVM)

Open Software

Software	Developer	Language	Environment	Algorithms	URL
SVMFu	R. Rifkin M. Nadermann (MIT)	C++	Unix-like system	Osuna *et al.*, SMO(Platt)	http://www.ai. mit.edu
LIBSVM	C.C. Chang, C.H. Lin (National Taiwan Univ.)	C++, Java	Python, R, Matlab, Perl	SMO(Platt), SVMLight(Joachims)	http://www.csie. ntu.edu.tw/ libsym
SVMLight	T. Joachims, (Univ. of Dortmund)	C	Solaris, Linux, IRIX, Windows NT	T. Joachims	http://www.svmlight. joachims.org
SVMTorch	R. Collobert, (IDIAP, Switzerland)	C, C++	Windows	R. Collobert	http://www.idap.ch /learning/SVMTorch. html

Demo



- http://www.eee.metu.edu.tr/~alatan/Courses/Demo/AppletSVM.html
- http://www.csie.ntu.edu.tw/~cjlin/libsvm/#GUI

Applications

- Biometrics
- Object Detection and Recognition
- Character Recognition
- Information and Image Retrieval
- Other Applications

References

S.Haykin, *Neural Networks A Comprehensive Foundation*, Prentice Hall, 1999. Chap. 6

Burges, C. J. C., "*A Tutorial on Support Vector Machines for Pattern Recognition*," Data Mining and Knowledge Discovery, Vol. 2, pp. 121~167, 1998.

John C. Platt, "*Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*," Microsoft Research, Technical Report MSR-TR-98-14, 1998

C.-W. Hsu and C.-J. Lin. "*A Comparison of Methods for Multi-Class Support Vector Machines*," IEEE Transactions on Neural Networks, 13, 415-425, 2002.

Books

Vladimir N. Vapnik, *The Nature of Statistical Learning* Theory, Springer-Verlag, 1995.

Nello Cristianini and John Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, 2000.

Bernhard Scholkopf and Alexander J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2001.

Links

http://videolectures.net/
http://www.kernel-machines.org/

감사합니다