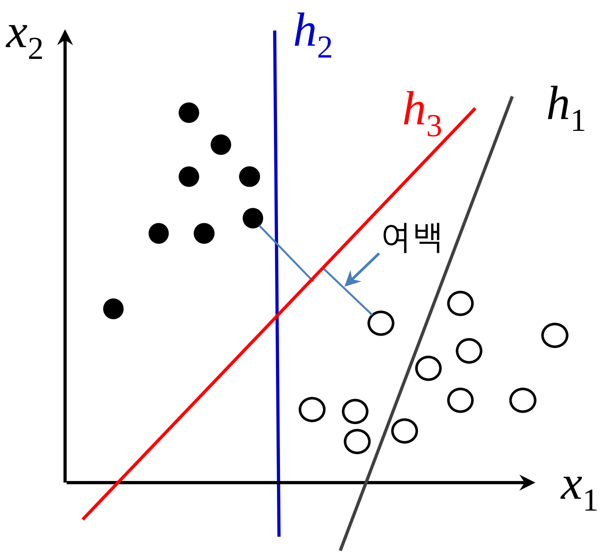
기계 학습

Part IV. 서포트 벡터 머신(SVM)

10.1 SVM

❖ **Support Vector Machine** (SVM)

- Vladimir Vapnik이 제안
- 분류 오차를 줄이면서 동시에 **여백**을 **최대**로 하는 **결정 경계(decision boundary)**를 찾는 **이진 분류기**(binary classifier)



h_3가 h_2보다 우수

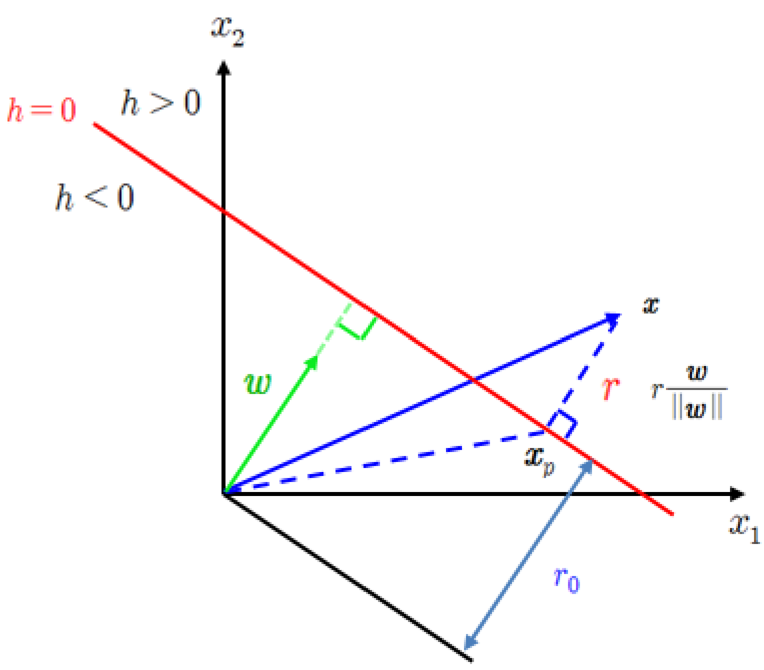
- **여백**(margin)
  - 결정 경계와 가장 가까이에 있는 학습 데이터까지의 거리
- **서포트 벡터**(support vector)
  - 결정 경계로부터 가장 가까이에 있는 학습 데이터들

SVM

- ❖ **초평면 기하학**
  - **초평면(hyperplane)**
    - 4차원이상의 공간에서 선형 방정식으로 표현되는 결정 경계

입력 데이터 $\boldsymbol{x} = [x_1\ x_2\ \cdots\ x_d]^\top$

$$h(\boldsymbol{x}) = w_1x_1 + w_2x_2 + \cdots + w_dx_d + b = \boldsymbol{w}^\top \boldsymbol{x} + b = 0$$



$$\boldsymbol{x} = \boldsymbol{x}_p + r\frac{\boldsymbol{w}}{||\boldsymbol{w}||}$$

$$\boldsymbol{w}^\top \boldsymbol{x} = \boldsymbol{w}^\top \boldsymbol{x}_p + r\frac{\boldsymbol{w}^\top \boldsymbol{w}}{||\boldsymbol{w}||}$$

$$\boldsymbol{w}^\top \boldsymbol{x} = \boldsymbol{w}^\top \boldsymbol{x}_p + r||\boldsymbol{w}||$$

$$\boldsymbol{w}^\top \boldsymbol{x} + b = \boldsymbol{w}^\top \boldsymbol{x}_p + b + r||\boldsymbol{w}||$$

$$h(\boldsymbol{x}) = r||\boldsymbol{w}|| \qquad r = \frac{h(\boldsymbol{x})}{||\boldsymbol{w}||}$$

$$r_0 = \frac{h(\boldsymbol{0})}{||\boldsymbol{w}||} = \frac{b}{||\boldsymbol{w}||}$$

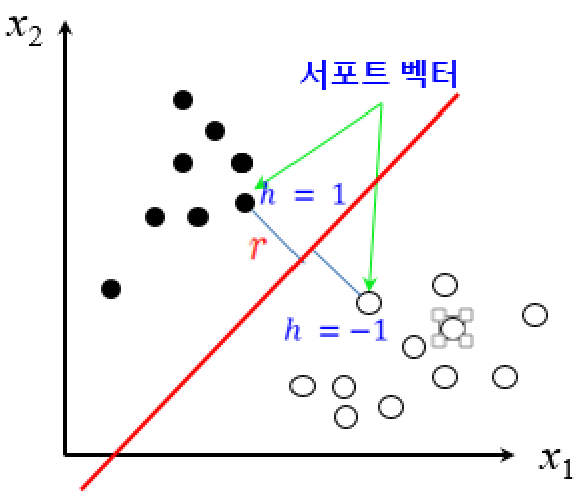
10.2 SVM의 학습

❖ SVM의 학습 목표

- 학습데이터 $X=\{(\boldsymbol{x}_1,t_1),(\boldsymbol{x}_2,t_2),\cdots,(\boldsymbol{x}_N,t_N)\}$ $t_i \in \{1,-1\},\quad i=1,\cdots,N$
- 분류를 위한 **초평면**의 만족조건

❶ $t_i h(\boldsymbol{x}_i) \geq 1,\quad i=1,\cdots,N$

❷ 서포트 벡터와의 거리, 즉 여백을 최대로 한다.



조건 ❶ 서포트 벡터 $\boldsymbol{x}'$에서의 $|h(\boldsymbol{x}')|=1$

$h(\boldsymbol{x})>0$인 공간에 $t_i=1$

$h(\boldsymbol{x})<0$인 공간에 $t_i=-1$

조건 ❷ $r=\dfrac{h(\boldsymbol{x})}{\|\boldsymbol{w}\|}$ 서포트 벡터에 대해서는 $h(\boldsymbol{x})=1$

$r=\dfrac{1}{\|\boldsymbol{w}\|}$

Find $\boldsymbol{w}, b$ which minimizes $J(\boldsymbol{w})=\frac{1}{2}\|\boldsymbol{w}\|^2$

subject to $t_i h(\boldsymbol{x}_i) \geq 1,\quad i=1,\cdots,N$

$$h(\boldsymbol{x})=\boldsymbol{w}^{\top}\boldsymbol{x}+b=0$$

10.3 SVM의 최적화 문제

❖ 제약조건 최적화

$$\boxed{\begin{aligned} &\text{Find } \boldsymbol{x} \text{ which minimizes } f(\boldsymbol{x}) \\ &\quad \text{subject to } g(\boldsymbol{x}) = 0 \\ &\quad\quad h(\boldsymbol{x}) \le 0 \end{aligned}}$$

- 라그랑주 함수

$$L(\boldsymbol{x}, \alpha, \lambda) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x}) + \alpha h(\boldsymbol{x})$$

$$(\alpha \ge 0)$$

$$\boxed{\begin{aligned} &\text{Find } \boldsymbol{w}, b \text{ which minimizes } J(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &\quad \text{subject to } t_i h(\boldsymbol{x}_i) \ge 1, \quad i = 1, \cdots, N \end{aligned}}$$

$$\Downarrow$$

$$\boxed{\begin{aligned} &\text{Find } \boldsymbol{w}, b \text{ which minimizes } J(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &\quad \text{subject to } 1 - t_i h(\boldsymbol{x}_i) \le 0, \quad i = 1, \cdots, N \end{aligned}}$$

- 라그랑주 함수

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \alpha_i \left(1 - t_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)\right)$$

$$\alpha_i \ge 0$$

SVM의 최적화 문제

❖ SVM의 최적화 문제

$$\boxed{\begin{aligned} &\text{Find } \boldsymbol{w}, b \text{ which minimizes } J(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &\qquad\qquad \text{subject to } 1 - t_i h(\boldsymbol{x}_i) \le 0, \quad i = 1, \cdots, N \end{aligned}}$$

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \alpha_i (1 - t_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)) \quad \alpha_i \ge 0$$

- 해에 대한 **KKT(Karush-Kuhn-Tucker) 조건**

  ① $\alpha_i \ge 0, \quad i = 1, \cdots, N$

  ② $1 - t_i h(\boldsymbol{x}_i) = 1 - t_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \le 0, \quad i = 1, \cdots, N$

  ③ $\alpha_i (1 - t_i h(\boldsymbol{x}_i)) = 0, \quad i = 1, \cdots, N$ **상보적 여유성(complementary slackness)**

  ④ $\frac{\partial L}{\partial \boldsymbol{w}} = 0, \ \frac{\partial L}{\partial b} = 0$

- **쌍대 함수(dual function)**

$$\tilde{L}(\boldsymbol{\alpha}) = \min_{\boldsymbol{w}, b} L(\boldsymbol{w}, b, \boldsymbol{\alpha})$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0, \ \frac{\partial L}{\partial b} = 0$$

SVM의 최적화 문제

❖ SVM의 최적화 문제

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}||\boldsymbol{w}||^2 + \sum_{i=1}^{N} \alpha_i (1 - t_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)) \qquad \alpha_i \geq 0$$

- **쌍대함수**(dual function)

$$\tilde{L}(\boldsymbol{\alpha}) = \min_{\boldsymbol{w}, b} L(\boldsymbol{w}, b, \boldsymbol{\alpha})$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0, \quad \frac{\partial L}{\partial b} = 0$$

- $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$의 미분을 0으로 하면

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{N} \alpha_i t_i \boldsymbol{x}_i = 0 \qquad \boxed{\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i t_i \boldsymbol{x}_i}$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{N} \alpha_i t_i = 0 \qquad \boxed{\sum_{i=1}^{N} \alpha_i t_i = 0}$$

SVM의 최적화 문제

❖ SVM의 최적화 문제

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i t_i \boldsymbol{x}_i \qquad \sum_{i=1}^{N} \alpha_i t_i = 0$$

- **초평면 함수식**

$$h(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b = \boldsymbol{w} \cdot \boldsymbol{x} + b = \sum_{i=0}^{N} \alpha_i t_i \boldsymbol{x_i} \cdot \boldsymbol{x} + b$$

- $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$에 넣어 전개하면

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}||\boldsymbol{w}||^2 + \sum_{i=1}^{N} \alpha_i (-t_i(\boldsymbol{w}^\top \boldsymbol{x_i} + b) + 1)$$

$$\tilde{L}(\boldsymbol{\alpha}) = \frac{1}{2}(\sum_{i=1}^{N} \alpha_i t_i \boldsymbol{x_i})(\sum_{j=1}^{N} \alpha_j t_j \boldsymbol{x_j})^\top - \sum_{i=1}^{N} \alpha_i t_i \sum_{j=1}^{N} \alpha_j t_j \boldsymbol{x_j} \boldsymbol{x_i}^\top - b\sum_{i=1}^{N} \alpha_i t_i + \sum_{i=1}^{N} \alpha_i$$

$$= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j t_i t_j \boldsymbol{x}_i \boldsymbol{x}_j^\top + \sum_{i=1}^{N} \alpha_i$$

$$= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j t_i t_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j + \sum_{i=1}^{N} \alpha_i$$

SVM의 최적화 문제

❖ SVM의 최적화 문제

$$\text{Find } \boldsymbol{w}, b \text{ which minimizes } J(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\text{subject to } 1 - t_i h(\boldsymbol{x}_i) \le 0, \quad i = 1, \cdots, N$$

- **라그랑주 함수**로 표현한 **본 문제**(**primal problem**)

$$\text{Find } \boldsymbol{w}, b \text{ which minimizes } L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \alpha_i (1 - t_i(\boldsymbol{w}^\top \boldsymbol{x_i} + b))$$
$$\text{subject to } \alpha_i \ge 0, \quad i = 1, \cdots, N$$

- w와 b가 없는 **쌍대 문제**(**dual problem**)

$$\text{Find } \boldsymbol{\alpha} \text{ which maximizes } \widetilde{L}(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j + \sum_{i=1}^{N}\alpha_i$$
$$\text{subject to } \sum_{i=1}^{N}\alpha_i t_i = 0, \quad i = 1, \cdots, N$$
$$\alpha_i \ge 0, \quad i = 1, \cdots, N$$

- **문제 복잡도**
  - 데이터 개수 N에 관계
  - 데이터 차원과 무관

SVM의 최적화 문제

❖ **SVM의 최적화 문제 해결**

- **쌍대 문제**(dual problem)의 **최소화 문제 변환**

$$\text{Find } \boldsymbol{\alpha} \text{ which maximizes } \widetilde{L}(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j + \sum_{i=1}^{N}\alpha_i$$

$$\text{subject to } \sum_{i=1}^{N}\alpha_i t_i = 0, \quad i = 1, \cdots, N$$

$$\alpha_i \geq 0, \quad i = 1, \cdots, N$$

⇩

$$\text{Find } \boldsymbol{\alpha} \text{ which minimizes } \widetilde{L}(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j - \sum_{i=1}^{N}\alpha_i$$

$$\text{subject to } \sum_{i=1}^{N}\alpha_i t_i = 0, \quad i = 1, \cdots, N$$

$$\alpha_i \geq 0, \quad i = 1, \cdots, N$$

이차식 계획법(quadratic programming) 문제

SVM의 최적화 문제

❖ 이차식 계획법(quadratic programming)

$$\text{Find } \boldsymbol{\alpha} \text{ which } \textcolor{red}{\text{minimizes}} \ \widetilde{L}(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j - \sum_{i=1}^{N}\alpha_i$$

$$\text{subject to } \sum_{i=1}^{N}\alpha_i t_i = 0, \quad i = 1, \cdots, N$$

$$\alpha_i \geq 0, \quad i = 1, \cdots, N$$

$$\widetilde{L}(\boldsymbol{\alpha}) = \frac{1}{2}\begin{bmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_N\end{bmatrix}\begin{bmatrix} t_1 t_1 \boldsymbol{x_1}\cdot\boldsymbol{x_1} & t_1 t_2 \boldsymbol{x_1}\cdot\boldsymbol{x_2} & \cdots & t_1 t_N \boldsymbol{x_1}\cdot\boldsymbol{x_N} \\ t_2 t_1 \boldsymbol{x_2}\cdot\boldsymbol{x_1} & t_2 t_2 \boldsymbol{x_2}\cdot\boldsymbol{x_2} & \cdots & t_2 t_N \boldsymbol{x_2}\cdot\boldsymbol{x_N} \\ \vdots & \vdots & \ddots & \vdots \\ t_N t_1 \boldsymbol{x_N}\cdot\boldsymbol{x_1} & t_N t_2 \boldsymbol{x_N}\cdot\boldsymbol{x_2} & \cdots & t_N t_N \boldsymbol{x_N}\cdot\boldsymbol{x_N} \end{bmatrix}\begin{bmatrix}\alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N\end{bmatrix} - \sum_{i=1}^{N}\alpha_i$$

$$= \frac{1}{2}\boldsymbol{x}^{\top}H\boldsymbol{x} - \sum_{i=1}^{N}\alpha_i$$

- 선형대수학의 **quadratic problem solver** 라이브러리 이용
  - MatLab/Octave의 **quadprog( )**
  - Python의 **solvers.qp( )**

SVM의 최적화 문제

❖ 이차식 계획법(quadratic programming) – cont.

- MatLab/Octave의 **quadprog( )**
  - **이차식 계획법의 표준식(canonical form)**

$$\text{Find } \boldsymbol{\alpha} \text{ which minimizes } L_D(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^\top H\boldsymbol{\alpha} + \boldsymbol{f}^\top\boldsymbol{\alpha}$$

$$\text{subject to } A\boldsymbol{\alpha} \le \boldsymbol{a} \text{ and } B\boldsymbol{\alpha} = \boldsymbol{b}$$

- **α = quadprog(H, f, A, a, B, b)**

  - **SVM의 쌍대문제**

$$\tilde{L}(\boldsymbol{\alpha}) = \frac{1}{2}\begin{bmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_N\end{bmatrix}\begin{bmatrix} t_1t_1\boldsymbol{x_1}\cdot\boldsymbol{x_1} & t_1t_2\boldsymbol{x_1}\cdot\boldsymbol{x_2} & \cdots & t_1t_N\boldsymbol{x_1}\cdot\boldsymbol{x_N} \\ t_2t_1\boldsymbol{x_2}\cdot\boldsymbol{x_1} & t_2t_2\boldsymbol{x_2}\cdot\boldsymbol{x_2} & \cdots & t_2t_N\boldsymbol{x_2}\cdot\boldsymbol{x_N} \\ \vdots & \vdots & \ddots & \vdots \\ t_Nt_1\boldsymbol{x_N}\cdot\boldsymbol{x_1} & t_Nt_2\boldsymbol{x_N}\cdot\boldsymbol{x_2} & \cdots & t_Nt_N\boldsymbol{x_N}\cdot\boldsymbol{x_N}\end{bmatrix}\begin{bmatrix}\alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N\end{bmatrix} - \sum_{i=1}^{N}\alpha_i$$

$$= \frac{1}{2}\boldsymbol{\alpha}^\top H\boldsymbol{\alpha} - \sum_{i=1}^{N}\alpha_i$$

$$\text{subject to } \sum_{i=1}^{N}\alpha_i t_i = 0, \quad i = 1, \cdots, N$$

$$\alpha_i \ge 0, \quad i = 1, \cdots, N \qquad \Rightarrow \quad -\alpha_i \le 0, \quad i = 1, \cdots, N$$

SVM의 최적화 문제

❖ 이차식 계획법(quadratic programming) – cont.

- MatLab/Octave의 **quadprog( )**

$$\text{Find } \boldsymbol{\alpha} \text{ which minimizes } L_D(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^\top H \boldsymbol{\alpha} - \sum_{i=1}^{N} \alpha_i$$

$$\text{subject to } -\alpha_i \le 0 \text{ and } \sum_{i=1}^{N} \alpha_i t_i = 0, \; i = 1, \cdots, N$$

$$\text{Find } \boldsymbol{\alpha} \text{ which minimizes } L_D(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^\top H \boldsymbol{\alpha} + \boldsymbol{f}^\top \boldsymbol{\alpha}$$

$$\text{subject to } A\boldsymbol{\alpha} \le \boldsymbol{a} \text{ and } B\boldsymbol{\alpha} = \boldsymbol{b}$$

  - **α = quadprog(H, f, A, a, B, b)**

- **SVM의 쌍대문제**

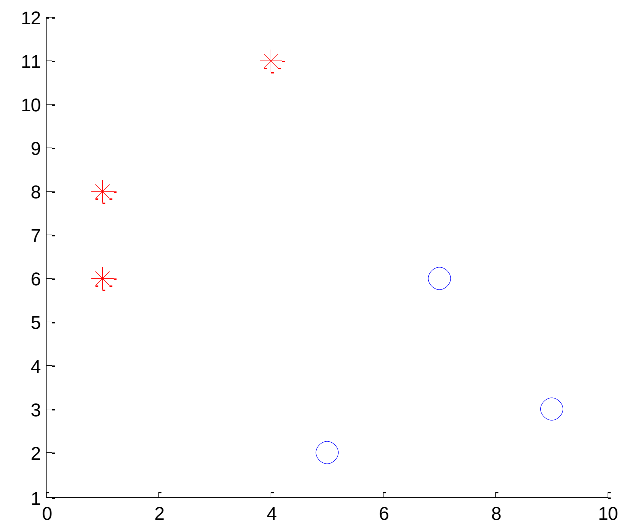
$$H = \begin{bmatrix} t_1 t_1 \boldsymbol{x_1} \cdot \boldsymbol{x_1} & t_1 t_2 \boldsymbol{x_1} \cdot \boldsymbol{x_2} & \cdots & t_1 t_N \boldsymbol{x_1} \cdot \boldsymbol{x_N} \\ t_2 t_1 \boldsymbol{x_2} \cdot \boldsymbol{x_1} & t_2 t_2 \boldsymbol{x_2} \cdot \boldsymbol{x_2} & \cdots & t_2 t_N \boldsymbol{x_2} \cdot \boldsymbol{x_N} \\ \vdots & \vdots & \ddots & \vdots \\ t_N t_1 \boldsymbol{x_N} \cdot \boldsymbol{x_1} & t_N t_2 \boldsymbol{x_N} \cdot \boldsymbol{x_2} & \cdots & t_N t_N \boldsymbol{x_N} \cdot \boldsymbol{x_N} \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}^\top .* \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \vdots \\ \boldsymbol{x_N} \end{bmatrix} \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \vdots \\ \boldsymbol{x_N} \end{bmatrix}^\top$$

$$\boldsymbol{f} = \begin{bmatrix} -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix} \quad A = \begin{bmatrix} -1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 \end{bmatrix} \quad \boldsymbol{a} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad B = \begin{bmatrix} t_1 & t_2 & \cdots & t_N \end{bmatrix} \quad \boldsymbol{b} = \begin{bmatrix} 0 \end{bmatrix}$$

SVM의 학습 예

❖ MatLab/Octave의 사용한 예

- 부류 1 (+1) : (1,6), (1,8), (4,11)
- 부류 2 (–1) : (5,2), (7,6), (9,3)



- 입력데이터 **X**
  - `X = [1 6; 1 8; 4 11; 5 2; 7 6; 9 3]`

$$X = \begin{bmatrix} 1 & 6 \\ 1 & 8 \\ 4 & 11 \\ 5 & 2 \\ 7 & 6 \\ 9 & 3 \end{bmatrix}$$

- 출력 데이터 t
  - `t = [1; 1; 1; -1; -1; -1]`

$$t = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$$

- 행렬 H

`H = (X*X').*(t*t')`

$$H = \begin{bmatrix} 37 & 49 & 70 & -17 & -43 & -27 \\ 49 & 65 & 92 & -21 & -55 & -33 \\ 70 & 92 & 137 & -42 & -94 & -69 \\ -17 & -21 & -42 & 29 & 47 & 51 \\ -43 & -55 & -94 & 47 & 85 & 81 \\ -27 & -33 & -69 & 51 & 81 & 90 \end{bmatrix}$$

SVM의 학습 예

❖ MatLab/Octave의 사용한 예 – cont.

- `f = -ones(6,1)`
- `A = -eye(6)`
- `a = zeros(6,1)`
- `B = t'`
- `b = [0]`

$$f = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \quad A = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \quad a = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

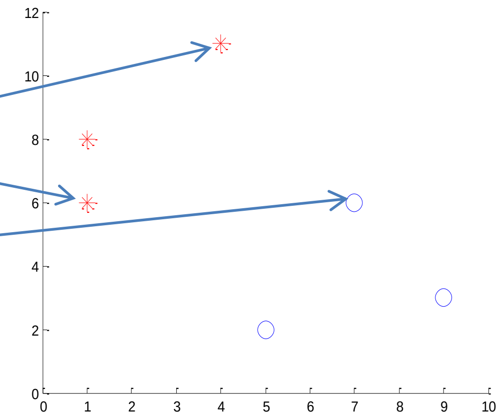
$$B = \begin{bmatrix} 1 & 1 & 1 & -1 & -1 & -1 \end{bmatrix}$$

▪ **이차식 계획법 사용**

- `α = quadprog(H+eye(6)*0.001, f, A, a, B, b)`

$$X = \begin{bmatrix} 1 & 6 \\ 1 & 8 \\ 4 & 11 \\ 5 & 2 \\ 7 & 6 \\ 9 & 3 \end{bmatrix} \qquad \boldsymbol{\alpha} = \begin{bmatrix} 0.0356 \\ 0.0000 \\ 0.0400 \\ 0.0000 \\ 0.0756 \\ 0.0000 \end{bmatrix}$$

서포트 벡터



SVM의 학습 예

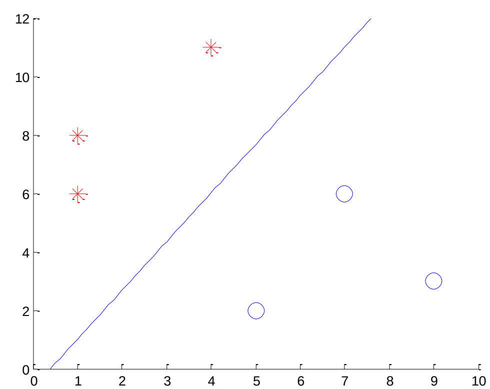
❖ SVM의 최적화 문제 해결

- 선형대수학의 **quadratic problem solver** 라이브러리를 이용
  - 최적화 문제의 해인 $\alpha_1, \alpha_2, \ldots, \alpha_N$ 계산
- KKT 조건의 $\alpha_i(1 - h(\boldsymbol{x}_i)) = 0$ 때문에, $\alpha_i \neq 0$이라면 $h(\boldsymbol{x}_i) = 1$
- $\alpha_i \neq 0$인 $\boldsymbol{x}_i$가 서포트 벡터
- $\boldsymbol{w}$ 의 계산

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i t_i \boldsymbol{x}_i = \begin{bmatrix} -0.333 \\ 0.200 \end{bmatrix}$$

$$h(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b$$
$$= \sum_{i=1}^{N} a_i t_i \boldsymbol{x}_i \cdot \boldsymbol{x} + b$$

$$X = \begin{bmatrix} 1 & 6 \\ 1 & 8 \\ 4 & 11 \\ 5 & 2 \\ 7 & 6 \\ 9 & 3 \end{bmatrix} \qquad \boldsymbol{\alpha} = \begin{bmatrix} 0.0356 \\ 0.0000 \\ 0.0400 \\ 0.0000 \\ 0.0756 \\ 0.0000 \end{bmatrix}$$



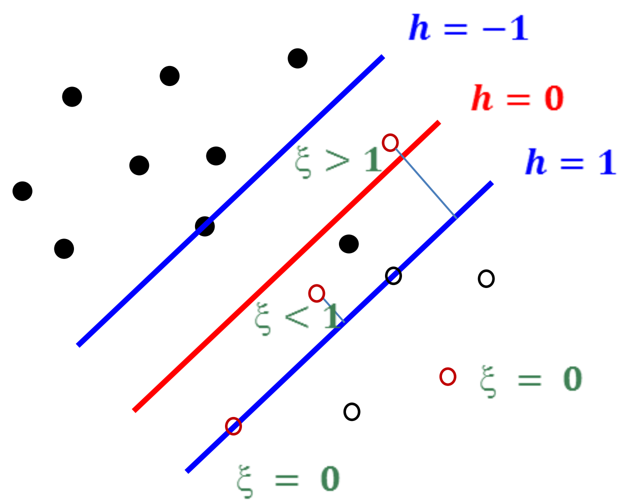
$$h(x_1, x_2) = -0.333x_1 + 0.2x_2 + 0.13$$

- b의 계산
  - 서포트 벡터 하나를 $t_i(\boldsymbol{w}^\top \boldsymbol{x}_i - b) = 1$ 에 넣어 계산

$$\begin{bmatrix} -0.333 \\ 0.200 \end{bmatrix}^\top \begin{bmatrix} 1 \\ 6 \end{bmatrix} - b = 1 \qquad b = 0.13$$

10.4 선형 분리불가 문제의 SVM

❖ 선형 분리불가 문제의 SVM



- **슬랙변수**(slack variable) ξ_i
  - 학습 데이이터별로 하나씩 생성
  - SVM의 분할 초평면에서 서포트벡터보다 멀리 위치하면, $\xi_i = 0$
  - 이외의 경우, $\xi_i = |t_i - h(\boldsymbol{x}_i)|$

$$\boxed{t_i h(\boldsymbol{x}_i) \geq 1 - \xi_i, \quad i = 1, \cdots, N}$$

선형 분리불가 문제의 SVM

- 선형분리가 되지 않는 데이터에 대한 SVM
  - **최적화 문제**
    - 슬랙 변수를 허용하는 **제약조건**

$$t_i h(\boldsymbol{x}_i) \geq 1 - \xi_i, \quad i = 1, \cdots, N$$

    - **슬랙 변수 값**의 **합**을 **최소화**

$$\begin{aligned} &\text{Find } \boldsymbol{w}, b \text{ which minimizes } J(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\ &\qquad \text{subject to } t_i h(\boldsymbol{x}_i) \geq 1 - \xi_i, \quad i = 1, \cdots, N \\ &\qquad\qquad \xi_i \geq 0, \quad i = 1, \cdots, N \\ &\qquad\qquad C > 0 \end{aligned}$$

    - **라그랑주 함수**

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i(1 - t_i h(\boldsymbol{x}_i) - \xi_i) - \sum_{i=1}^{N}\beta_i\xi_i$$

선형 분리불가 문제의 SVM

❖ 선형분리가 되지 않는 데이터에 대한 SVM

- 최적화 문제 - cont.
  - 라그랑주 함수

$$L(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i(1-t_ih(\boldsymbol{x_i})-\xi_i) - \sum_{i=1}^{N}\beta_i\xi_i$$

  - 쌍대 문제(dual problem)

$$\tilde{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_jt_it_j\boldsymbol{x_i}\cdot\boldsymbol{x_j}$$

$$\sum_{i=1}^{N}\alpha_it_i = 0$$

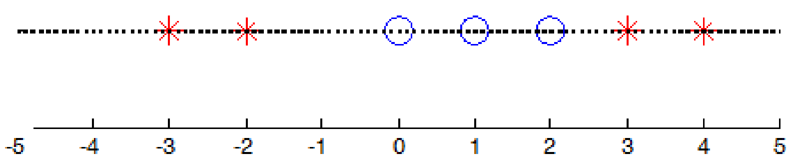
$$0 \le \alpha_i \le C$$ 상한(upper bound)에 대한 제약조건 추가

10.5 비선형 SVM

❖ **선형 SVM** (linear SVM)
- **선형**인 **초평면**으로 **공간 분할**
- **슬랙변수**를 도입하더라도 **한계**

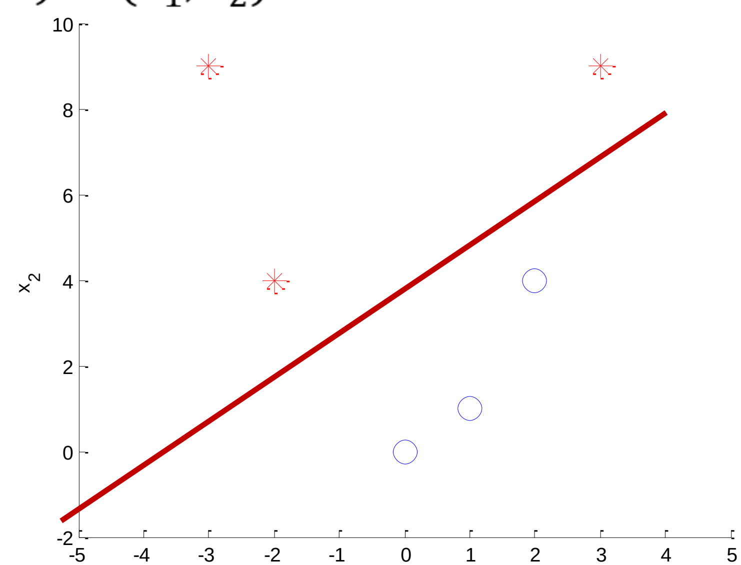
❖ **데이터의 고차원 사상**(high-dimensional mapping)
- 데이터를 고차원의 사상하면 선형 분리 가능

$$\Phi(x) = (x, x^2) = (x_1, x_2)$$



$$h(x_1, x_2) = w_1 x + w_2 x^2 + b$$

비선형 함수



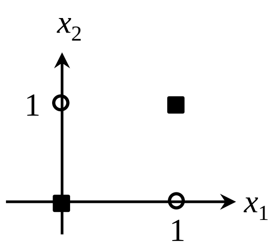
$$h(x_1, x_2) = w_1 x_1 + w_2 x_2 + b$$

선형 함수

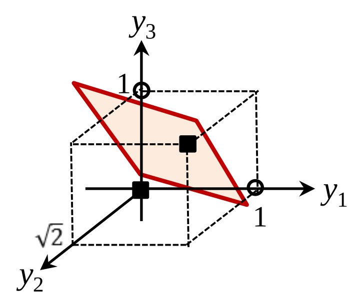
비선형 SVM

❖ 데이터의 고차원 사상 – cont.

- **XOR 문제**



$$\Phi(\boldsymbol{x}) = (x_1^2, \sqrt{2}\,x_1x_2, x_2^2) = (y_1, y_2, y_3)$$



❖ 고차원 변환의 문제점

- **차원의 저주**(curse of dimensionality) 문제 발생
- 테스트 데이터에 대한 **일반화**(generalization) **능력 저하** 가능
  - **여백(margin) 최대화**를 통해 일반화 능력 유지
- **계산 비용** 증가
  - **커널 트릭(kernel trick)** 사용으로 해결

비선형 SVM

❖ SVM의 최적화 문제

- **선형 SVM**

$$\text{Find } \boldsymbol{\alpha} \text{ which } \textcolor{red}{\text{minimizes}} \quad \widetilde{L}(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boxed{\boldsymbol{x}_i \cdot \boldsymbol{x}_j} - \sum_{i=1}^{N}\alpha_i$$

$$\text{subject to } \sum_{i=1}^{N}\alpha_i t_i = 0, \quad i = 1, \cdots, N$$

$$\alpha_i \geq 0, \quad i = 1, \cdots, N$$

$$h(\boldsymbol{x}) = \sum_{i=0}^{N}\alpha_i t_i \boxed{\boldsymbol{x}_i \cdot \boldsymbol{x}} + b$$

- **데이터의 고차원 변환**

$$\boldsymbol{x}_i \rightarrow \Phi(\boldsymbol{x}_i)$$

- **비선형 SVM**

$$\widetilde{L}(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boxed{\Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)} + \sum_{i=1}^{N}\alpha_i$$

$$h(\boldsymbol{x}) = \sum_{i=1}^{N}a_i t_i \boxed{\Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x})} + b$$

비선형 SVM

❖ 커널 트릭(kernel trick)

- $\boldsymbol{\Phi}(\boldsymbol{x}_i) \cdot \boldsymbol{\Phi}(\boldsymbol{x}_j)$를 고차원으로 **변환하여 계산하지 않고**, **원래 데이터에서 계산**
- 고차원 변환없이 계산할 수 있는 **커널 함수**(kernel function) $\boldsymbol{K}$ 사용
  - $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\Phi}(\boldsymbol{x}_i) \cdot \boldsymbol{\Phi}(\boldsymbol{x}_j)$

❖ 커널 함수의 예

- $K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^T \boldsymbol{y})^2$
  - $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$
  - $(\boldsymbol{x}^T \boldsymbol{y})^2 = (x_1 y_1 + x_2 y_2)^2 = (x_1 y_1)^2 + 2(x_1 y_1)(x_2 y_2) + (x_2 y_2)^2$
    $= \left[(x_1)^2 \quad \sqrt{2} x_1 x_2 \quad (x_2)^2\right]\left[(y_1)^2 \quad \sqrt{2} y_1 y_2 \quad (y_2)^2\right]^T$
    $= \boldsymbol{\Phi}(\boldsymbol{x}) \cdot \boldsymbol{\Phi}(\boldsymbol{y})$
  - $\boldsymbol{\Phi}(\boldsymbol{x}) = \begin{bmatrix} (x_1)^2 \\ \sqrt{2} x_1 x_2 \\ (x_2)^2 \end{bmatrix} \quad \boldsymbol{\Phi}(\boldsymbol{y}) = \begin{bmatrix} (y_1)^2 \\ \sqrt{2} y_1 y_2 \\ (y_2)^2 \end{bmatrix}$

비선형 SVM

❖ **대표적인 커널 함수**(kernel function)

- **다항식 커널**(polynomial kernel)

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j + 1)^p, \quad p\text{는 양의정수}$$

- **RBF**(radial basis function) **커널**

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j) = e^{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/(2\sigma^2)}$$

- **쌍곡 탄젠트**(hyperbolic tangent) **커널**

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j) = \tanh(\alpha \boldsymbol{x}_i \cdot \boldsymbol{x}_j + \beta)$$

비선형 SVM

❖ 커널 트릭을 사용할 때의 최적화 문제

$$\widetilde{L}(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boxed{\Phi(\boldsymbol{x}_i)\cdot\Phi(\boldsymbol{x}_j)} + \sum_{i=1}^{N}\alpha_i$$

$$h(\boldsymbol{x}) = \sum_{i=1}^{N} a_i t_i \boxed{\Phi(\boldsymbol{x}_i)\cdot\Phi(\boldsymbol{x})} + b$$

- 커널 함수 적용

$$\widetilde{L}(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boxed{K(\boldsymbol{x}_i, \boldsymbol{x}_j)} + \sum_{i=1}^{N}\alpha_i$$

$$h(\boldsymbol{x}) = \sum_{i=1}^{N} a_i t_i \boxed{K(\boldsymbol{x}_i, \boldsymbol{x})} + b$$

비선형 SVM

❖ 초평면 결정

- **선형 SVM**

$$\widetilde{L}(\boldsymbol{\alpha}) = \frac{1}{2}\begin{bmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_N\end{bmatrix}\begin{bmatrix} t_1t_1\boldsymbol{x_1}\cdot\boldsymbol{x_1} & t_1t_2\boldsymbol{x_1}\cdot\boldsymbol{x_2} & \cdots & t_1t_N\boldsymbol{x_1}\cdot\boldsymbol{x_N} \\ t_2t_1\boldsymbol{x_2}\cdot\boldsymbol{x_1} & t_2t_2\boldsymbol{x_2}\cdot\boldsymbol{x_2} & \cdots & t_2t_N\boldsymbol{x_2}\cdot\boldsymbol{x_N} \\ \vdots & \vdots & \ddots & \vdots \\ t_Nt_1\boldsymbol{x_N}\cdot\boldsymbol{x_1} & t_Nt_2\boldsymbol{x_N}\cdot\boldsymbol{x_2} & \cdots & t_Nt_N\boldsymbol{x_N}\cdot\boldsymbol{x_N} \end{bmatrix}\begin{bmatrix}\alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N\end{bmatrix} - \sum_{i=1}^{N}\alpha_i$$

- **비선형 SVM**

$$\widetilde{L}(\boldsymbol{\alpha}) = \frac{1}{2}\begin{bmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_N\end{bmatrix}\begin{bmatrix} t_1t_1K(\boldsymbol{x_1},\boldsymbol{x_1}) & t_1t_2K(\boldsymbol{x_1},\boldsymbol{x_2}) & \cdots & t_1t_NK(\boldsymbol{x_1},\boldsymbol{x_N}) \\ t_2t_1K(\boldsymbol{x_2},\boldsymbol{x_1}) & t_2t_2K(\boldsymbol{x_2},\boldsymbol{x_2}) & \cdots & t_2t_NK(\boldsymbol{x_2},\boldsymbol{x_N})) \\ \vdots & \vdots & \ddots & \vdots \\ t_Nt_1K(\boldsymbol{x_N},\boldsymbol{x_1}) & t_Nt_2K(\boldsymbol{x_N},\boldsymbol{x_2}) & \cdots & t_Nt_NK(\boldsymbol{x_N},\boldsymbol{x_N} \end{bmatrix}\begin{bmatrix}\alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N\end{bmatrix} - \sum_{i=1}^{N}\alpha_i$$

- **이차식 계획법 라이브러리 사용**
  - quadprog( ), solver.qp()
  - $\alpha_1, \alpha_2, \dots, \alpha_N$ 계산
  - 고차원의 초평면 함수 결정

$$h(\boldsymbol{x}) = \sum_{i=1}^{N} a_i t_i \Phi(\boldsymbol{x}_i)\cdot\Phi(\boldsymbol{x}) + b$$

$$h(\boldsymbol{x}) = \sum_{i=1}^{N} a_i t_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b$$

SVM

❖ 비선형 SVM에 의한 결정 경계 및 서포트 벡터