

# Support Vector Machine

Jeonghun Yoon

# Concepts

Decision Hyperplane

조건부 최적화 문제

- Affine set / function
- Convex set / function
- Convex optimization
- Lagrangian / Dual problem / Wolfe duality
- KKT conditions

Soft / Hard margin SVM

Kernel Function

# Classification 분류

Supervised Learning

Given samples  $X = ((\mathbb{x}_1, y_1), (\mathbb{x}_2, y_2), \dots, (\mathbb{x}_N, y_N))$



find!  $f : \mathbb{X} \rightarrow Y$

# Demo

<http://cs.stanford.edu/people/karpathy/svmjs/demo/>

# Think about

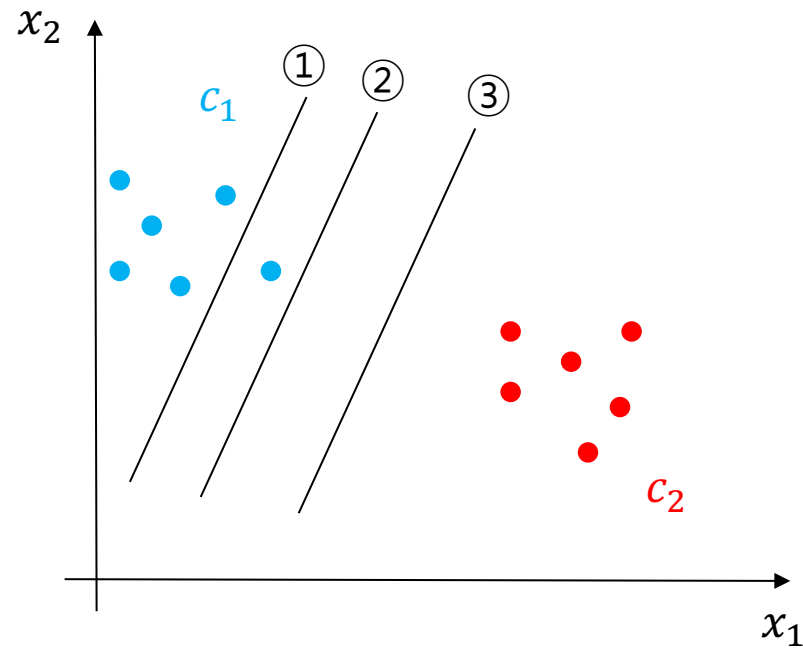
Question 1) 좋은 분류기(classifier)란?

Question 2) 분류기의 분류 기준은?

- Bayesian Classifier : Error, Risk를 minimize  
(Likelihood or MAP 분포에서의 오류영역)

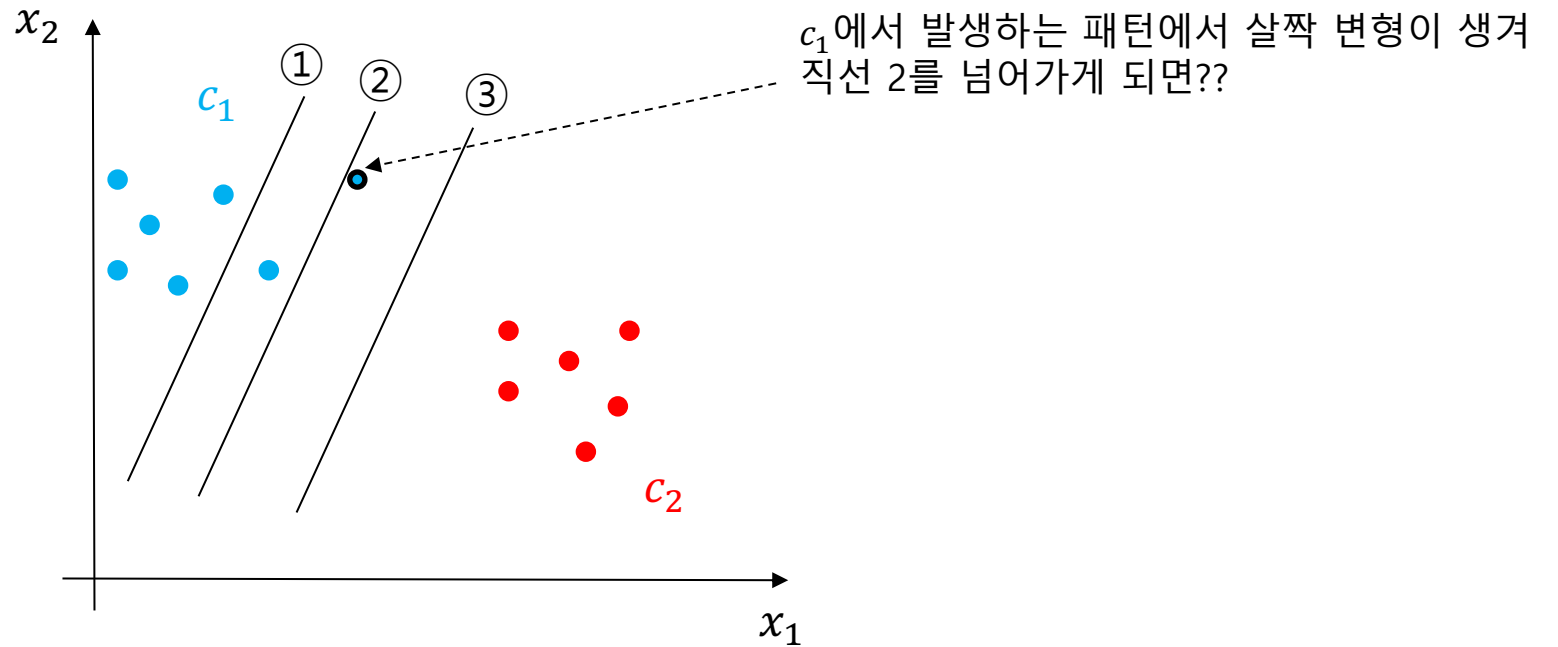
# Classifier의 기준

1. 데이터는 2차원 공간의 point라고 가정 : feature가 2개
2. 하늘색 point  $\in c_1$ , 빨간색 point  $\in c_2$  라고 가정



어떤 분류 직선을 선택하겠는가?

# Classifier의 기준

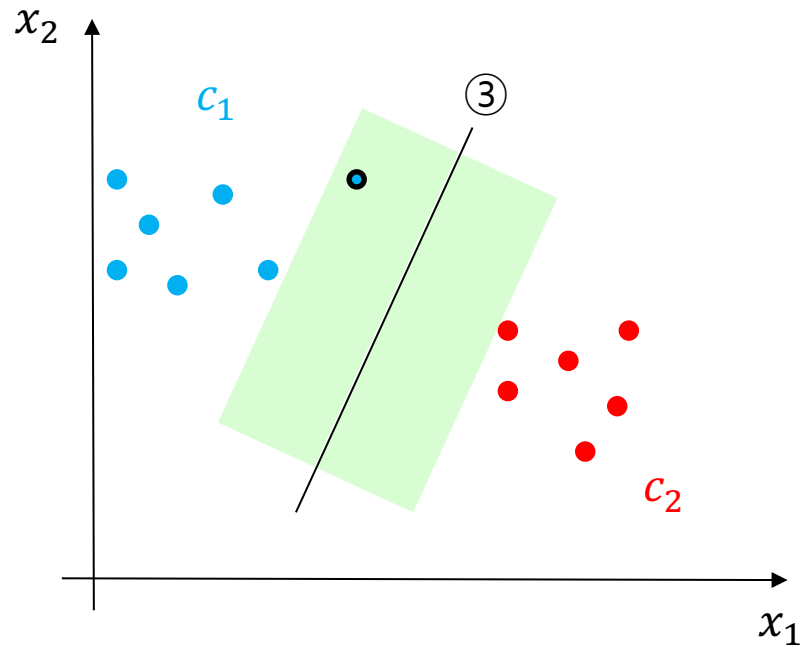


분류 직선을 선택함에 있어서, 중요하게 고려되어야 할 요소 : **Generalization**  
(미래에 발생할 미지의 패턴을 얼마나 잘 분류하는가?)

# Classifier의 기준

Generalization을 극대화

(패턴에서 어느정도의 변형이 발생해도 오분류 되지 않도록 한다.)



부류 사이에 여유를 두자. 즉 여백(margin)을 크게 만들고, 반으로 나누자.



# Decision Hyperplane 결정 초평면

$$\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n, \mathbf{w} = (w_1, w_2, \dots, w_n) \in \mathbb{R}^n, b \in \mathbb{R}$$

$$d(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = 0 \quad \text{결정 초평면 (decision hyperplane)}$$

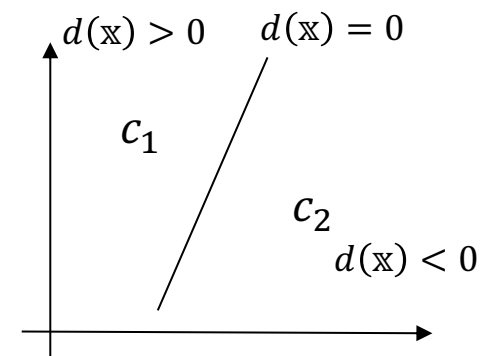
$d(\mathbf{x})$ 은 특징 공간을 두 영역으로 분할한다.

- $c_1$ 은  $d(\mathbf{x}) > 0$ 인 영역
- $c_2$ 은  $d(\mathbf{x}) < 0$ 인 영역

하나의 초평면을 표현하는 식은 여럿 있다.

$\mathbf{w}$  : normal vector(초평면의 방향),  $b$  : 초평면의 위치

임의의 점  $\mathbf{x}_1$ 에서 초평면까지의 거리 :  $\mathbf{h} = \frac{d(\mathbf{x}_1)}{|\mathbf{w}|}$



# 조건부 최적화 문제

Constrained optimization problem (primal problem)

$$\begin{array}{ll}\text{minimize} & f_0(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & h_i(\mathbf{x}) = 0, \quad i = 1, \dots, p\end{array}$$

- $\mathbf{x} \in \mathbb{R}^n$  is the optimization variable.
- $f_0: \mathbb{R}^n \rightarrow \mathbb{R}$  is the objective function or cost function. (목적 함수)
- $f_i: \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \dots, m$ , are inequality constraint functions.
- $h_i: \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \dots, p$ , are equality constraint functions.
- Optimal value  $p^*$

$$p^* = \inf\{f_0(\mathbf{x}) \mid f_i(\mathbf{x}) \leq 0, i = 1, \dots, m, h_i(\mathbf{x}) = 0, i = 1, \dots, p\}$$

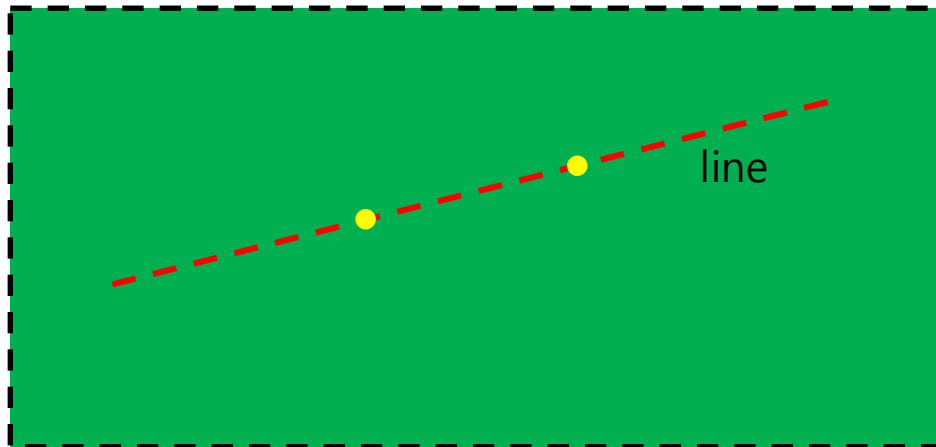
# Affine / Convex

## Affine set

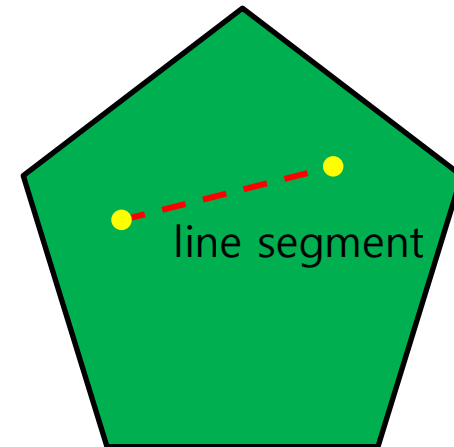
- $C \subset \mathbb{R}^n$  is **affine**  $\rightarrow$   $C$  안의 2개의 서로 다른 point를 지나는 line이  $C$ 에 속할 때.
- 쉬운 예로, 평면 or hyperplane 등이 있다.

## Convex set

- $C \subset \mathbb{R}^n$  is **convex**  $\rightarrow$   $C$  안의 2개의 서로 다른 point를 지나는 line segment가  $C$ 에 속할 때.
- 쉬운 예로, 원 or 오각형 등이 있다.



Affine set (평면)



convex set (오각형)

# Affine function

## Affine function

- a function composed of a linear function and constant (translation)
- in 1-dim :  $y = Ax + C$
- in 2-dim :  $f(x, y) = Ax + By + C$
- in 3-dim :  $f(x, y, z) = Ax + By + Cz + D$

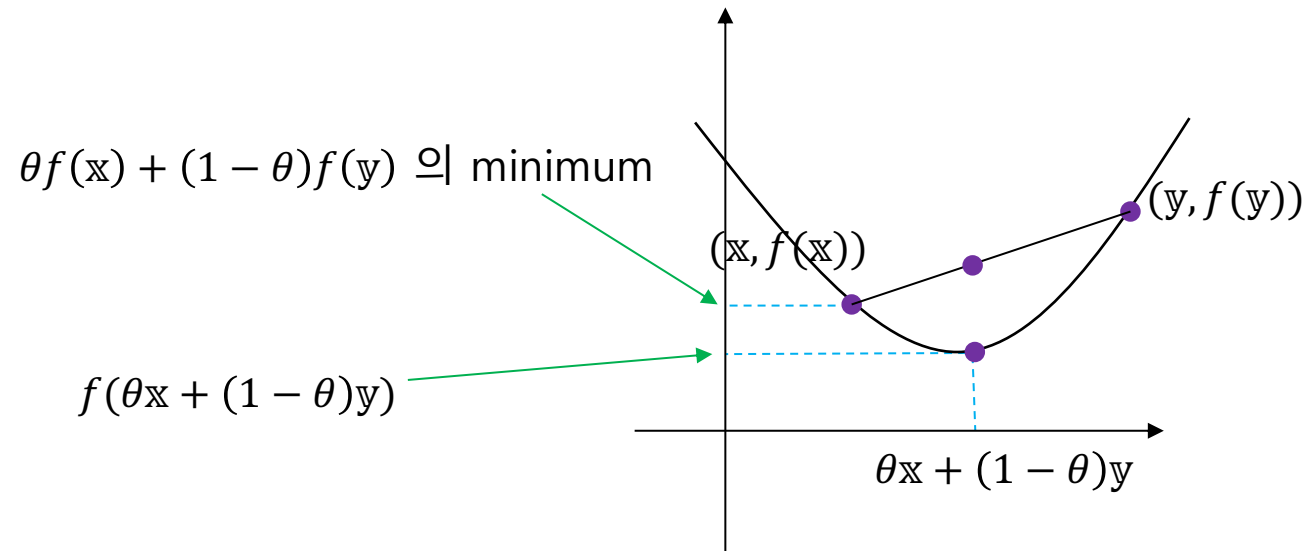
Translation : a transformation consisting of a constant offset with no rotation or distortion

# Convex function

$f: \mathbb{R}^n \rightarrow \mathbb{R}$  is **convex** if  $\text{dom } f$  is convex and

$$f(\theta \mathbf{x} + (1 - \theta)\mathbf{y}) \leq \theta f(\mathbf{x}) + (1 - \theta)f(\mathbf{y})$$

for  $\forall \mathbf{x}, \mathbf{y} \in \text{dom } f, 0 \leq \theta \leq 1$ .



$f$  is concave if  $-f$  is convex.

**dom  $f$**  : 함수  $f$ 의 유효한 입력 값의 집합. 이 영역을  $f$ 가 define되는 영역이라고도 표현 한다.

# Convex optimization

convex optimization problem

$$\begin{array}{ll}\text{minimize} & f_0(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & \mathbf{a}_i^T \mathbf{x} = b_i, \quad i = 1, \dots, p\end{array}$$

- $f_0, \dots, f_m$  are **convex**.
- $h_i(\mathbf{x}) = \mathbf{a}_i^T \mathbf{x} - b_i, \quad i = 1, \dots, p$  are **affine**.
- The feasible set of a convex optimization problem is convex.
  - Since  $f_1, \dots, f_m$  are convex,  $\bigcap_{i=1}^m \text{dom } f_i$  is convex.
  - Since  $\{\mathbf{x} \mid \mathbf{a}_i \mathbf{x} = b_i\}$  is hyperplane,  $\bigcap_{i=1}^p \{\mathbf{x} \mid \mathbf{a}_i \mathbf{x} = b_i\}$  is affine (i.e convex).
- In a convex optimization, we minimize a convex objective function over a convex set.

# Lagrangian

**Lagrangian**  $\mathcal{L}: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$ ,

$$\mathcal{L}(\mathbb{x}, \Lambda, V) = f_o(\mathbb{x}) + \sum_{i=1}^m \lambda_i f_i(\mathbb{x}) + \sum_{i=1}^p v_i h_i(\mathbb{x})$$

with,  $\text{dom } \mathcal{L} = (\cap_{i=1}^m \text{dom } f_i) \cap (\cap_{i=1}^p \text{dom } h_i) \times \mathbb{R}^m \times \mathbb{R}^p$

- $\lambda_i$  is **Lagrange multiplier** associated with  $f_i(\mathbb{x}) \leq 0$  and  $\Lambda = (\lambda_1, \dots, \lambda_m)$
- $v_i$  is **Lagrange multiplier** associated with  $h_i(\mathbb{x}) = 0$  and  $V = (v_1, \dots, v_p)$

# Lagrange dual function / Lagrange dual problem

Lagrange dual function  $g: \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$

$$g(\Lambda, V) = \inf_{\mathbf{x} \in \mathcal{D}} \mathcal{L}(\mathbf{x}, \Lambda, V) = \inf_{\mathbf{x} \in \mathcal{D}} \left( f_0(\mathbf{x}) + \sum_{i=1}^m \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^p v_i h_i(\mathbf{x}) \right)$$

Lagrange dual problem

$$\begin{array}{ll} \text{maximize} & g(\Lambda, V) \\ \text{subject to} & \Lambda \succeq 0 \end{array}$$

- $g(\Lambda, V)$  is concave and constraints are convex, so this is convex optimization problem.
- Lagrange dual problem의 해  $(\Lambda^*, V^*)$ 는 primal problem  $(\mathbf{x}^*)$  해의 **lower bound**를 제공한다.

$$\inf \mathcal{L}(\mathbf{x}^*, \Lambda^*, V^*) \leq f_0(\mathbf{x}^*)$$

*How to make it coincide?*



# KKT condition and zero duality

If

- $f_i$  are convex (목적함수, inequality constraint)
- $h_i$  are affine (equality constraint)
- $\mathbf{x}^*, \Lambda^*, V^*$  satisfy KKT condition

Then

- $\mathbf{x}^*$  is primal optimal and  $(\Lambda^*, V^*)$  is dual optimal with zero duality gap.
- 두 개의 해가 같아진다.

# KKT conditions

primal constraints  $f_i(\mathbb{x}') \leq 0, i = 1, \dots, m$

$$h_i(\mathbb{x}') = 0, i = 1, \dots, p$$

dual constraints  $\lambda'_i \geq 0, i = 1, \dots, m$

complementary slackness  $\lambda'_i f_i(\mathbb{x}') = 0, i = 1, \dots, m$

gradient Lagrangian  
with respect to  $\mathbb{x}$  vanishes at  $\mathbb{x}'$

$$\nabla f_0(\mathbb{x}') + \sum_{i=1}^m \lambda'_i \nabla f_i(\mathbb{x}') + \sum_{i=1}^p v'_i \nabla h_i(\mathbb{x}') = 0$$

# Wolfe duality

## Wolfe duality problem

- $f_0$  is convex.
- $f_i, g_i$  are differentiable. (Lagrange dual problem  $\rightarrow$  Wolfe duality problem)

$$\begin{aligned} \text{maximize} \quad & f_0(\mathbf{x}) + \sum_{i=1}^m \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^p v_i h_i(\mathbf{x}) \quad \leftarrow \mathcal{L}(\mathbf{x}, \Lambda, V) \\ \text{subject to} \quad & \Lambda \geq 0 \\ & \nabla f_0(\mathbf{x}) + \sum_{i=1}^m \lambda'_i \nabla f_i(\mathbf{x}) + \sum_{i=1}^p v'_i \nabla h_i(\mathbf{x}) = 0 \end{aligned}$$

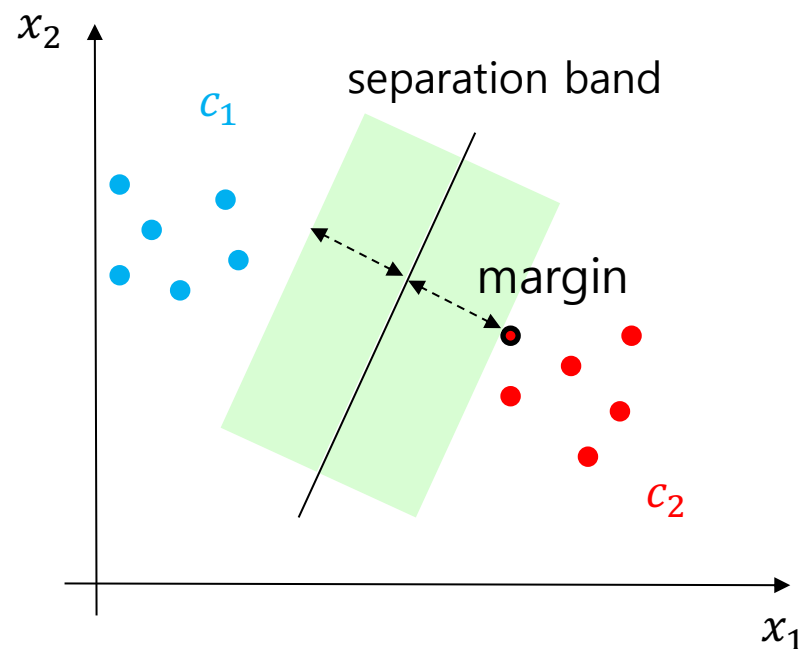
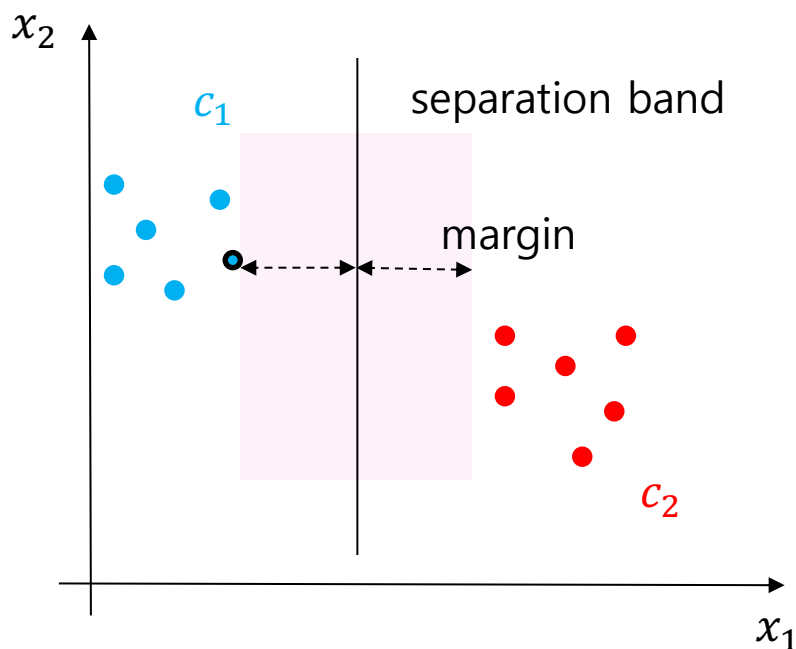
- 제약조건이 KKT conditions에 있는 조건들에 이미 포함되므로, 이 문제의 해를 구하는 것이 primal problem의 해를 구하는 것과 동일하다.

# Idea of SVM

SVM의 concept은 여백(margin)에서 시작한다.

두 부류의 샘플 사이의 여백(margin)의 크기를 극대화하는 분류직선을 찾는다.

- 여백(margin) : 결정 초평면(직선)으로부터 가장 가까운 샘플까지의 거리의 2배



*How to find the best margin?*

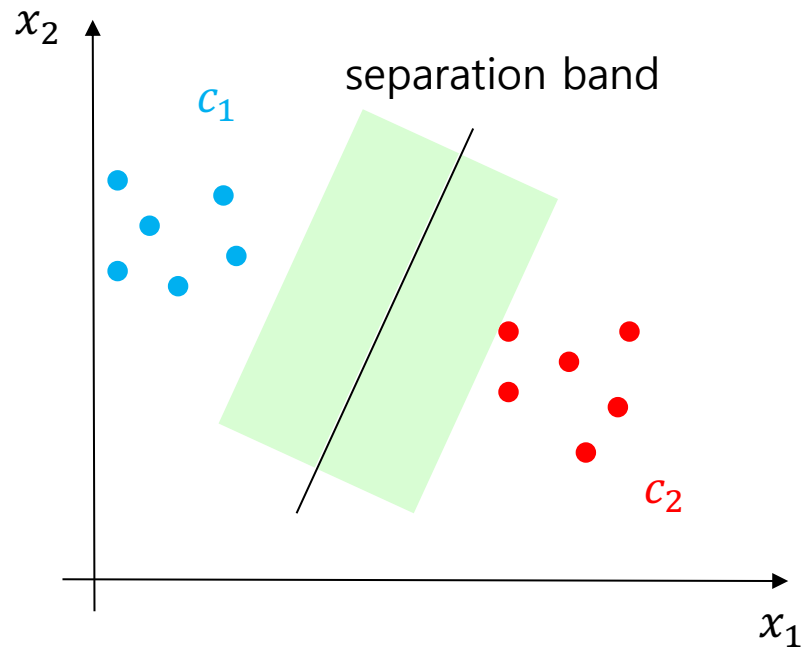
# SVM의 종류

## 선형 SVM

- 선형 분리 가능한 상황(Hard margin)
- 선형 분리 불가능한 상황(Soft margin)

## 비선형 SVM

# 선형 분리 가능한 상황



(분할 띠 안에 데이터가 존재하지 않을 때)

# Problem of SVM

**Problem** : margin을 극대화하는 분류직선을 찾아라.

- support vector,  $\mathbf{x}_s$ : 분류직선으로부터 가장 가까운 sample
- margin의 크기 =  $\frac{2|d(\mathbf{x}_s)|}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$  (직선의 방정식은 rescale이 가능함. 편의상 1로 계산)
- 훈련집합  $X = \{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \dots, (\mathbf{x}_n, t_n)\}$ 
  - if  $\mathbf{x}_i \in c_1$ , then  $t_i = 1$
  - if  $\mathbf{x}_i \in c_2$ , then  $t_i = -1$

$$\begin{aligned} &\text{maximize} && \frac{2}{\|\mathbf{w}\|} \\ &\text{subject to} && \mathbf{w}^T \mathbf{x}_i + b \geq 1 \quad \forall \mathbf{x}_i \in c_1 \\ &&& \mathbf{w}^T \mathbf{x}_i + b \leq -1 \quad \forall \mathbf{x}_i \in c_2 \end{aligned}$$

※ 가정 : 모든 샘플을 옳게 분류한다. 즉 모든 샘플이 분할 띠의 바깥에 놓여있다.

# Primal problem of SVM

## Primal problem (조건부 최적화 문제)

- $\frac{2}{\|\mathbf{w}\|}$ 를 최대화하는 문제는 비용함수  $\frac{\|\mathbf{w}\|^2}{2}$ 를 최소화하는 문제로 변형할 수 있다.

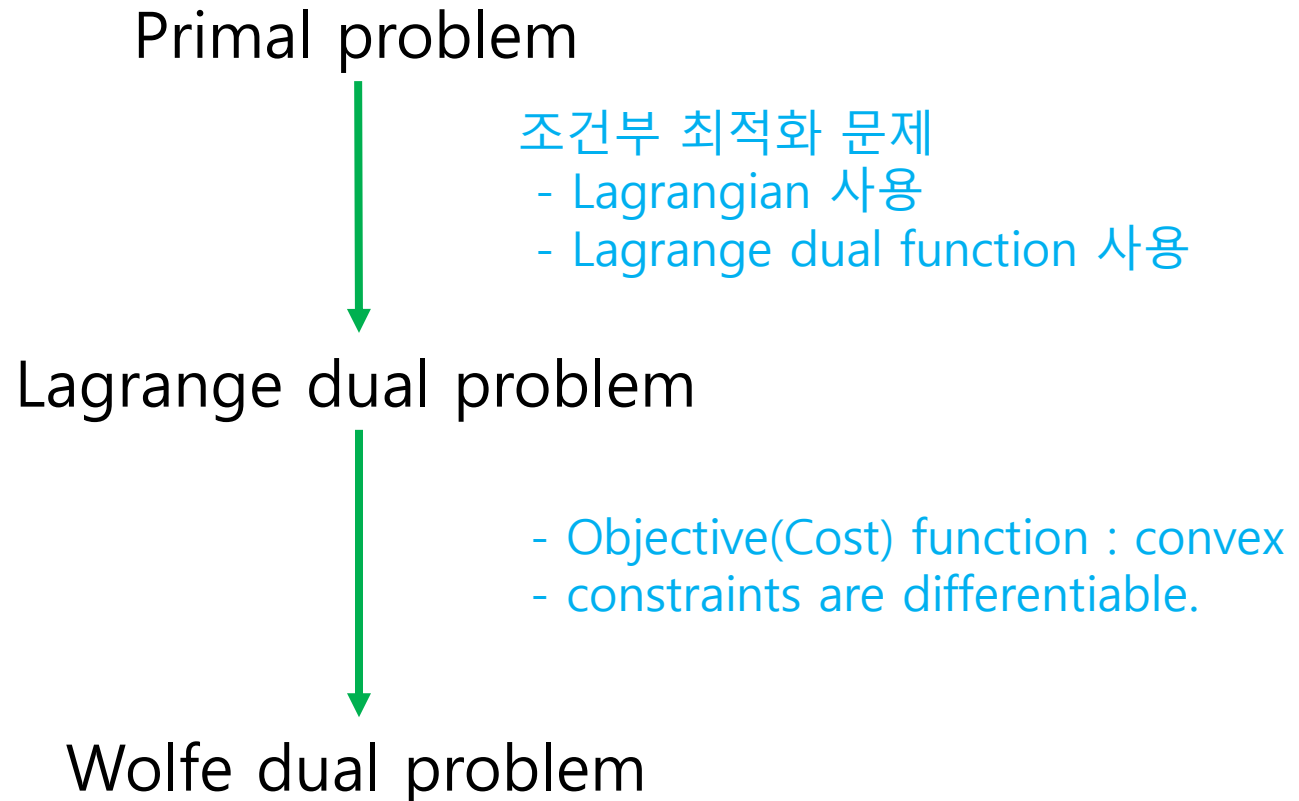
$$\begin{aligned} \text{minimize} \quad & J(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{subject to} \quad & t_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad i = 1, \dots, N \end{aligned}$$

- $J(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2$ 는  $\|\mathbf{w}\|$ 의 2차식으로 이루어져 있기 때문에 **convex 함수**이다.
- **inequality constrain**가 모두 선형이기 때문에 **convex 함수**이다.
- inequality constrain을 만족시키는 영역(**feasible**)은 **convex set**이다.
- Primal problem은 **convex optimization problem**이다.
- Primal problem의 해는 **global solution**이 된다.



# Primal problem of SVM

Primal problem을 아래와 같은 순서로 변형시킬 것이다.



# Lagrangian / Lagrange dual problem of SVM

Lagrangian of primal problem

- $\mathfrak{a} = (\alpha_1, \dots, \alpha_N)^T$  를 Lagrange multiplier라고 하자.  $\alpha_i \geq 0$  for  $i = 1, \dots, N$

$$\mathcal{L}(\mathfrak{w}, b, \mathfrak{a}) = \frac{1}{2} \|\mathfrak{w}\|^2 - \sum_{i=1}^N \alpha_i (t_i (\mathfrak{w}^T \mathfrak{x}_i + b) - 1)$$

Lagrange dual function of primal problem

$$g(\mathfrak{a}) = \inf_{\mathfrak{w}, b} \mathcal{L}(\mathfrak{w}, b, \mathfrak{a}) = \inf_{\mathfrak{w}, b} \left( \frac{1}{2} \|\mathfrak{w}\|^2 - \sum_{i=1}^N \alpha_i (t_i (\mathfrak{w}^T \mathfrak{x}_i + b) - 1) \right)$$

Lagrange dual problem of primal problem

$$\begin{array}{ll} \text{maximize} & g(\mathfrak{a}) = \inf_{\mathfrak{w}, b} \mathcal{L}(\mathfrak{w}, b, \mathfrak{a}) \\ \text{subject to} & \mathfrak{a} \succcurlyeq 0 \end{array}$$

# Wolfe dual problem of SVM

## Wolfe dual problem

- 목적함수  $\frac{1}{2} \|\mathbb{w}\|^2$  는 convex 함수이다.
- 목적함수와 제약조건식이 모두 미분 가능하다.

$$\begin{aligned} \text{maximize} \quad & \mathcal{L}(\mathbb{w}, b, \mathfrak{a}) = \frac{1}{2} \|\mathbb{w}\|^2 - \sum_{i=1}^N \alpha_i (t_i (\mathbb{w}^T \mathbb{x}_i + b) - 1) \\ \text{subject to} \quad & \mathfrak{a} \geq 0, \frac{\partial \mathcal{L}}{\partial \mathbb{w}} = 0, \frac{\partial \mathcal{L}}{\partial b} = 0 \end{aligned}$$

$$\begin{aligned} \alpha_i &\geq 0, i = 1, \dots, N & \mathbb{w} &= \sum_{i=1}^N \alpha_i t_i \mathbb{x}_i & \sum_{i=1}^N \alpha_i t_i &= 0 \end{aligned}$$



$$\begin{aligned} \text{maximize} \quad & \mathcal{L}(\mathfrak{a}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j t_i t_j \mathbb{x}_i^T \mathbb{x}_j \\ \text{subject to} \quad & \alpha_i \geq 0, i = 1, \dots, N \\ & \sum_{i=1}^N \alpha_i t_i = 0 \end{aligned}$$

# Wolfe dual problem of SVM

$$\begin{aligned} \text{maximize} \quad & \mathcal{L}(\mathfrak{a}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j t_i t_j \mathbb{x}_i^T \mathbb{x}_j \\ \text{subject to} \quad & \alpha_i \geq 0, i = 1, \dots, N \\ & \sum_{i=1}^N \alpha_i t_i = 0 \end{aligned}$$

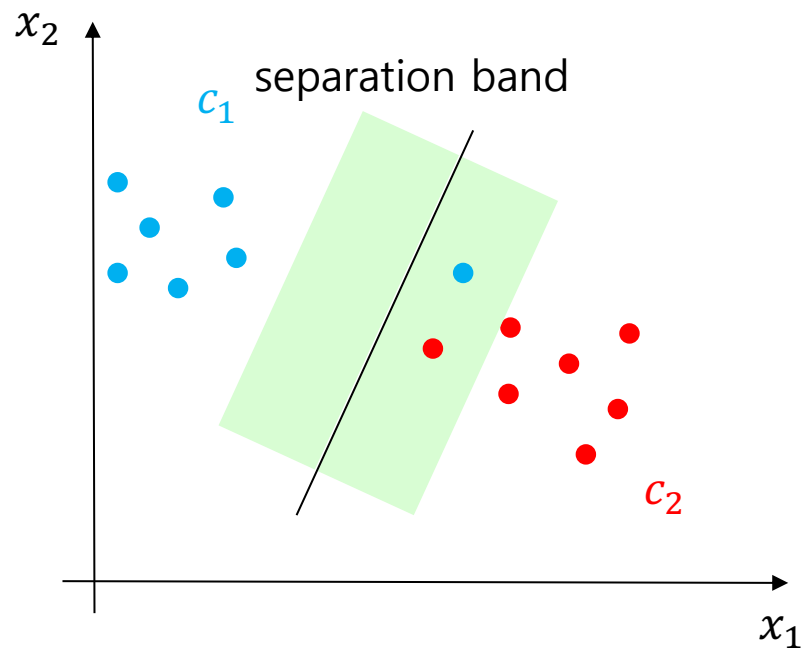
한 개의 등식 조건과  $N$ 개의 부등식 조건을 가진 2차(quadratic) 목적 함수의 최대화 문제이다.

$\mathfrak{w}$ ,  $b$ 를 구하는 문제가 아닌, Lagrange multiplier  $\mathfrak{a}$ 를 구하는 문제로 변경된다.

목적 함수에서 특징 벡터  $\mathbb{x}_i$ 가 혼자 나타나지 않고, 두 개의 특징 벡터의 내적  $\mathbb{x}_i^T \mathbb{x}_j$ 으로 나타난다.

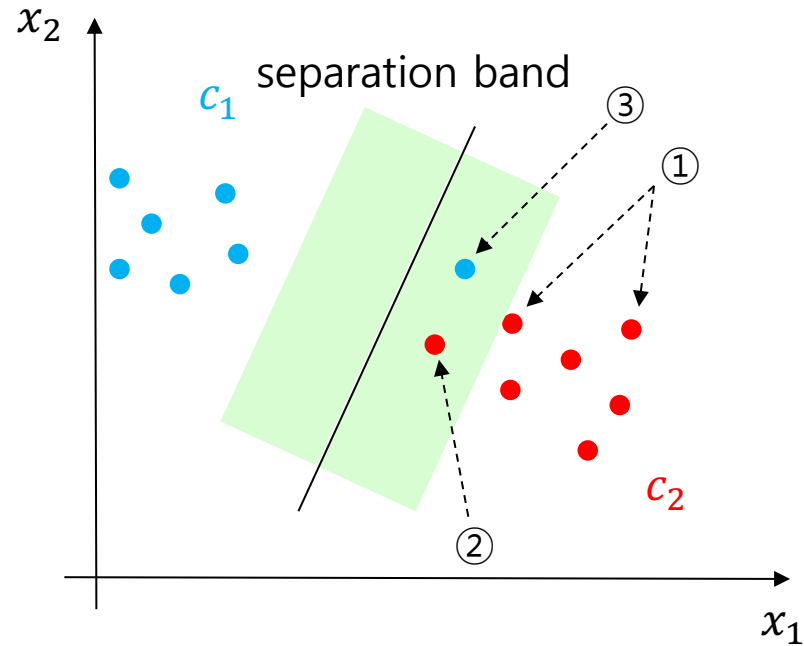
$N^2$ 의 항을 계산해야 한다.

# 선형 분리 불가능한 상황<sub>soft margin</sub>



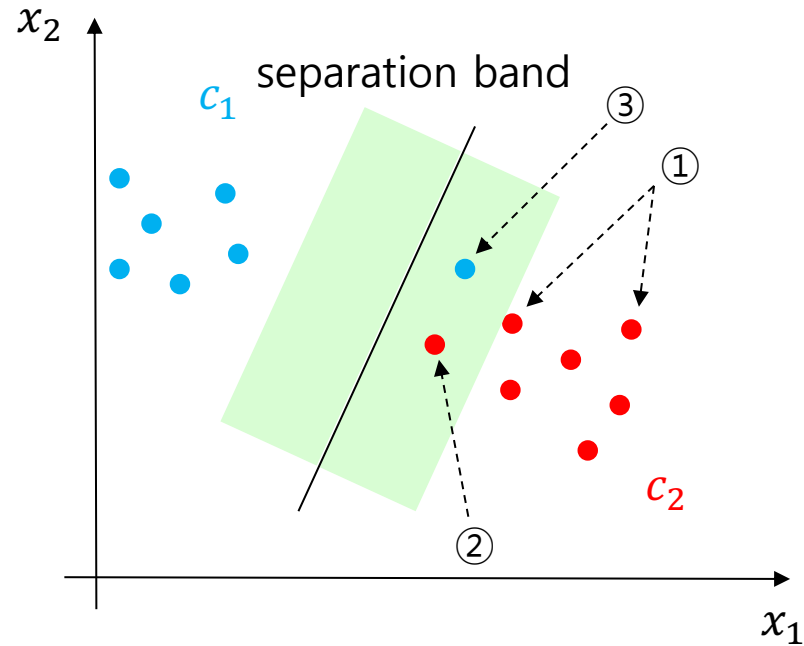
(분할 때 안에 데이터가 존재하거나, 분류오류가 생기는 경우)

# 샘플의 위치



- ① 분할 띠의 경계에 있거나(support vector), 분할 띠의 바깥에 있다.  
 $\Rightarrow \mathbf{1} \leq \mathbf{t}(\mathbf{w}^T \mathbf{x} + \mathbf{b})$
- ② 분할 띠의 안쪽에 있고, 자기가 속한 부류의 영역에 있다.  
 $\Rightarrow \mathbf{0} \leq \mathbf{t}(\mathbf{w}^T \mathbf{x} + \mathbf{b}) < \mathbf{1}$
- ③ 결정 경계를 넘어 자신이 속하지 않은 부류의 영역에 있다.  
 $\Rightarrow \mathbf{t}(\mathbf{w}^T \mathbf{x} + \mathbf{b}) < \mathbf{0}$

# 샘플의 위치



slack 변수  $\xi$ 를 사용하여, 3가지 경우를, 하나의 식으로 표현

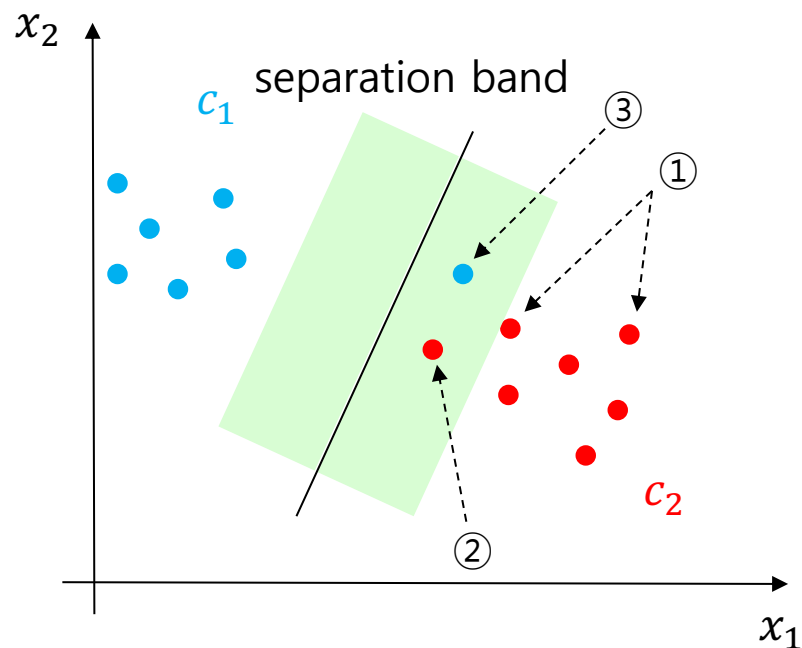
$$\textcircled{1} \quad \mathbf{1} \leq t(\mathbf{w}^T \mathbf{x} + \mathbf{b}) \Rightarrow \xi = 0$$

$$t(\mathbf{w}^T \mathbf{x} + \mathbf{b}) \geq \mathbf{1} - \xi$$

$$\textcircled{2} \quad \mathbf{0} \leq t(\mathbf{w}^T \mathbf{x} + \mathbf{b}) < \mathbf{1} \Rightarrow 0 < \xi \leq 1$$

$$\textcircled{3} \quad t(\mathbf{w}^T \mathbf{x} + \mathbf{b}) < \mathbf{0} \Rightarrow \xi > 1$$

# Objective function of soft margin SVM

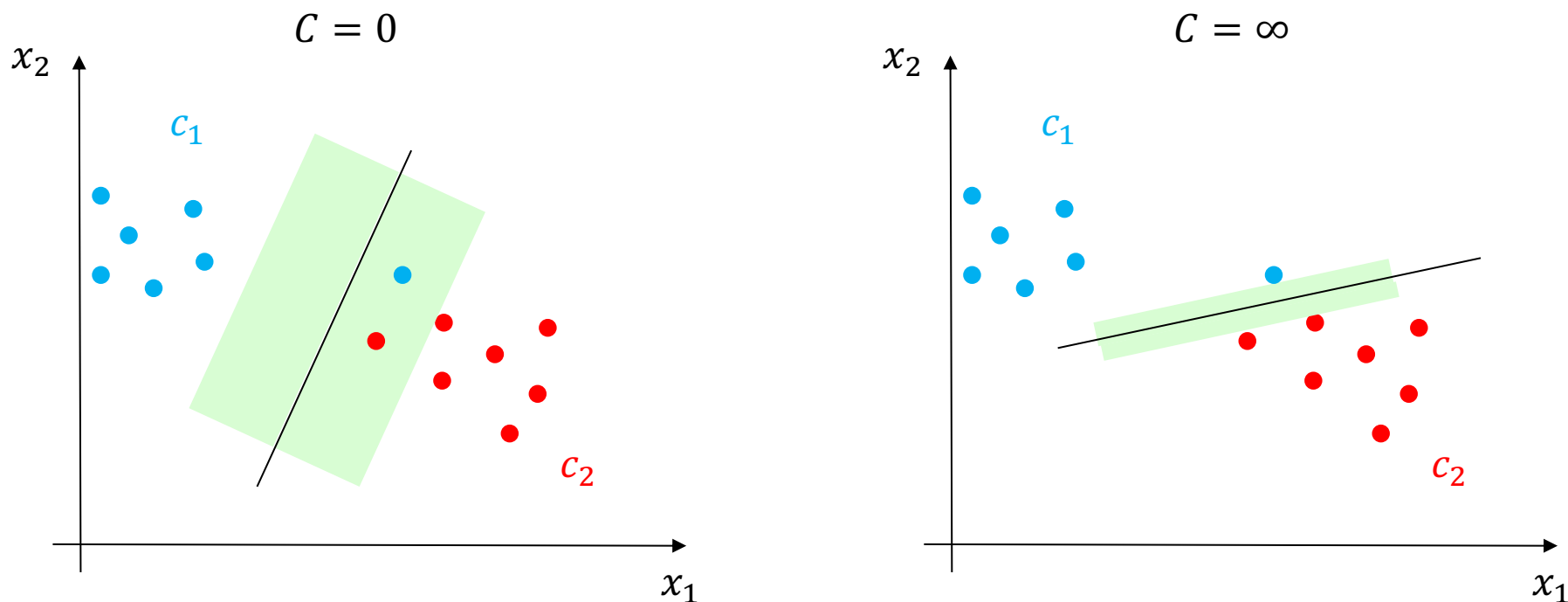


우리가 해결해야 할 문제는 다음 2개의 목적을 만족시켜야 한다.

- 목적 1 : margin을 최대화 시켜야 한다.
- 목적 2 : ②, ③ 경우에 해당하는 샘플의 수를 최소한으로 줄여야 한다.



# Objective function of soft margin SVM



2가지의 목적에 맞게 목적함수를 아래와 같이 변경하도록 한다.

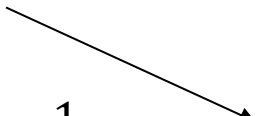
$\Xi = (\xi_1, \dots, \xi_N)$ 라고 하자.

두 가지 목적중 어느 것에 비중을 둘지를 결정하는 매개 변수.  
 $C = 0$ 이면 목적2를 무시한다.  $C = \infty$ 이면 목적 2만 고려한다.

$$J(\mathbf{w}, \Xi) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i$$

# Primal problem of soft margin SVM

따라서  $c$ 는 적절히 조절되어야 한다.

$$J(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2} \|\mathbf{w}\|^2 + c \sum_{i=1}^N \xi_i$$


목적함수가 주어졌으므로, 조건부 최적화 문제를 유도할 수 있다.

Primal problem (선형 분리 불가능한 상황)

$$\begin{aligned} \text{minimize} \quad & J(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 + c \sum_{i=1}^N \xi_i \\ \text{subject to} \quad & t_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \dots, N \\ & \xi_i \geq 0, \quad i = 1, \dots, N \end{aligned}$$

- 목적함수  $J(\mathbf{w})$ 는 convex함수이다.
- 조건식 모두 미분가능하다.

# Wolfe dual problem of soft margin SVM

Lagrangian

$$\mathcal{L}(\mathbf{w}, b, \Xi, \mathbf{a}, \mathbf{b}) = \left( \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \right) - \left( \sum_{i=1}^N \alpha_i (t_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i) + \sum_{i=1}^N \beta_i \xi_i \right)$$

•  $\mathbf{a} = (\alpha_1, \dots, \alpha_N), \mathbf{b} = (\beta_1, \dots, \beta_N), \Xi = (\xi_1, \dots, \xi_N)$

Wolfe dual problem

maximize  $\mathcal{L}(\mathbf{w}, b, \Xi, \mathbf{a}, \mathbf{b})$   
 $= \left( \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \right) - \left( \sum_{i=1}^N \alpha_i (t_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i) + \sum_{i=1}^N \beta_i \xi_i \right)$

subject to  $\mathbf{a} \geq 0, \mathbf{b} \geq 0, \frac{\partial \mathcal{L}}{\partial \mathbf{w}} = 0, \frac{\partial \mathcal{L}}{\partial b} = 0, \frac{\partial \mathcal{L}}{\partial \Xi} = 0$

$\alpha_i \geq 0, i = 1, \dots, N$   
 $\beta_i \geq 0, i = 1, \dots, N$   
 $\mathbf{w} = \sum_{i=1}^N \alpha_i t_i \mathbf{x}_i$   
 $\sum_{i=1}^N \alpha_i t_i = 0$   
 $C = \alpha_i + \beta_i$

# Wolfe dual problem of soft margin SVM

Wolfe dual problem

$$\begin{aligned} \text{maximize} \quad & \mathcal{L}(\mathbf{a}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j t_i t_j \mathbf{x}_i^T \mathbf{x}_j \\ \text{subject to} \quad & \sum_{i=1}^N \alpha_i t_i = 0 \\ & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, N \end{aligned}$$

- soft margin 최적화 문제의 솔루션은 훈련 샘플을 세 가지 경우로 분류
  - $\alpha_i = 0$  : 샘플들은 support vector이거나, margin 바깥쪽에 있다.
  - $0 < \alpha_i < C$  : 샘플들은 support vector
  - $\alpha_i = C$  : 샘플들은 support vector이거나, margin 안쪽에 있다.

# Wolfe dual problem of soft margin SVM

문제의 솔루션은 KKT 조건을 만족한다.

- $C = \alpha_i + \beta_i$
- $\beta_i \xi_i = 0$
- $\alpha_i (y_i (\mathbf{w}^T \mathbf{x}_i + b) - (1 - \xi_i)) = 0$
- $\xi_i \geq 0$

$$\alpha_i = C$$

- $\beta_i = 0, \xi_i \geq 0$
- $(y_i (\mathbf{w}^T \mathbf{x}_i + b) - (1 - \xi_i)) = 0$   
 $\Rightarrow y_i (\mathbf{w}^T \mathbf{x}_i + b) \leq 1$

$$\alpha_i = 0$$

- $\beta_i = C, \xi_i = 0$
- $(y_i (\mathbf{w}^T \mathbf{x}_i + b) - (1 - \xi_i)) \geq 0$   
 $\Rightarrow y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1$

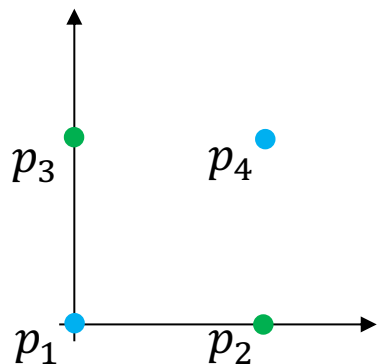
$$0 < \alpha_i < C$$

- $0 < \beta_i < C, \xi_i = 0$
- $(y_i (\mathbf{w}^T \mathbf{x}_i + b) - (1 - \xi_i)) = 0$   
 $\Rightarrow y_i (\mathbf{w}^T \mathbf{x}_i + b) = 1$

# 비선형 SVM의 Idea

실제 세계에서 발생하는 분류 문제에서는 선형 분류기로 높은 성능을 얻는다는 것은 가능성이 낮다.

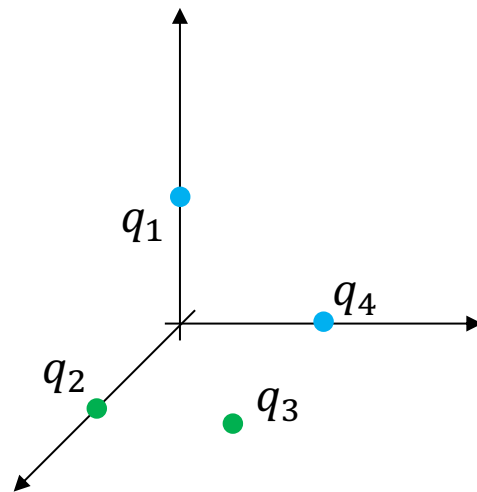
원래 특징 공간에서는 선형 분리 가능하지 않는데 이 공간을 더 높은 차원의 새로운 공간으로 매핑하여 선형 분리 가능하게 만들 수 있다.



원래 공간 : 선형 분리 불가능

$$\Phi: L \rightarrow H$$

$$\Phi(p_i) = q_i$$



변형된 공간 : 선형 분리 가능

# Kernel Function

공간  $L$ 에서 공간  $H$ 로의 공간 매핑

$$\underline{\Phi: L \rightarrow H}$$

- 보통, 공간  $H$ 의 차원은  $L$ 보다 훨씬 높다.

**커널 함수**(Kernel function)

- 두 벡터  $\mathbf{x}, \mathbf{y} \in L$
- 두 벡터  $\Phi(\mathbf{x}), \Phi(\mathbf{y}) \in H$
- 함수  $K: L \times L \rightarrow R$ (실수공간)

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

- 커널 함수의 값과 매핑된 두 벡터의 내적의 값이 같아야 한다.

# Kernel Function 예제

$$\mathbb{x} = (x_1, x_2)^T, \mathbb{y} = (y_1, y_2)^T, \mathbb{x}, \mathbb{y} \in L$$

$$K(\mathbb{x}, \mathbb{y}) = (\mathbb{x} \cdot \mathbb{y})^2 = x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2$$

$$\Phi(\mathbb{x}) = (x_1^2, \sqrt{2}x_1 x_2, x_2^2) \in H$$

$$\begin{aligned} K(\mathbb{x}, \mathbb{y}) &= (\mathbb{x} \cdot \mathbb{y})^2 \\ &= x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2 \\ &= (x_1^2, \sqrt{2}x_1 x_2, x_2^2)(y_1^2, \sqrt{2}y_1 y_2, y_2^2) \\ &= \Phi(\mathbb{x}) \cdot \Phi(\mathbb{y}) \end{aligned}$$



# Kernel substitution(trick)

## Kernel substitution

- Kernel trick이라고도 한다.
- 어떤 수식이 벡터의 내적을 포함하고 있을 때, 그 내적을 커널 함수로 대체하여 계산하는 기법이다.
- 실제 계산은  $L$ 공간에서 커널 함수  $K$ 의 계산으로 이루어진다. 하지만 실제로는  $\Phi$ 로 매핑된 고차원 공간  $H$ 에서 작업하는 효과를 얻는다.
- 차원의 저주를 피해가는 셈이다.
- 내적 연산인 경우에만 사용할 수 있다.

# SVM-커널 대치 도입

선형 분류기로 원래 특징 공간  $L$ 에서 높은 성능을 기대하기 어렵다면  $L$ 공간에서 작업하는 대신 매핑 함수  $\Phi$ 로 더 높은 차원의 (즉, 선형 분리가 더 유리한) 새로운 공간  $H$ 로 매핑하고  $H$ 에서 작업

공간  $L$ 에서의 SVM 분류기는 아래의 목적함수를 이용

$$\mathcal{L}(\mathbf{a}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j t_i t_j \mathbf{x}_i^T \mathbf{x}_j$$

- $\mathbf{x}_1, \mathbf{x}_2 \in L, \mathbf{a} = (\alpha_1, \dots, \alpha_N)$

공간  $H$ 로 매핑되면, SVM 분류기는 아래의 목적함수를 이용

$$\mathcal{L}(\mathbf{a}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j t_i t_j \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$$

- $\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2) \in H$
- $\text{dom} K \in L$

실제 계산에 사용되는 함수  
계산은 원래 공간에서 수행

$$\sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j t_i t_j K(\mathbf{x}, \mathbf{y})$$

목표로 하는 고차원에서의 계산을, 커널함수를 사용하여 저차원에서 수행.  
고차원에서의 결과값과 동일함을 달성

# SVM-커널 함수의 종류

다항식 커널

$$K(\mathbf{x}, \mathbf{y}) = (\gamma \mathbf{x} \cdot \mathbf{y} + \gamma)^p$$

Gaussian Radial Basis Function 커널

$$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{\gamma \|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}}$$

하이퍼볼릭 탄젠트 커널 (Sigmoid 커널)

$$K(\mathbf{x}, \mathbf{y}) = \tanh(\alpha \mathbf{x} \cdot \mathbf{y} + \beta)$$

- 모든  $\alpha, \beta$ 에 대하여 매핑함수가 존재하지는 않는다.  $\alpha = 2, \beta = 1$ 이 적절한 값이다.

# Mercer Theorm

머서의 정리에 따르면 함수  $K(\mathbf{a}, \mathbf{b})$ 가 Mercer's condition를 만족할 때,  $\mathbf{a}$ 와  $\mathbf{b}$ 를 더 높은 차원의 다른 공간에 매핑하는  $K(\mathbf{a}, \mathbf{b}) = \phi(\mathbf{a}^T)^T \phi(\mathbf{b})$ 와 같은 함수  $\phi$ 가 존재.  $\phi$ 를 모르더라도  $\phi$ 가 존재하는 것은 알기 때문에  $K$ 를 커널로 사용할 수 있음.

- $K$ 가 매개변수에 대해 연속
- $K$ 는 대칭  $K(\mathbf{a}, \mathbf{b}) = K(\mathbf{b}, \mathbf{a})$

Gaussian RBF 커널의 경우  $\phi$ 는 각 훈련 샘플을 무한 차원의 공간에 매핑하기 때문에, 실제로 매핑을 하여 볼 수 없음.

자주 사용되는 Sigmoid 커널은 머서의 조건을 모두 따르지 않지만 일반적인 실전에서는 잘 작동함.

# M부류 SVM

## 1대 M-1방법

- $M$ 개의 이진 분류기를 생성
  - $j$ 번째 이진 분류기는 부류  $c_j$ 와 나머지  $M - 1$ 개 부류를 분류
  - $c_j$ 에 속하는 샘플은 양의 영역, 나머지 부류에 속하면 음의 영역으로 구분
- 결정 초평면 함수의 값이 가장 큰 부류로 분류
  - $k = \arg \max_j d_j(\mathbf{x})$

## 1대 1방법

- 부류 쌍을 분류하는 이진 분류기를  $\frac{M(M-1)}{2}$ 개 생성
  - $d_{ij}(\mathbf{x})$ 는  $c_i$ 와  $c_j$ 를 분류하는 이진 분류기의 결정 초평면
  - $\mathbf{x}$ 에 대하여  $d_{ij}(\mathbf{x})$ 가  $c_i$ 로 분류하면 부류  $c_i$ 이 한표, 반대면  $c_j$ 가 한표를 얻음
  - 2번 스텝을  $\frac{M(M-1)}{2}$ 번 반복하여 가장 많은 표를 얻은 부류로  $\mathbf{x}$ 를 분류

# How to compute $\mathbf{a} = (\alpha_1, \dots, \alpha_N)$

Sequential Minimal Optimization (순차적 최소 최적화) 를 이용한다.

# SMO의 기본 아이디어

커다란 최적화 문제를, 작은 문제들로 나누어 해결한다.

- 작은 문제들은 쉽게 해결할 수 있으며, 순차적으로 해결된다.
- 순차적으로 해결된 문제들의 답은 모두 같게 되고, 모든 문제를 다함께 처리한 것과 같은 효과가 있다.

커다란 최적화 문제 :  $\mathbf{a} = (\alpha_1, \dots, \alpha_N)$ 의 최적화

나누어진 작은 문제 : 적당한  $\alpha_i, \alpha_j$ 의 쌍을 최적화

# SMO의 기본 아이디어

[Platt이 제안한 알고리즘](#) (강의노트에서는 simple version을 다루겠다.)

$$\sum_{i=1}^N \alpha_i t_i = 0$$

수학적 아이디어는 Coordinate ascent를 사용한다.

```
Loop until convergence : {  
    For  $i = 1, \dots, m$ , {  
         $\alpha_i := \arg \max_{\hat{\alpha}} W(\alpha_1, \dots, \alpha_{i-1}, \hat{\alpha}, \alpha_{i+1}, \dots, \alpha_N)$   
    }  
}
```

- $\alpha_i$ 의 최적값을 찾기위해,  $\alpha_i$ 를 제외한 값들을 상수값으로 고정시킨 후,  $W$ 를 optimize 시킨다.



# SMO의 기본 아이디어

초기값을 0으로 하는  $\mathbf{a} = (\alpha_1, \dots, \alpha_N)$  생성

$$\sum_{i=1}^N \alpha_i t_i = 0$$

(while) 반복 횟수  $M$ 이 임계값보다 작을 때 반복

(for)  $\alpha_i, i = 1, \dots, N$  에 대해 반복 (라그랑지 승수에 대해)

라그랑지 승수  $\alpha_i$ 가 최적화 될 수 있다면

임의로 다른 라그랑지 승수  $\alpha_j$ 를 선택

## Appendix 1

새로운  $\alpha_j$ 를 계산 (새로운  $\alpha_j$ 는 목적함수의 optimization으로부터 나옴)

$\alpha_j$ 를 최적화 할 수 없다면 next  $i$  continue

## Appendix 2

$\alpha_j$ 를 최적화 할 수 있다면  $\alpha_i, b$ 를 새롭게 계산

최적화된 라그랑지 승수의 쌍이 하나도 없다면, 반복 횟수  $M$ 을 1 증가 시킨다.

# Simplified SMO 알고리즘

$\alpha_i, \alpha_j$ 를 선택

- $\alpha_i$ 가 최적화 될 수 있는 조건
  - 라그랑지 승수  $\alpha_i$ 에 대응되는 샘플의 참 값( $t_i$ )과, SVM계산 결과( $\sum_{j=1}^N \alpha_j t_j \mathbb{x}_j^T \mathbb{x}_i + b$ )의 차이가 허용 가능한 수치(tolerance) 이상인 경우
  - 이 경우에  $\alpha_j$ 를 random하게 선택

새로운  $\mathfrak{a}$ 에 대하여  $b$ 를 구한다.

- 새로운  $\alpha_i$ 와  $\alpha_j$ 가 적용된  $\mathfrak{a}$

$\mathfrak{a}$ 가 수렴할 때까지 위의 과정을 반복한다.

- 최적화되는(변경되는) 라그랑지 승수의 쌍이 발견되지 않는 상황이, 미리 설정해 놓은 임계치 이상 반복될 때.

# Summary

여백(margin) 개념을 분류기 설계에 도입하고 여백을 극대화하는 결정 초평면을 찾아내는 것  $\Rightarrow$  뛰어난 일반화 능력을 확보하는 것

커널과 관련된 매개 변수,  $C$  값을 다양하게 설정하여 성능 실험을 하고 그 중 가장 뛰어난 값을 선택 (휴리스틱)

SVM은 일반화 능력이 뛰어남.

# Appendix<sub>(Advanced)</sub>

$\alpha_j$  구하기

$\alpha_i, b$  구하기

# $\alpha_j$ 구하기

$0 < \alpha_i < C$ 를 만족시키기 위하여 새로운  $\alpha_j$ 는 다음의 범위를 만족시켜야 한다.

- $\alpha_i + \alpha_j = \gamma \ (t_i = t_j)$

- $\gamma > C$

- $\gamma - C \leq \alpha_j \leq C$

- $\gamma < C$

- $0 \leq \alpha_j \leq \gamma$

$$L = \max(0, \alpha_j + \alpha_i - C), H = \min(C, \alpha_j + \alpha_i)$$

- $\alpha_i - \alpha_j = \gamma \ (t_i \neq t_j)$

- $\gamma > 0$

- $0 \leq \alpha_j \leq C - \gamma$

- $\gamma < 0$

- $-\gamma \leq \alpha_j \leq C$

$$L = \max(0, \alpha_j - \alpha_i), H = \min(C, C + \alpha_j - \alpha_i)$$

# $\alpha_j$ 구하기

From here!

$$\mathcal{L}(\mathfrak{a}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j t_i t_j \mathbb{X}_i^T \mathbb{X}_j$$

$\alpha_i$ 와  $\alpha_j$ 를 제외하고는 모두 고정된 상수로 표현

$$t_i \alpha_i + t_j \alpha_j = \text{Const.}$$

위의 식을 목적함수에 대입하여  $\alpha_j$ 의 2차식으로 표현

1계 도함수, 2계 도함수 사용

# $\alpha_j$ 구하기

$$\alpha_j := \overset{\alpha_j^{old}}{\alpha_j} - \frac{t_j(E_i - E_j)}{\eta}$$

where

$$E_j = \sum_{i=1}^N \alpha_i t_i \mathbb{x}_i^T \mathbb{x}_j + b - t_j$$
$$\eta = 2\mathbb{x}_i^T \mathbb{x}_j - \mathbb{x}_i \mathbb{x}_i - \mathbb{x}_j \mathbb{x}_j$$

$$\alpha_j := \begin{cases} H & \text{if } \alpha_j > H \\ \alpha_j & \text{if } L \leq \alpha_j \leq H \\ L & \text{if } \alpha_j < L \end{cases}$$

새로운  $\alpha_j$

# $\alpha_i, b$ 구하기

새로운  $\alpha_i$   $\alpha_i^{old}$

$$\alpha_i := \alpha_i + t_i t_j (\alpha_j^{old} - \alpha_j)$$

$$b_1 = b - E_i - t_i (\alpha_i - \alpha_i^{old}) \mathbb{X}_i^T \mathbb{X}_i - t_j (\alpha_j - \alpha_j^{old}) \mathbb{X}_i^T \mathbb{X}_j$$

$$b_2 = b - E_j - t_i (\alpha_i - \alpha_i^{old}) \mathbb{X}_i^T \mathbb{X}_j - t_j (\alpha_j - \alpha_j^{old}) \mathbb{X}_j^T \mathbb{X}_j$$

새로운  $b$

$$b := \begin{cases} b_1 & \text{if } 0 < \alpha_i < C \\ b_2 & \text{if } 0 < \alpha_j < C \\ \frac{b_1 + b_2}{2} & \text{if otherwise} \end{cases}$$