

투빅스 3주차 교육

ToBig's 8기 김은서

# Support Vector Machine

# contents

---

Unit 01 | SVM Intro

---

Unit 02 | Hyperplane

---

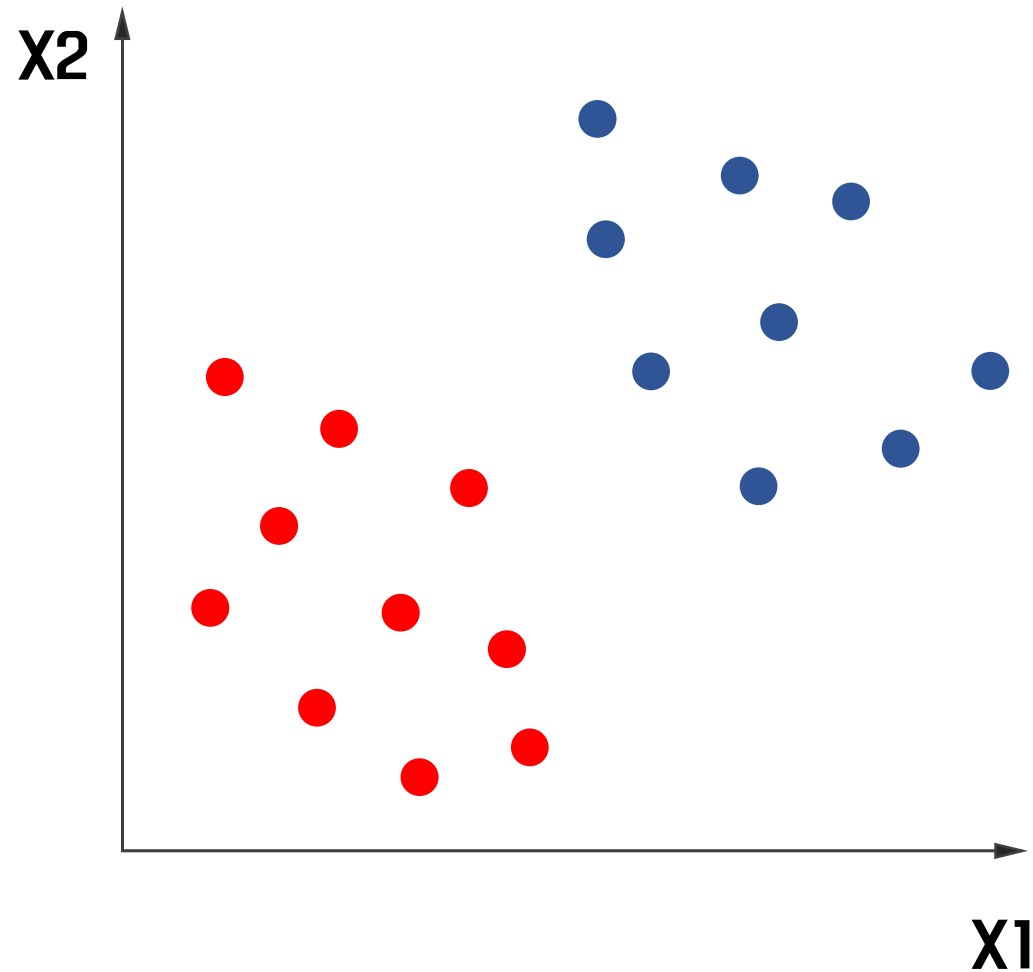
Unit 03 | SVM의 종류

---

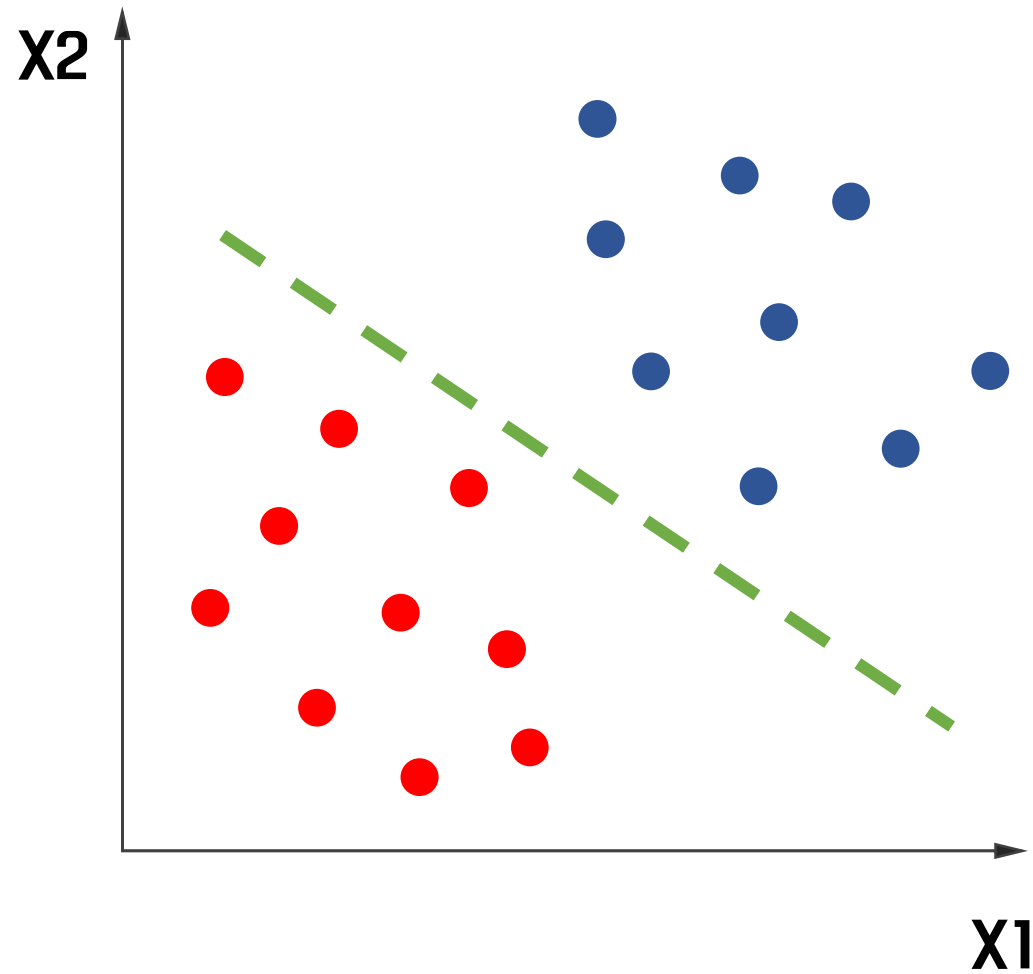
Unit 04 | Kernel

---

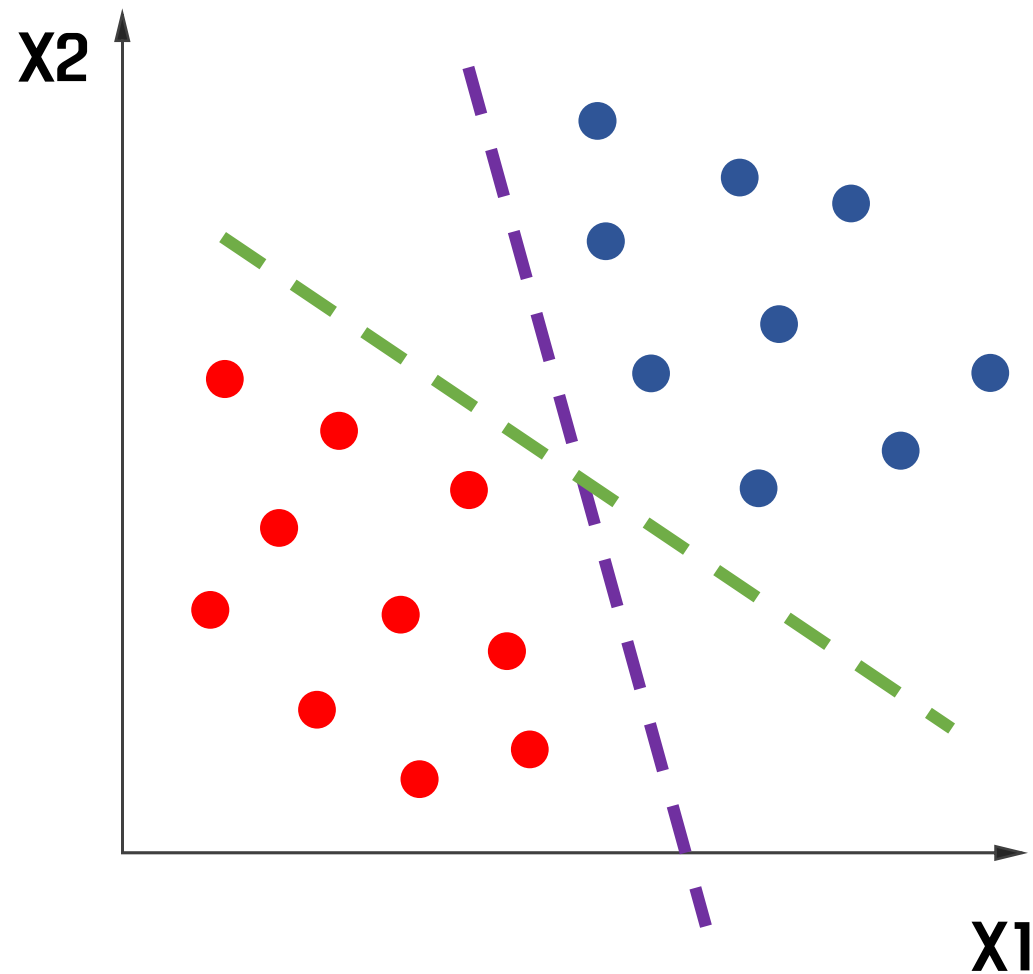
## Unit 01 | SVM Intro



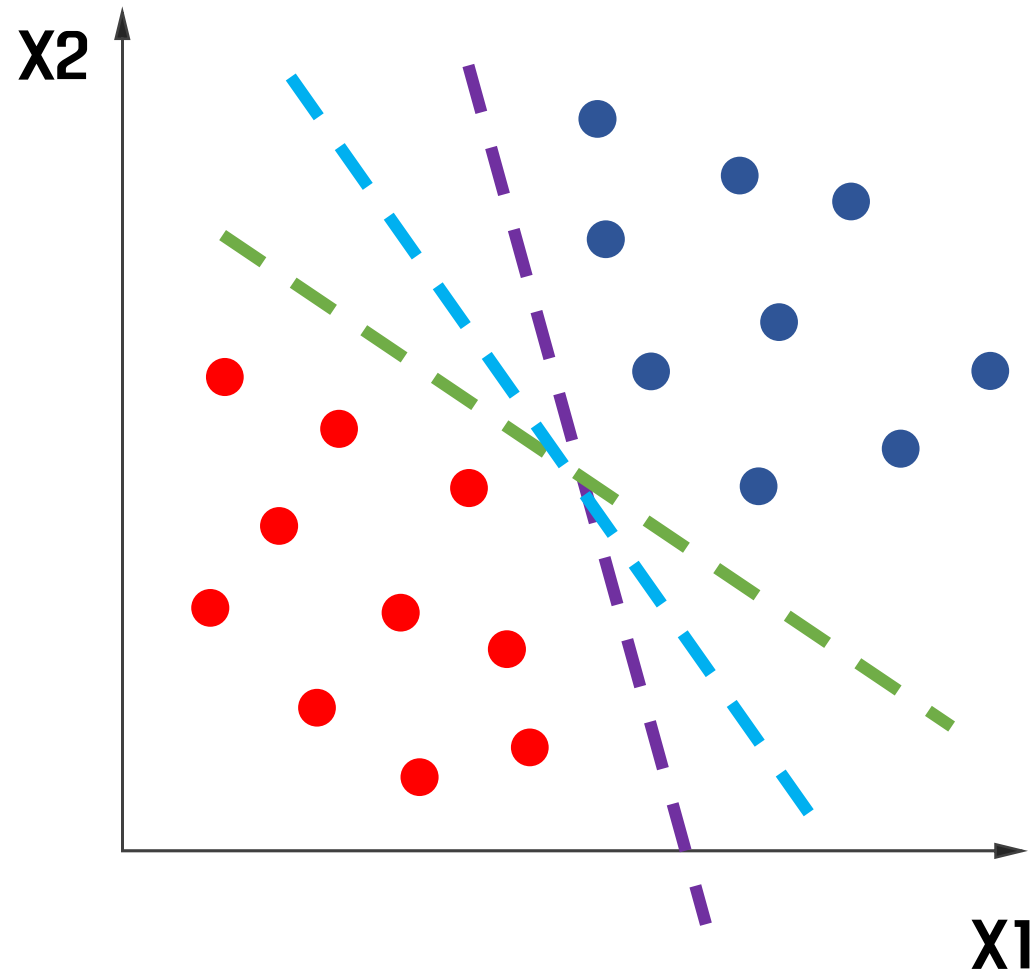
## Unit 01 | SVM Intro



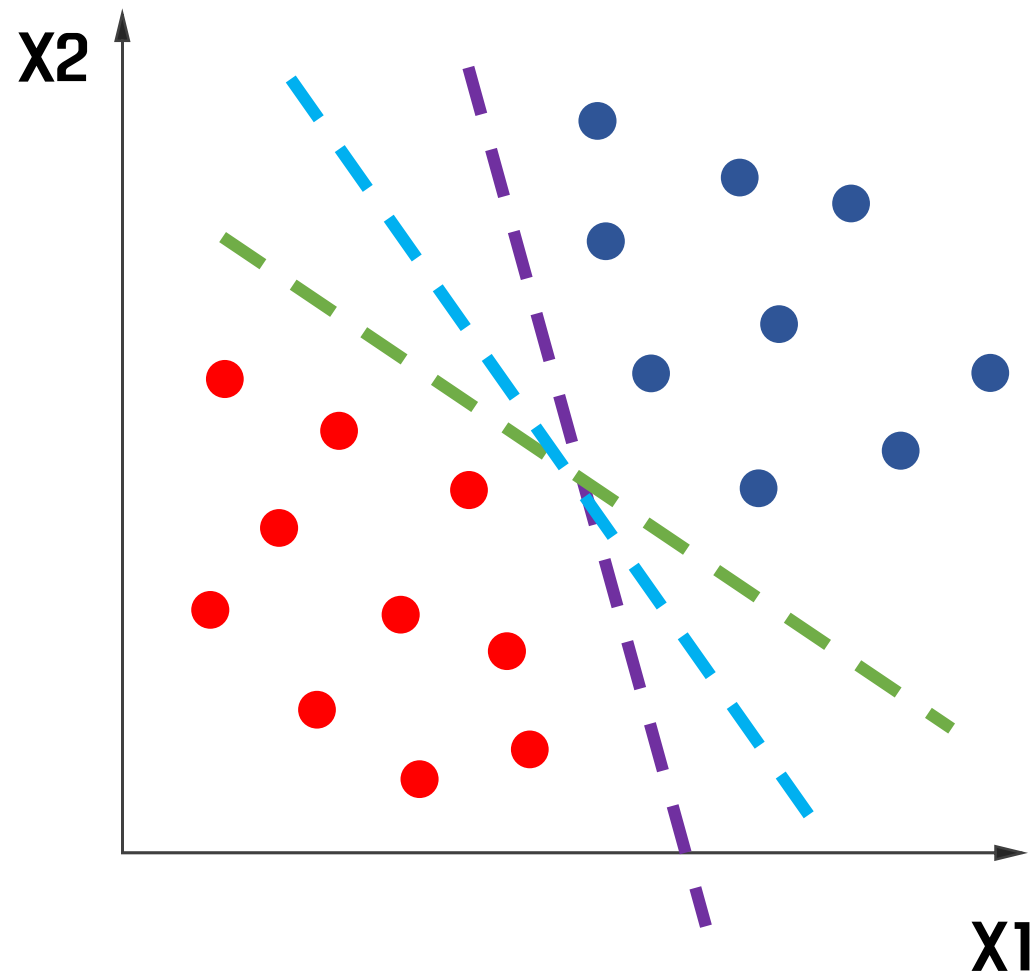
## Unit 01 | SVM Intro



## Unit 01 | SVM Intro

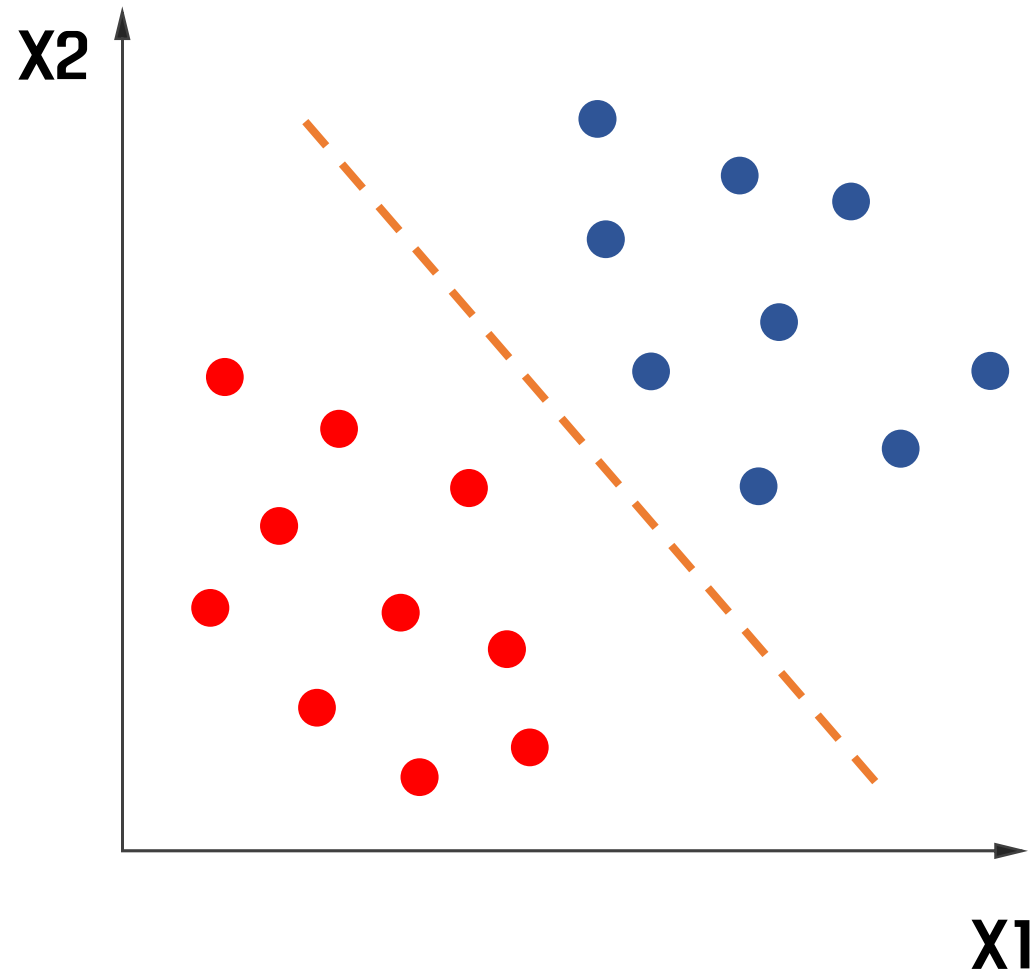


## Unit 01 | SVM Intro



어떤 게 <sup>.</sup><sub>.</sub> 좋은 분류기일까?

## Unit 01 | SVM Intro

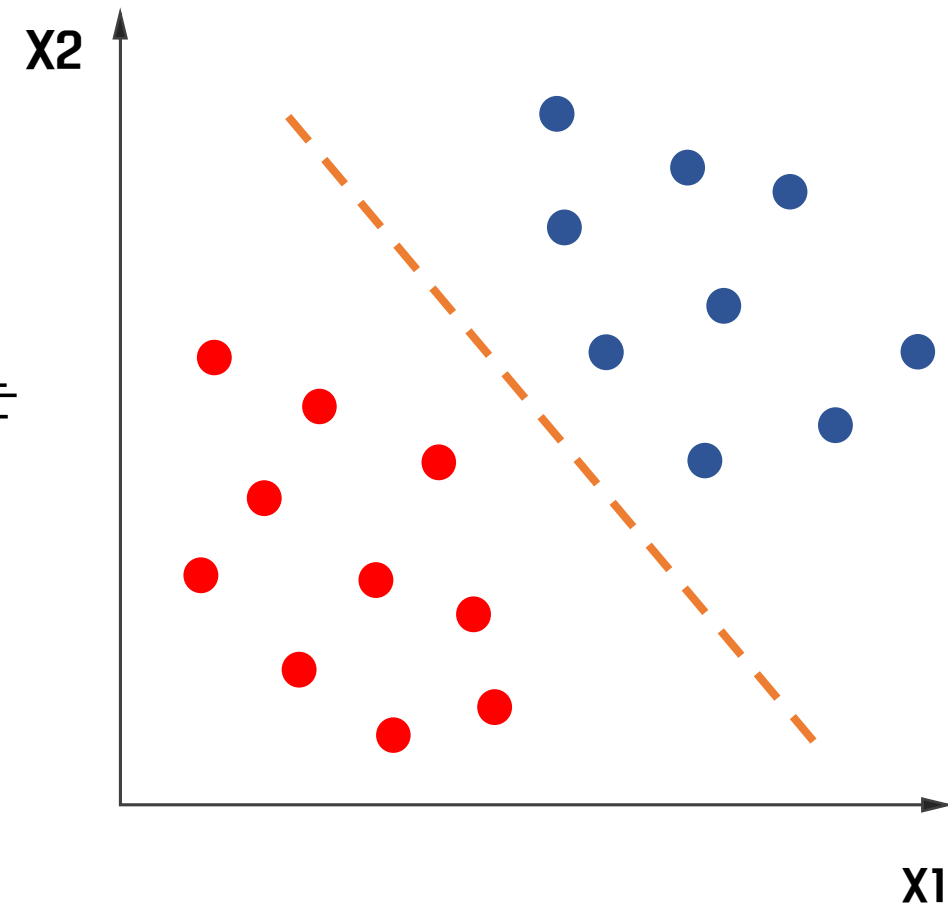




## Unit 01 | SVM Intro

# Support Vector Machine

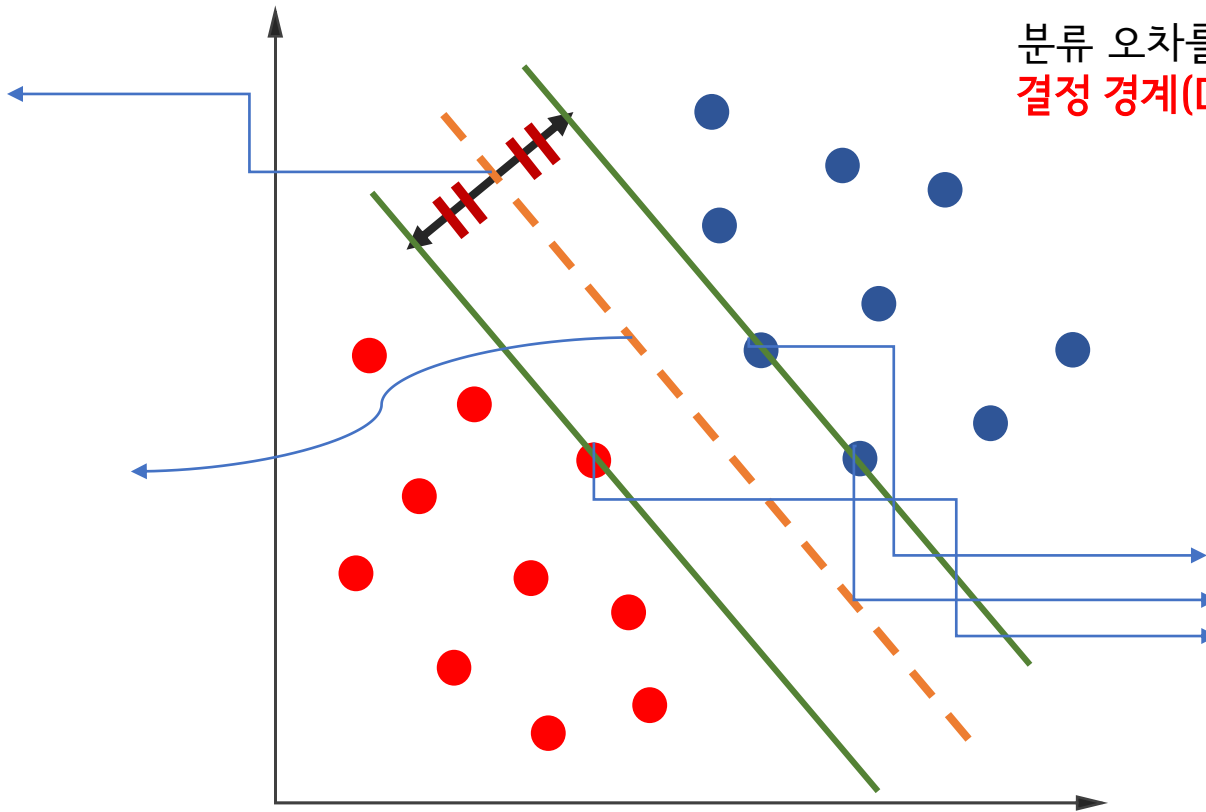
분류 오차를 줄이면서 동시에 **여백(Margin)**을 **최대로** 하는  
**결정 경계(Decision Boundary)**를 찾는 분류기



## Unit 01 | SVM Intro

### Support Vector Machine

분류 오차를 줄이면서 동시에 **여백(Margin)**을 **최대로** 하는  
**결정 경계(Decision Boundary)**를 찾는 분류기



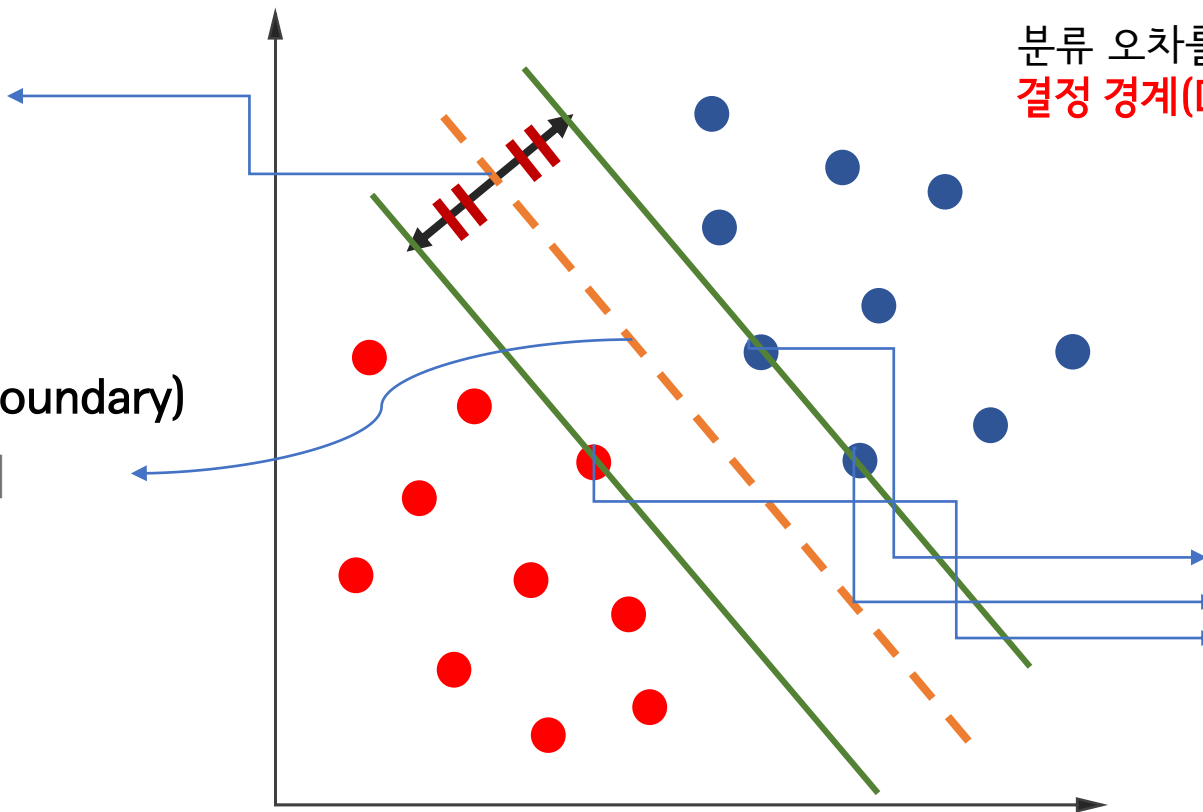
## Unit 01 | SVM Intro

### Support Vector Machine

분류 오차를 줄이면서 동시에 **여백(Margin)**을 **최대로** 하는  
**결정 경계(Decision Boundary)**를 찾는 분류기

결정 경계(Decision boundary)

분류 기준이 되는 경계  
(Hyperplane)



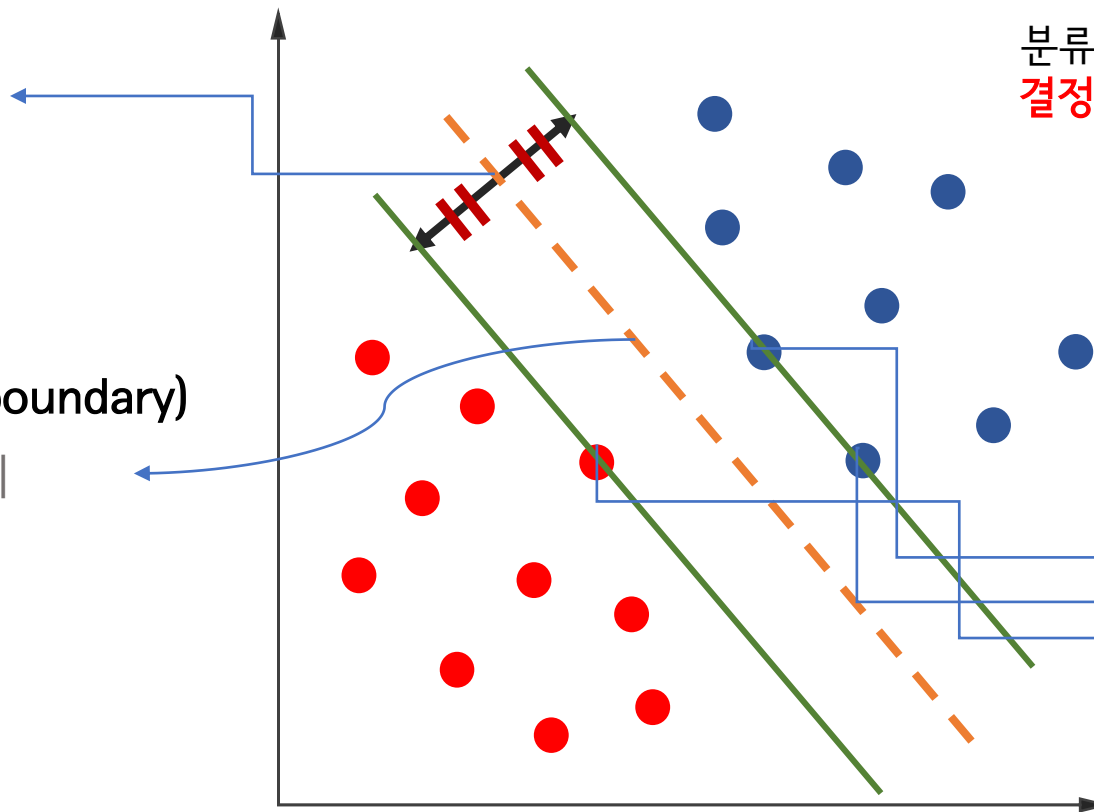
## Unit 01 | SVM Intro

### Support Vector Machine

분류 오차를 줄이면서 동시에 **여백(Margin)**을 **최대로** 하는  
**결정 경계(Decision Boundary)**를 찾는 분류기

결정 경계(Decision boundary)

분류 기준이 되는 경계  
(Hyperplane)



Support Vectors

결정 경계와 가장 가까운 벡터

## Unit 01 | SVM Intro

여백 (Margin)

결정 경계와  
서포트 벡터 사이의 거리  $\times 2$

결정 경계(Decision boundary)

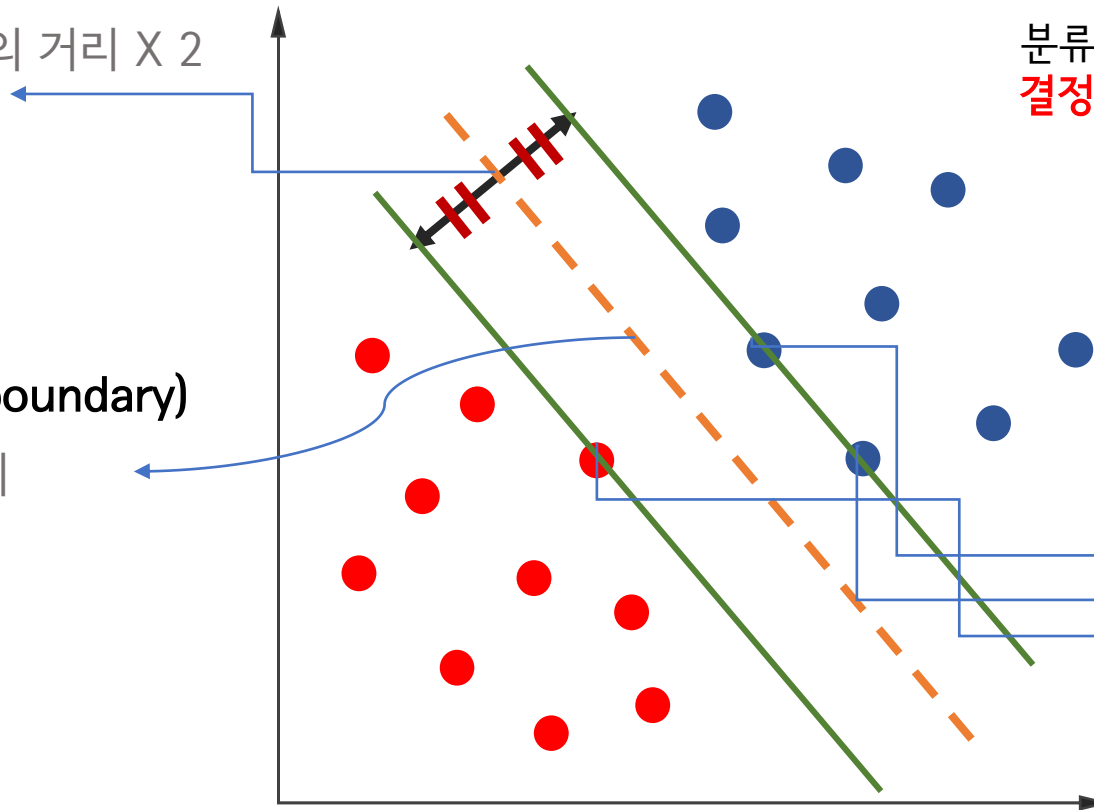
분류 기준이 되는 경계  
(Hyperplane)

## Support Vector Machine

분류 오차를 줄이면서 동시에 **여백(Margin)**을 **최대로** 하는  
**결정 경계(Decision Boundary)**를 찾는 분류기

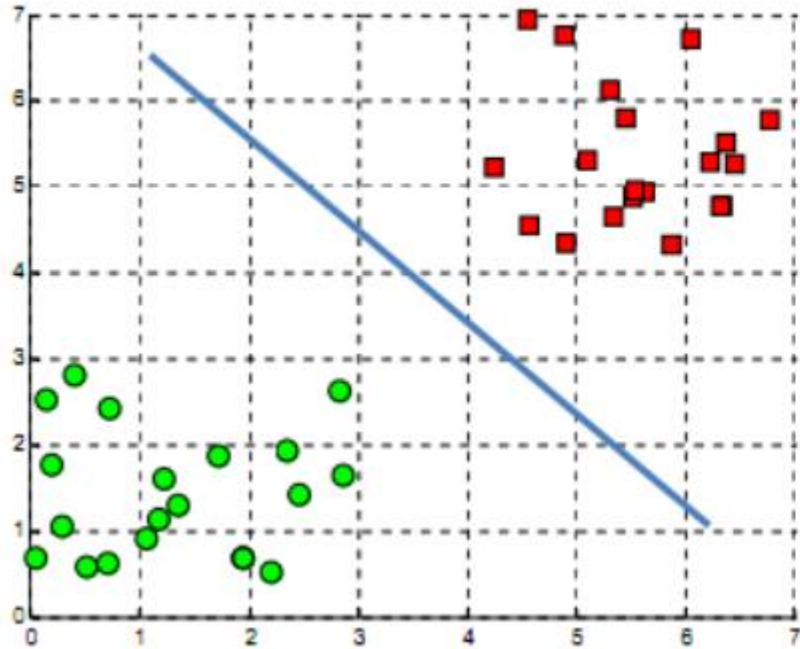
Support Vectors

결정 경계와 가장 가까운 벡터

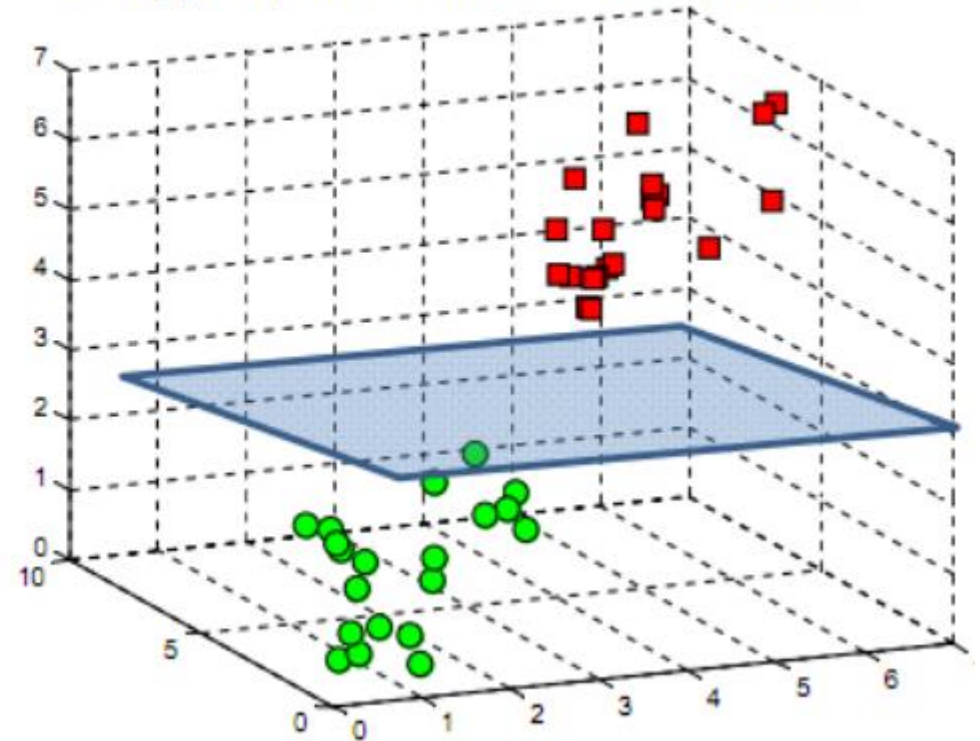


## Unit 02 | Hyperplane

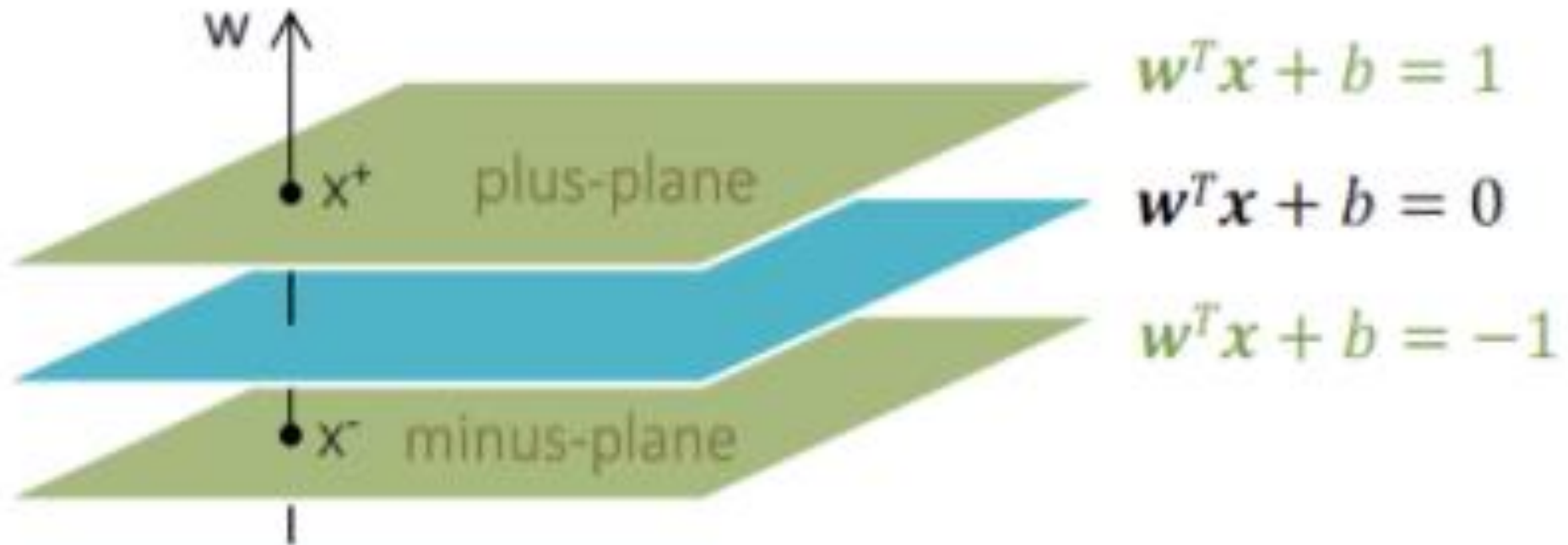
A hyperplane in  $\mathbb{R}^2$  is a line



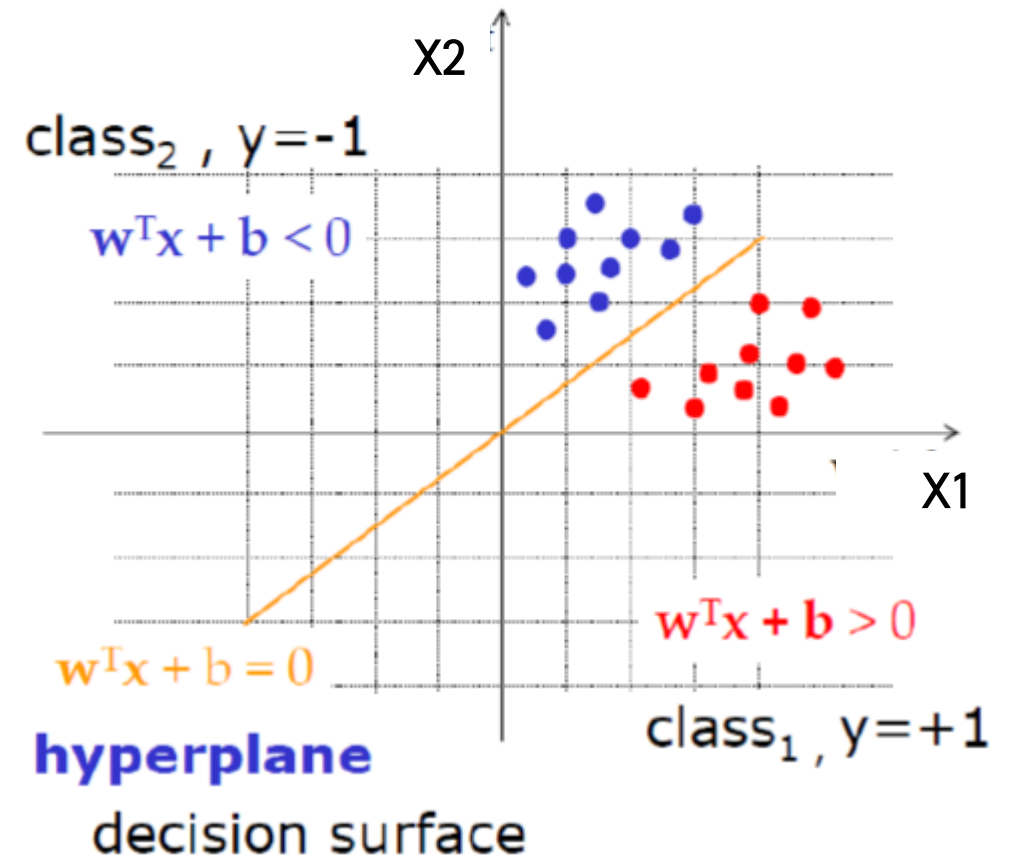
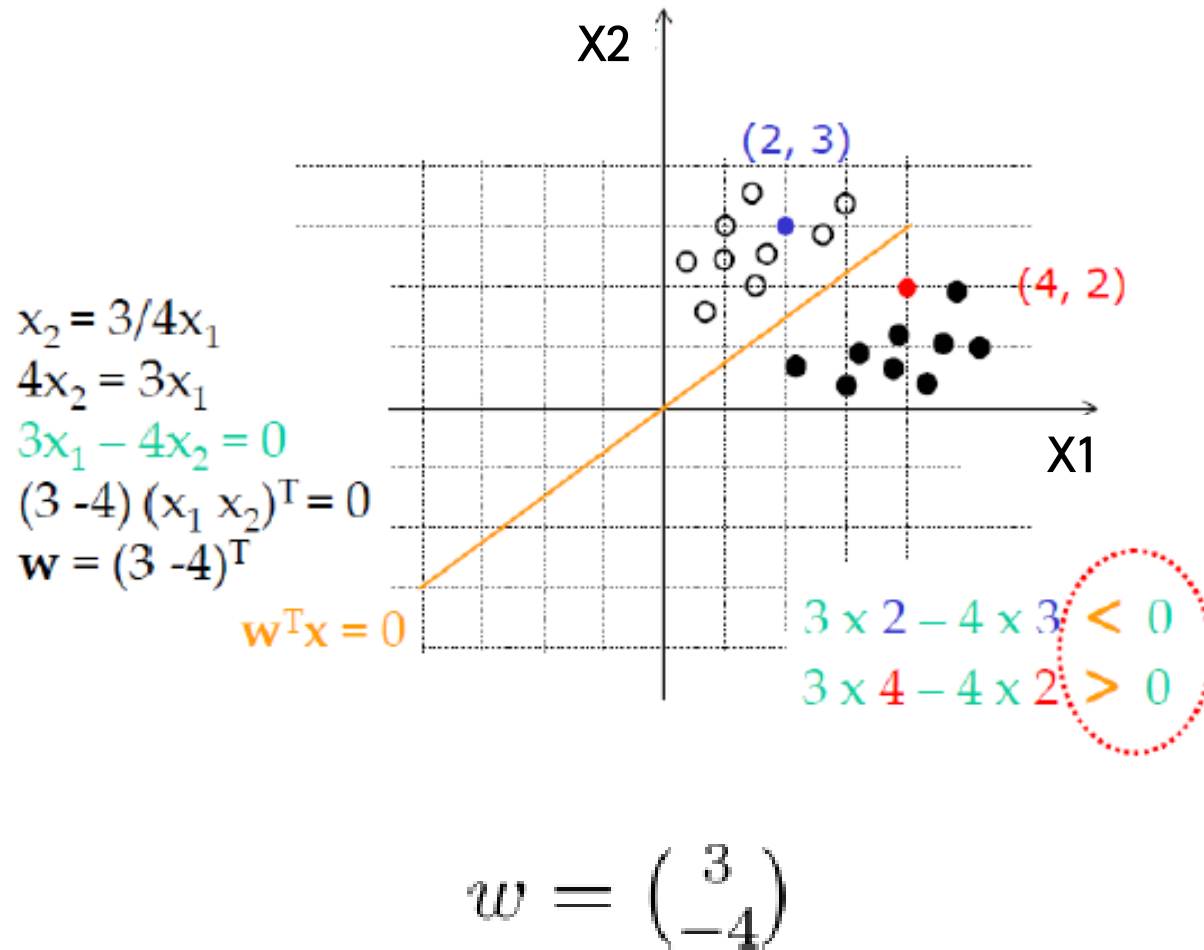
A hyperplane in  $\mathbb{R}^3$  is a plane



## Unit 02 | Hyperplane



## Unit 02 | Hyperplane





## Unit 02 | Hyperplane

Let's say

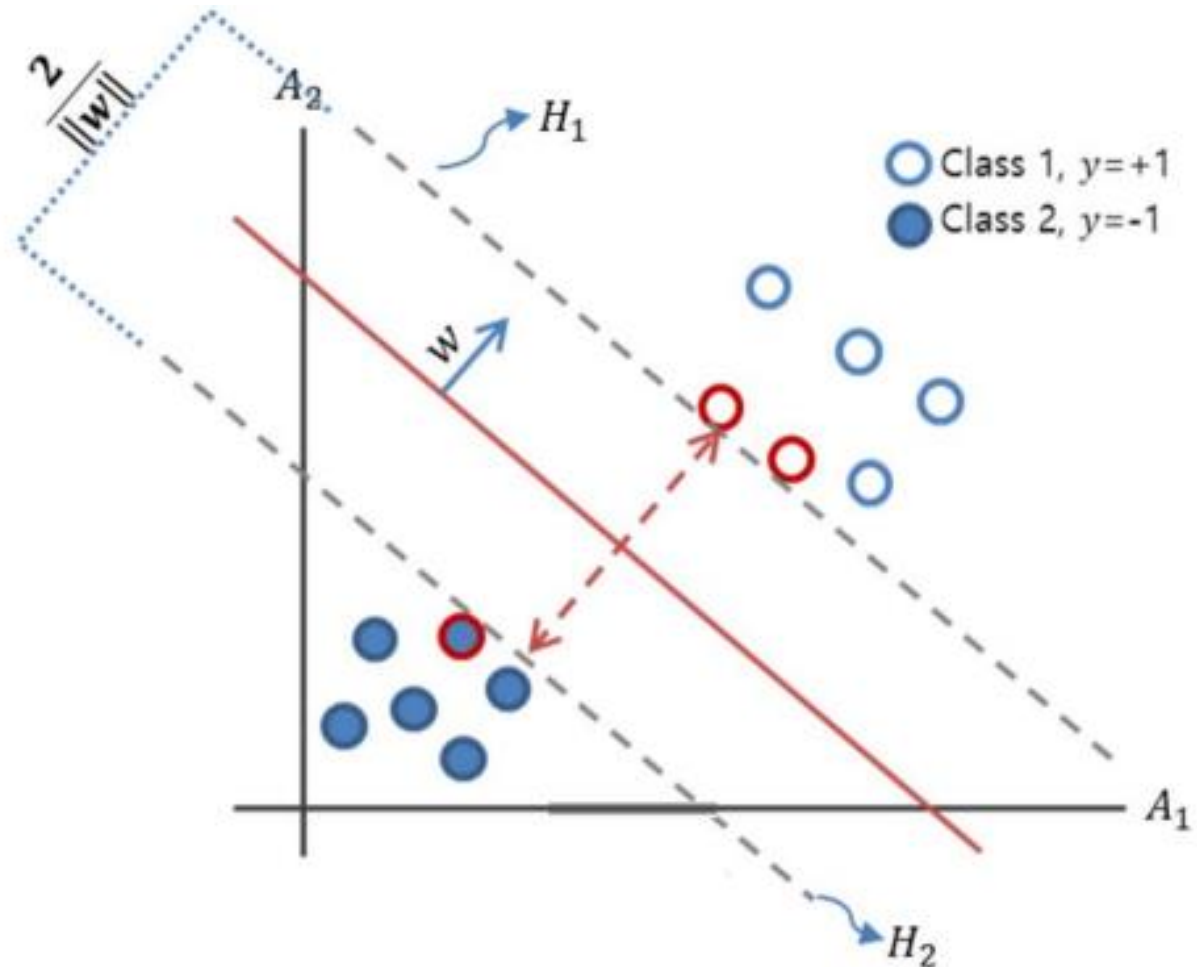
$$f(x) = w * x + b$$

A point  $x$  on the boundary has

$$f(x) = w * x + b = 0$$

A positive point  $x$  has

$$f(x) = w * x + b = a, \quad a > 0$$

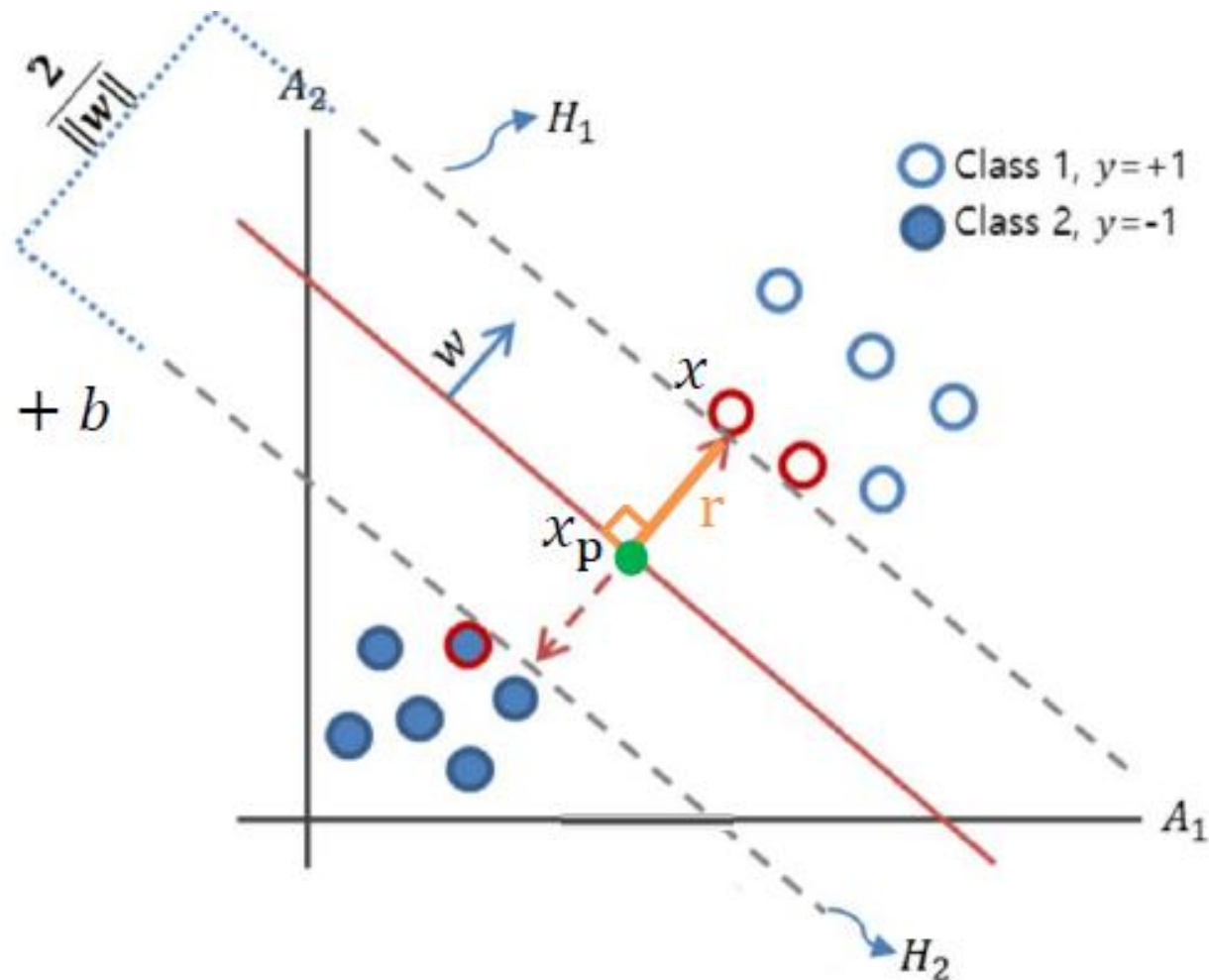


## Unit 02 | Hyperplane

$$x = x_p + r * \frac{w}{||w||}, \quad f(x_p) = 0$$

$$\begin{aligned} f(x) &= w * x + b = w \left( x_p + r * \frac{w}{||w||} \right) + b \\ &= w * x_p + b + r * \frac{w * w}{||w||} \\ &= r * ||w|| = a \end{aligned}$$

$$\text{거리} |r| = \frac{f(x)}{||w||} = \frac{a}{||w||}$$



## Unit 02 | Hyperplane

$$\text{거리 } r = \frac{f(x)}{\|w\|} = \frac{a}{\|w\|}$$

여기서  $a$  는 임의의 상수,  
 $a = 1$  해도 문제 X

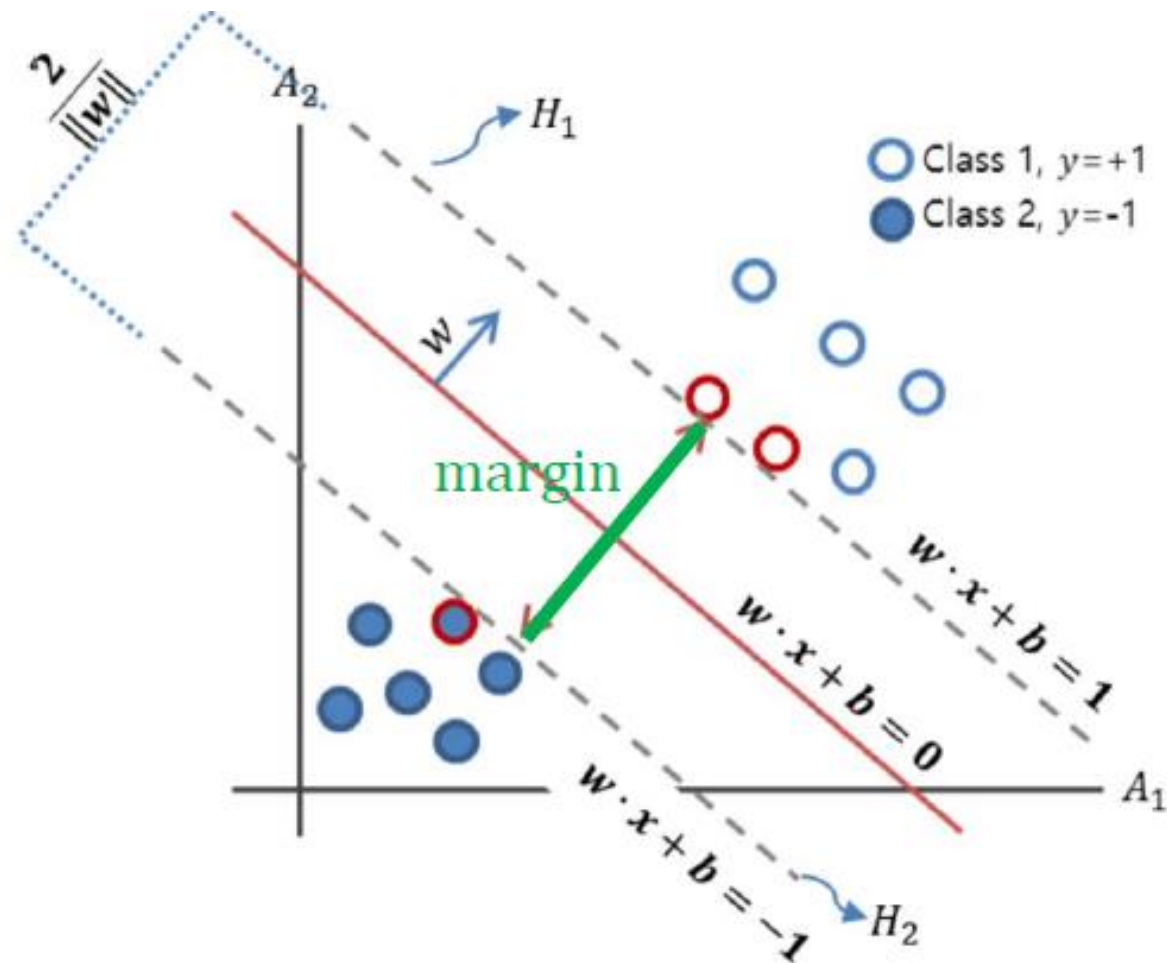
$$\text{margin} = 2r = \frac{2}{\|w\|}$$

SVM에서 목적은 바로 이 **margin**의 최대화

→  $\|w\|$ 의 최소화

→  $\frac{\|w\|^2}{2}$ 의 최소화

- Quadratic programming으로 해결가능



## Unit 03 | SVM의 종류

**분류 기준 #1**  
**Error 허용 여부**

**Hard Margin SVM**  
Error case가 하나도 없는 SVM

**Soft Margin SVM**  
Error case를 허용하는 SVM

**분류 기준 #2**  
**결정 경계의 형태**

**선형 SVM**

**비선형 SVM**

## Unit 03 | SVM의 종류

### Hard Margin SVM

Error case가 하나도 없는 SVM

### Soft Margin SVM

Error case를 허용하는 SVM

## Unit 03 | SVM의 종류

### Hard Margin SVM

Error case가 하나도 없는 SVM

### Soft Margin SVM

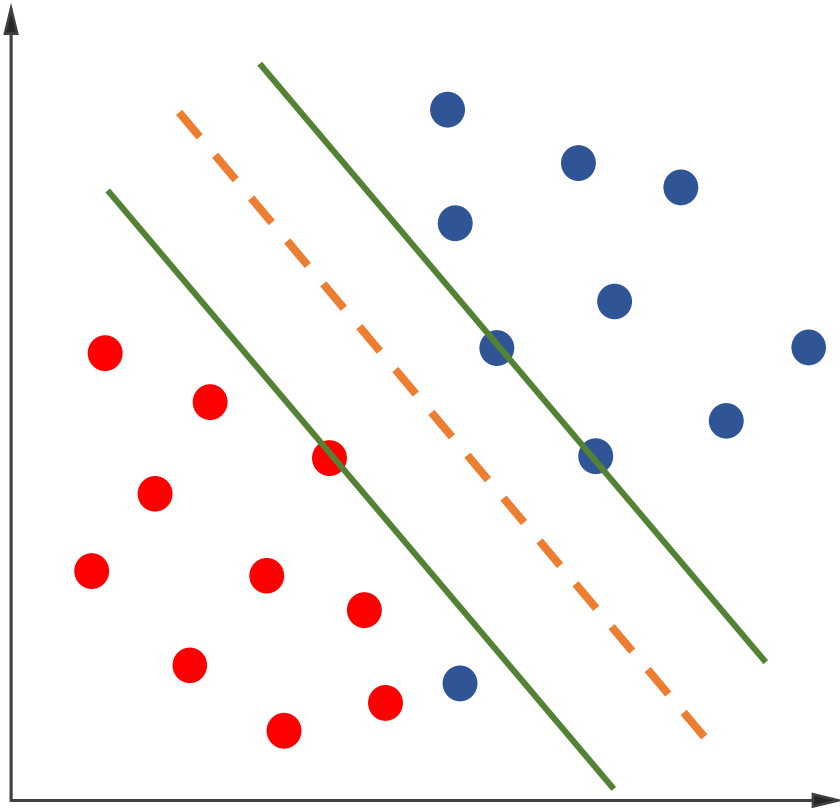
Error case를 허용하는 SVM

지금까지 설명한 SVM 방식

- ✓ 매우 엄격하게 두 개의 클래스를 분리하는 초평면을 구하는 방법.
- ✓ 모든 값들은 이 초평면을 사이에 두고 무조건 한 클래스에 속해야 하는데, 몇 개의 노이즈로 인해 두 그룹을 구별하는 초평면을 잘 못 구할 수도 있고, 찾지 못하는 경우도 발생!

→ 설계는 가능하나 현실에서는 적용 힘들.

## Unit 03 | SVM의 종류



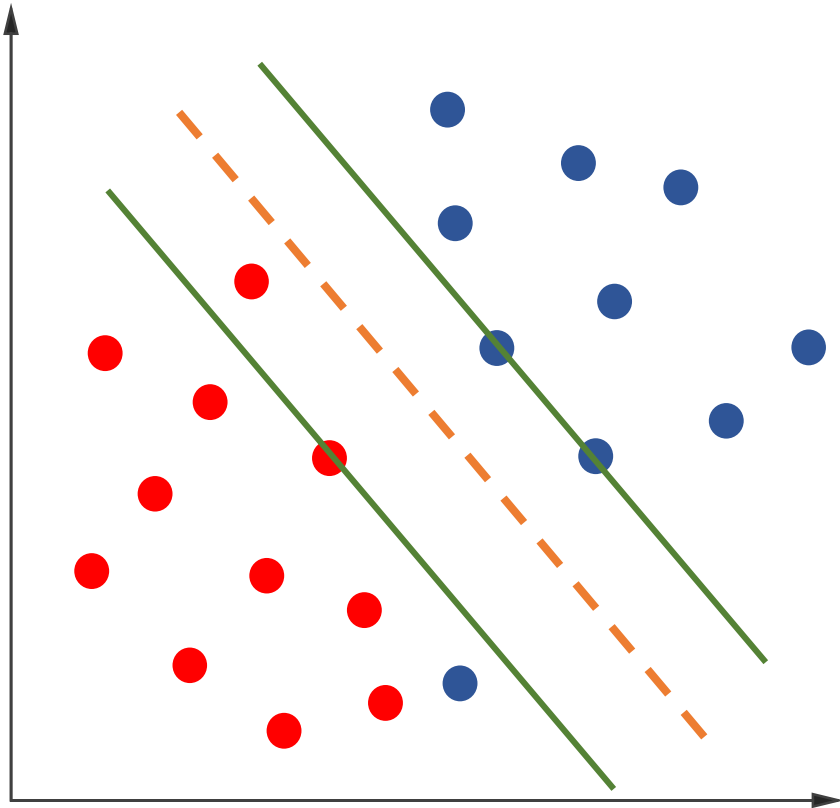
**Soft Margin SVM**  
Error case를 허용하는 SVM

**에러를 어느 정도 인정하고 최소화 시키도록!**

**→ Loss function 사용해서 계산.**

**두 가지 방법!!**

## Unit 03 | SVM의 종류



### <Soft Margin SVM>

Option 1)

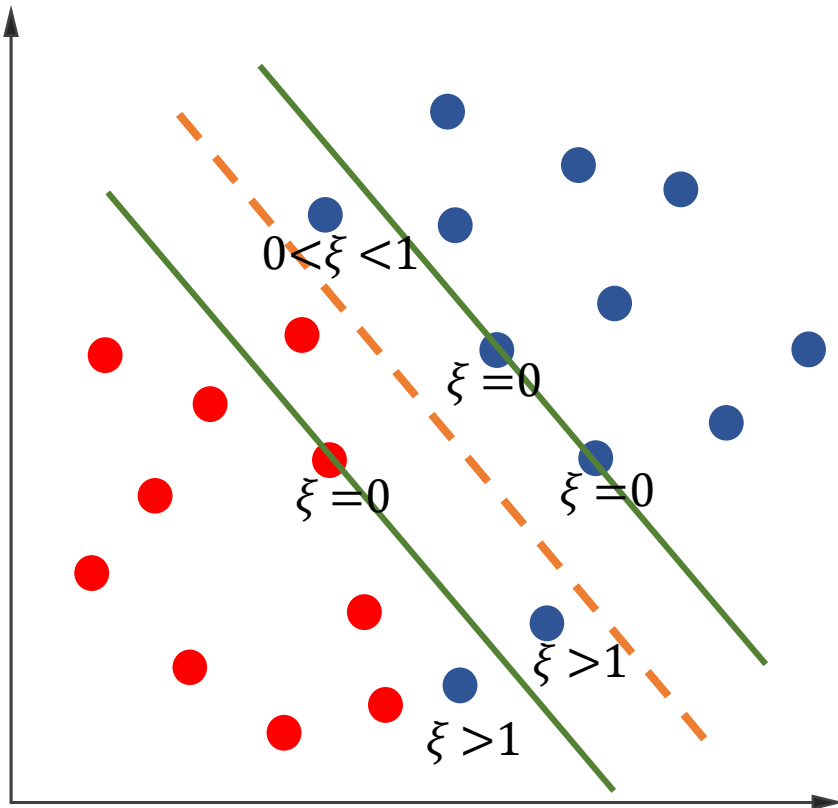
에러 발생 개수 만큼 패널티를 계산하자

$$\min ||w|| + C * \#error$$

→ 0-1 Loss



## Unit 03 | SVM의 종류



### <Soft Margin SVM>

Option 2)

에러 개수로만 판단하기엔 비합리적이므로  
Decision boundary에서 얼마나 떨어져 있는지  
거리 계산 + 에러의 개수로 판단하자

**Slack variable** 도입을 통해!

$\xi$  = 잘못 분류된 정도

$$\min ||w|| + C * \sum \xi_j$$

→ Hinge Loss

## Unit 03 | SVM의 종류

### Soft Margin SVM

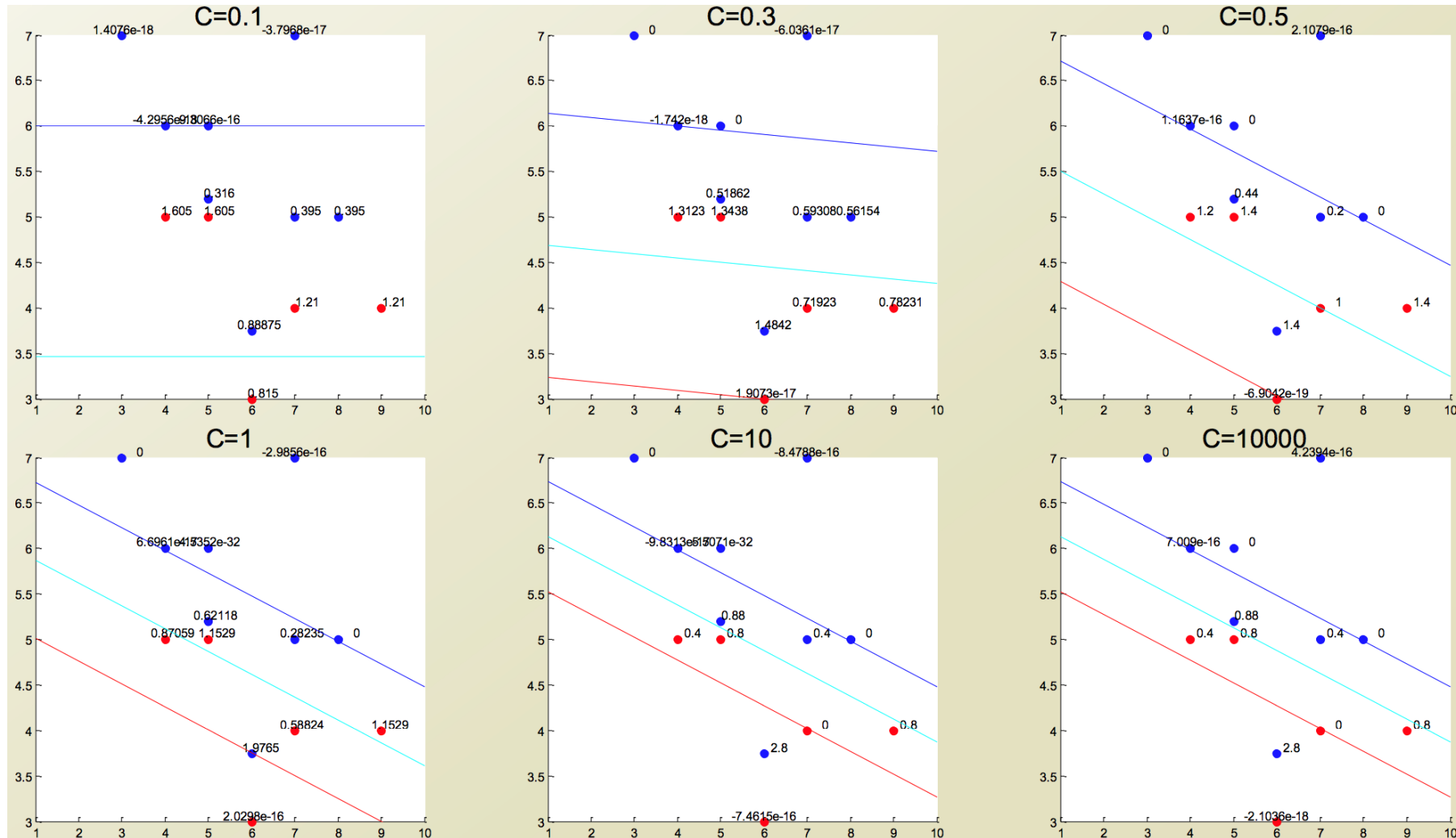
Error case를 허용하는 SVM

- 0-1 Loss  $\rightarrow \min ||w|| + C * \#error$
- Hinge Loss  $\rightarrow \min ||w|| + C * \sum \xi_j$

C는 constant term으로 패널티를 주는 강도를 의미함

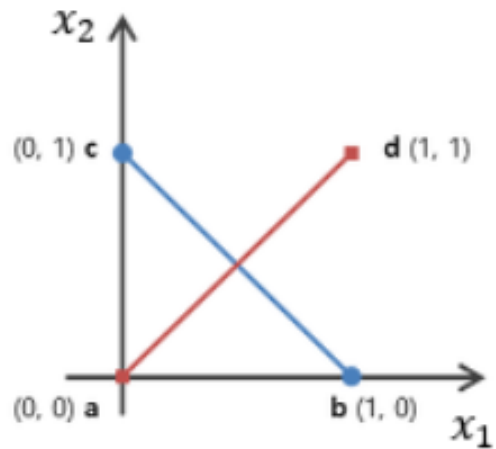
$\rightarrow$  C라는 새로운 parameter를 감당해야 하는 단점!!

## Unit 03 | SVM의 종류

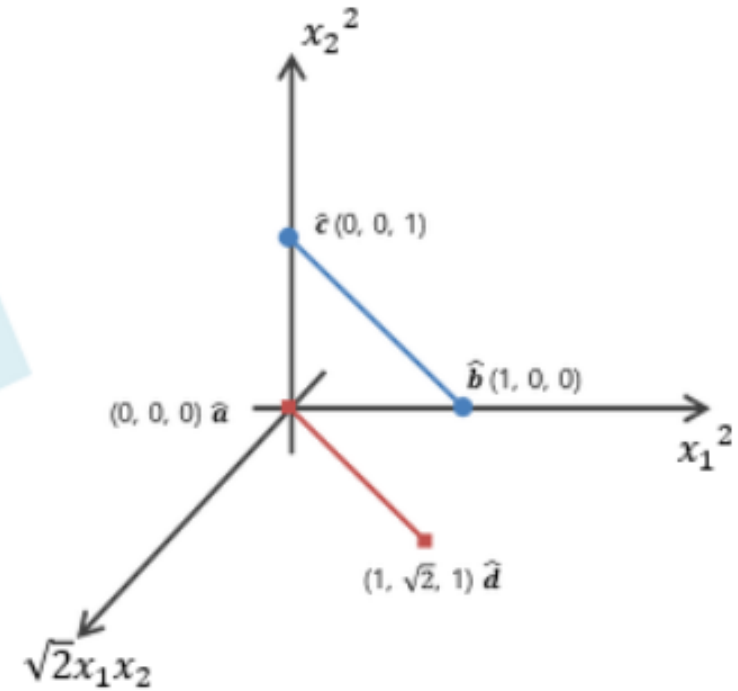


## Unit 03 | SVM의 종류

### 비선형 SVM

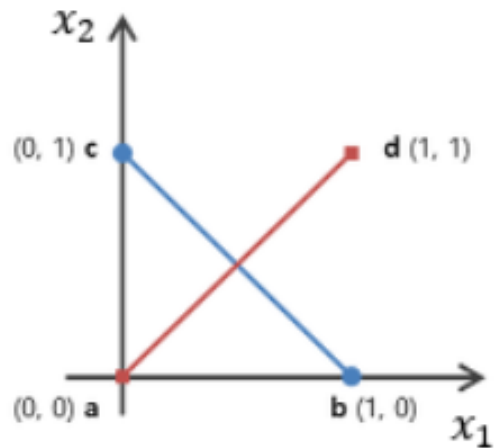


$$\varphi(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$$

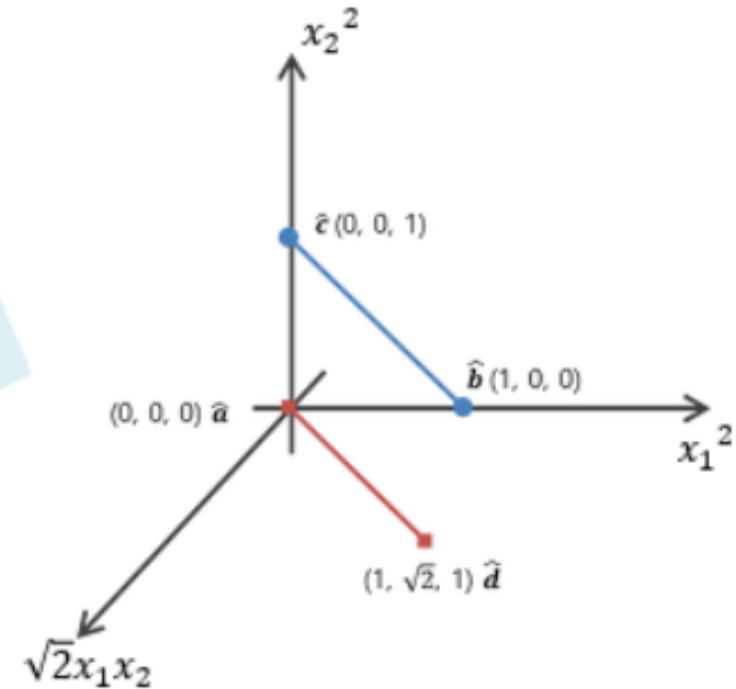


## Unit 03 | SVM의 종류

### 비선형 SVM



$$\varphi(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$$

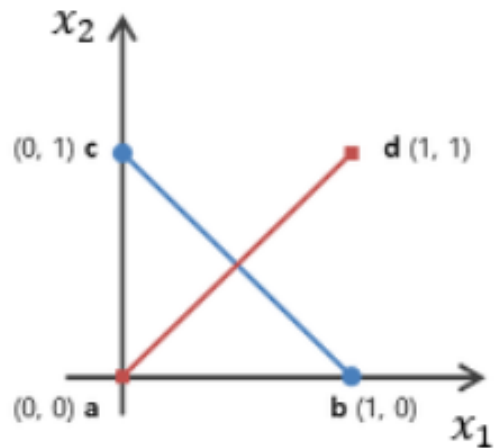


차원을 높여 선형인 것처럼!

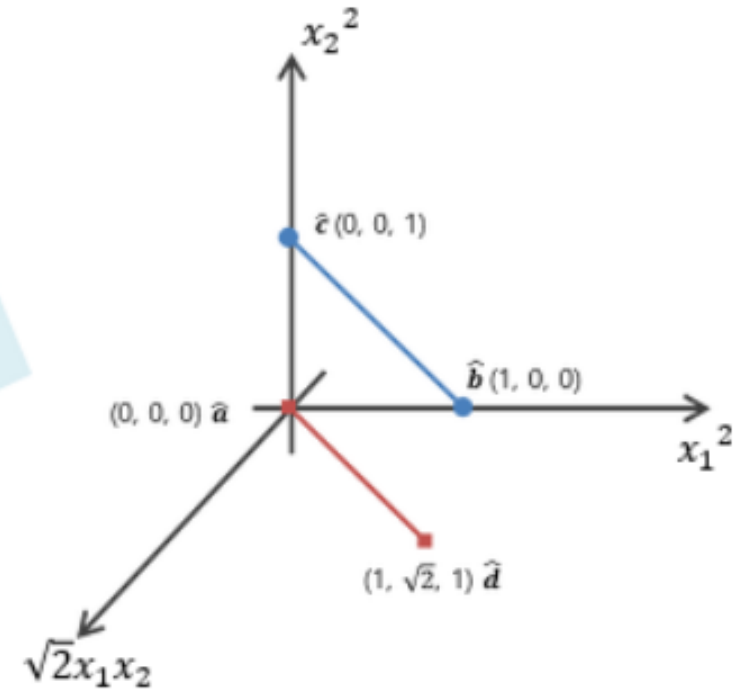
→ 차원이 늘어날수록 계산량이 기하급수적으로  
증가하기 때문에 좋은 방법 X

## Unit 04 | SVM의 종류

### 비선형 SVM



$$\varphi(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$$



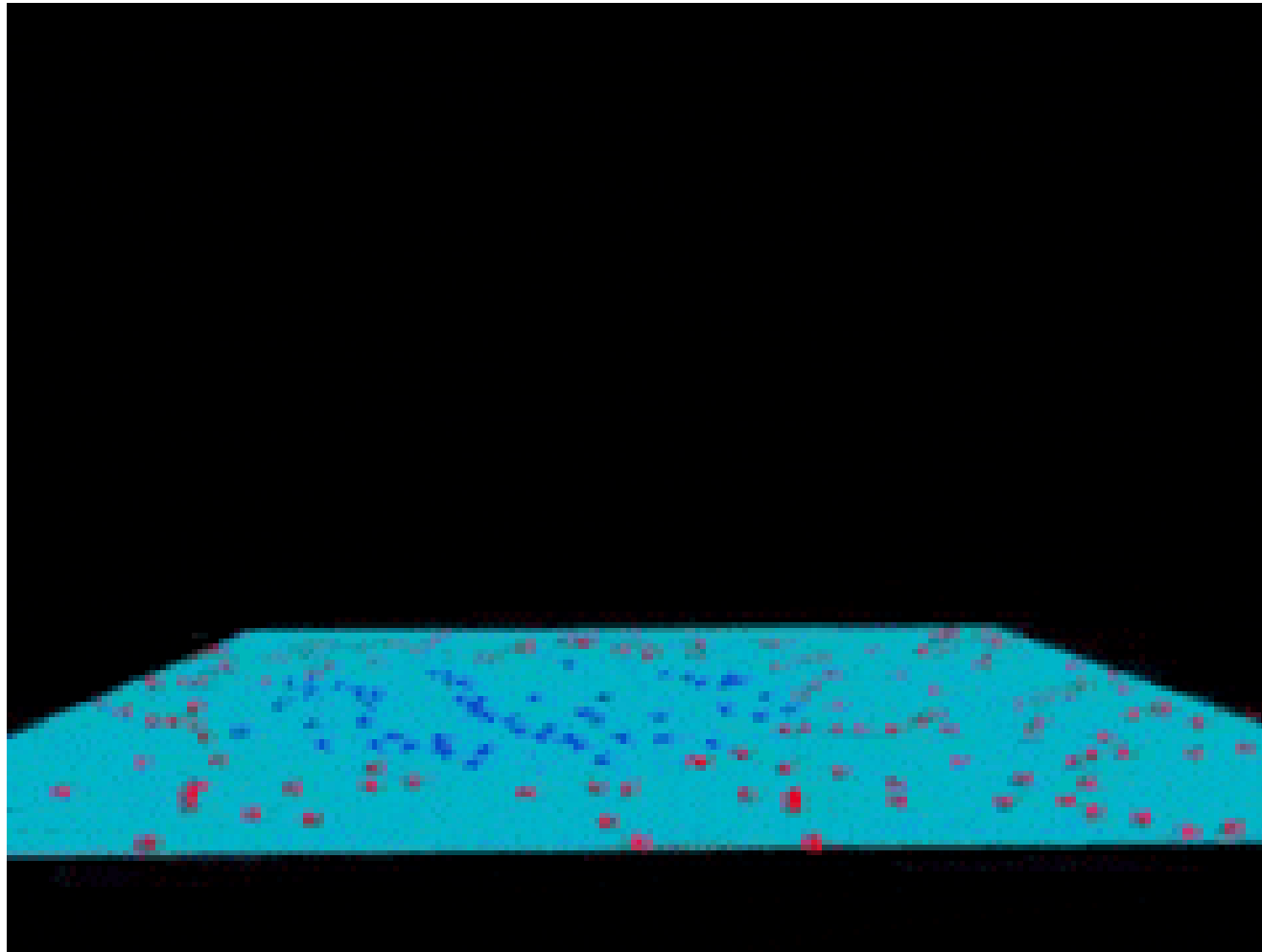
차원을 높여 선형인 것처럼!

→ 차원이 늘어날수록 계산량이 기하급수적으로 증가하기 때문에 좋은 방법 X

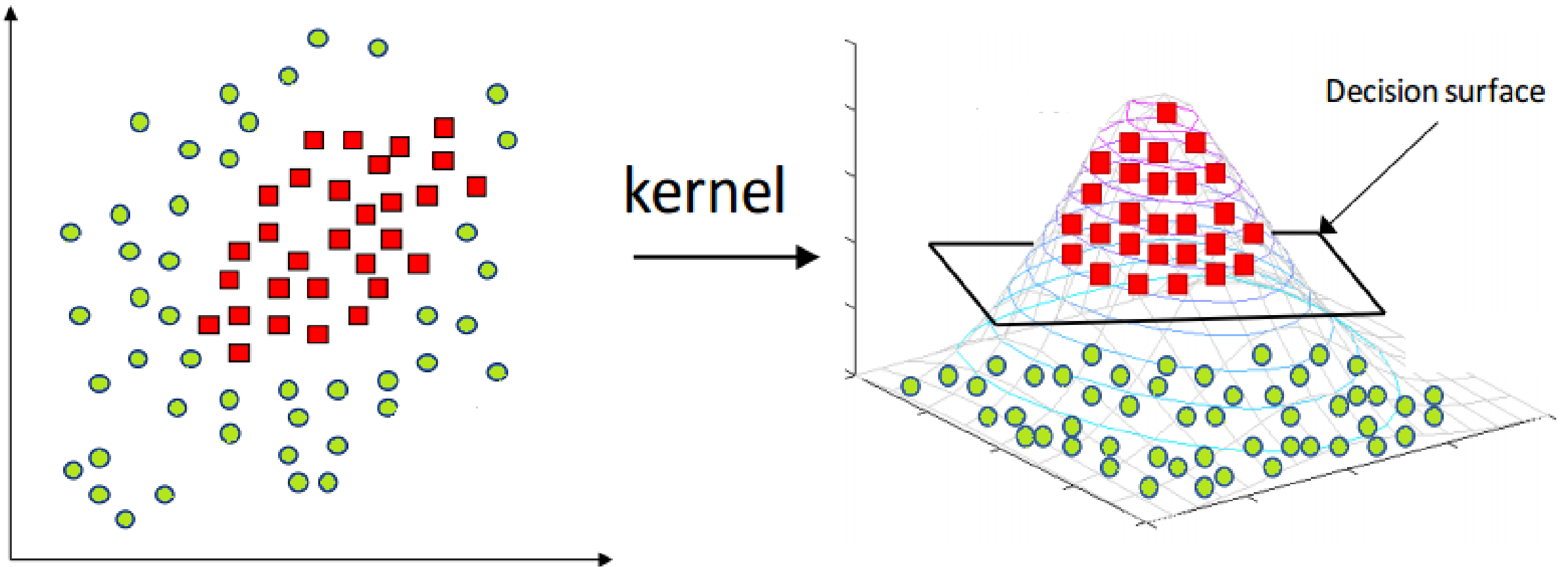


**Kernel 이용해서 해결!**

## Unit 04 | Kernel



## Unit 04 | Kernel



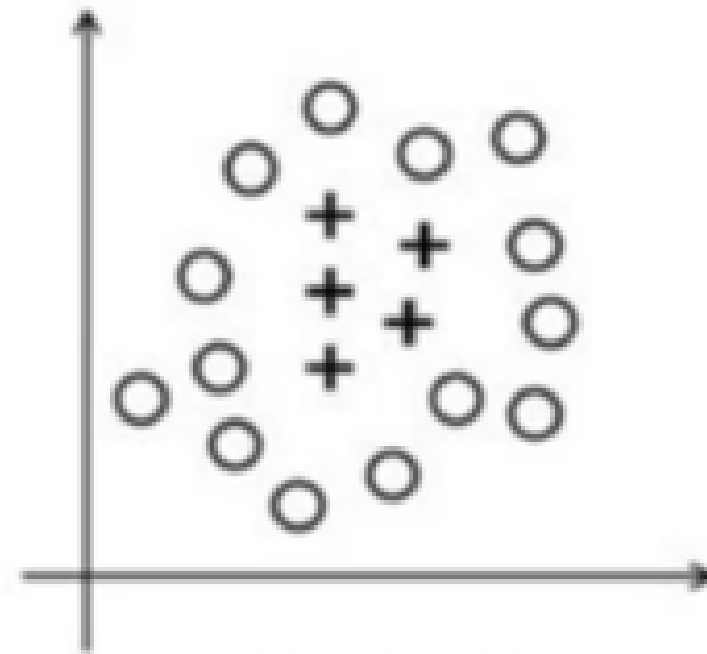


## Unit 03 | Kernel

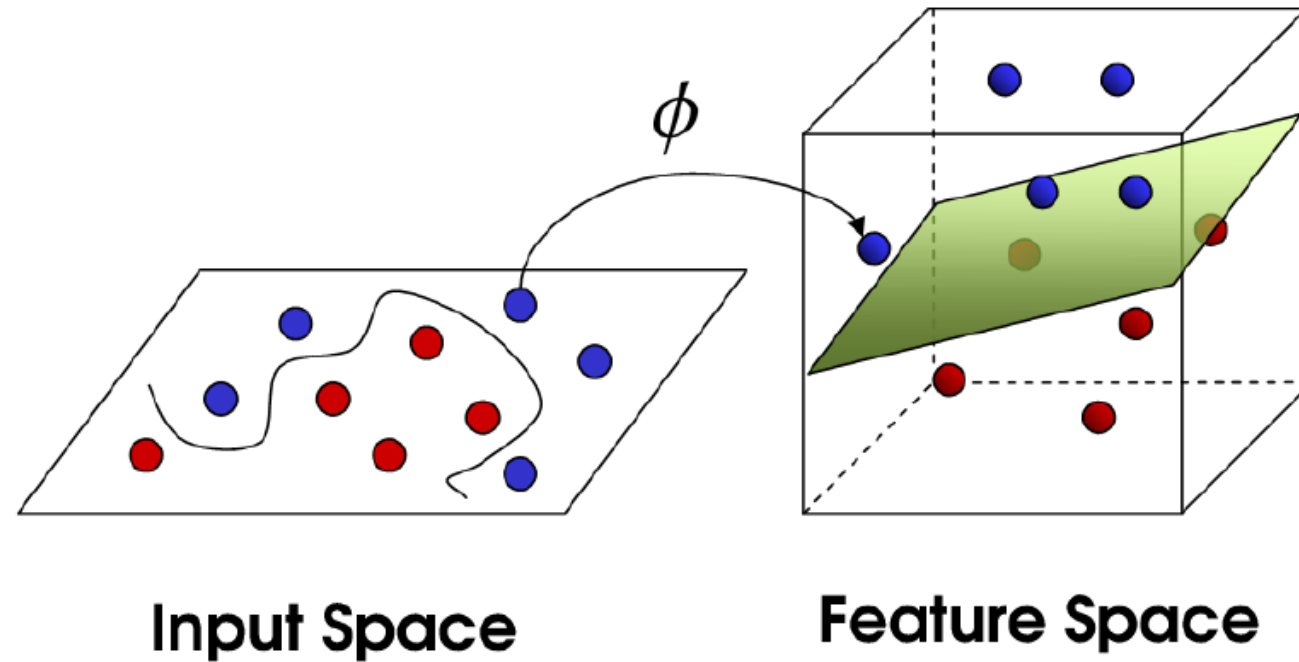
# Kernel? 커널??

SVM은 두 범주를 잘 분류하면서 Margin이 최대화된 초평면(hyperplane)을 찾는 기법!

기본적으로 선형 분류. 하지만 어떤 직선을 그어도 두 범주를 완벽하게 분류하기 어려운 경우가 多



## Unit 04 | Kernel



원공간(input space)의 데이터를 선형분류가 가능한 고차원 공간 (Feature space)으로 매핑한 뒤, 두 범주를 분류하는 초평면을 찾는다.

## Unit 04 | Kernel

### Kernel? 커널??

두 벡터를 각각 다른차원으로 보내서 내적(연산)

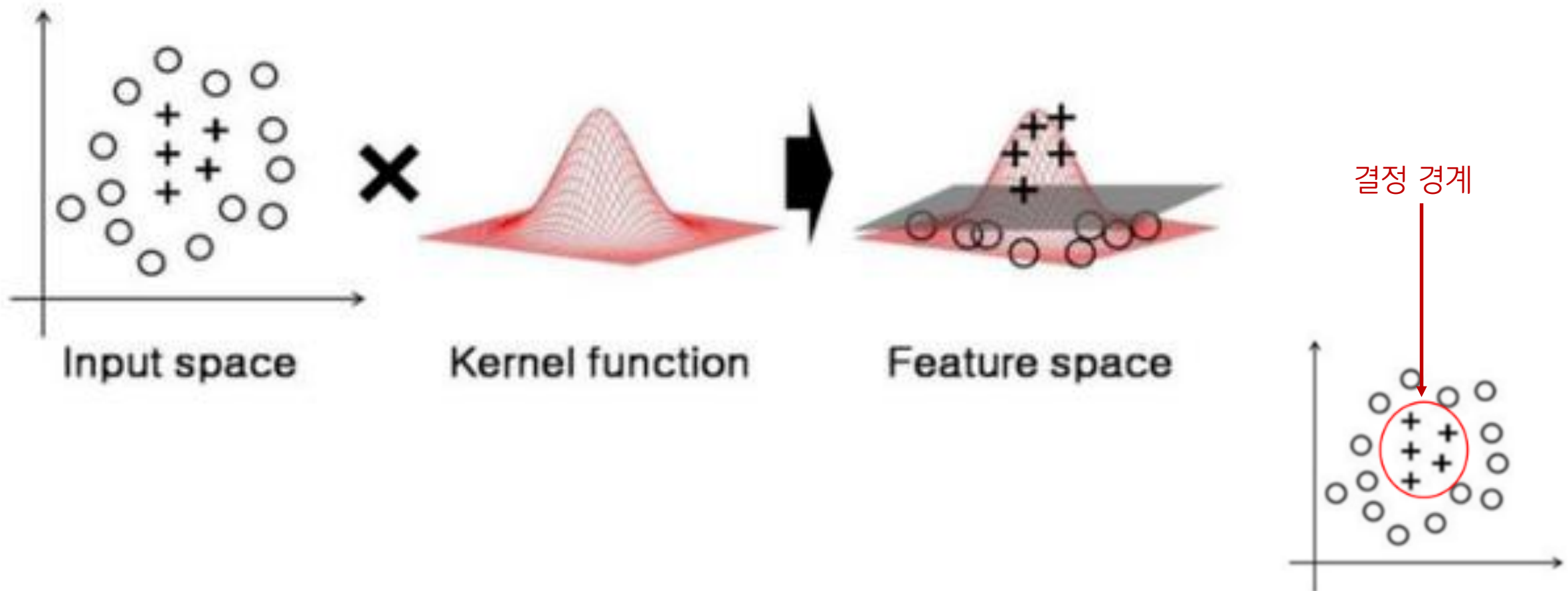
$$K(x_i, x_j) = \varphi(x_i) \cdot (x_j)$$

### Kernel trick?

고차원으로 보낸 뒤 벡터의 내적을 계산하는 것과 내적을 한 뒤 고차원으로 보내는 것은 결과적으로 같은 값

→ 고차원 변환 없이 계산할 수 있는 커널 함수 사용

## Unit 04 | Kernel

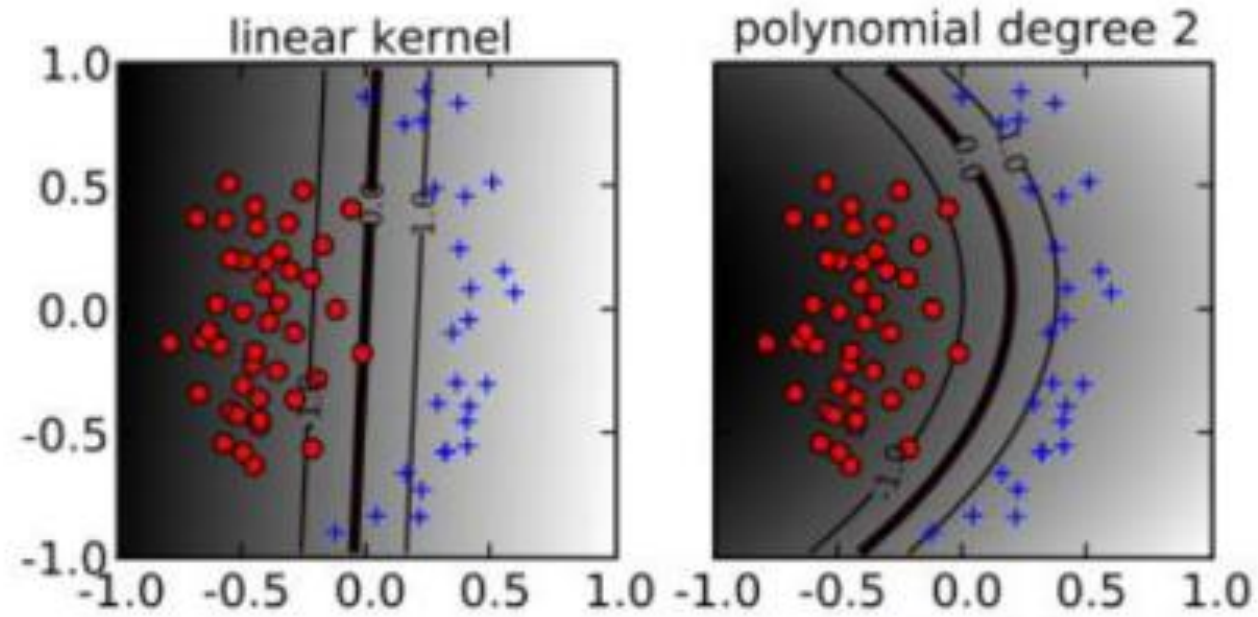


## Unit 04 | Kernel

Kernel함수의 종류

$$K(\mathbf{X}_i, \mathbf{X}_j) = \left\{ \begin{array}{ll} \mathbf{X}_i \cdot \mathbf{X}_j & \text{Linear} \\ (\gamma \mathbf{X}_i \cdot \mathbf{X}_j + C)^d & \text{Polynomial} \\ \exp(-\gamma \|\mathbf{X}_i - \mathbf{X}_j\|^2) & \text{RBF} \\ \tanh(\gamma \mathbf{X}_i \cdot \mathbf{X}_j + C) & \text{Sigmoid} \end{array} \right\}$$

## Unit 04 | Kernel



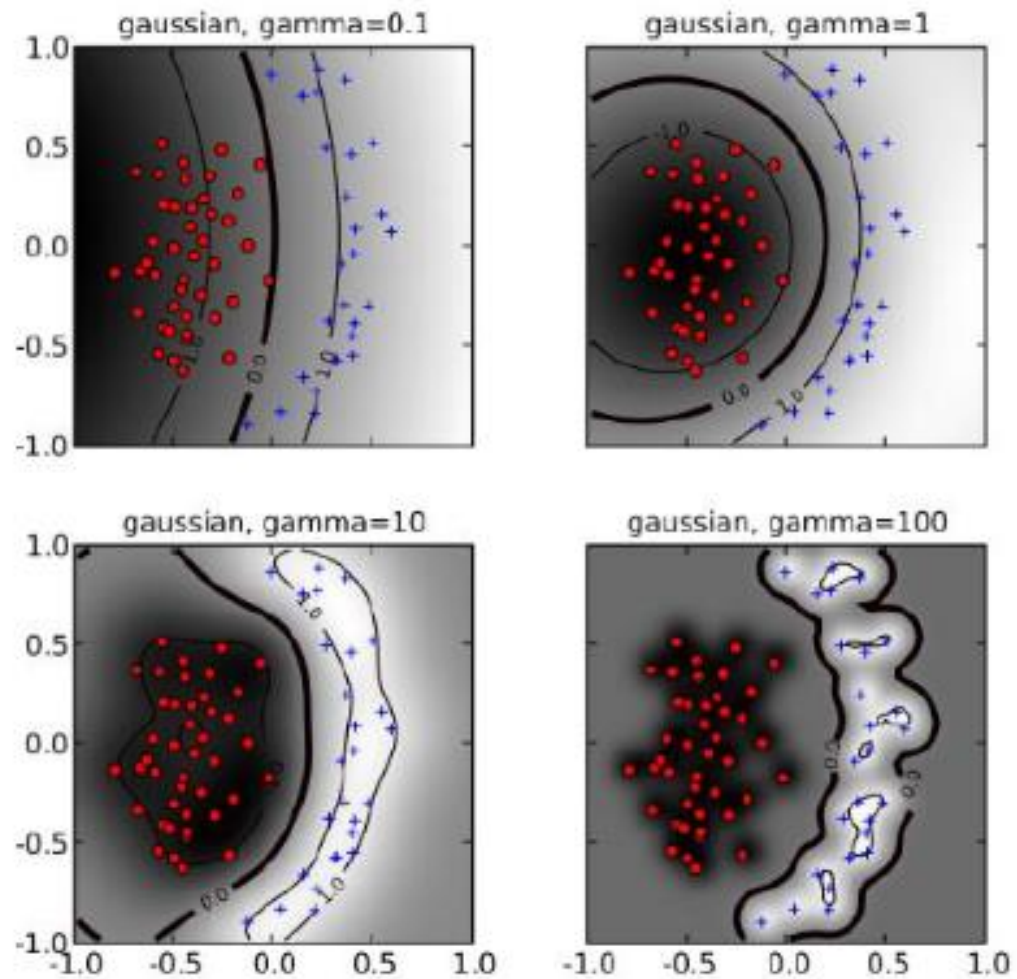
## Unit 04 | Kernel

Kernel함수의 종류

$$K(\mathbf{X}_i, \mathbf{X}_j) = \left\{ \begin{array}{ll} \mathbf{X}_i \cdot \mathbf{X}_j & \text{Linear} \\ (\gamma \mathbf{X}_i \cdot \mathbf{X}_j + C)^d & \text{Polynomial} \\ \exp(-\gamma |\mathbf{X}_i - \mathbf{X}_j|^2) & \text{RBF} \\ \tanh(\gamma \mathbf{X}_i \cdot \mathbf{X}_j + C) & \text{Sigmoid} \end{array} \right\}$$

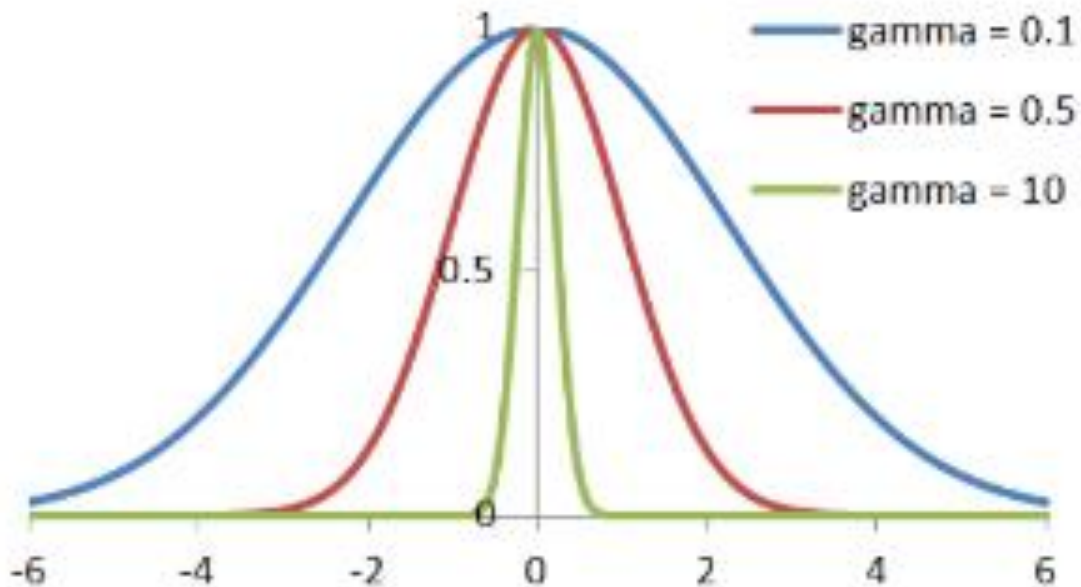
→ 성능이 좋아 자주 사용되는  
가우시안 RBF 커널

## Unit 04 | Kernel





## Unit 04 | Kernel



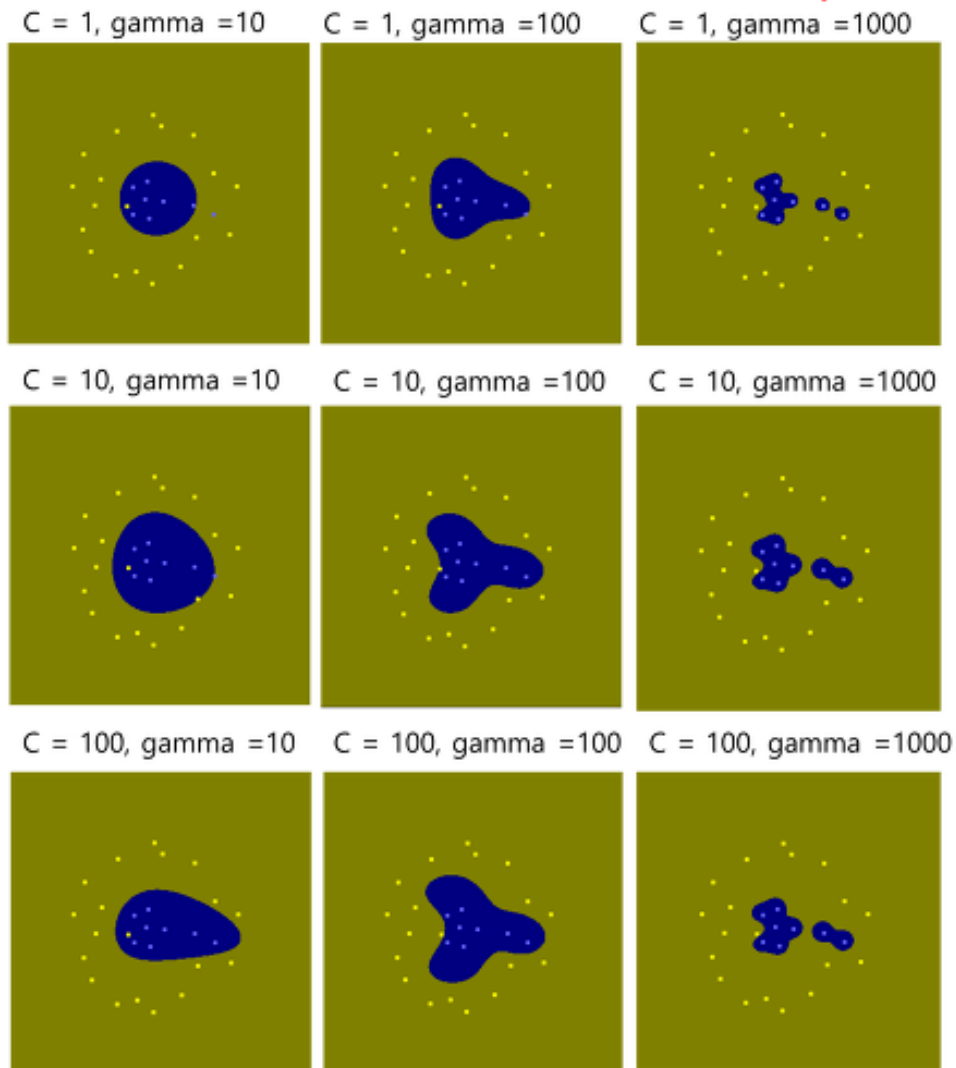
각 커널마다 최적화 도와주는 매개변수들 존재

RBF 커널의 경우 ,  $\gamma$  변수 조정 필요!

$\gamma$ ?  $\rightarrow$  하나의 데이터 샘플이 영향력을 행사하는 거리를 결정!

Gaussian 함수의 표준편차와 관련이 있는데, **클수록 작은 표준편차**를 갖는다.

## Unit 04 | Kernel



**Gamma 매개변수는  
결정 경계의 곡률을 조정한다!**

**C 매개변수는  
데이터 샘플들이 다른 클래스에  
놓이는 걸 허용하는 정도를 결정!**

**Gamma와 C  
두 parameter 모두  
적정값을 찾아야 함!**

## 참고 자료

- ① <http://yamalab.tistory.com/40> [티스토리 svm 포스팅]
- ② <http://blog.naver.com/PostView.nhn?blogId=tjdudwo93&logNo=221051481147>
- ③ <http://kkn1220.tistory.com/140>
- ④ <https://www.dtreg.com/solution/view/20>
- ⑤ <https://www.youtube.com/watch?v=m2a2K4lprQw> [gamma parameter 설정 Youtube]
- ⑥ <http://www.csie.ntu.edu.tw/~cjlin/libsvm/#GUI> → 좋은 자료!!

Q & A

들어주셔서 감사합니다.