Linear SVM

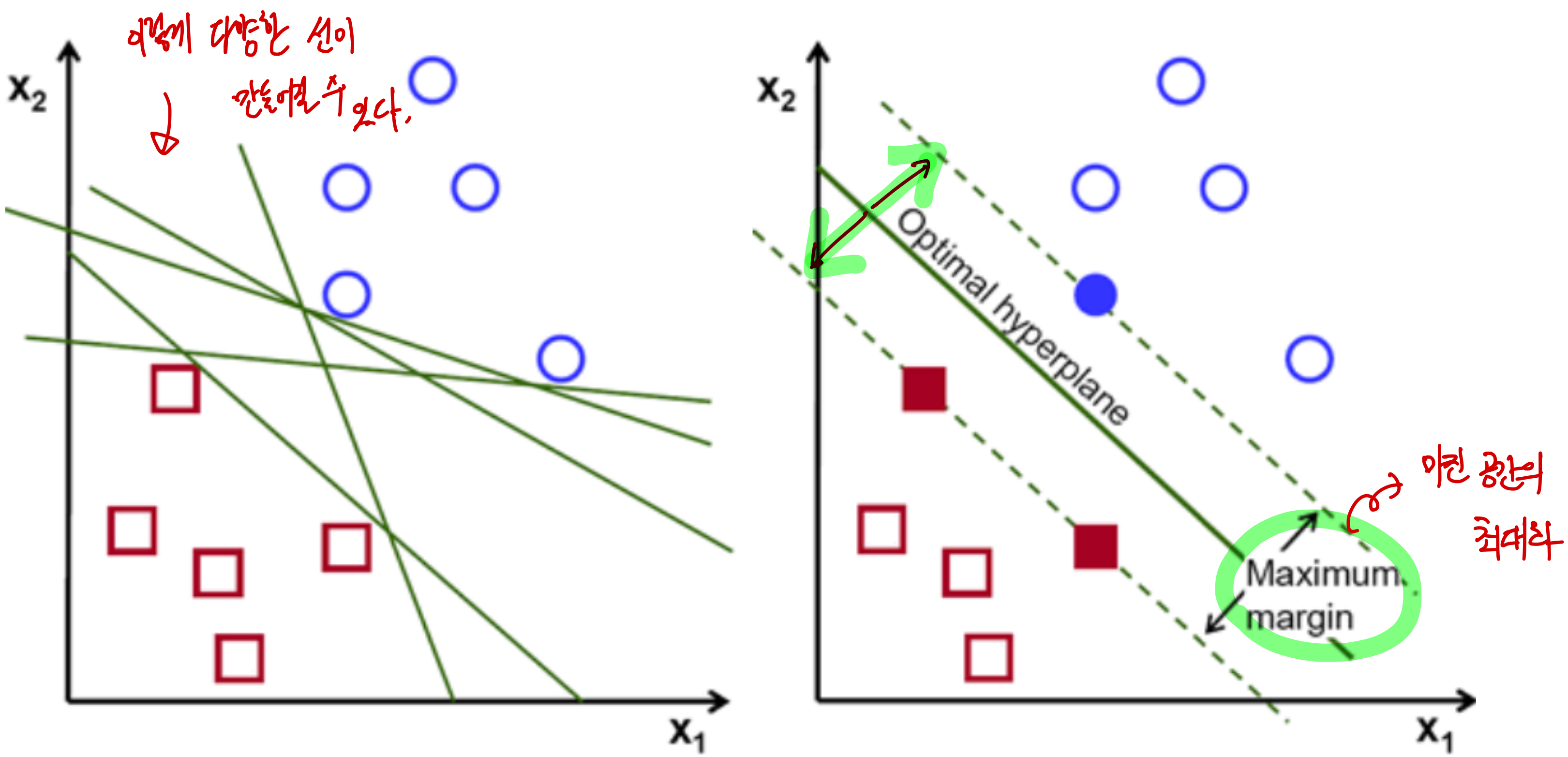
Hard Margin SVM

서포트 백터 머신 (SVM)

- **SVM 이란?**
  - 널리 사용되는 기계학습 방법론
  - 패턴인식, 자료 분석을 위한 **지도 학습 모델** (Supervised Model)
  - **분류**와 **회귀 문제**에 사용 (주로 분류 문제 사용)
  - 두 카테고리 중 어느 하나에 속한 데이터의 집합이 주어졌을 때, 주어진 데이터 집합을 바탕으로 하여 새로운 데이터가 어느 카테고리에 속할지 판단하는 **비-확률적 이진 선형 분류 모델**
  - 커널 트릭(Kernel Trick)을 활용하여 **비선형 분류 문제에도 사용 가능**

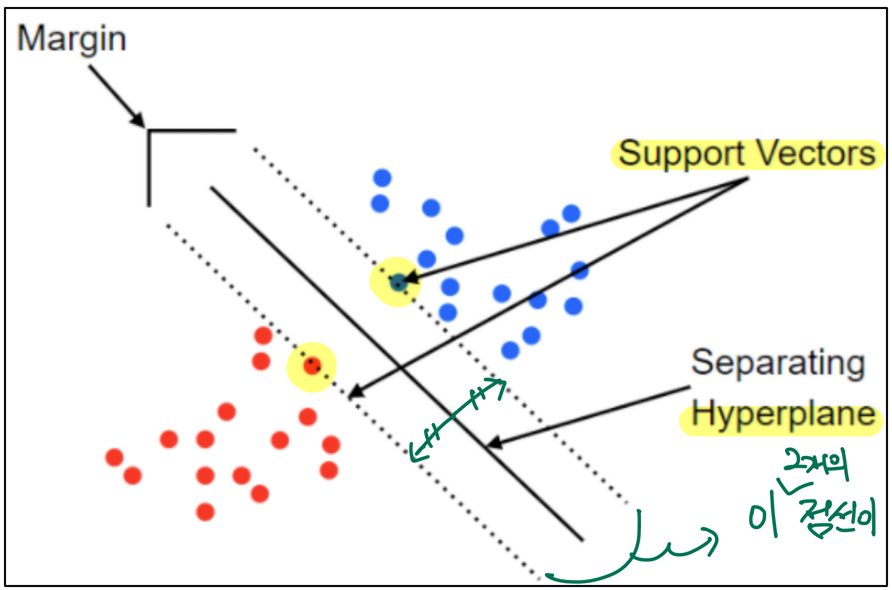
서포트 벡터 머신 (SVM)

- **서포트 벡터 머신 (Support Vector Machine)**
  - SVM 학습 방향 : 마진(Margin)의 최대화
    - 결정 경계(Hyperplane)는 주변 데이터와의 거리가 최대가 되어야 함
      (∵ 결정 경계 근처에 위치하는 새로운 데이터가 들어와도 강인한 분류)



서포트 백터 머신 (SVM)

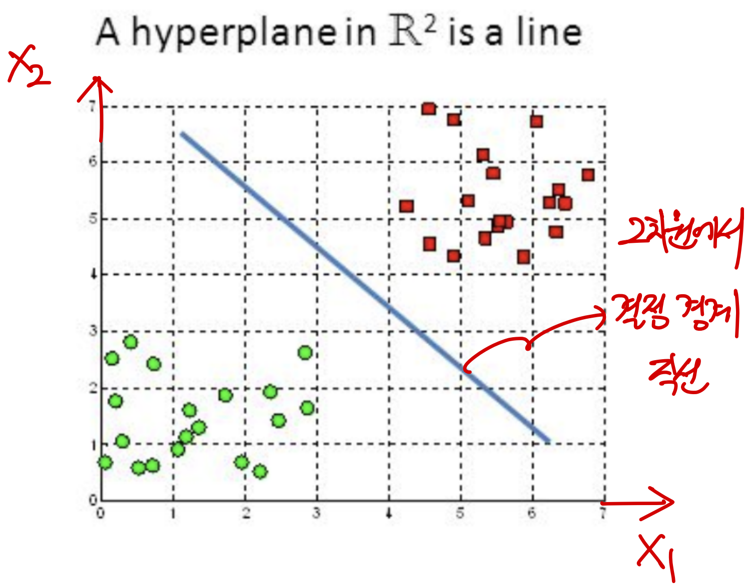
- 용어
  - Hyperplane
  - 결정 경계 (Huperplane)
    - 서로 다른 클래스를 완벽하게 분류하는 기준
  - 서포트 벡터(Support Vector)
    - 결정 경계선에 가장 가까이에 있는 각 클래스의 데이터
  - 마진(Margin)
    - 어떤 데이터도 포함하지 않는 영역, 서포트 벡터와 직교하는 직선과의 거리

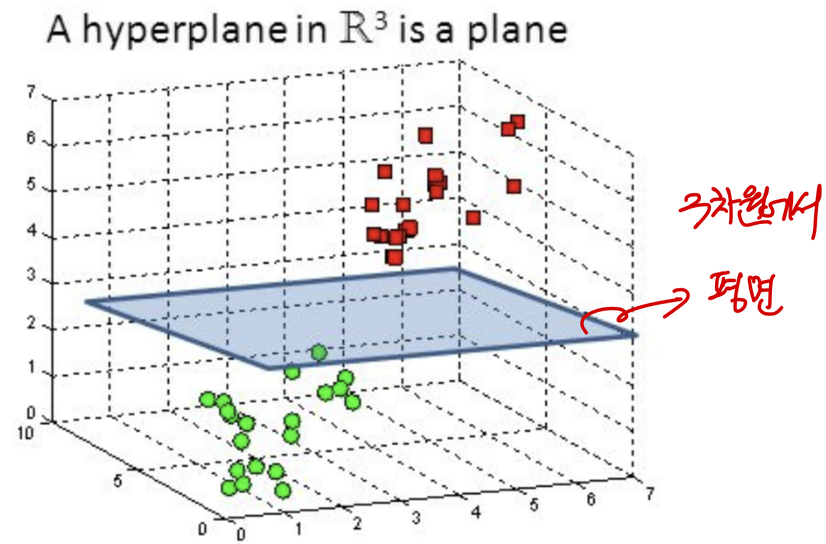


이 2개의 점선이 먼저 결정되고 그 지점에 Hyperplane이 설정되는 것!

서포트 백터 머신 (SVM)

- **결정 경계 (Hyper Plane)**
    - 데이터 임베딩 공간보다 1차원 낮은 부분 공간
        - 2차원 데이터 공간의 결정 경계: 직선
        - 3차원 데이터 공간의 결정 경계: 평면
        - 4차원 이상 데이터 공간의 결정 경계: 초평면





서포트 벡터 머신 (SVM)

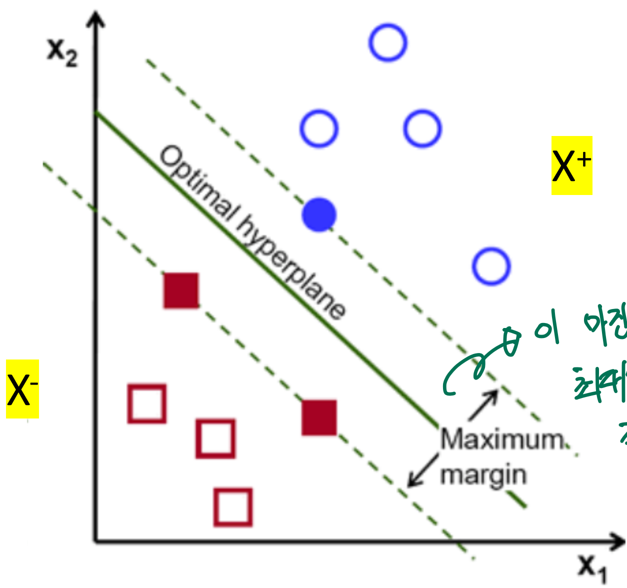
- **수학적 표현**
  - 결정 경계(초 평면) : $W^TX+b = 0$
  - (파랑) 서포트 벡터를 지나는 초 평면: $W^TX+b = 1$, (파랑) 데이터: $W^TX^+ + b >= 1$
  - (빨강) 서포트 벡터를 지나는 초 평면: $W^TX+b = -1$, (빨강) 데이터: $W^TX^- + b <= -1$
  - 초 평면의 법선 벡터 : W^T
  - **서로 다른 클래스 데이터의 관계 : $X^+ = X^- + \lambda W$**

이용치환

$W^TX^+ + b = 1$

$W^T(X^- + \lambda W) + b = 1$

⋮



이 마진을 최대화하는 것이 주목적.

$W^TX^+ + b = 1$

$W^T(X^- + \lambda W) + b = 1$

$(W^TX^- + b) + \lambda W^TW = 1$

$-1 + \lambda W^TW = 1$

$\lambda = \frac{2}{W^TW}$

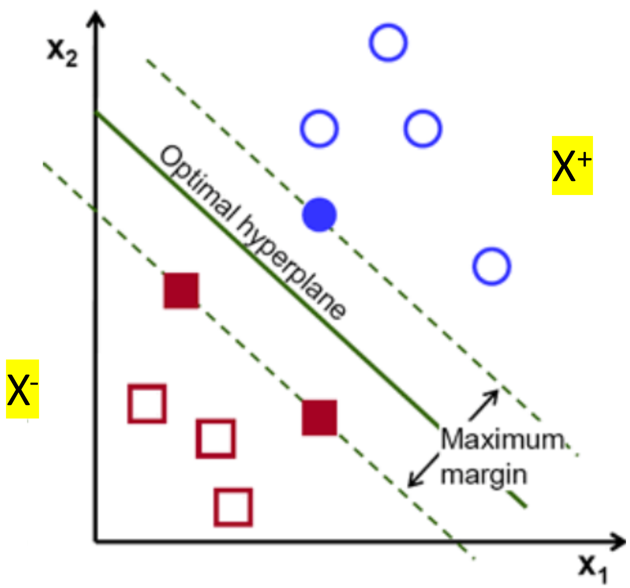
$y = ax + b$

$W^TX + b = 0$

2차원

서포트 벡터 머신 (SVM)

- **수학적 표현**
  - 결정 경계의 초 평면 : $W^TX+b = 0$
  - (파랑) 서포트 벡터를 지나는 초 평면: $W^TX+b = 1$, (파랑) 데이터: $W^TX^+ + b \rangle{=}1$
  - (빨강) 서포트 벡터를 지나는 초 평면: $W^TX+b = -1$, (빨강) 데이터: $W^TX^- +b \langle{=}{-1}$
  - 초 평면의 법선 벡터 : W^T
  - 서로 다른 클래스 데이터의 관계 : $X^+=X^-+\lambda W$



Margin

$$
\begin{aligned}
&= \text{distance}(X^+, X^-) \\
&= \|X^+ - X^-\|_2 \\
&= \|(\cancel{X^-}+\lambda W) - \cancel{X^-}\|_2 \\
&= \|\lambda W\|_2 \\
&= \lambda\sqrt{W^TW} \\
&= \frac{2}{W^TW}\sqrt{W^TW} \\
&= \frac{2}{\sqrt{WTW}} = \frac{2}{\|W\|_2}
\end{aligned}
$$

$$L_2 = \sqrt{\sum_i^n x_i^2} = \sqrt{x_1^2 + x_2^2 + x_3^2 + \ldots + x_n^2}$$

$\max\left(\frac{2}{\|W\|_2}\right)$

↳ 이것이 SVM 함수의 목적

서포트 백터 머신 (SVM)

- **SVM의 목적 함수**
  - **마진(Margin)의 최대화**

$$max\ \text{Margin} = \boxed{max\ \frac{2}{\|W\|_2}}$$

$$max\ \frac{2}{\|W\|_2} = min\ \frac{\|W\|_2}{2}$$

최대화 문제 최소화 문제로 변경

$$min\ \frac{\|W\|_2}{2} \approx min\ \frac{\|w\|_2^2}{2}$$

→ 계산편의를 위해 제곱한 것

계산상의 편의

$w^T x + b = 0$

$$\boxed{\min_{w,b}\ \frac{1}{2}\|w\|_2^2}$$

$$subject\ to\ \ y_i(w^T x_i + b) \geq 1, i = 1,2, \ldots, n$$

조건

$w^T x^+ + b \geq 1$, $y = 1$

$w^T x^- + b \leq -1$, $y = -1$

→ y (-1) 곱해서 식 부등호 같게 만들어주면 됨.

서포트 벡터 머신 (SVM)

- 서포트 벡터 머신 (Support Vector Machine)
  - SVM학습 방향 : 마진(Margin)의 최대화

Original Problem

$$\min_{w,b} \frac{1}{2}\|w\|_2^2$$ ~> objective function

~> constraints function

$$subject\ to\ \ y_i(w^T x_i + b) \geq 1, i = 1,2, \dots, n$$

  - 〈라그랑지 승수〉 해법으로 해결 → 이것을 통해 해결
    - 제약식(constraints)을 목적식(objective function)에 포함

Primal Problem

$$\max_{\alpha} \min_{w,b} \mathcal{L}(w,b,\alpha) = \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{n} \alpha_i (y_i(w^T x_i + b) - 1)$$

$$subject\ to\ \alpha_i \geq 0, i - 1,2, \dots, n$$

[참고] 모두를 위한 컨벡스 최적화: https://wikidocs.net/19932

서포트 벡터 머신 (SVM)

- Primal Problem 을 Dual Problem 으로 변경하여 해결

Dual Problem

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

$$subject\ to \sum_{i=1}^{n} \alpha_i y_i = 0, \alpha_i \geq 0, i - 1,2,\dots,n$$

W는 α, x, y 로 변경

w, b 변수 소거되었다.

← 제약 조건.

- (W, b, α)가 Lagrangian dual problem의 최적해가 되기 위한 조건
  - KKT (Karush-Kuhn-Tucker) Conditions
    - Stationarity $\quad w = \sum_{i=1}^{n} \alpha_i y_i x_i \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$
    - Primal feasibility $\quad y_i(w^T x_i + b) \geq 1, i = 1,2,\dots,n$
    - Dual feasibility $\quad \alpha_i \geq 0, i - 1,2,\dots,n$
    - Complementary slackness $\quad \alpha_i(y_i(w^T x_i + b) - 1) = 0$

서포트 벡터 머신 (SVM)

$\alpha \geq 0,$

둘 중 하나가 0 $\quad \geq 1$

- **Complementary slackness** $\quad \alpha_i(y_i(w^T x_i + b) - 1) = 0$

$\alpha_i \neq 0$

- $\alpha_i > 0,\ y_i(w^T x_i + b) - 1 = 0$ → ✓ x_i가 support vector(SV) 인 경우
- $\alpha_i = 0,\ y_i(w^T x_i + b) - 1 \neq 0$ → ✓ x_i가 support vector(SV)가 아닌 경우
  - ✓ Hyperplane 구축에 영향을 미치지 않음
  - ✓ SVM이 outlier 에 강인한 이유이기도 함

소음

서포트 벡터

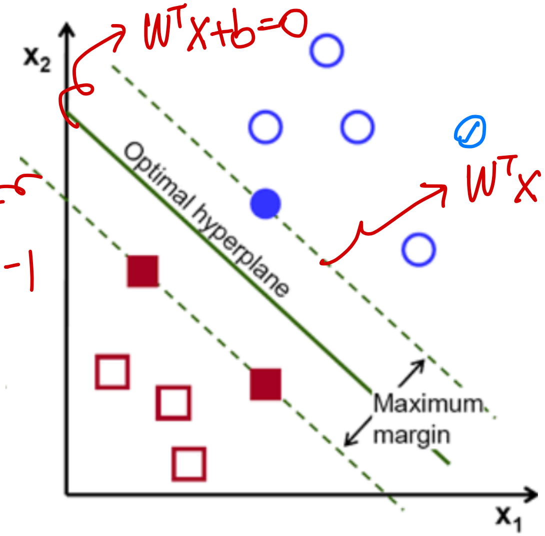
$y_i(W^T x_i + b) = 1$

①,② 식은 의미

$y_i < \begin{matrix} +1 \\ -1 \end{matrix}$

α_i가 서포트 벡터가 아니다!!

- **결정 경계 결정법: 서포트 벡터를 가지고 계산**



$$W = \sum_{i=1}^{n} \alpha_i y_i x_i = \sum_{i \in SV} \alpha_i y_i x_i$$

$$b = y_{sv} - \sum_{i=1}^{n} \alpha_i y_i x_i^T x_{sv}$$

$$y_{new} = sign(W^T X_{new} + b)$$

모델!!

$W^T \cdot X_{new} + b > 0$ ↳ 1 → +

< 0 ↳ −1 → −

Toy example

Linear SVM

Soft Margin SVM

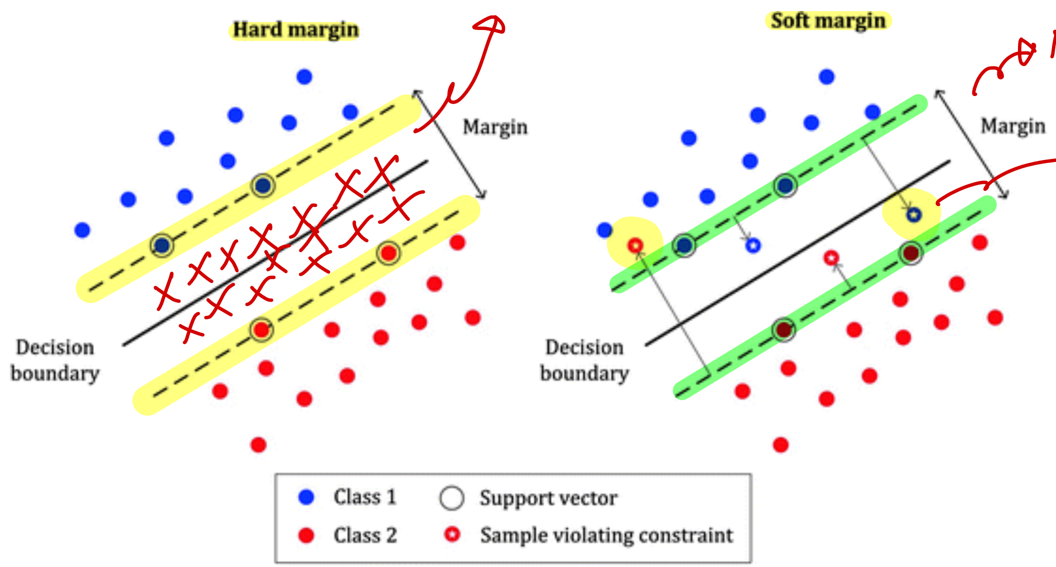
지난 시간에는 Hard Margin SVM

서포트 백터 머신 (SVM)

- Hard Margin SVM vs Soft Margin SVM
  - Hard Margin SVM
    - 선형 분리 가능한 문제
  - Soft Margin SVM
    - 선형 분리 불가능 문제
    - 학습 데이터의 에러가 0 이 되도록 완벽하게 나누는 것을 불가능

      ➔ **에러를 허용 하자**

아무것도 없다 = 학습데이터의 에러가 0.

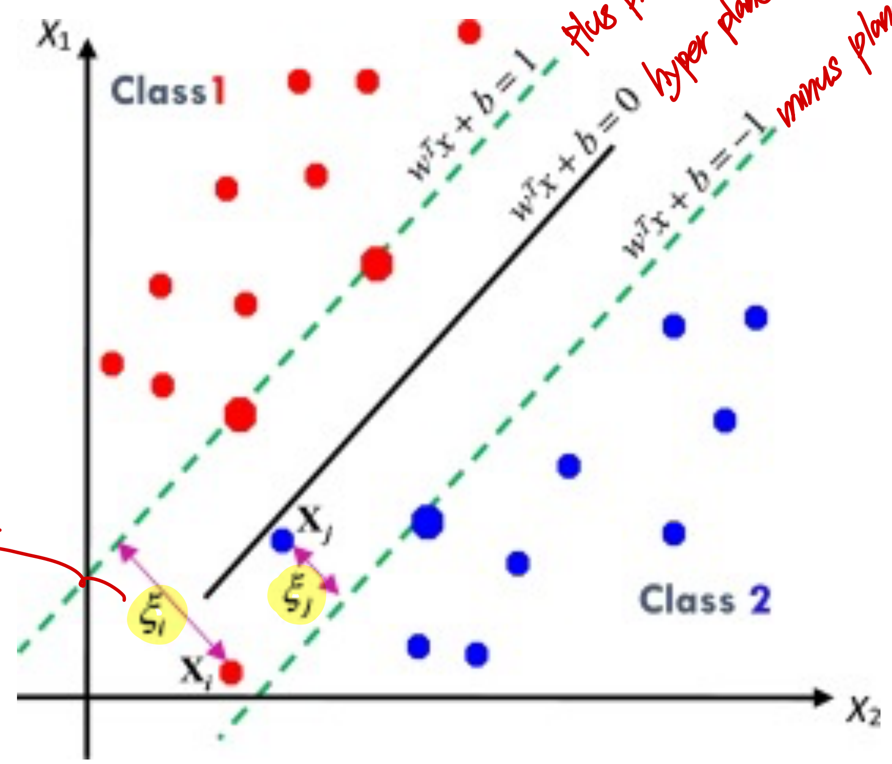


Margin 조금 에러가 있다.

이런식으로 조금 섞여 있다.
이런 경우, 비선형 분류를 사용 or soft SVM!!

Soft Margin SVM

- Soft Margin SVM
  - 선형 분리 불가능 문제
  - 학습 데이터의 에러가 0 이 되도록 완벽하게 나누는 것을 불가능
    ➔ **에러를 허용 하자**



plus plane

hyper plane

minus plane.

에러

에러에
해당

Soft Margin SVM

$\min_{w,b} \frac{1}{2}\|w\|_2^2$

subject to $y_i(w^T x_i + b) \geq 1$, $i=1,\cdots,n$

- Soft Margin SVM
  - 목적 함수: 에러(penalty term)를 허용하면서 마진을 최대화 하는 것

Original Problem

$$\min_{w,b,\xi} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}\xi_i$$

Penalty term

$$subject\ to\ \ y_i(w^T x_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0,\ i = 1,2,\ldots,n$$

“C” parameter

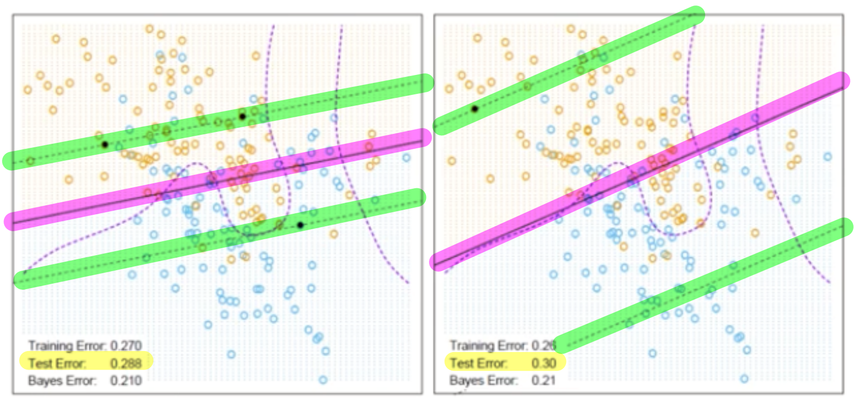
에러 거리에 해당

  - C를 통해 에러의 허용 정도 조절
    - C가 크면 학습 에러를 상대적으로 허용하지 않음 (Overfitting) ➔ 마진 작아짐
    - C가 작으면 학습 에러를 상대적으로 허용 (Underfitting) ➔ 마진이 커짐

C=10000

C↑ 에러 허용 ↓ ··· 에러↓

C↓ 에러 허용↑ ··· 에러↑



C=0.01

But, C를 무조건 크게 하는 것이 옳지 X

Soft Margin SVM

- Soft Margin SVM
  - 목적 함수: 에러(penalty term)를 허용하면서 마진을 최대화 하는 것

Original Problem

$$\min_{w,b,\xi} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}\xi_i$$

Penalty term

$$subject\ to\ yi(W^T X_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0, i = 1,2,\dots,n$$

  - 〈라그랑지 승수〉 해법으로 해결
    - 제약식(constraints)을 목적식(objective function)에 포함

Primal Problem

$$\max_{\alpha}\min_{w,b} \mathcal{L}(w,b,\alpha,\xi,\gamma) = \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{n}\alpha_i(y_i(w^T x_i + b) - 1 + \xi_i) - \sum_{i=1}^{n}\gamma_i\xi_i$$

$$subject\ to\ \alpha_i\gamma_i \geq 0, i - 1,2,\dots,n$$

Soft Margin SVM

- **Primal Problem 을 Dual Problem 으로 변경하여 해결**

Dual Problem

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

W는 α, y, x 로 변경

$$subject\ to\ \sum_{i=1}^{n} \alpha_i y_i = 0, 0 \le \alpha_i \le C, i = 1,2,\dots,n$$

$0 \le \alpha$

- 주의사항

$\boxed{0 \le \alpha_i \le C}, from\ \alpha_i \ge 0, \gamma_i \ge 0, C - \alpha_i - \gamma_i = 0$

Hard Margin SVM vs Soft Margin SVM

Hard Margin SVM

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

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

$$\mathbf{0 \leq \boldsymbol{\alpha_i}},\ i = 1,2,\ldots,n$$

Soft Margin SVM

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

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

$$\mathbf{0 \leq \boldsymbol{\alpha_i} \leq \boldsymbol{C}},\ i = 1,2,\ldots,n$$

★★★

- 선형 분류의 두 가지 케이스 모두 quadratic programming 을 풀어 α 를 구함
  - 간적접으로 W를 구한 셈

$\alpha \rightarrow w, b \rightarrow$ 모델

Nonlinear SVM

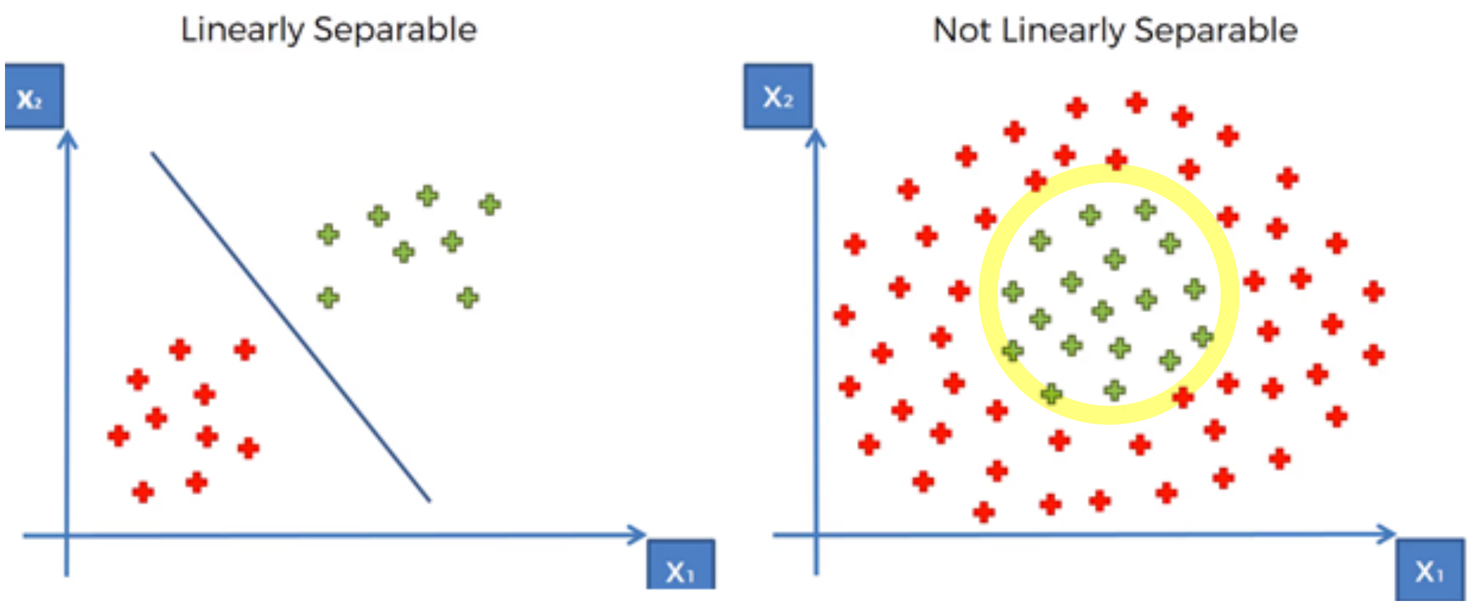
Kernel SVM

비선형 SVM

서포트 백터 머신 (SVM)

- **선형 SVM vs 비선형 SVM**
  - **선형 SVM**
    - 하드 마진(Hard margin) SVM, 소프트 마진(Soft margin) SVM
  - **비선형 SVM**
    - 커널(Kernel) SVM

분류하고자 하는 데이터 타입에 의존.

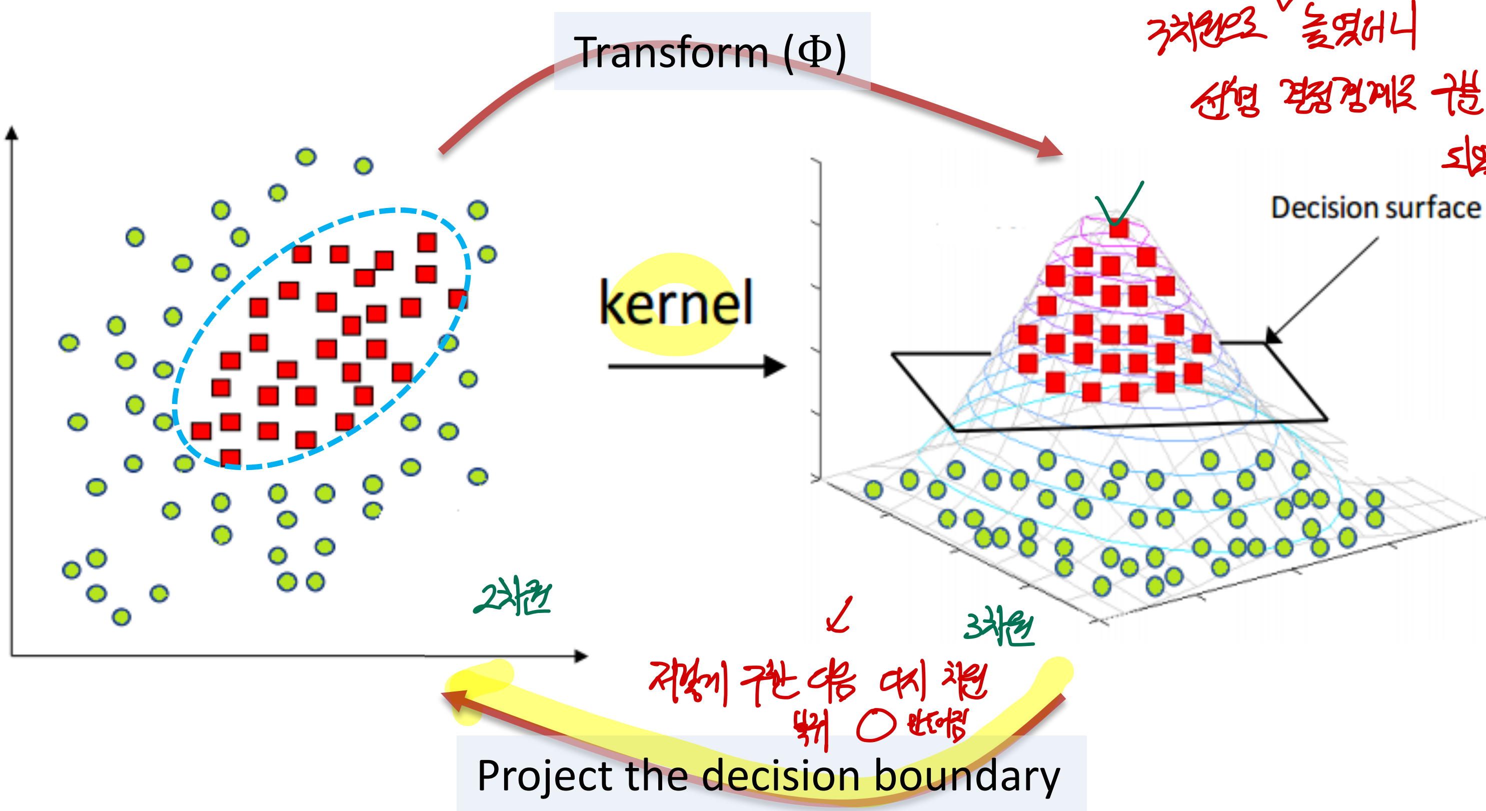


비선형 SVM

- 비선형 SVM
  - 데이터를 선형으로 분류하기 위해 차원을 높이는 방법을 사용
  - Feature Map (Φ)을 통해 차원을 높임. 즉, X 대신 Φ(X)를 사용
  - 커널: Feature Map 의 내적 $\Phi(x) \cdot \Phi(x)$

이 역할을 수행하는 함수

아래와 같이 3차원으로 늘였더니 선형 결정경계로 구분 가능하게 되었다!!



적당히 구한 다음 다시 차원 낮춰 ○ 만들어짐

비선형 SVM

- **비선형 SVM의 해법**
  - SVM 모델을 Original Space 가 아닌 Feature Space에서 학습 (X➔ $\Phi(\mathbf{X})$)
  - Original Space 에서 Nonlinear decision boundary ➔ Feature Space에서 linear decision boundary
  - 고차원 Feature space에서는 분류가 더 쉬울 수 있음을 입증함
  - **고차원 Feature space를 효율적으로 계산할 수 있는 방법이 있음**

projection

$\Phi: X \to Z = \Phi(X)$

[예시] 2D ➔ 5D

$\Phi: (x_1, x_2) \to (x_1, x_2, x_1^2\ x_2^2, x_1 x_2)$

$\Phi(x) =$

비선형 SVM

- **비선형 SVM의 목적함수**

Dual Problem

x_i $\quad$ x_j

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \Phi(x_i)^T \Phi(x_j)$$

feature map을 통해 변형된 데이터가 들어감!

$$subject\ to \sum_{i=1}^{n} \alpha_i y_i = 0, 0 \le \alpha_i \le C, i = 1,2, \dots, n$$

Dual Problem

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

'kernel'

↳ 연산 효율성 ↑

$$subject\ to \sum_{i=1}^{n} \alpha_i y_i = 0, 0 \le \alpha_i \le C, i = 1,2, \dots, n$$

비선형 SVM

- Kernel Mapping 의 예

$$X = (x_1, x_2), \mathrm{Y} = (y_1, y_2)$$

$$\Phi(X) = (x_1^2, x_2^2, \sqrt{2}x_1x_2), \Phi(Y) = (y_1^2, y_2^2, \sqrt{2}y_1y_2)$$

$$< \Phi(X), \Phi(Y) > = x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2$$

처음은 단순히 2번의 단계를 통해 도달
But, 실제로는 복잡↑
연산량↑

- 커널 사용을 통해 명시적(explicitly)으로 $\Phi(X), \Phi(Y)$를 각각 계산하지 않고 암묵적(implicitly)으로 $< \Phi(X), \Phi(Y) >$를 바로 계산하여 연산 효율을 높일 수 있음 !?

$$\boxed{(X,Y)^2} =< (x_1, x_2), (y_1, y_2) >^2$$

$$= < x_1y_1 + x_2y_2 >^2$$

$$= x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2$$

$$\boxed{= < \Phi(X), \Phi(Y) >}$$

- $\boxed{(X,Y)^2} = < (x_1, x_2), (y_1, y_2) >^2$ = $\boxed{\mathrm{K(X,Y)}}$ ➔ Kernel Function

비선형 SVM

- Kernel Function의 예
  - Linear Kernel

    $K(x_1, x_2) = \langle x_1, x_2 \rangle$

  - Polynomial Kernel

    $K(x_1, x_2) = (a\langle x_1, x_2 \rangle + b)^d$

  - Sigmoid Kernel

    $K(x_1, x_2) = \tanh(a\langle x_1, x_2 \rangle + b)$

  - Gaussian Kernel (Radial basis function (R~~G~~B) Kernel)

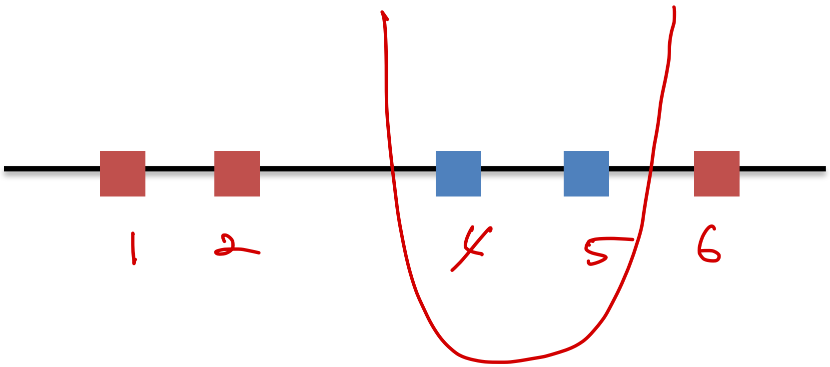
    RBF

    $$K(x_1, x_2) = \exp\left(\frac{-\| x_1 - x_2 \|_2^2}{2\sigma^2}\right)$$

비선형 SVM의 예

1차원 X　　라벨

X	Y
1	+1
2	+1
4	−1
5	−1
6	+1



- 선형 분류가 불가능하므로 Kernel 적용
- Low degree polynomial kernel function 사용
  - $K(x, y) = (xy+1)^2$
- Tuning parameter C = 100

비선형 SVM의 예

- **비선형 SVM의 α 계산**

$n=5$, $K(x,x) = (x^2+1)^2$

$\Rightarrow K(x,y) = (xy+1)^2$

$$\max_{\alpha} \sum_{i=1}^{n} \boldsymbol{\alpha}_i - \frac{1}{2}\sum_{i=1}^{5} \boldsymbol{\alpha}_i\boldsymbol{\alpha}_j\boldsymbol{y}_i\boldsymbol{y}_j(\boldsymbol{x}_i\boldsymbol{x}_j + 1)^2, \quad subject\ to \sum_{i=1}^{5} \boldsymbol{\alpha}_i\boldsymbol{y}_i = 0,\ 0 \le \boldsymbol{\alpha}_i \le 100$$

$$f(x) = \sum_{i \in SV} \boldsymbol{\alpha}_i\boldsymbol{y}_i K\langle \Phi(x_i), \Phi(x_{new})\rangle + b$$

$\alpha =$

↑ 매트랩, 기타 솔루션 이용 α값 구할 수 있다.

X	Y
1	+1
2	+1
4	−1
5	−1
6	−1

α_1	α_2	α_3	α_4	α_5
0	2.5	0	7.333	4.833
-	SV	-	SV	SV

$\alpha_i \neq 0 \Rightarrow \alpha_i > 0$, $x_i \Rightarrow SV$

$\alpha_i = 0$, $x_i \neq SV$

비선형 SVM의 예

- **비선형 SVM의 α 계산 ➔ b 계산 ➔ 모델 학습 완료**

$$\underset{\alpha}{\text{maximize}} \sum_{i=1}^{n} \boldsymbol{\alpha}_i - \frac{1}{2} \sum_{i=1}^{5} \boldsymbol{\alpha}_i \boldsymbol{\alpha}_j \boldsymbol{y}_i \boldsymbol{y}_j (\boldsymbol{x}_i \boldsymbol{x}_j + 1)^2 , \quad subject\ to \sum_{i=1}^{5} \boldsymbol{\alpha}_i \boldsymbol{y}_i = 0,\ 0 \le \boldsymbol{\alpha}_i \le 100$$

$$SVM\ model = f(x) = \sum_{i \in SV} \boldsymbol{\alpha}_i \boldsymbol{y}_i K\langle \Phi(x_i), \Phi(\boldsymbol{x}_{new}) \rangle + b$$

	X_i	Y_i
i=1	1	+1
i=2	2	+1
i=3	4	−1
i=4	5	−1
i=5	6	−1
	8	?

대입

α_1	α_2	α_3	α_4	α_5
0	2.5	0	7.333	4.833
-	SV	-	SV	SV

$f(x) = 2.5(+1)(2x+1)^2 + 7.333(-1)(5x+1)^2 + 4.833(+1)(6x+1)^2 + b$
$= 0.667x^2 - 5.333x + b$

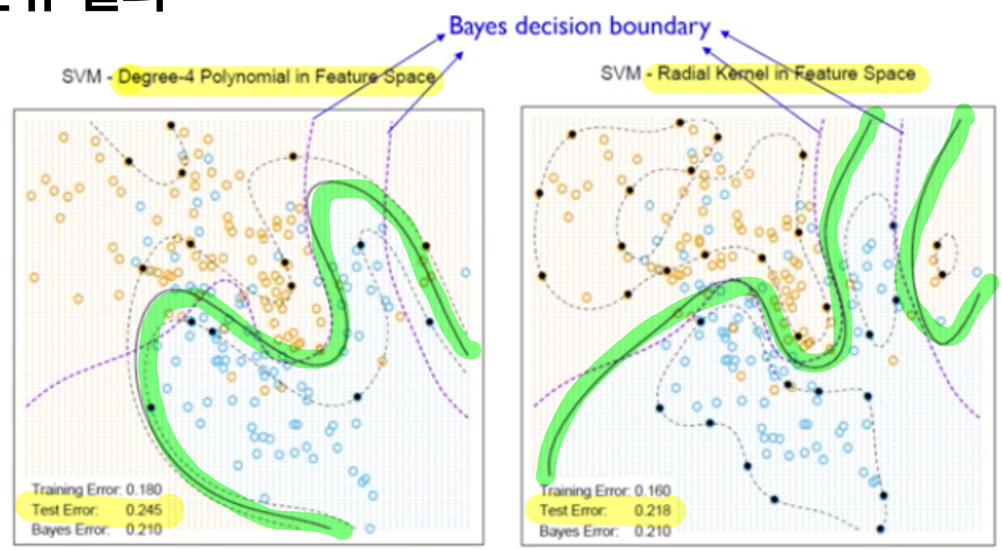
$f(2) = 0.667(2^2) - 5.333(2) + b = 1,\ b \approx 9$

$f(x) = 0.667x^2 - 5.333x + 9$ ← 우리가 얻은 모델 !!

비선형 SVM

- **비선형 SVM 의 커널 선정 법**
  - SVM Kernel 을 결정하는 것은 어려운 문제
    - 정해진 기준이 없으므로, 실험적으로 결정
  - 사용하는 Kernel에 따라 feature space의 특징이 달라지기 때문에 데이터의 특성에 맞는 Kernel을 결정하는 것은 중요함
    - 일반적으로 RBF Kernel, Sigmoid Kernel, Low Degree Polynomial Kernel (4차 미만) 등이 주로 사용됨

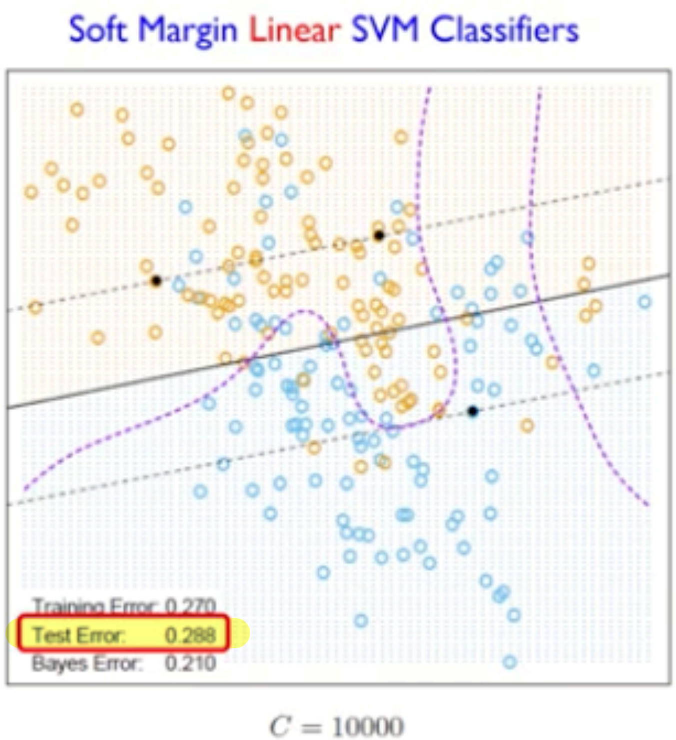
- **커널에 따른 분류 결과**

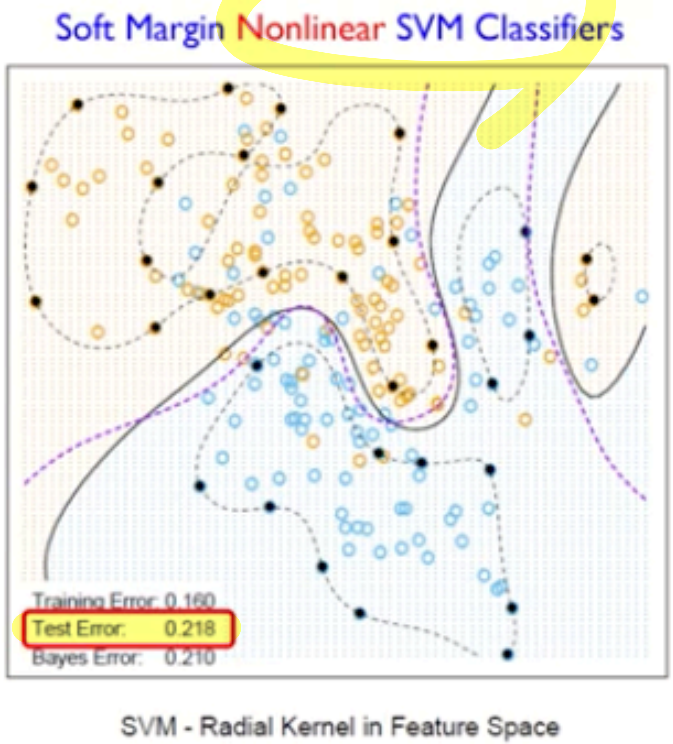


테스트 에러보고
성능 판단 !!

비선형 SVM

- 선형SVM 과 비선형 SVM 분류 결과





SVM 다계층 분류

Multi-Classification

~ 지금까지 2개의 클래스만 분류
(이진 분류)

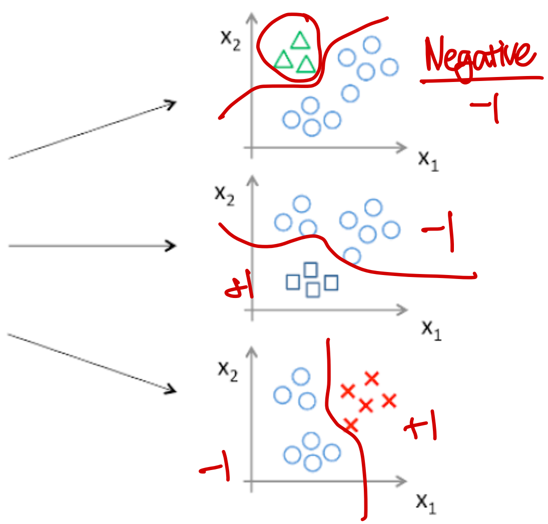
SVM: Multiple Classification

* 어떻게 하면 이진 분류기를 다중 분류기로?

- 하나-나머지 방법(One-vs-Rest) 또는 하나-하나 방법(One-vs-One)

  "OVR" "OVO"
  - 하나-나머지 방법
    - 이항 분류 값(hypothesis function)이 가장 큰 값을 그룹으로 할당
  - 하나-하나 방법
    - 주어진 특성 자료에 대해 가장 많이 할당된 그룹으로 할당 (voting 방식)

One-vs-all (one-vs-rest):

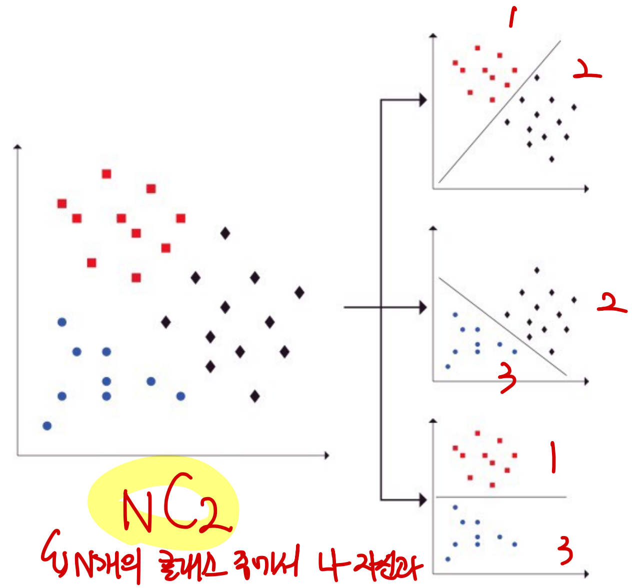


Class 1: Green
Class 2: Blue
Class 3: Red

N개
클래스 수만큼 분류기 필요.

N.

$$\max_i h_\theta^{(i)}(x)$$



$_NC_2$

↳ N개의 클래스 중에서 나 자신과 다른 놈을 뽑아서~

$= {}_3C_2 = \frac{3\times2}{2\times1} = 3$

모든 분류기에 대해서 voting

0 2 1
1 2 3

↳ 그러면 이 새로운 클래스는 class 2로 분류!!

Today We learned.

SVM 분류기

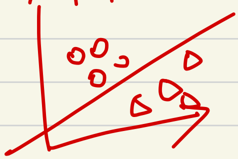
- 이진 분류기
- Deep Learning 전에 널리 사용되던 기법
  기반

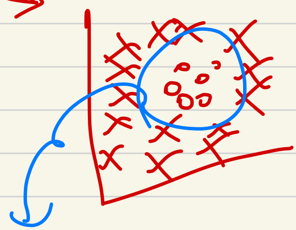
데이터 타입에 따라

→ 선형 SVM ← Hard Margin SVM
　　　　　　　Soft Margin SVM

★ → 비선형 SVM : kernel 도입 ∴ kernel SVM

거리 X





차원 높여서 봄

결정 경계 → 선형 분류

- C : 얼마나 에러를 허용하지 않을 것인가?
- Kernel : Linear, Poly—, RBF, Sigmoid, etc—
- OVO, OVR → 다중분류 시.