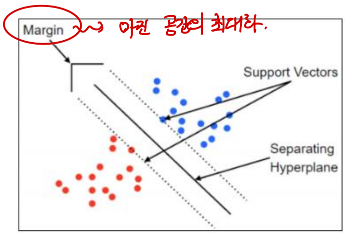
Linear SVM

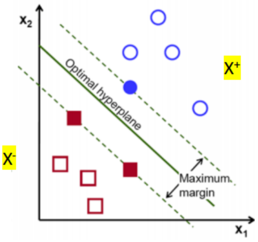
무인이동체공학 17011882 김 우 혁

- 서포트 벡터 머신 (SVM) ⤳ 널리 사용되는 기계학습 방법론
  - 패턴인식, 자료 분석을 위한 지도 학습 모델 → 주로 분류 문제에 사용
  - 주어진 데이터 집합을 바탕으로 하여 새로운 데이터가 어느 카테고리에 속할지 판단하는 비-확률적 이진 선형 분류 모델
  - ☆ 커널 트릭을 활용하여 비선형 분류 문제에도 사용 가능
  - 학습 방향: 마진(Margin)의 최대화 → 결정 경계는 주변 데이터와의 거리 최대가 되어야 함 → 새로운 데이터가 들어와도 강인한 분류
  - 결정 경계: 서로 다른 클래스를 완벽하게 분류하는 기준
  - 서포트 벡터: 결정 경계선 가장 가까이에 있는 각 클래스의 데이터
  - 마진(Margin): 어떤 데이터도 포함하지 않는 영역

Margin ⤳ 마진 공간의 최대화.



- **수학적 표현**
  - 결정 경계의 초 평면 : $W^TX+b = 0$
  - (파랑) 서포트 벡터를 지나는 초 평면: $W^TX+b = 1$, (파랑) 데이터: $W^TX^+ + b \geq 1$
  - (빨강) 서포트 벡터를 지나는 초 평면: $W^TX+b = -1$, (빨강) 데이터: $W^TX^- + b \leq -1$
  - 초 평면의 법선 벡터 : W^T
  - 서로 다른 클래스 데이터의 관계 : $X^+ = X^- + \lambda W$ ⤳ $w^T x^+ + b = 1$, $w^T(x^- + \lambda w) + b = 1$ → $(w^T x^- + b) + \lambda w^T w = 1$, $-1 + \lambda w^T w = 1$ ⇨ $\boxed{\lambda = \frac{2}{w^T w}}$



Margin
$= distance(X^+, X^-)$
$= \|X^+ - X^-\|_2$
$= \|(X^- + \lambda W) - X^-\|_2$
$= \|\lambda W\|_2$
$= \lambda \sqrt{W^T W}$
$= \frac{2}{W^T W}\sqrt{W^T W}$
$= \frac{2}{\sqrt{W^T W}} = \frac{2}{\|W\|_2}$

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

☆ $max\left(\frac{2}{\|W\|_2}\right)$
↳ 이것이 SVM 함수의 목적!

- **SVM의 목적 함수**
  - 마진(Margin)의 최대화

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

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

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

⤳ 계산 편의를 위해 제곱.

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

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

조건

- **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

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

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

Primal Problem

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

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

- Primal Problem 을 Dual Problem 으로 변경하여 해결

Dual Problem

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

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

W는 α, x, y 로 변경

w,b 변수 소거됨.

- (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 $\quad y_i(w^T x_i + b) \geq 1, i = 1,2,\ldots,n$
- Dual feasibility $\quad \alpha_i \geq 0, i - 1,2,\ldots,n$
- Complementary slackness $\quad \alpha_i(y_i(w^T x_i + b) - 1) = 0$

서포트 백터 머신 (SVM)

둘 중 하나가 0

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

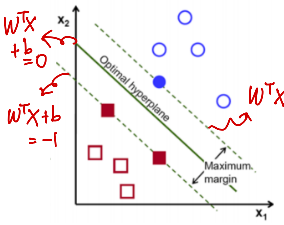
$\alpha \geq 0$, ≥ 1

- $\alpha_i > 0$, $y_i(w^T x_i + b) - 1 = 0$ ✓ 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

- 결정 경계 결정법: 서포트 백터를 가지고 계산



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

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

모델.

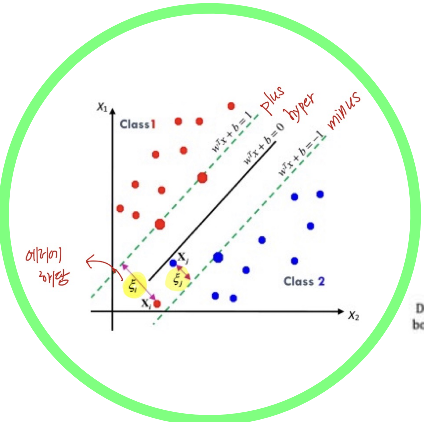
$$y_{new} = sign(W^T X_{new} + b)$$

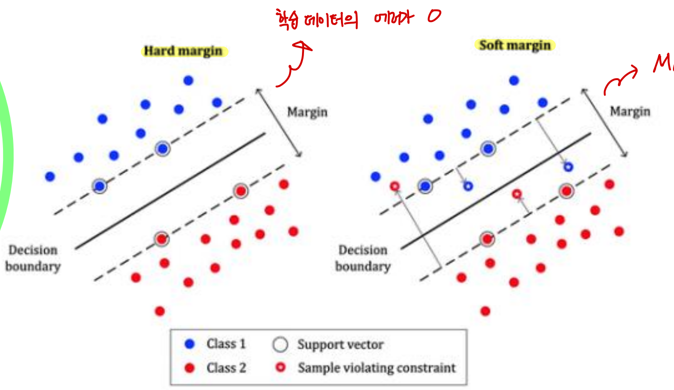
$> 0 \Rightarrow 1$

$< 0 \Rightarrow -1$

Toy example

- Hard Margin SVM – 선형 분리 가능한 문제
- Soft Margin SVM – 선형 분리 불가능 문제, 학습 데이터의 에러가 0이 되도록 완벽하게 나누는 것은 불가능 → 에러를 허용하자!





Margin에 에러가 조금 있다.

Soft SVM 대신, 비선형 분류로 사용할 수도 있다 등

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

↳ 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

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

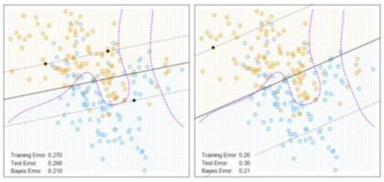
$$subject\ to\ \ y_i(w^T x_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0,\ i = 1,2,\ldots,n$$

↳ 에러 거리에 해당

"C"

- **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=0.01

★ C를 무조건 크게 하는 것이 옳지 X

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

Original Problem

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

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

- **<라그랑지 승수> 해법으로 해결**
  - 제약식(constraints)을 목적식(objective function)에 포함

Primal Problem

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

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

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

Dual Problem

$$\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j \quad \text{W는 } \alpha, y, x \text{ 로 변경}$$

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

- 주의사항

  $0 \leq \alpha_i \leq C,\ from\ \alpha_i \geq 0, \gamma_i \geq 0, C - \alpha_i - \gamma_i = 0$

Hard Margin SVM	Soft Margin SVM
$\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j$	$\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j$
$subject\ to\ \sum_{i=1}^{n}\alpha_i y_i = 0,$	$subject\ to\ \sum_{i=1}^{n}\alpha_i y_i = 0,$
$\mathbf{0 \leq \alpha_i}, i = 1,2,\ldots,n$	$\mathbf{0 \leq \alpha_i \leq C}, i = 1,2,\ldots,n$

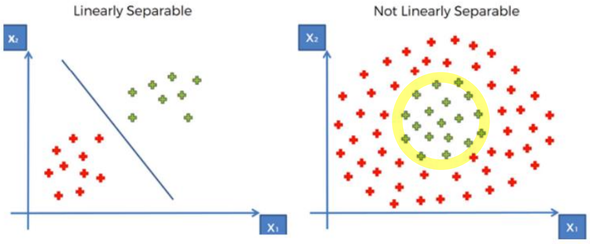
★★★

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

$\alpha \to w, b \to$ 모델

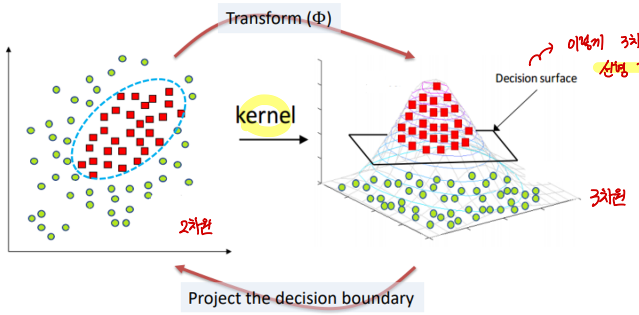
- 서포트 벡터 머신 (SVM)
  - 선형 SVM: 하드 마진(Hard margin)
    소프트 마진(Soft margin) SVM
  - 비선형 SVM: 커널(Kernel) SVM

분류하고자 하는 데이터 타입에 의존



- 비선형 SVM
  - 데이터를 선형으로 분류하기 위해 차원을 높이는 방법을 사용 → 'Feature Map'을 통해 차원을 높임 ➔ 커널: 'Feature Map'의 내적

$\Phi(x)$



이렇게 3차원으로 늘렸더니 선형 결정 경계로 구분 가능하게 되었다.

- **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

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

[예시] 2D ➔ 5D

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

- **비선형 SVM의 목적함수**

Dual Problem

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

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

x_i x_j

feature map을 통해 변형된 데이터가 들어감

Dual Problem

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

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

'kernel'

↳ 연산 효율성↑

비선형 SVM

- Kernel Mapping 의 예

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

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

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

- 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

  $(X,Y)^2 = < (x_1, x_2), (y_1, y_2) >^2$

  $= < x_1y_1 + x_2y_2 >^2$

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

  $= < \Phi(X), \Phi(Y) >$

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

- 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

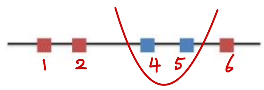
    $K(x_1, x_2) = \tanh(a\langle x_1, x_2 \rangle + b)$

  - Gaussian Kernel (Radial basis function (RBF ~~RGB~~) Kernel)

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

\- 비선형 SVM의 예 →

X (1차원)	Y (라벨)
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 α 계산

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

$$\underset{\alpha}{\text{maximize}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{5} \alpha_i\alpha_j y_i y_j (x_i x_j + 1)^2, \quad subject\ to \sum_{i=1}^{5} \alpha_i y_i = 0,\ 0 \le \alpha_i \le 100$$

$$f(x) = \sum_{i \in SV} \alpha_i y_i K\langle \Phi(x_i), \Phi(x_{new}) \rangle + b$$

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

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

$\alpha_i = 0$, $x_i \not\Rightarrow SV$

- 비선형 SVM의 α 계산 ➔ b 계산 ➔ 모델 학습 완료

$$\underset{\alpha}{\text{maximize}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{5} \alpha_i \alpha_j y_i y_j (x_i x_j + 1)^2, \quad subject\ to \sum_{i=1}^{5} \alpha_i y_i = 0,\ 0 \le \alpha_i \le 100$$

$$SVM\ model = f(x) = \sum_{i \in SV} \alpha_i y_i K\langle \Phi(x_i), \Phi(x_{new}) \rangle + b$$

	X_i	Y_i
i = 1	1	+1
2	2	+1
3	4	-1
4	5	-1
5	6	-1

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

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

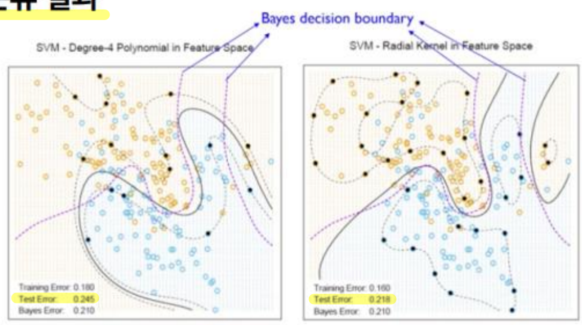
$f(2) = 0.667(2^2) - 5.333(2) + b = 1,\ b \approx 9$

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

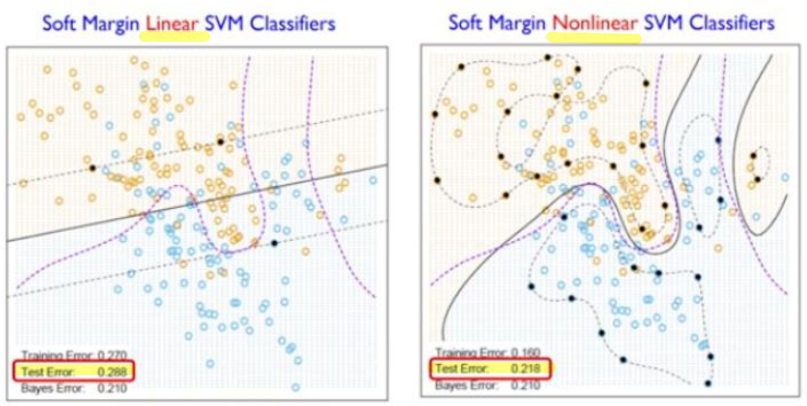
어려운 문제 !

- 비선형 SVM의 커널 선정 법 → 정해진 기준이 없으므로, 실험적으로 결정
- 사용하는 Kernel에 따라 feature space의 특징이 달라짐 → 데이터의 특성에 맞는 Kernel을 결정하는 것은 중요!!
  - 일반적으로 RBF Kernel, Sigmoid Kernel, Low Degree Polynomial Kernel(4차 미만)

커널에 따른 분류 결과



- **선형SVM 과 비선형 SVM 분류 결과**



$C = 10000$

SVM - Radial Kernel in Feature Space

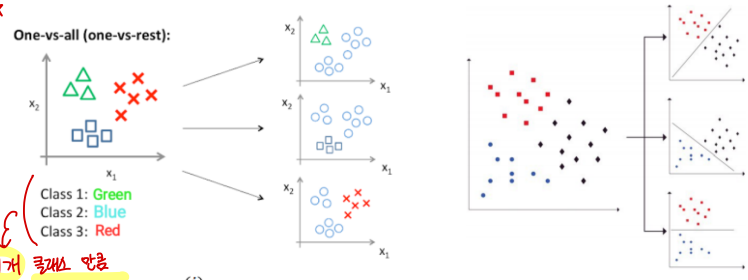
클래스가 3개 이상 !

- SVM 다계층 분류 (Multi - Classification)

SVM: Multiple Classification

OVR　　　　OVO

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



N개 클래스 만큼 분류기 필요.

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