

Linear SVM

Hard Margin SVM

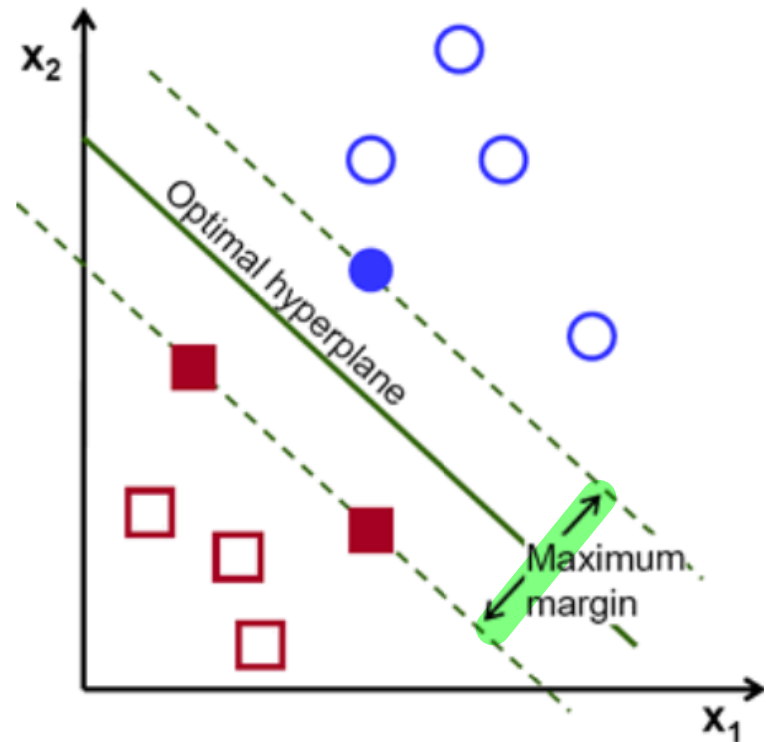
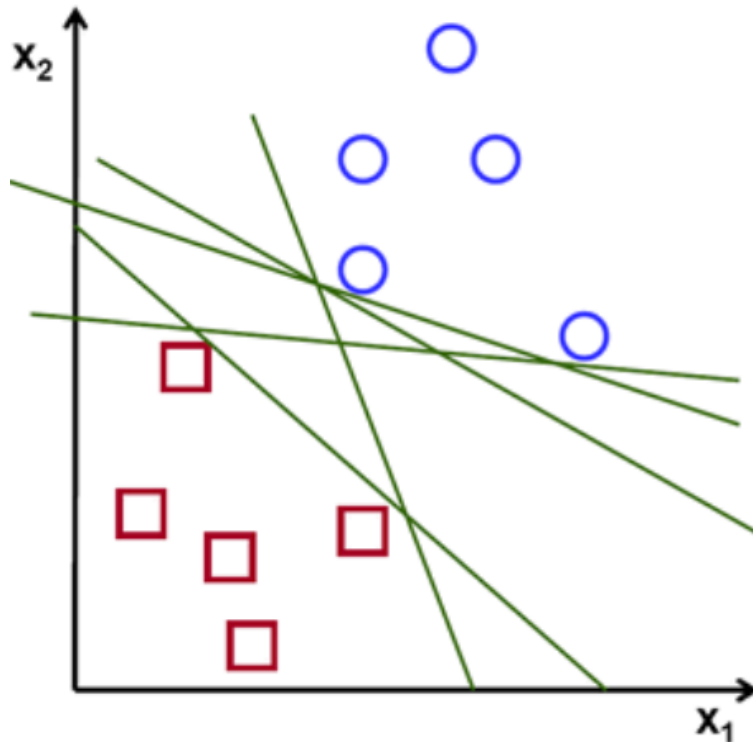
서포트 벡터 머신 (SVM)

- SVM이란?

- 널리 사용되는 기계학습 방법론
- 패턴인식, 자료 분석을 위한 지도 학습 모델 (Supervised Model)
- 분류와 회귀 문제에 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)

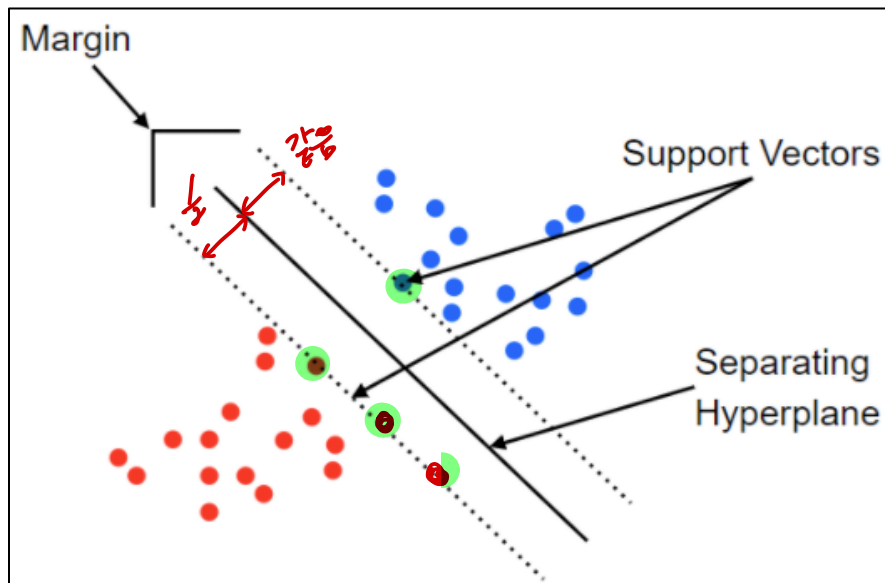
- 서포트 벡터 머신 (Support Vector Machine)
 - SVM 학습 방향 : 마진(Margin)의 최대화
 - 결정 경계(Hyperplane)는 주변 데이터와의 거리가 최대가 되어야 함
(\because 결정 경계 근처에 위치하는 새로운 데이터가 들어와도 강인한 분류)



서포트 벡터 머신 (SVM)

■ 용어

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



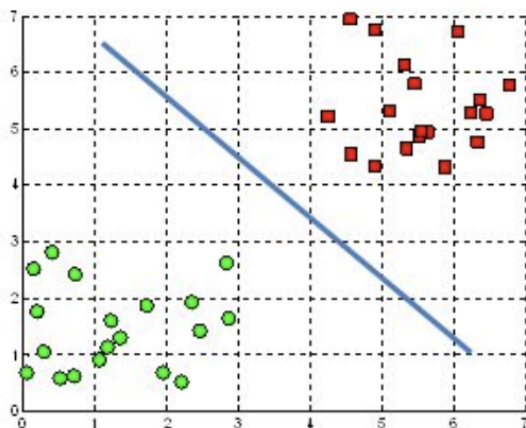
점선먼저 결정 → $\frac{1}{2}$ 지점 hyperplane

서포트 벡터 머신 (SVM)

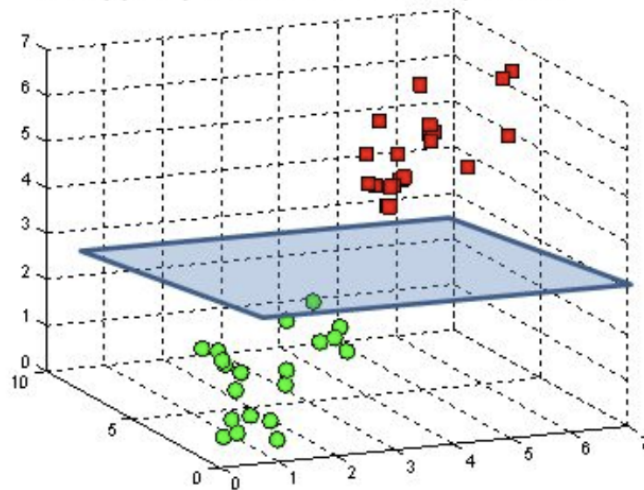
- 결정 경계 (Hyper Plane)

- 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

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



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

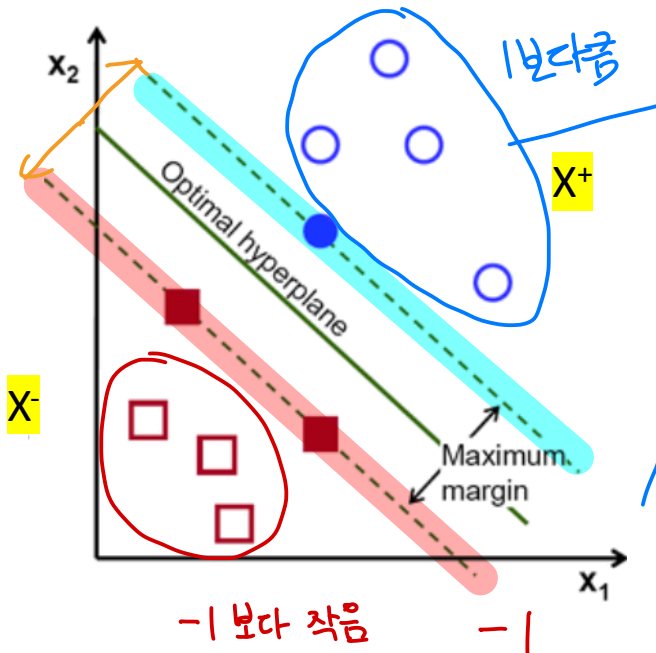


서포트 벡터 머신 (SVM)

■ 수학적 표현

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

→ 2차원 : $y = ax + b$



$$\begin{aligned}
 W^T X^+ + b &= 1 \\
 W^T (X^- + \lambda W) + b &= 1 \\
 (W^T X^- + b) + \lambda W^T W &= 1 \\
 -1 + \lambda W^T W &= 1 \\
 \lambda &= \frac{2}{W^T W}
 \end{aligned}$$

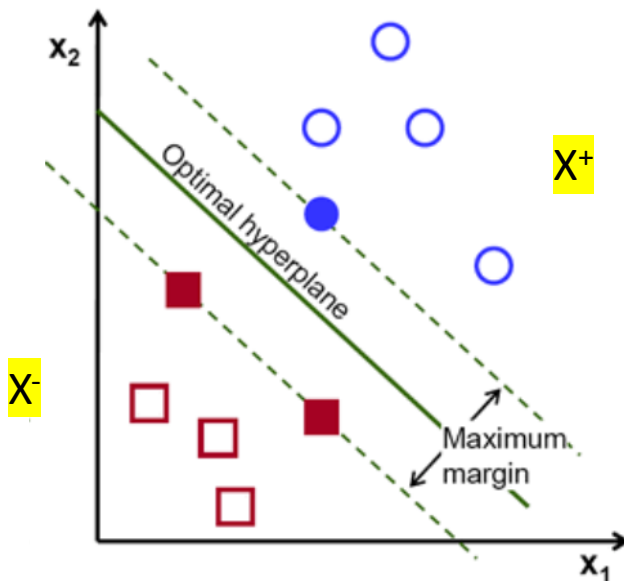
치환

$$\begin{aligned}
 W^T X^+ + b &= 1 \\
 W^T (X^- + \lambda W) + b &= 1 \\
 W^T X^- + W^T \lambda W + b &= 1 \\
 \underbrace{(W^T X^- + b)}_{-1} + \lambda W^T W &= 1 \\
 -1 + \lambda W^T W &= 1 \\
 \lambda W^T W &= 2 \\
 \therefore \lambda &= \frac{2}{W^T W}
 \end{aligned}$$

서포트 벡터 머신 (SVM)

■ 수학적 표현

- 결정 경계의 초 평면 : $W^T X + b = 0$
- (파랑) 서포트 벡터를 지나는 초 평면: $W^T X + b = 1$, (파랑) 데이터: $W^T X + b \geq 1$
- (빨강) 서포트 벡터를 지나는 초 평면: $W^T X + b = -1$, (빨강) 데이터: $W^T X + b \leq -1$
- 초 평면의 법선 벡터 : W^T
- 서로 다른 클래스 데이터의 관계 : $X^+ = X^- + \lambda W$



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

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

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

최대화하는 것 이) SVM 함수의 목적

서포트 벡터 머신 (SVM)

- SVM의 목적 함수
 - 마진(Margin)의 최대화

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

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

최대화 문제 최소화 문제로 변경

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

계산상의 편의

☆

$y \dots ?$

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

$w^T x + b = 0$ 가르기

최소화하는 w, b 구하기

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

조건 $\left\{ \begin{array}{l} w^T x^+ + b \geq 1, y = 1 \\ w^T x^- + b \leq -1, y = -1 \end{array} \right\}$

합해주면 한변이 4표현

서포트 벡터 머신 (SVM)

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

Original Problem

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

object function 목적식

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

constraints function 제약식

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

↓ 주식을 하나로 합친 식

Primal Problem

$$\max_{\alpha} \min_{w,b} \mathcal{L}(w, b, \alpha) = \frac{1}{2} \|w\|_2^2 - \sum_{i=1}^n \alpha_i (y_i (w^T x_i + b) - 1)$$
$$\text{subject to } \alpha_i \geq 0, i = 1, 2, \dots, n$$

서포트 벡터 머신 (SVM)

- Primal Problem 을 Dual Problem 으로 변경하여 해결

Dual Problem

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \\ \text{subject to} \quad & \sum_{i=1}^n \alpha_i y_i = 0, \alpha_i \geq 0, i = 1, 2, \dots, n \end{aligned}$$

최대화할수있는 문제

W는 α, x, y 로 변경

- (W, b, α)가 Lagrangian dual problem의 최적해가 되기 위한 조건

- KKT (Karush-Kuhn-Tucker) Conditions

- Stationarity

$$w = \sum_{i=1}^n \alpha_i y_i x_i \quad \sum_{i=1}^n \alpha_i y_i = 0$$

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

서포트 벡터 머신 (SVM)

Complementary slackness

$$\alpha_i \geq 0 \quad \text{조건}$$

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

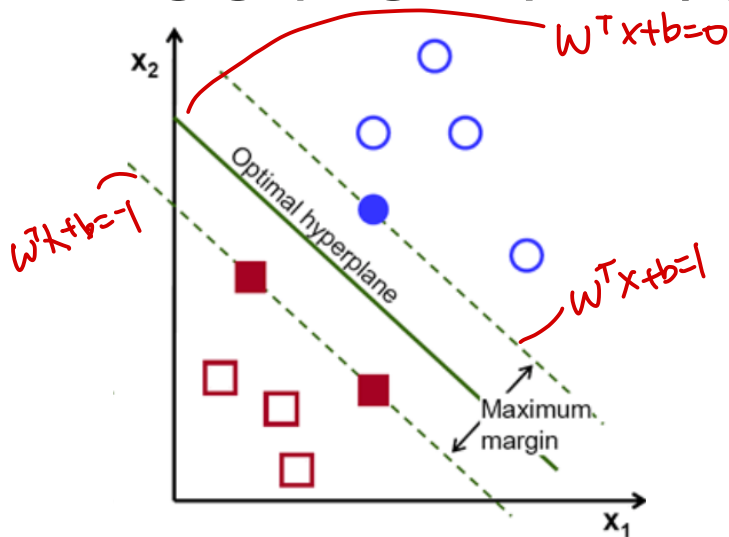
$$\begin{matrix} +1 & -1 \end{matrix}$$

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

$$\alpha_i(y_i(w^T x_i + b) - 1) = 0 \quad \therefore \alpha_i = 0 \text{ or } y_i(w^T x_i + b) - 1 = 0$$

- ✓ x_i 가 support vector(SV) 인 경우
- ✓ x_i 가 support vector(SV)가 아닌 경우
- ✓ Hyperplane 구축에 영향을 미치지 않음
- ✓ SVM이 outlier에 강인한 이유이기도 함
= noise

결정 경계 결정법: 서포트 벡터를 가지고 계산



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

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

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

$$W^T X_{new} + b > 0 \rightarrow 1$$

$$< 0 \rightarrow -1$$

Toy example

Linear SVM

Soft Margin SVM

서포트 벡터 머신 (SVM)

- Hard Margin SVM vs Soft Margin SVM

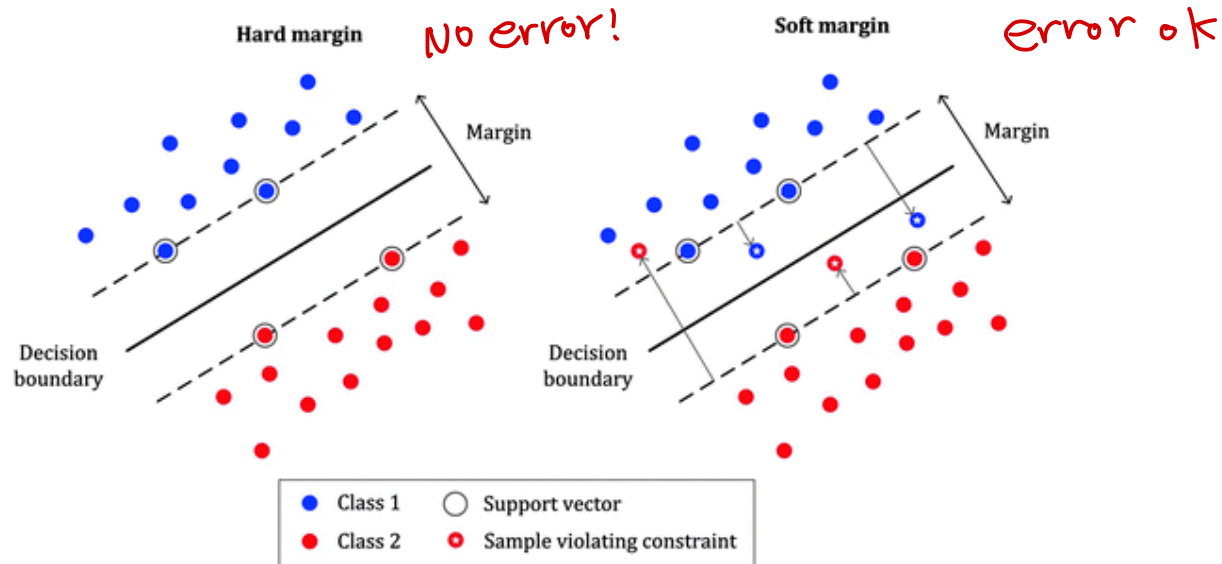
- Hard Margin SVM

- 선형 분리 가능한 문제

- Soft Margin SVM

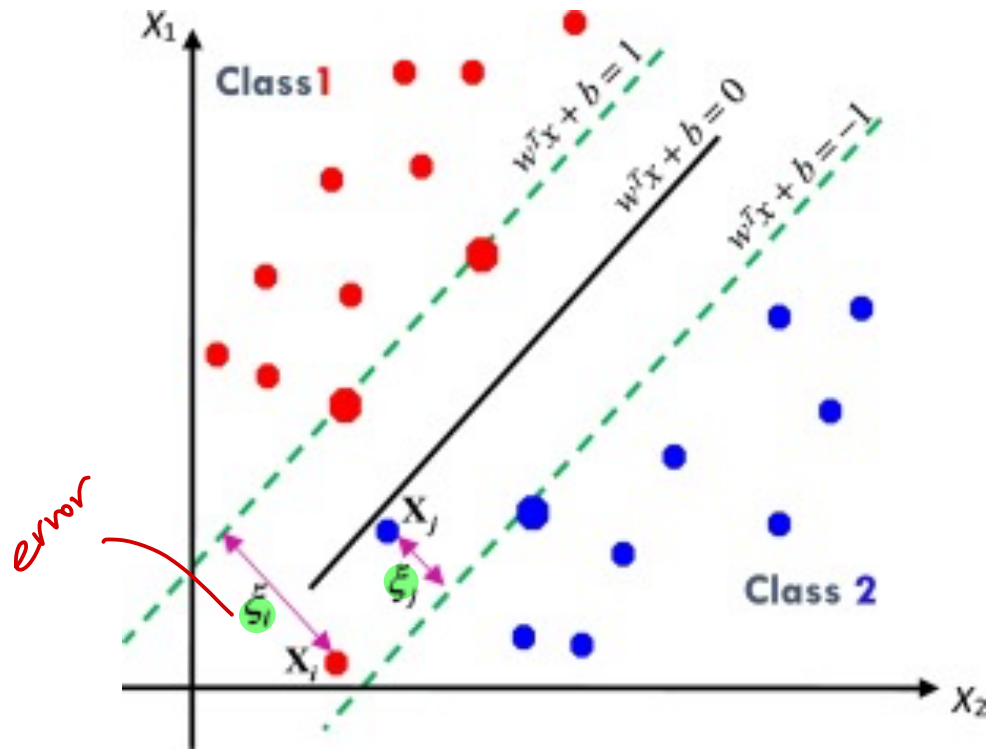
- 선형 분리 불가능 문제
 - 학습 데이터의 에러가 0 이 되도록 완벽하게 나누는 것을 불가능

→ 에러를 허용 하자



Soft Margin SVM

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



Soft Margin SVM

$$\begin{aligned} \min_{w, b} \quad & \frac{1}{2} \|w\|_2^2 \\ \text{subject to} \quad & y_i(w^T x_i + b) \geq 1, \quad i=1, \dots, n \end{aligned}$$

Soft Margin SVM

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

Original Problem

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

Penalty term

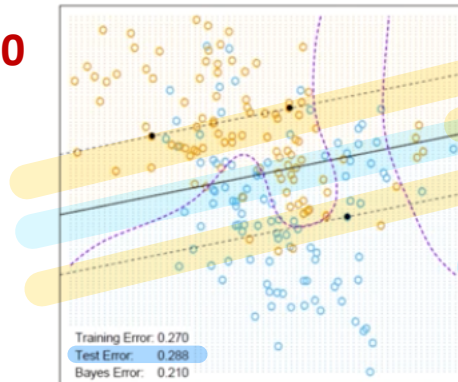
$$\text{subject to } y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, 2, \dots, n$$

"C" Parameter

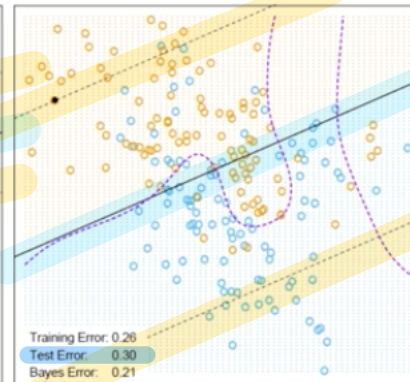
C를 통해 에러의 허용 정도 조절

- C가 크면 학습 에러를 상대적으로 허용하지 않음 (Overfitting) → 마진 작아짐
- C가 작으면 학습 에러를 상대적으로 허용 (Underfitting) → 마진이 커짐

C=10000



C=0.01



C ↑ 에러 ↓

C ↓ 에러 ↑

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

$$\text{subject to } y_i(w^T X_i + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, 2, \dots, n$$

- 〈라그랑지 승수〉 해법으로 해결

- 제약식(constraints)을 목적식(objective function)에 포함

Primal Problem

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

$$\text{subject to } \alpha_i \gamma_i \geq 0, i = 1, 2, \dots, n$$

Soft Margin SVM

- Primal Problem 을 Dual Problem 으로 변경하여 해결

Dual Problem

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

W 는 α, y, x 로 변경

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

hard margin 차이점

- 주의사항

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

Hard Margin SVM vs Soft Margin SVM

Hard Margin SVM

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

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

$$0 \leq \alpha_i, i = 1, 2, \dots, n$$

Soft Margin SVM

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

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

$$0 \leq \alpha_i \leq C, i = 1, 2, \dots, n$$

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

$\alpha \rightarrow w, b \rightarrow \text{모델}$

Nonlinear SVM

Kernel SVM

서포트 벡터 머신 (SVM)

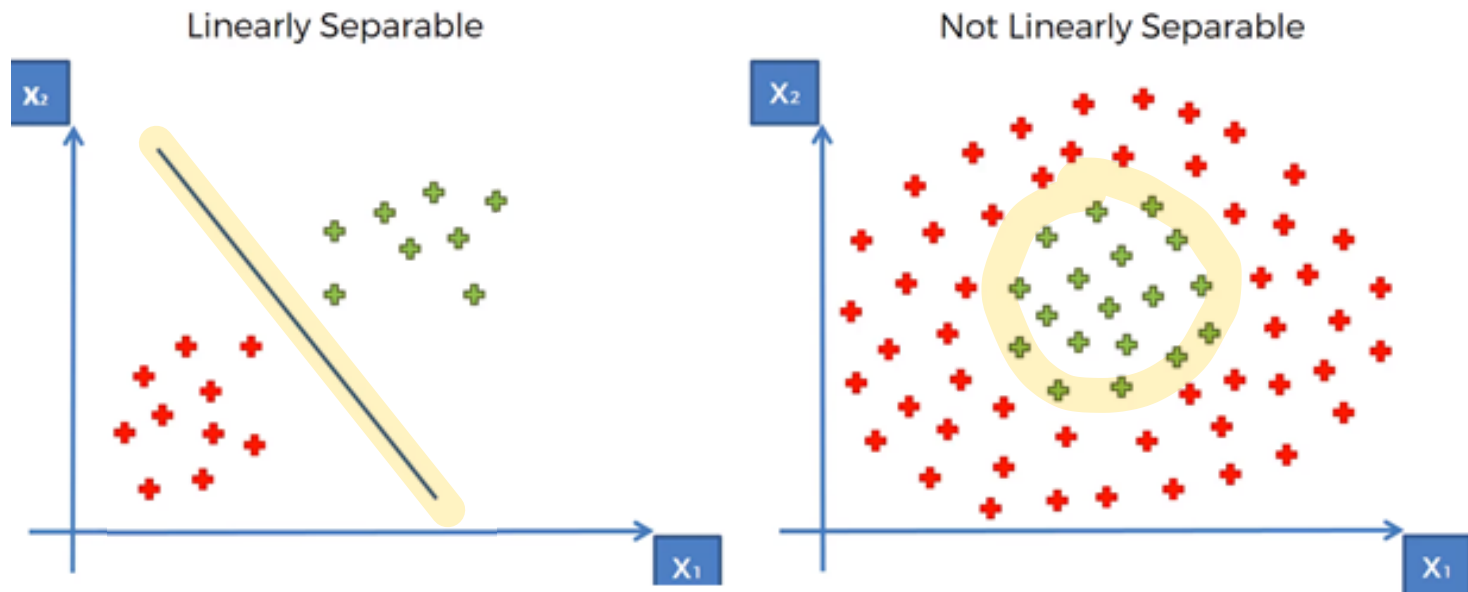
- 선형 SVM vs 비선형 SVM

- 선형 SVM

- 하드 마진(Hard margin) SVM, 소프트 마진(Soft margin) SVM

- 비선형 SVM

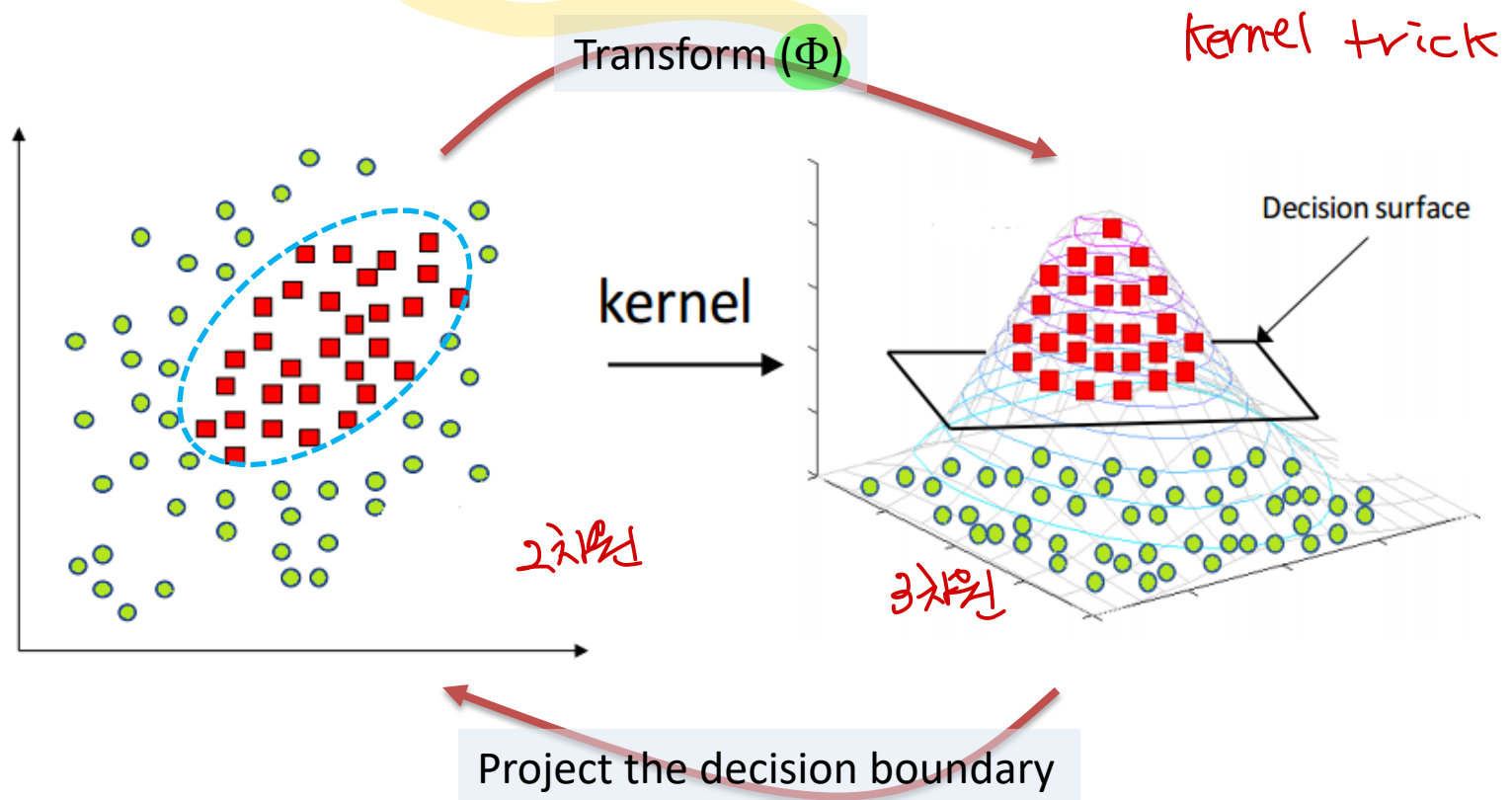
- 커널(Kernel) SVM



비선형 SVM

- 비선형 SVM

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



비선형 SVM

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

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

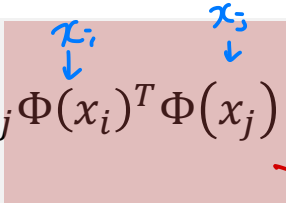
[예시] 2D \rightarrow 5D

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

비선형 SVM

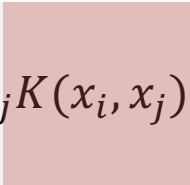
■ 비선형 SVM의 목적함수

Dual Problem

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


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

Dual Problem

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


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

내적

"kernel"
평상효용성

비선형 SVM

- Kernel Mapping 의 예

$$X = (x_1, x_2), Y = (y_1, y_2)$$

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

$$\langle \Phi(X), \Phi(Y) \rangle = \underline{x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2} \quad \text{연산량이 너무 큼}$$

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

$$\begin{aligned}(X, Y)^2 &= \langle (x_1, x_2), (y_1, y_2) \rangle^2 \\&= \langle x_1y_1 + x_2y_2 \rangle^2 \\&= x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2 \\&= \langle \Phi(X), \Phi(Y) \rangle\end{aligned}$$

- $(X, Y)^2 = \langle (x_1, x_2), (y_1, y_2) \rangle^2 = K(X, Y) \rightarrow \text{Kernel Function}$

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

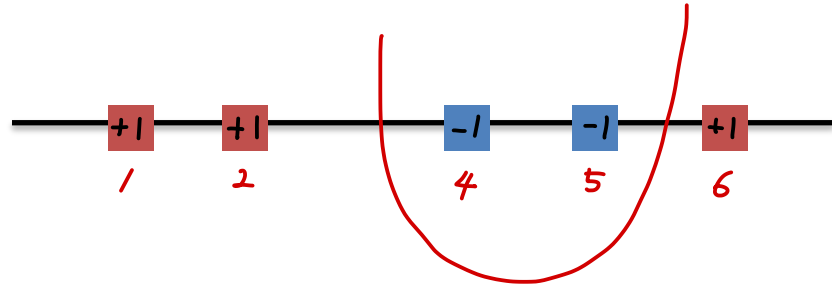
$$K(x_1, x_2) = \tanh(a\langle x_1, x_2 \rangle + b)$$

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

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

비선형 SVM의 예

차원 data	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

비선형 SVM 목적함수

$n=5, k(x, x) \Rightarrow k(x, y) = (xy+1)^2$

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

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

$\lambda =$

hyper Plane을 결정짓는
Support vector

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

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

$\lambda_i = 0, x_i \neq SV$

비선형 SVM의 예

- 비선형 SVM의 α 계산 \rightarrow b 계산 \rightarrow 모델 학습 완료

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

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

	X_i	Y_i
$\bar{1}=1$	1	+1
$\bar{1}=2$	2	+1
$\bar{1}=3$	4	-1
$\bar{1}=4$	5	-1
$\bar{1}=5$	6	-1

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

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

$$= 0.667x^2 - 5.333x + b$$

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

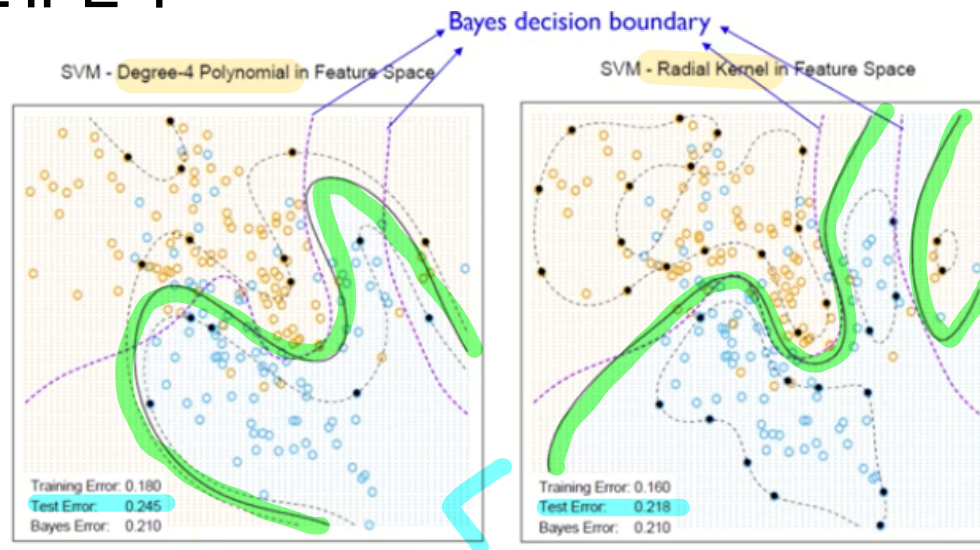
$$f(x) = 0.667x^2 - 5.333x + 9 \quad \text{모델}$$

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

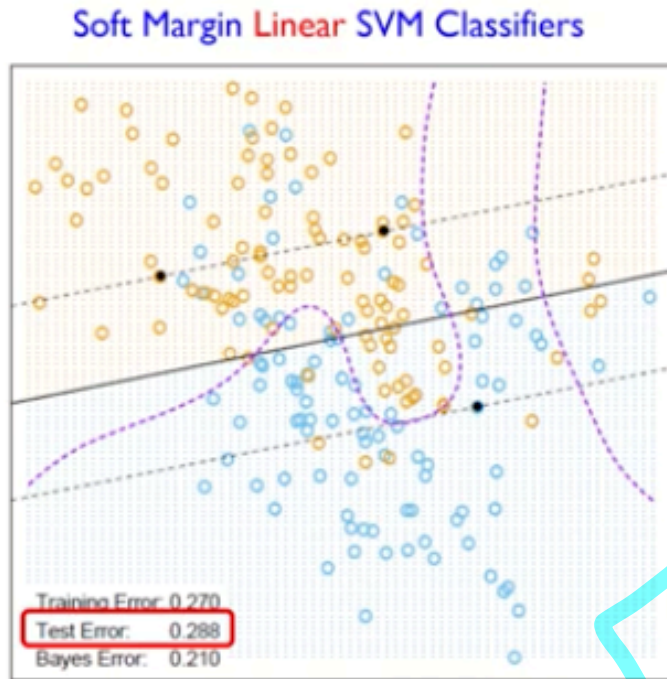
■ 커널에 따른 분류 결과



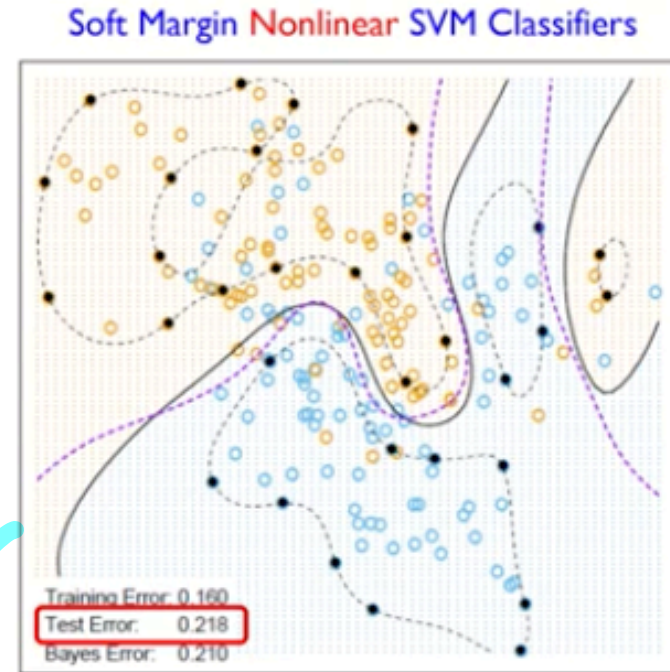
- 더 좋음

비선형 SVM

- 선형SVM 과 비선형 SVM 분류 결과



$C = 10000$



SVM - Radial Kernel in Feature Space

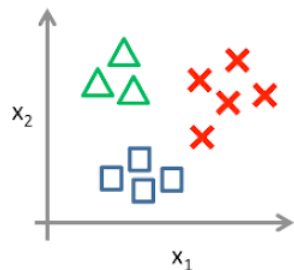
SVM 다계층 분류

Multi-Classification

SVM: Multiple Classification

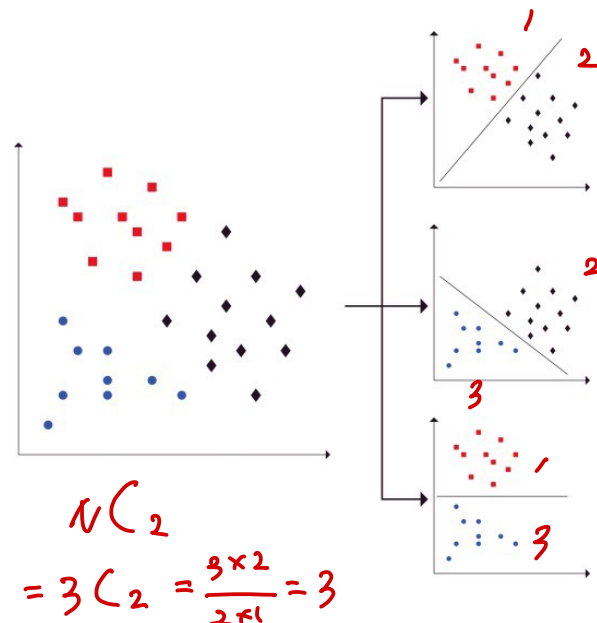
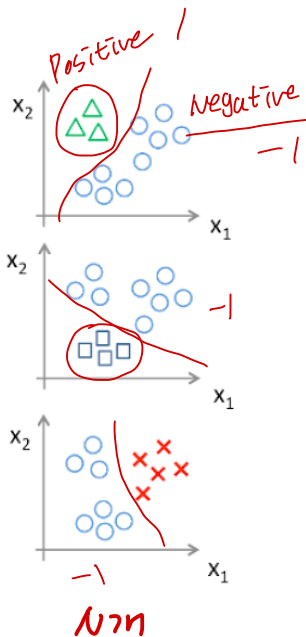
- 하나-나머지 방법 (One-vs-Rest) 또는 하나-하나 방법 (One-vs-One)**
 - 하나-나머지 방법 "OVR"**
 - 이항 분류 값(hypothesis function)이 가장 큰 값을 그룹으로 할당
 - 하나-하나 방법 "OVO"**
 - 주어진 특성 자료에 대해 가장 많이 할당된 그룹으로 할당 (voting 방식)

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



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

$$\max_i h_{\theta}^{(i)}(x)$$
 클래스 수 = 분류기 수



SVM 분류기

- 이진 분류기

- Deep Learning 전, 널리 사용되던 기법

- 선형 SVM $\begin{cases} \text{Hard Margin SVM} \\ \text{Soft Margin SVM : error 허용} \end{cases}$

★★
- 비선형 SVM : kernel SVM
 x - 고차원 $\rightarrow \Phi(x)$

C : 얼마나 에러를 허용할 것인가?
- kernel: Linear, Poly, RBF, Sigmoid, etc
- OVO, OVR