Linear SVM

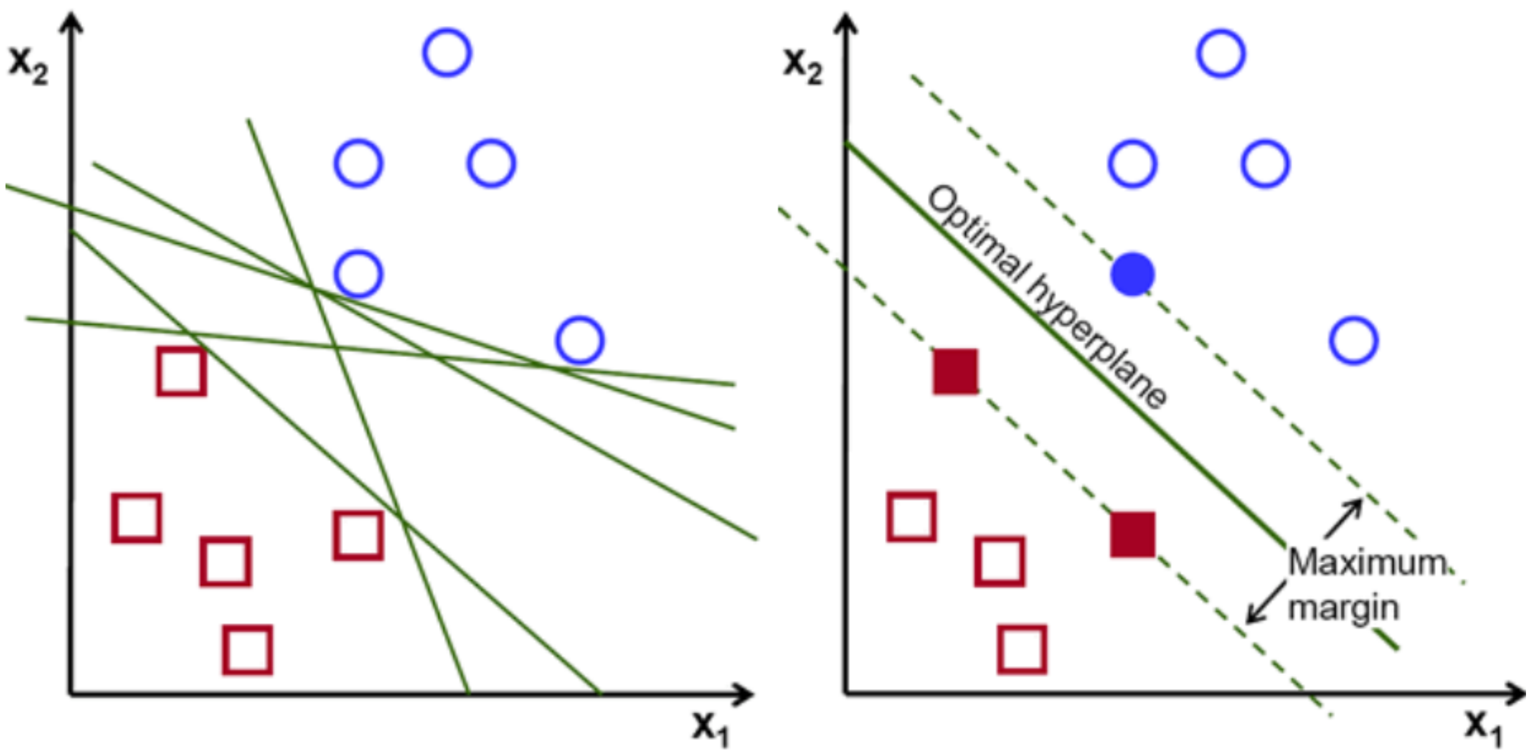
Hard Margin SVM

서포트 벡터 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**

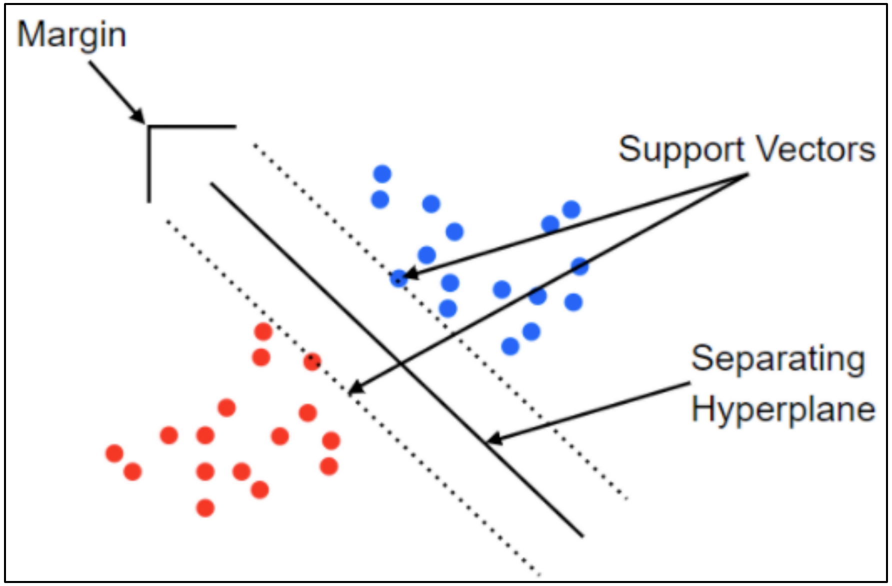
서포트 벡터 머신 (SVM)

- **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 벡터 머신 (SVM)

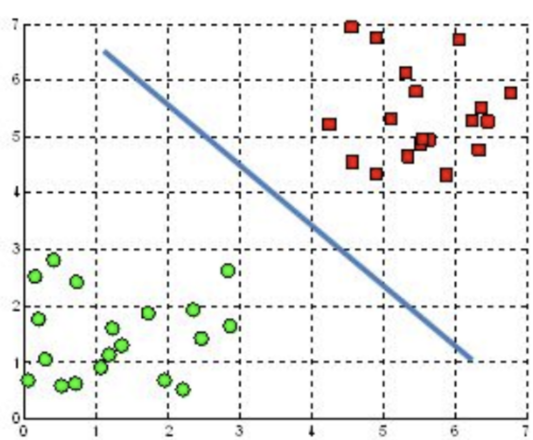
- **용어**
  - 결정 경계 (Huperplane)
    - 서로 다른 클래스를 완벽하게 분류하는 기준
  - 서포트 벡터 (Support Vector)
    - 결정 경계선에 가장 가까이에 있는 각 클래스의 데이터
  - 마진 (Margin)
    - 어떤 데이터도 포함하지 않는 영역, 서포트 벡터와 직교하는 직선과의 거리



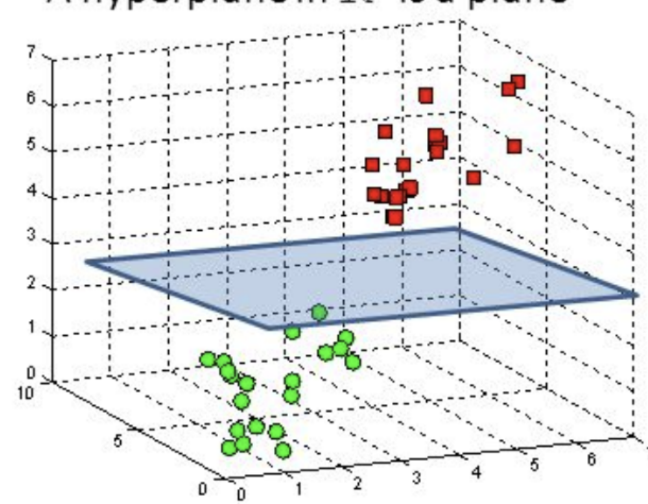
서포트 백터 머신 (SVM)

- **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

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

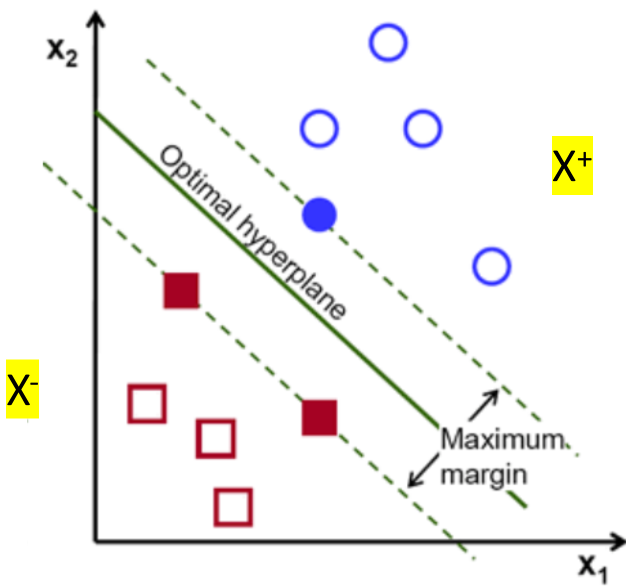


A hyperplane in $\mathbb{R}^3$ is a plane



서포트 벡터 머신 (SVM)

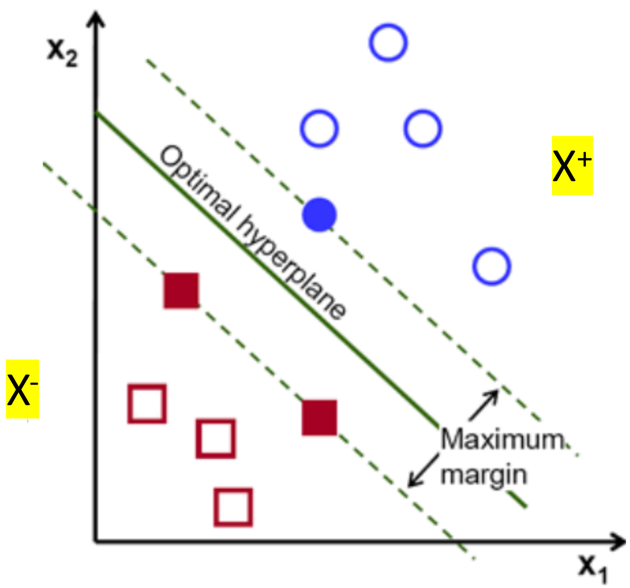
- **수학적 표현**
  - 결정 경계(초 평면) : $W^TX+b = 0$
  - (파랑) 서포트 벡터를 지나는 초 평면: $W^TX+b = 1$, (파랑) 데이터: $W^TX^+ + b$ 〉=1
  - (빨강) 서포트 벡터를 지나는 초 평면: $W^TX+b = -1$, (빨강) 데이터: W^TX^-+b 〈=−1
  - 초 평면의 법선 벡터 : W^T
  - **서로 다른 클래스 데이터의 관계 : $X^+=X^-+\lambda W$**



$$W^TX^+ + b = 1$$
$$W^T(X^-+\lambda W) + b = 1$$
$$(W^TX^-+b) + \lambda W^TW = 1$$
$$-1+ \lambda W^TW = 1$$
$$\lambda=\frac{2}{W^TW}$$

서포트 백터 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

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

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

서포트 백터 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

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

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

서포트 백터 머신 (SVM)

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

Original Problem

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

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

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

Primal Problem

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

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

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

서포트 백터 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, α)가 Lagrangian dual problem의 최적해가 되기 위한 조건**
  - KKT (Karush-Kuhn-Tucker) Conditions
    - Stationarity $$w = \sum_{i=1}^{n} \alpha_i y_i x_i \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$$
    - 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 백터 머신 (SVM)

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

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

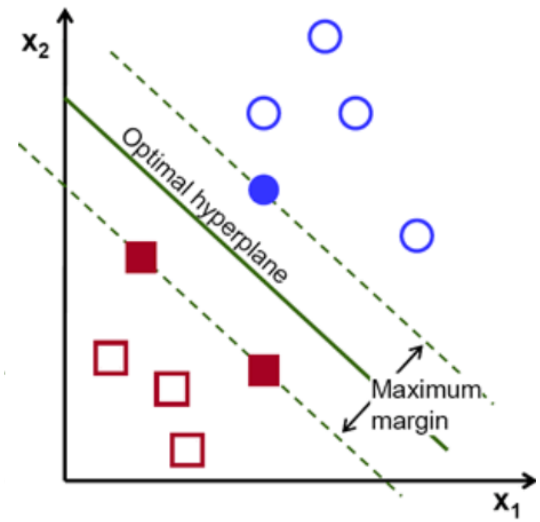
✓ x_i가 support vector(SV) 인 경우

✓ 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



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

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

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

Toy example

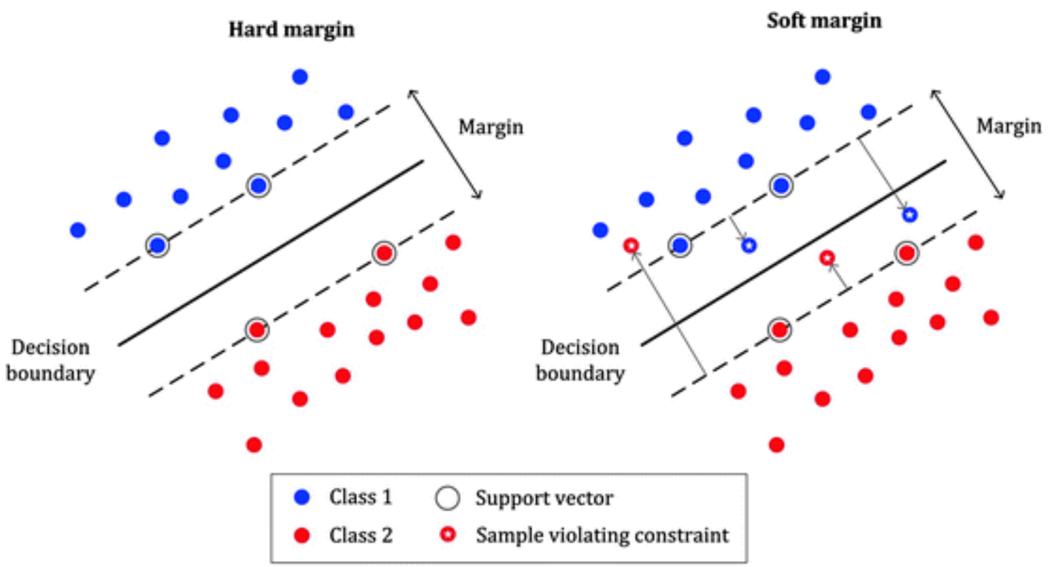
Linear SVM

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

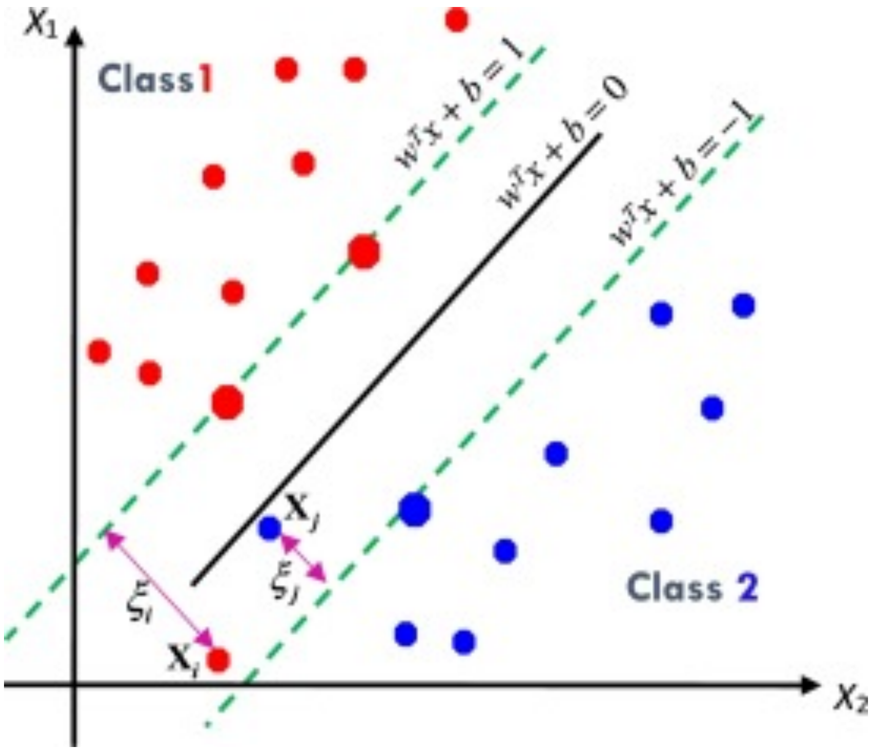
      ➔ **에러를 허용 하자**



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

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

Original Problem

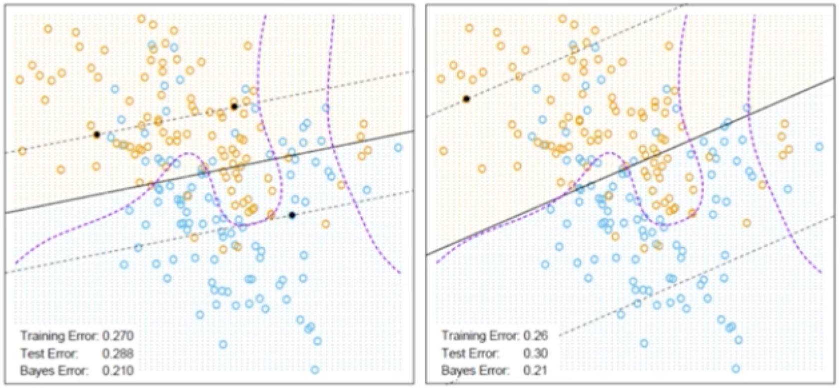
$$\min_{w,b,\xi} \frac{1}{2}\|w\|_2^2 + \boxed{C\sum_{i=1}^{n}\xi_i}$$

Penalty term

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

  - 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

Soft Margin SVM

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

Original Problem

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

Penalty term

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

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

Primal Problem

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

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

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

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

- 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 vs Soft Margin SVM

Hard Margin SVM

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

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

$$\mathbf{0 \le \boldsymbol{\alpha}_i},\ i = 1,2,\ldots,n$$

Soft Margin SVM

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

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

$$\mathbf{0 \le \boldsymbol{\alpha}_i \le \boldsymbol{C}},\ i = 1,2,\ldots,n$$

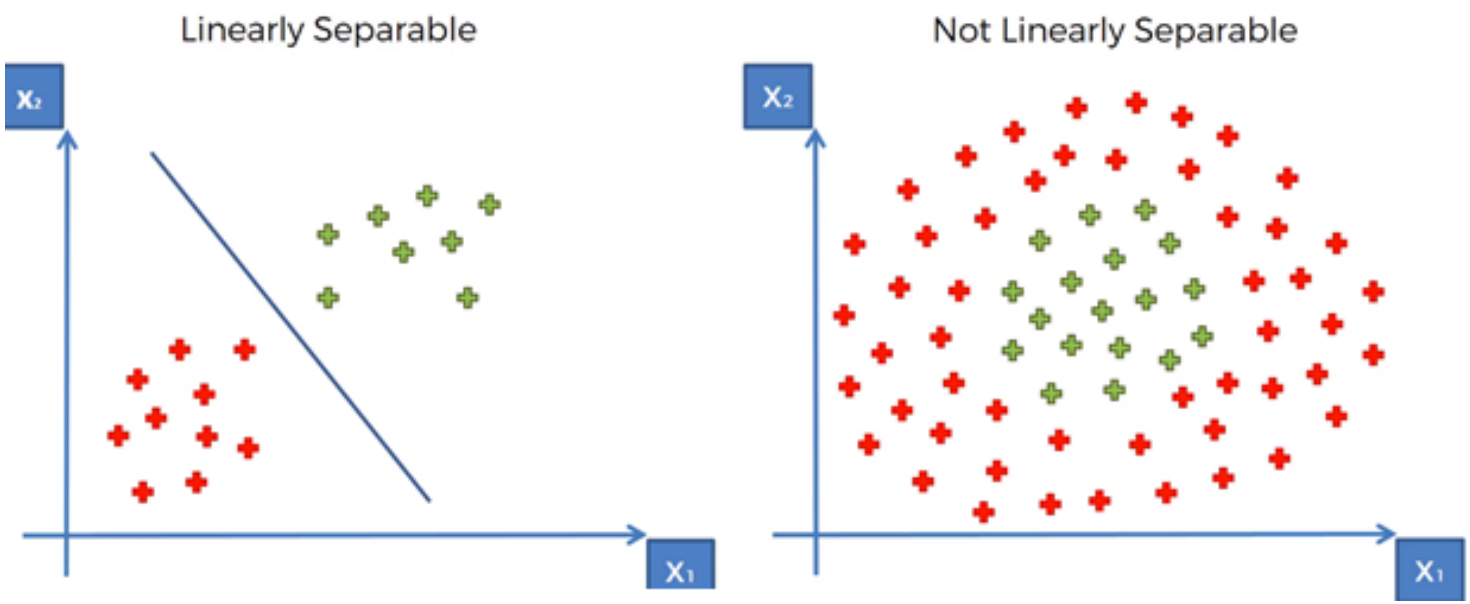
- 선형 분류의 두 가지 케이스 모두 quadratic programming 을 풀어 α 를 구함
  - 간접적으로 W를 구한 셈

Nonlinear SVM

Kernel SVM

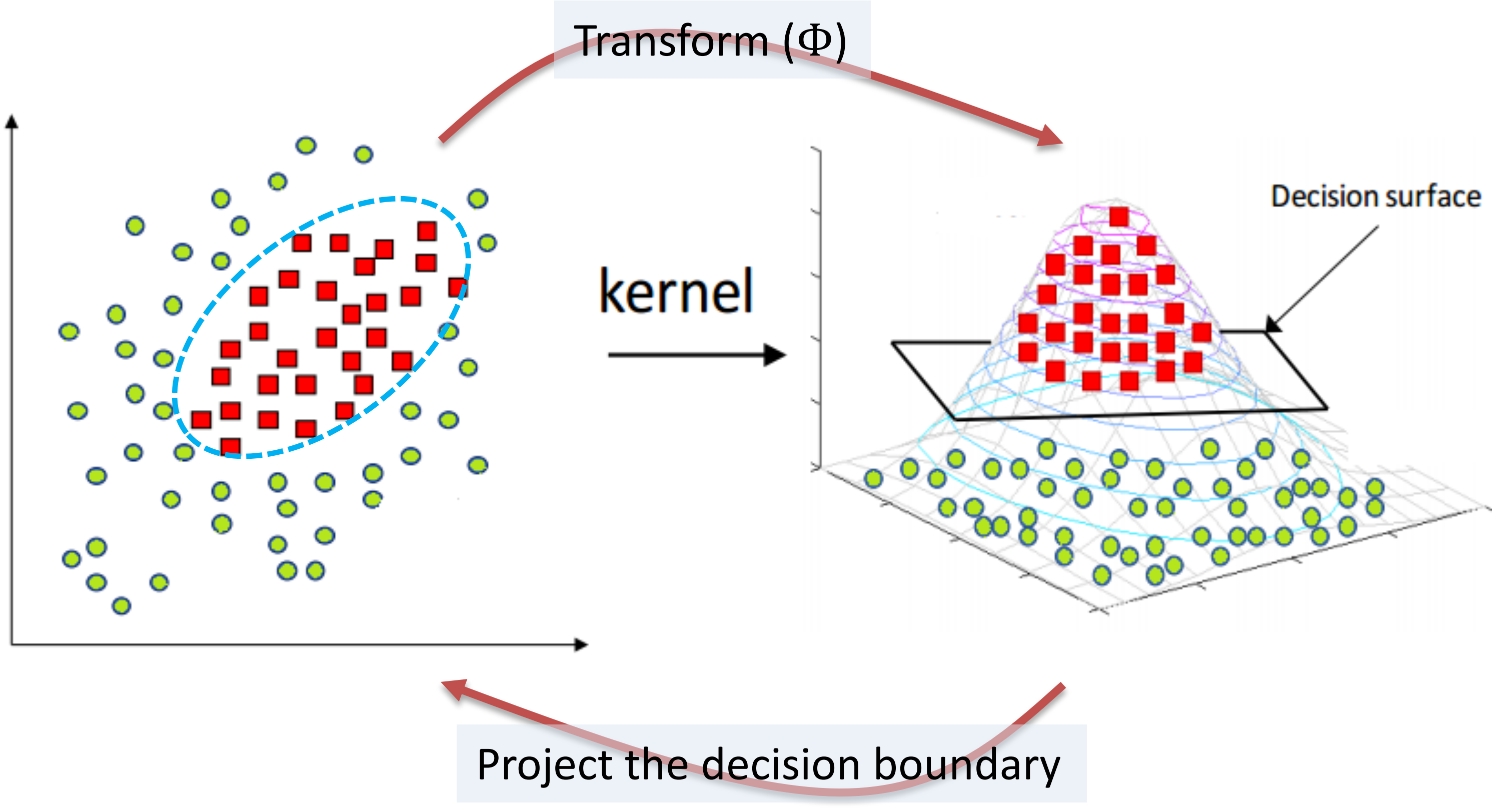
서포트 백터 머신 (SVM)

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



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**



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**

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

[예시] 2D ➔ 5D

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

비선형 SVM

- **비선형 SVM의 목적함수**

Dual Problem

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

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

Dual Problem

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

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

비선형 SVM

- **Kernel Mapping 의 예**

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

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

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

- **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**

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 (RGB) Kernel)

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

비선형 SVM의 예

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**

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

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

X	Y
1	+1
2	+1
4	−1
5	−1
6	−1

$\boldsymbol{\alpha_1}$	$\boldsymbol{\alpha_2}$	$\boldsymbol{\alpha_3}$	$\boldsymbol{\alpha_4}$	$\boldsymbol{\alpha_5}$
0	2.5	0	7.333	4.833
-	SV	-	SV	SV

비선형 SVM의 예

- **비선형 SVM의 α 계산 ➔ b 계산 ➔ 모델 학습 완료**

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

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

X_i	Y_i
1	+1
2	+1
4	−1
5	−1
6	−1

$\boldsymbol{\alpha_1}$	$\boldsymbol{\alpha_2}$	$\boldsymbol{\alpha_3}$	$\boldsymbol{\alpha_4}$	$\boldsymbol{\alpha_5}$
0	2.5	0	7.333	4.833
-	SV	-	SV	SV

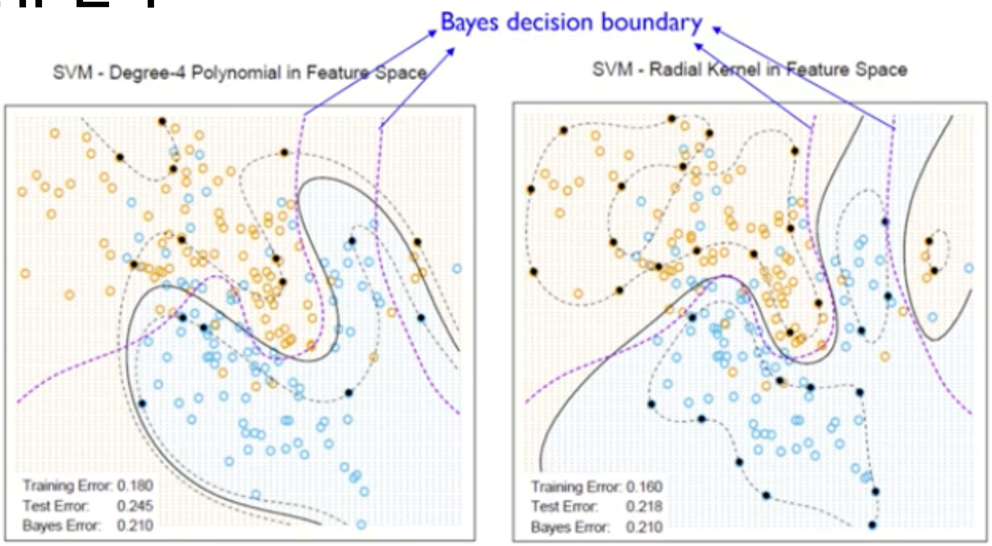
$f(x) = 2.5(+1)(2x+1)^2 + 7.333(-1)(5x+1)^2 + 4.833(+1)(6x+1)^2 + b$
$= 0.667x^2 - 5.333x + b$

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

$f(x) = 0.667x^2 - 5.333x + 9$

비선형 SVM

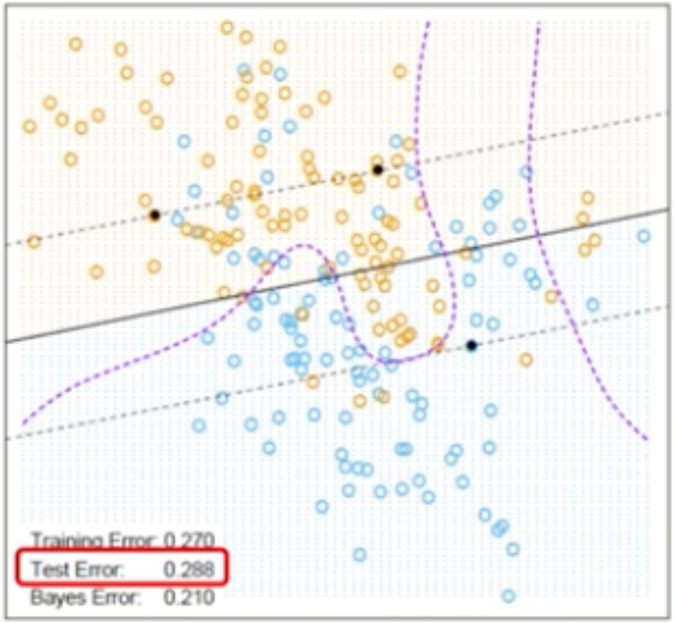
- **비선형 SVM 의 커널 선정 법**
  - SVM Kernel 을 결정하는 것은 어려운 문제
    - 정해진 기준이 없으므로, 실험적으로 결정
  - 사용하는 Kernel에 따라 feature space의 특징이 달라지기 때문에 데이터의 특성에 맞는 Kernel을 결정하는 것은 중요함
    - 일반적으로 RBF Kernel, Sigmoid Kernel, Low Degree Polynomial Kernel (4차 미만) 등이 주로 사용됨
- **커널에 따른 분류 결과**



비선형 SVM

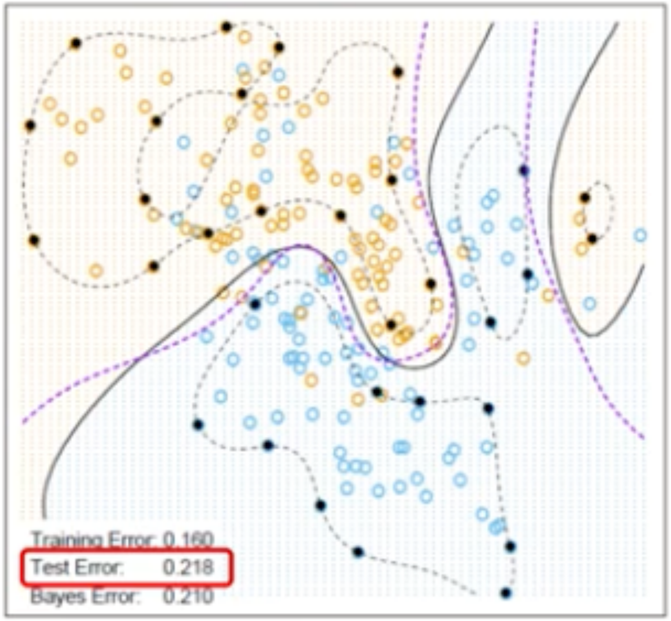
- **선형SVM 과 비선형 SVM 분류 결과**

Soft Margin Linear SVM Classifiers



$C = 10000$

Soft Margin Nonlinear SVM Classifiers



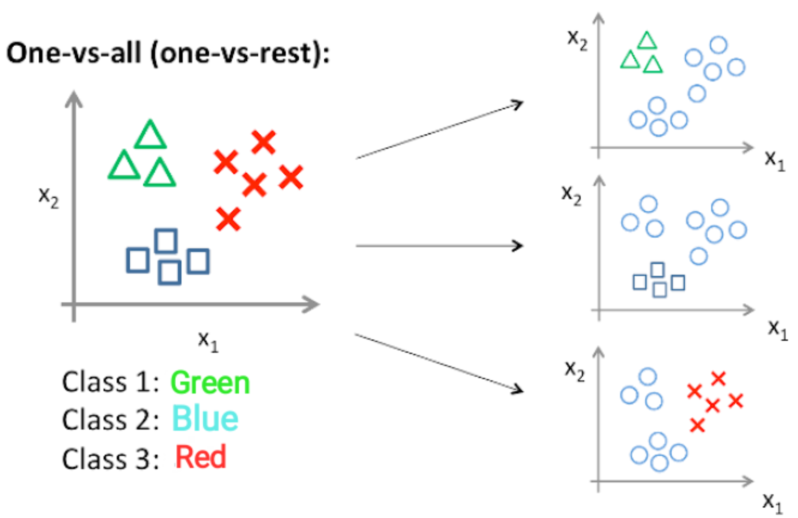
SVM - Radial Kernel in Feature Space

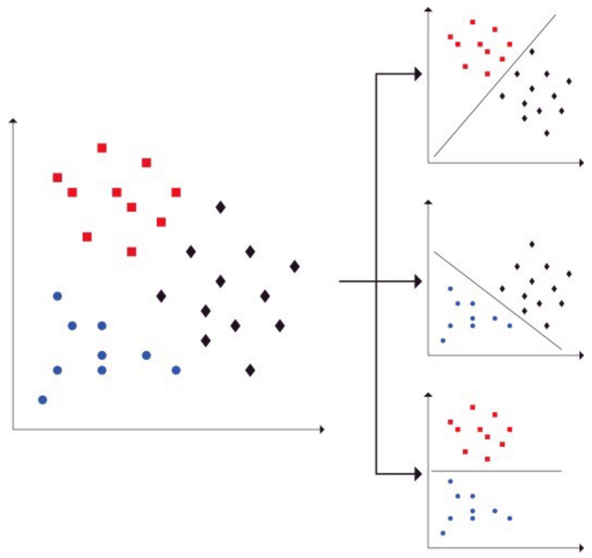
SVM 다계층 분류

Multi-Classification

SVM: Multiple Classification

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





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