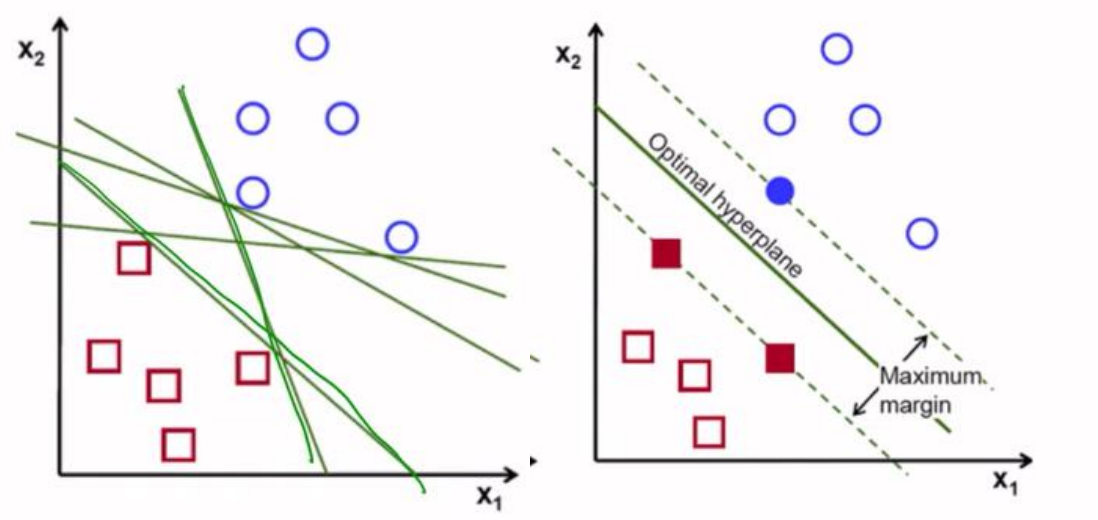
Linear SVM

Hard Margin Support Vector Machine (SVM)

- 회귀와 분류 모두 적용 가능한 지도 학습 모델(Supervised Model). 주로 분류에 사용하고 회귀에 사용할 땐(SVR) 마진에 최대한 많은 데이터를 포함시키는 방향으로 진행한다.
- Kernel Trick을 사용하여 비선형 분류 문제에도 사용 가능하다.
- 학습 방향 : 마진(Margin) 최대화



- 결정 경계(Decision boundary)는 데이터 공간의 차원보다 한 단계 낮은 차원의

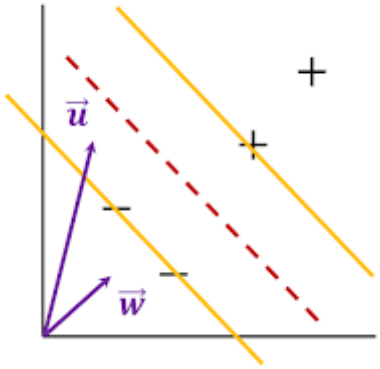
  Hyperplane. 주변 데이터와의 거리가 최대가 되도록(**margin의 최대화**) 해야 강인한 분류를 할 수 있다.

- 서포트 벡터(Support Vector) : 결정 경계선에 가장 가까이 있는 각 클래스의 데이터.

  결정 경계가 먼저가 아니라, 각 클래스의 데이터 중에 가까운 Support vector 들을 먼저 찾고 그 거리의 절반에 위치하는 결정 경계를 구한다.

마진(Margin)의 최대화 : SVM의 목적함수

Decision boundary의 위에 있는 + label의 값들과 아래의 – label을 갖는 값들은 어떤 Decision Rule을 가져야 할까?



어떤 한 샘플 $\vec{u}$ 가 있을 때 이 값이 +인지 -인지 구별하는 방법은 이렇게 표현할 수 있다.

$\vec{u}$를 Decision Boundary에 수직하는 벡터 $\vec{w}$(이 벡터의 크기는 모른다) 에 내적 한 뒤 ($\vec{u}$가 $\vec{w}$ 방향으로 얼마나 큰지) 이 값이 어떤 값 c 보다 크면(길을 건너면.) +일 것이고 못 건너면 -인 것이다. 이 c를 넘기면 아래와 같다.

$$\vec{w} \cdot \vec{u} \geq c$$

$$\vec{w} \cdot \vec{u} + b \geq 0 \quad \text{then '+'}$$

아직은 $\vec{w}$ 값의 크기도, b의 크기도 모른다. 그래서 이제 수학적 편리성을 위해 여러 조건들을 붙여간다. 먼저 Support vector의 조건을 붙여서, +와 – 값들이 최소한 아래와 같은 조건을 만족하도록 바꾼다.

(이 때 Support vector를 지나는 Hyperplane 들은 $\vec{w} \cdot \vec{x} + b = 1$, $\vec{w} \cdot \vec{x} + b = -1$)

$$\vec{w} \cdot \vec{x}_+ + b \geq 1$$

$$\vec{w} \cdot \vec{x}_- + b \leq -1$$

이제 계속해서 수학적 편리성을 위해 조작을 해보자. 위의 두 식을 하나의 식으로 합치기 위해서, x_i의 label에 따라 값을 곱해준다. 즉 아래와 같은 y_i를 곱해준다.

$$y_i = \begin{cases} 1 & for\ '+' \\ -1 & for\ '-' \end{cases}$$

그럼 아래와 같이 합쳐진다.

$$y_i(\vec{w} \cdot \vec{x_i} + b) \geq 1\ , i = 1,2,\ldots,n$$

이 때 Support Vector의 경우는 아래와 같다. 경계에 있으니까.

$$y_i(\vec{w} \cdot \vec{x_i} + b) - 1 = 0$$

그럼 이걸 통해 Margin의 폭 Width를 표현할 수 있다. + support 벡터와 – support 벡터의 차이에 w⃗의 단위벡터를 곱하면 되겠지.

$$\text{Width} = (\overrightarrow{x_+} - \overrightarrow{x_-}) \cdot \frac{\vec{w}}{\|\vec{w}\|}$$

이 때 서포트 벡터 두 녀석은 $y_i(\vec{w} \cdot \overrightarrow{x_i} + b) - 1 = 0$ 이 식을 이용하면 $\frac{2}{\vec{w}}$ 가 된다.

$$\therefore \text{Width} = \frac{2}{\|\vec{w}\|}$$

우리의 목적은 이 Margin의 Width를 Maximize하는 것이고 미분하기 편하게 역수를 취하고 제곱해서 Minimize 문제로 바꾼다. 이를 최소화하는 w와 b를 찾자.

$$\max \frac{1}{||\vec{w}||} \leftrightarrow \min ||\vec{w}|| \leftrightarrow \min \frac{1}{2}||\vec{w}||^2$$

이제 우리의 모델은

$$y_i(\vec{w} \cdot \overrightarrow{x_i} + b) - 1 = 0$$

이라는 조건에서의 최소화 문제가 되었는데, 이를 Lagrange 승수법을 통해 해결한다.

$$\min_{w,b} \mathcal{L}(w, b, \alpha) = \frac{1}{2}\|\vec{w}\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(\vec{w} \cdot \vec{x}_i + b) - 1]$$

$$\alpha \text{ is Lagrange multiplier}$$

위 식을 w와 b로 각각 미분 한 값을 0이라 둔다. 왜냐면 기울기(벡터)가 같은 접선을 구해야 하기 때문이다 (극 값). 그럼 아래와 같은 식을 얻을 수 있다.

$$\frac{\partial \mathcal{L}}{\partial w} = \vec{w} - \sum_i \alpha_i y_i \overrightarrow{x_i} = 0$$

$$\vec{w} = \sum_i \alpha_i y_i \overrightarrow{x_i}$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_i \alpha_i y_i = 0$$

$\mathcal{L}(w, b, \alpha)$ 에 대입해서 최솟값을 찾아내면 된다. 위에서 구한 I와 j에 대한 $\vec{w}$와 b를 대입해보자.

$$\mathcal{L}(w,b,\alpha) = \frac{1}{2}\left(\sum_i \alpha_i y_i \vec{x_i}\right)\cdot\left(\sum_j \alpha_j y_j \vec{x_j}\right) - \left(\sum_i \alpha\, ; y_i \vec{x}_i\right)\cdot\left(\sum_j \alpha_j y_j \vec{x}_j\right) - b\sum_i \alpha_i y_i\,(=0) + \sum_i \alpha_i$$
$$= \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

이제 α만 구하면 되는데, 이제 이 문제는 α의 max를 찾는 Dual problem이 된다.

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

$$subject\ to \sum_i \alpha_i y_i = 0, \qquad \alpha_i \geq 0, \qquad i = 1,2,\ldots,n$$

(w,b,α)가 Lagrangian Dual Problem의 최적해가 되기 위한 조건이 있다고 한다. 이를 KKT condition이라고 하는데 Primal과 Dual problem의 관계에서 나온 조건으로써 아래와 같다.

1. Stationarity : 미분해서 0이 되는 극점이 존재해야 한다.

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \;->\; \vec{w} = \sum_i \alpha_i y_i \vec{x_i}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \;->\; \sum_i \alpha_i y_i = 0$$

2. Primal feasibility : Primal의 조건

$$y_i(\vec{w}\cdot\vec{x_i} + b) \geq 1\,, i = 1,2,\ldots,n$$

3. Dual feasibility : Dual의 조건

$$\alpha_i \geq 0, \qquad i = 1,2,\ldots,n$$

4. Complementary slackness : 어떤 조건식에 slack(부등식의 차)이 있다면 그에 대응하는 dual variable이 0이라는 말이다. 이게 중요!

$$\alpha_i[y_i(\vec{w}\cdot\vec{x_i} + b) - 1] = 0$$

- $\alpha_i > 0$ 이면, $y_i(\vec{w}\cdot\vec{x_i} + b) - 1 = 0$ 이라는 뜻이고 이건 $\vec{x_i}$ **가 Support Vector인 경우이다.**
- $\alpha_i = 0$ 이면, $y_i(\vec{w}\cdot\vec{x_i} + b) - 1 \neq 0$ 이라는 뜻이고 이건 $\vec{x_i}$ **가 Support**

Vector 바깥에 있는 경우이다.

그러면 즉, $\overrightarrow{x_i}$ 이 Support Vector인 경우에만 $\alpha_i > 0$ 이므로 w를 구할 때, Support vector만 가지고 Optimal hyperplane (Decision Boundary)를 구할 수 있다. Outlier에 Robust 하다!

$$\vec{w} = \sum_i \alpha_i y_i \overrightarrow{x_i} = \sum_{i \in SV} \alpha_i y_i \overrightarrow{x_i}$$

$$\vec{w} \cdot \overrightarrow{x_{SV}} + b = y_{SV}$$

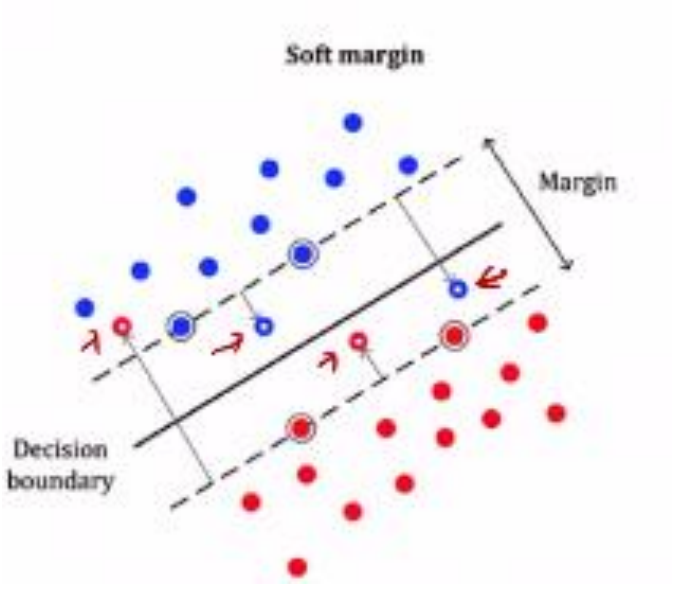
$$b = y_{SV} - \vec{w} \cdot \overrightarrow{x_{SV}} = y_{SV} - \sum_{i \in SV} \alpha_i y_i \overrightarrow{x_i} \cdot \overrightarrow{x_{SV}}$$

W와 b를 구했다. = Model을 구했다..

$$\vec{w} \cdot \overrightarrow{x_{new}} + b > 0 \text{ , } y_{new} = 1(+)$$

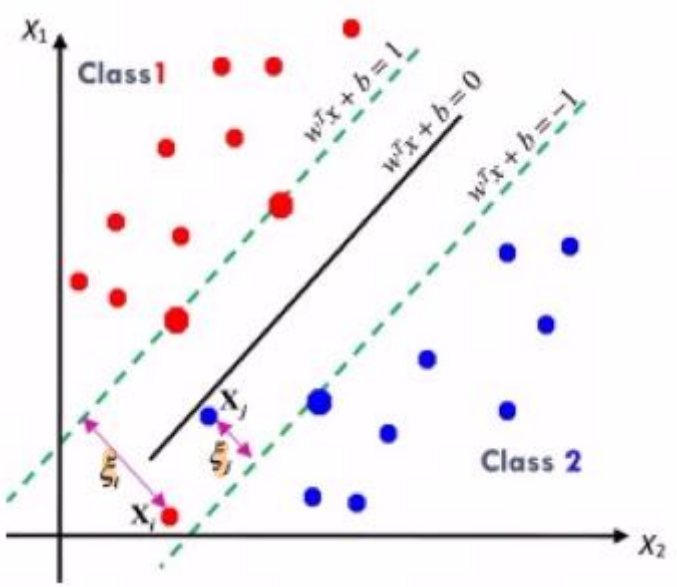
$$\vec{w} \cdot \overrightarrow{x_{new}} + b < 0 \text{ , } y_{new} = -1(-)$$

근데, 위의 방법으로 나눌 수 없는 데이터들은...? 즉 선형 분리가 안되면???



흠.... 그럼 이런 Error를 허용해주자.. 말랑말랑한 Margin~

Soft Margin SVM



ξ_i (크사이)는 error과 해당 label의 Margin plane의 거리이다. 그럼 이를 허용해 주면서 옛날의 그 $\min \frac{1}{2}||\vec{w}||^2$ 최소화 문제를 푸는 거다. $\frac{1}{2}||\vec{w}||^2$ 에 $C\sum_{i=1}^{n}\xi_i$ 을 더해준다.

Original Problem

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

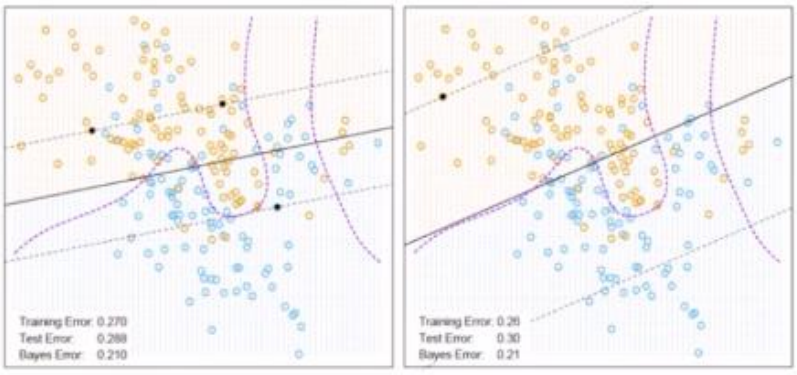
$$subject\ to\ y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1,2,\dots,n$$

C 라는 parameter는 무엇을 말하는가??

최소화 문제인데, 여기서 C를 크게 설정한다는 것은 그만큼 Error를 허용하지 않겠다 라는 것이고 좁은 margin을 형성하게 된다. 따라서 상대적으로 Overfit을 하게 되고

반대로 C를 작게 설정하면 그만큼 error를 허용함으로써 상대적으로 Underfit하게 된다.

C=10000



C=0.01

이제 Lagrange Multiplier Method 로 최적화 문제를 푼다.

Primal Problem

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

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

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

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

Convex, continuous이기 때문에 미분 = 0 에서 최소값을 가짐

① $\frac{\partial \mathcal{L}(w, b, \xi, \alpha, \gamma)}{\partial w} = 0 \longrightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i$

② $\frac{\partial \mathcal{L}(w, b, \xi, \alpha, \gamma)}{\partial b} = 0 \longrightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$

③ $\frac{\partial \mathcal{L}(w, b, \xi, \alpha, \gamma)}{\partial \xi_i} = 0 \longrightarrow C - \alpha_i - \gamma_i = 0, i = 1,2, \dots \dots, n$

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

$$= \frac{1}{2}\|w\|_2^2 - \sum_{i=1}^{n}\alpha_i\left(y_i\left(w^T x_i + b\right) - 1\right) + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\xi_i - \sum_{i=1}^{n}\gamma_i\xi_i$$

$$= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}(\alpha_i - \gamma_i)\xi_i$$

$$= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j + C\sum_{i=1}^{n}\xi_i - C\sum_{i=1}^{n}\xi_i$$

$$= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j$$

RECALL

$$\frac{\partial\mathcal{L}(w,b,\xi,\alpha,\gamma)}{\partial\xi_i} = 0 \longrightarrow C - \alpha_i - \gamma_i = 0, i = 1,2,\dots\dots,n$$

Hard Margin SVM

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

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

$$\mathbf{0} \le \boldsymbol{\alpha_i}, i = 1,2,\dots,n$$

Soft Margin SVM

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

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

$$\mathbf{0} \le \boldsymbol{\alpha_i} \le \boldsymbol{C}, i = 1,2,\dots,n$$

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

(w,b,ξ,α,γ)가 Lagrangian dual problem의 해가 되기 위한 조건 (KKT condition)

① $$\frac{\partial \mathcal{L}(w,b,\alpha)}{\partial w} = 0 \longrightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

② $$\frac{\partial \mathcal{L}(w,b,\alpha)}{\partial b} = 0 \longrightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$$

③ $$\frac{\partial \mathcal{L}(w,b,\xi,\alpha,\gamma)}{\partial \xi_i} = 0 \longrightarrow C - \alpha_i - \gamma_i = 0, i = 1,2,\dots\dots,n$$

Complementary slackness

④ $$\alpha_i\left(y_i\left(w^T x_i + b\right) - 1 + \xi_i\right) = 0, \quad \gamma_i \xi_i = 0, i = 1,2,\dots\dots,n$$

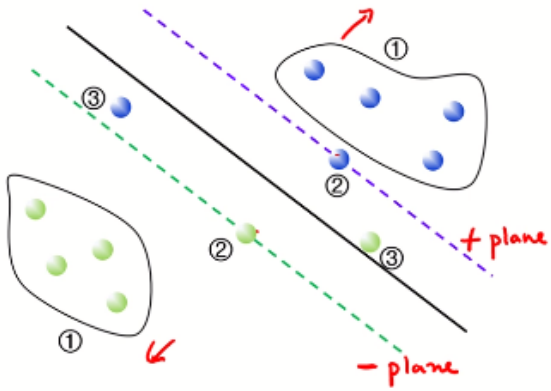
KKT condition으로부터 다음과 같은 정보를 얻을 수 있음:

$$\alpha_i\left(y_i\left(w^T x_i + b\right) - 1 + \xi_i\right) = 0,$$
$$\alpha_i = C - \gamma_i, \quad \gamma_i \xi_i = 0, i = 1,2,\dots\dots,n$$

① $\alpha_i = 0 \Rightarrow \gamma_i = C$
$\Rightarrow \xi_i = 0$
$\Rightarrow \left(y_i\left(w^T + b\right) - 1\right) \neq 0$
$\Rightarrow x_i$가 plus-plane 또는 minus-plane 위에 있지 않음

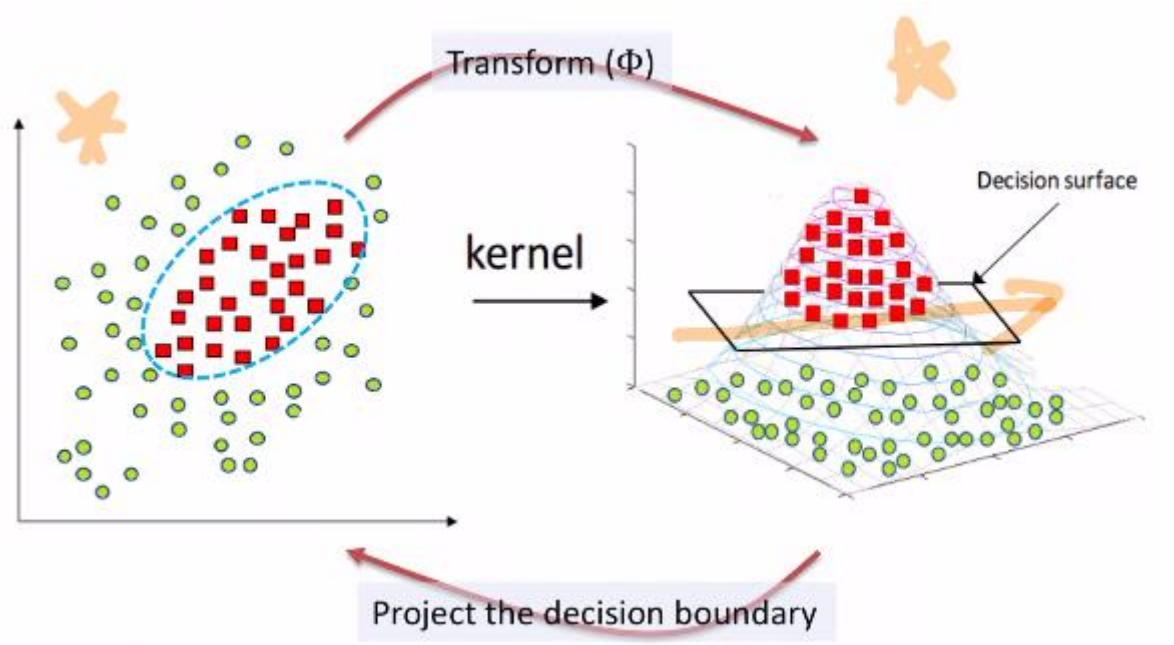
② $0 < \alpha_i < C \Rightarrow \gamma_i > 0$
$\Rightarrow \xi_i = 0, \gamma_i \xi_i = 0$
$\Rightarrow \left(y_i\left(w^T + b\right) - 1\right) = 0$
$\Rightarrow x_i$ 가 plus-plane 또는 minus-plane 위에 있음
(**support vector**)

③ $\alpha_i = C \Rightarrow \gamma_i = 0$
$\Rightarrow \xi_i > 0$
$\Rightarrow \alpha_i\left(y_i\left(w^T + b\right) - 1\right) = -\alpha_i \xi_i \neq 0$
$\Rightarrow x_i$가 plus-plane과 minus-plane 사이에 있음
(**support vector**)



Non-linear SVM

- 데이터를 선형으로 분류하기 위해 차원을 높여서 갈라버리는 거다.
- 변환함수 Feature Map (Φ)을 통해 차원을 높인다. X 대신 Φ(x)
- Kernel : Feature map을 내적 하여 차원을 변환하는 것.
- 데이터 차원을 변경한 것이지, 분류기를 바꾼 것이 아니다. (Kernel Trick)



- SVM 모델을 Feature Space에서 학습시키면 Feature Space에서의 Linear Decision Boundary가 Original Space에서 Nonlinear Decision Boundary로 나타난다.
- 고차원 Feature space에서는 분류가 더 쉬울 수 있음을 입증함.
- 고차원 Feature space를 효율적으로 계산할 수 있는 방법이 있다. 그렇지 않다면 연산 복잡도가 너무 높아서 적용하기 쉽지 않았을 것이다.

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

[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

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

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

Dual Problem

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

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

- **Kernel Mapping 의 예**

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

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

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

우리가 필요한 건 매핑한 x들의 내적인데, 각각 커널 매핑해서 내적하기엔 연산량이 너무 크니까 바로 내적을 계산해버리자!

- **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**

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

- $(X,Y)^2 = < (x_1, x_2), (y_1, y_2) >^2$ = 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) 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 불 가능하므로 Kernel 적용
- Low degree polynomial kernel function 사용
  - K(x, y) = (xy+1)^2
- Tuning parameter C =100

- **비선형 SVM의 α 계산**

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

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

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

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

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

알파 구했으니 이제 b를 구하자

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

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

X_i	Y_i
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

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

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

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

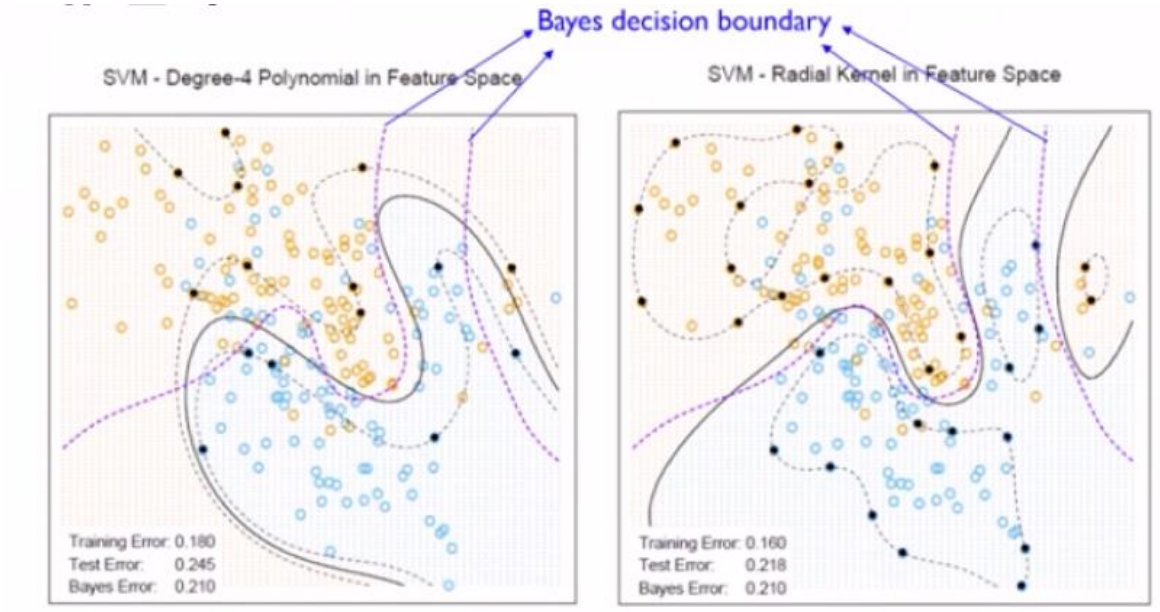
짜잔. 모델에 x를 넣었을 때, 0보다 크면 1, 작으면 -1 이다~!

비선형 SVM의 커널 선정 법

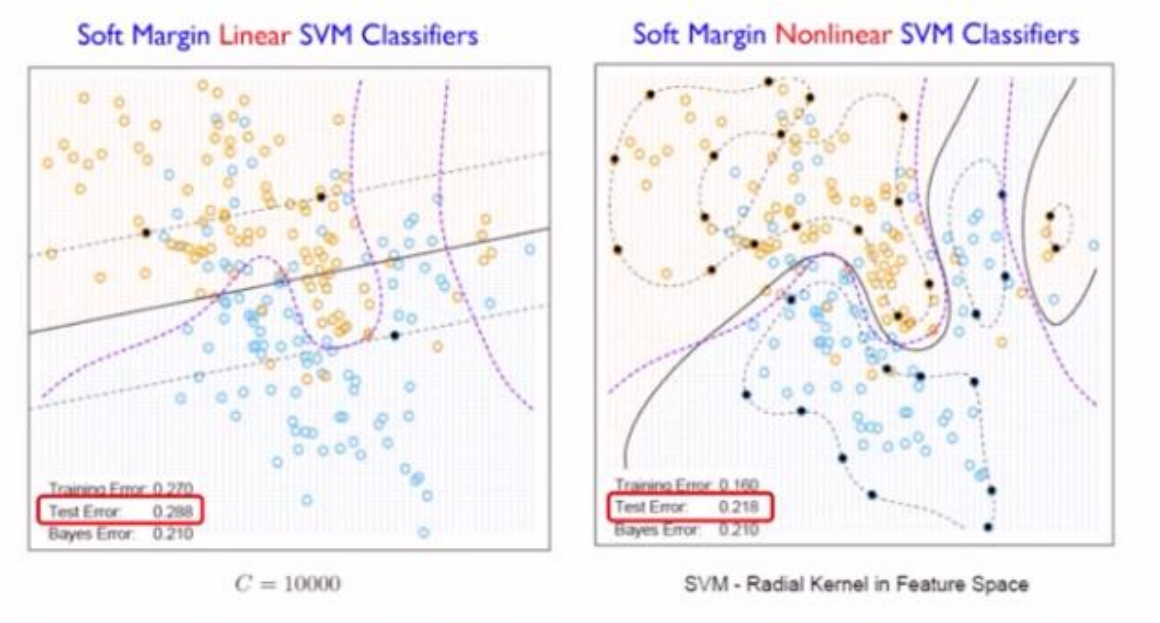
SVM Kernel을 결정하는 것은 어려운 문제, 정해진 기준이 없으므로 실험적으로 결정해야 한다.

사용하는 Kernel에 따라 Feature space의 특징이 달라지기 때문에 데이터의 특성에 맞는 Kenel을 결정해야 한다.

- 일반적으로 RBF Kernel, Sigmoid Kernel, Low Degree Polynomial Kernel (4차 미만) 등이 주로 사용된다.



검정 선이 결정경계. 위의 경우엔 우측의 RBF 커널이 더 잘 분류한다.



비 선형적인 데이터에선 당연히 비선형 SVM이 더 잘 분류하겠지만 경우에 따라 Soft Margin이

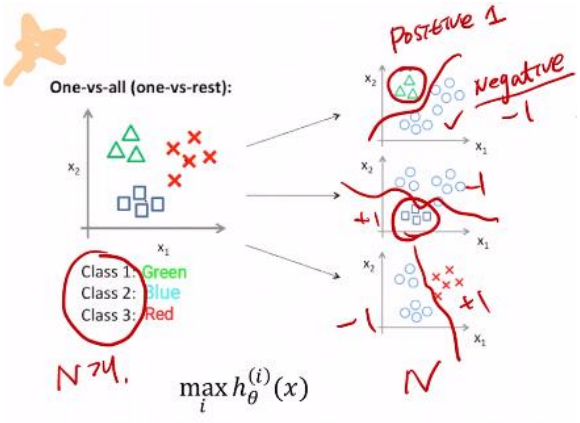
더 잘 분류하는 경우도 있겠다.

Multiple-Classification

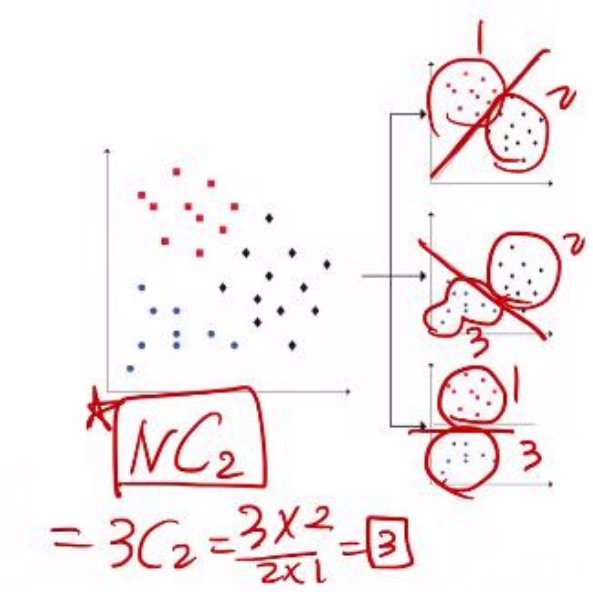
SVM은 Binary Classifier인데 이를 이용해 다계층 분류를 하는 방법은 아래와 같다.

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

- 하나-나머지 방법 : 하나의 Class와 나머지의 Class를 이항 분류하고 이를 각 Class 마다 반복. 새로운 데이터가 들어오면 N개의 분류기에 다 넣어보고 이항 분류 값 (Hypothesis function) 이 가장 큰 값을 그룹으로 할당한다.



- 하나-하나 방법 : 두 개의 Class 씩 뽑아서 ${}_nC_2$ 개의 분류기를 돌리고 새로운 데이터가 들어오면 분류기에 다 넣고 분류된 횟수가 가장 많은 그룹으로 할당한다.



SVM 정리

- 이진 분류기
- Deep Learning이 나오기 전, 널리 사용되던 기법
- 선형 SVM : Hard margin SVM / Soft margin SVM (Error 허용)
- 비선형 SVM : Kernel SVM (차원 올리고 선형 분리)
- Parameters : C (soft margin에서 에러 허용) / Kernel (Linear, Polynomial, RBF, Sigmoid..)

  OvR, OVO(Multiple)