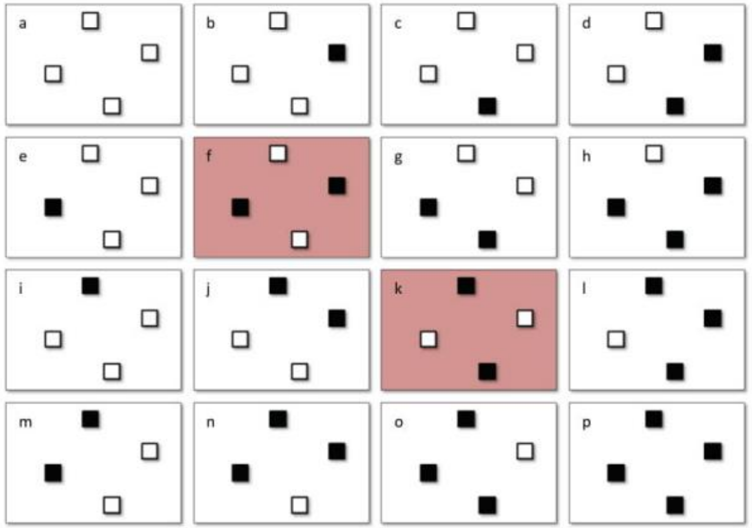
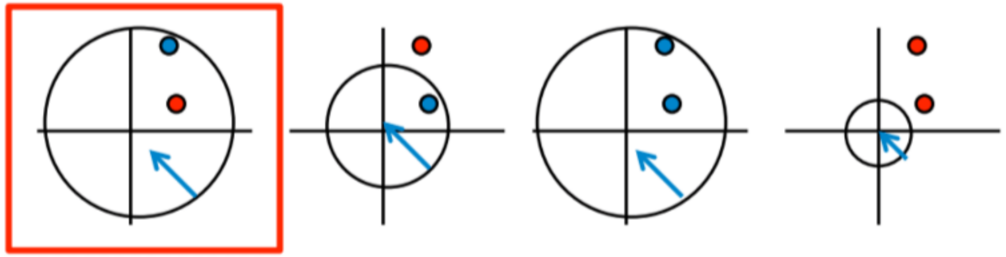
Kernel-based Learning

❖ Overview

- 질문: 선형 분류기는 d 차원에서 (d+2)개의 점을 shatter 하는 것이 가능한가?
- 원형 분류기 (원 안에 있으면 파란색, 원 밖에 있으면 빨간색으로 분류)는 2차원에서 2개의 점을 shatter 할 수 있는가?

• can $\mathcal{H}_1$ shatter four points in $\mathbb{R}^2$?

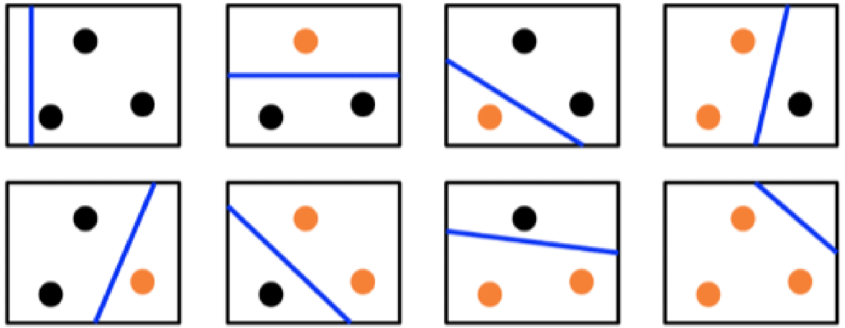


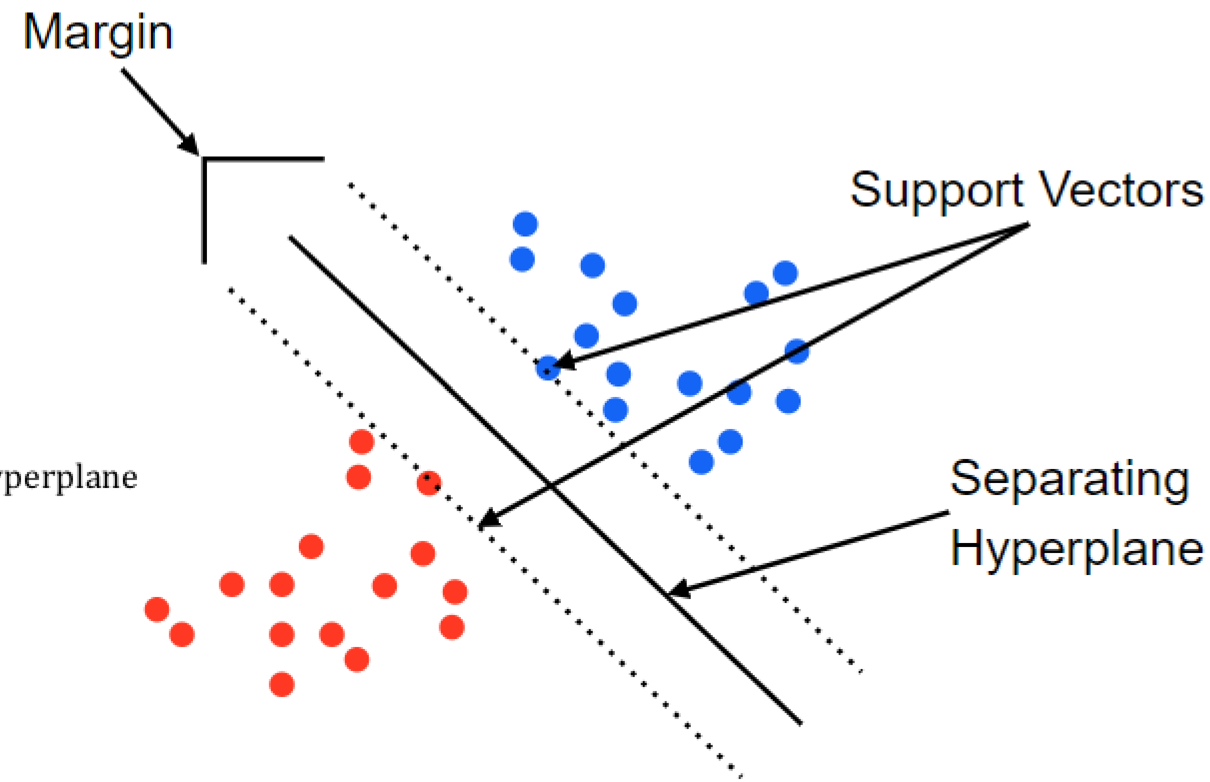


Kernel-based Learning

❖ Overview

- Support Vector Classifier





$$w^T x + b = 0$$

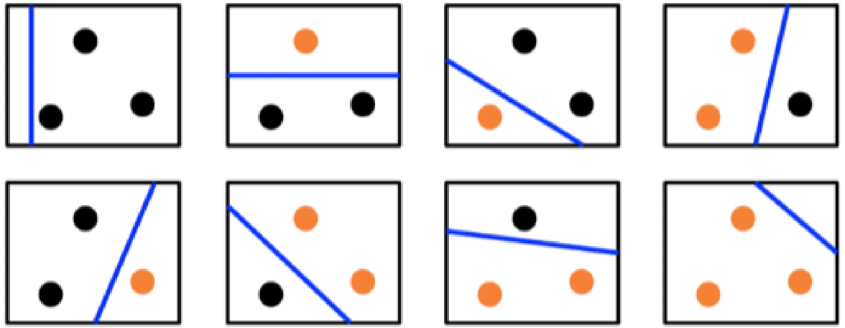
w : normal vector of the hyperplane
b : bias

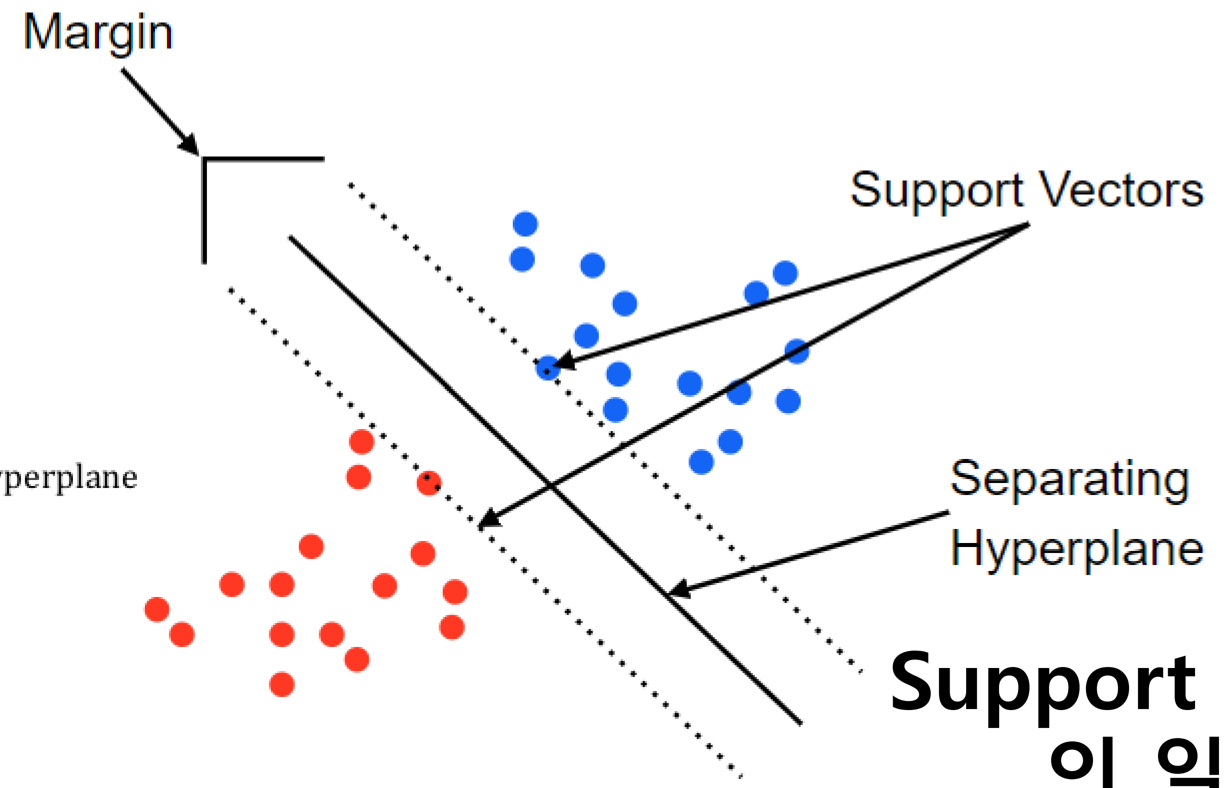
Training data (X,Y)를 가지고 w와 b를 찾자!

Kernel-based Learning

❖ Overview

- Support Vector Classifier





$$w^T x + b = 0$$

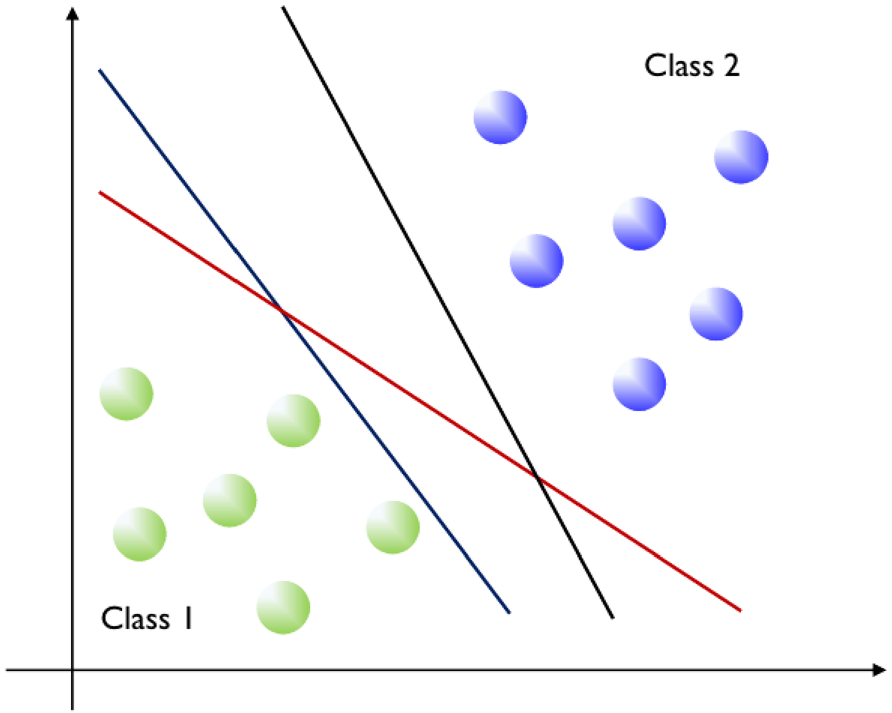
w : normal vector of the hyperplane
b : bias

Support Vector Classifier 의 일반화 = SVM

Training data (X,Y)를 가지고 w와 b를 찾자!

Kernel-based Learning

❖ Support Vector Machine



Two class classification 문제

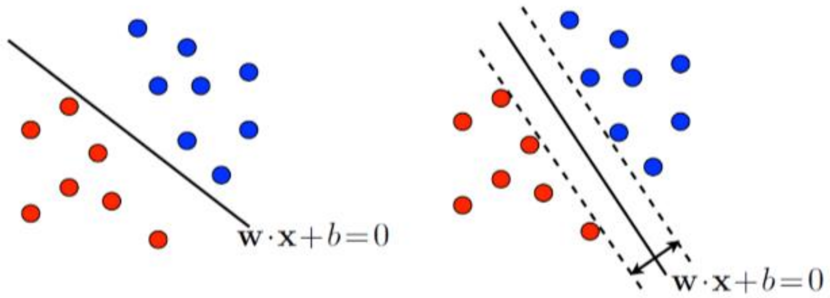
두 class를 나누는 hyperplane은 무한히 많음

어떤 hyperplane이 가장 "좋은" hyperplane 인가?

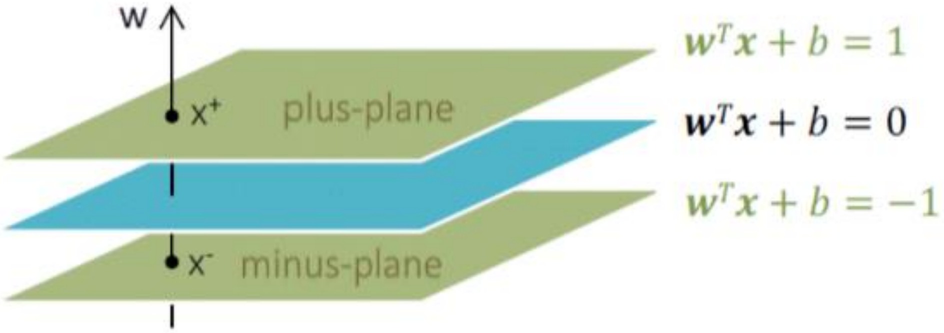
"좋다"는 것의 기준은?

Kernel-based Learning

❖ Support Vector Machine



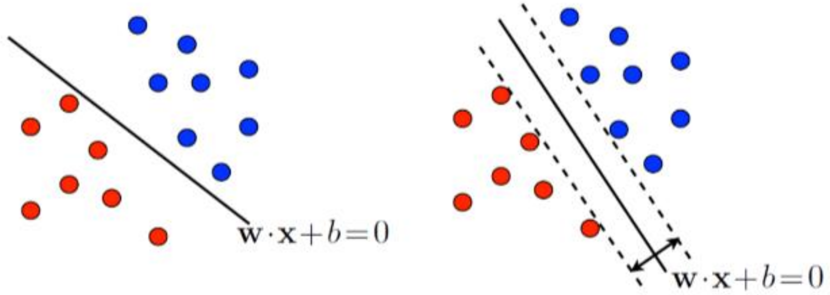
$$H = \{\mathbf{x} \rightarrow sign(\mathbf{w} \cdot \mathbf{x} + b : \ \mathbf{w} \in R^d, \ b \in R\}$$



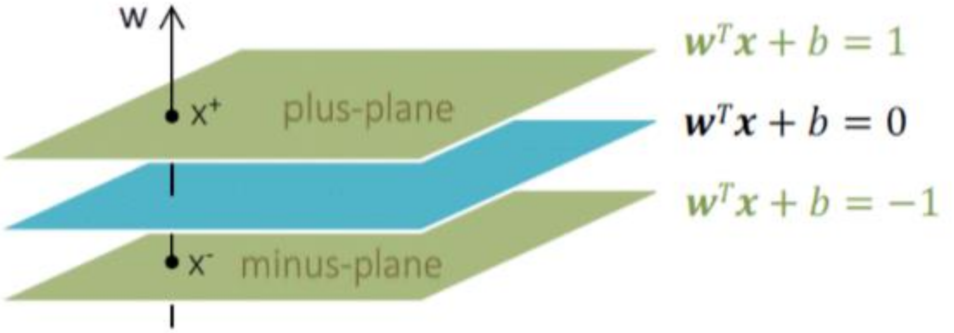
$$\text{margin} = \frac{1}{||\mathbf{w}||^2}$$

Kernel-based Learning

❖ Support Vector Machine



$$H = \{\mathbf{x} \rightarrow sign(\mathbf{w} \cdot \mathbf{x} + b : \ \mathbf{w} \in R^d, \ b \in R\}$$



$$\text{margin} = \frac{1}{||\mathbf{w}||^2}$$

Maximize the Margin!!!

Kernel-based Learning

❖ Support Vector Machine

$$\underset{w,b}{minimize}\ \frac{1}{2}\|w\|_2^2$$
$$subject\ to\ y_i(w^T x_i + b) \geq 1, i = 1, 2, \dots \dots, n$$

- Decision variable은 *w* 와 *b*
- Objective function은 separating hyperplane으로 부터 정의된margin의 역수
- Constraint는 training data를 <u>완벽하게</u> separating하는 조건
- Objective function is quadratic and constraint is linear → quadratic programming → convex optimization → globally optimal solution exists (전역최적해 존재)
- Training data가 linearly separable한 경우에만 해가 존재함

Kernel-based Learning

❖ Support Vector Machine

Original Problem

$$minimize\ \frac{1}{2}\|w\|_2^2$$

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

Lagrangian multiplier를 이용하여 Lagrangian primal문제로 변환

Lagrangian Primal

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

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

***KKT**

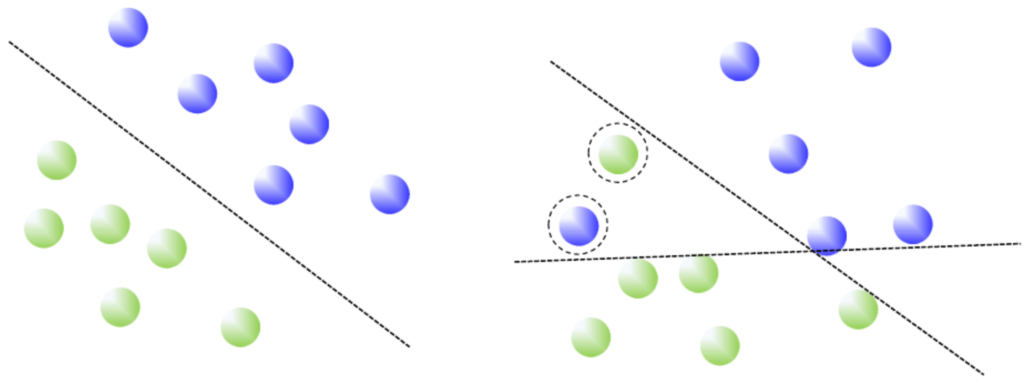
KKT condition은 strong duality를 만족하기 위한 필요조건.
Strong duality란 우리가 풀고 자하는

Primal problem과 dual problem의 optimal solution 값이 같은 것!!

(duality gap=0)
1. Stationarity
2. Primal Constraints
3. Dual Constraints
4. Complementary Slackness

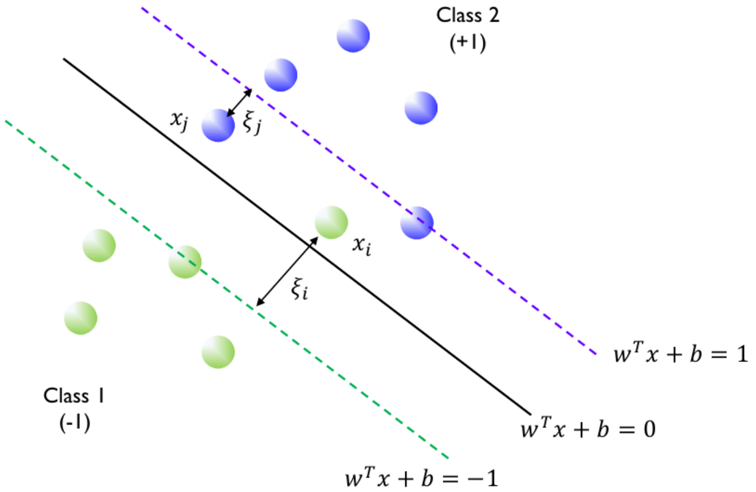
Kernel-based Learning

❖ Support Vector Machine



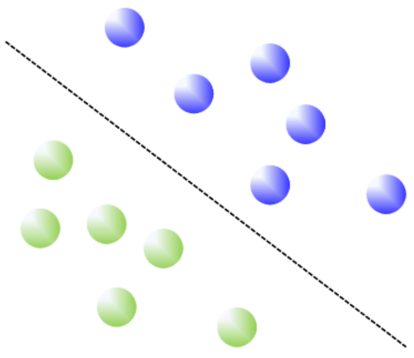
Linearly separable   Linearly nonseparable

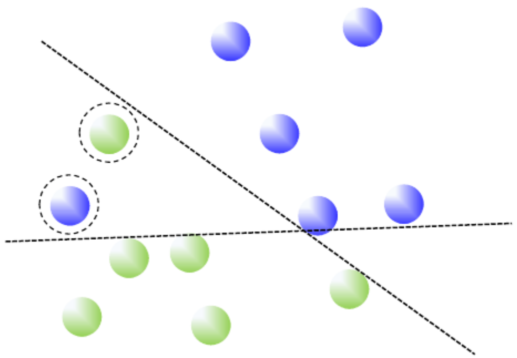
Linear decision boundary를 이용하여 완벽하게 나누는 것은 불가능
→ Error 허용

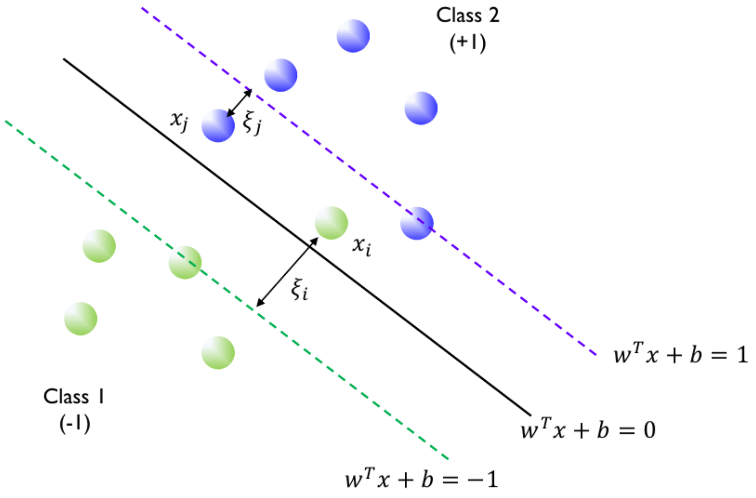


Kernel-based Learning

❖ Support Vector Machine







Linear decision boundary를 이용하여 완벽하게 나누는 것은 불가능
→ Error 허용

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

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

Kernel-based Learning

❖ Support Vector Machine

Dual Problem of Linearly Separable SVM (Hard Margin)

① $$L(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{m=1}^{N} \sum_{n=1}^{N} \alpha_m \alpha_n y_m y_n \mathbf{k}(\mathbf{x}_m, \mathbf{x}_n)$$

$$\text{subject to} \quad \alpha_n \geq 0, \quad \sum_{n=1}^{N} \alpha_n y_n = 0$$

Equation 5. Dual problem for SVM

① C가 ∞가 되면 ②는 ①번과 동일해짐.

↳ $0 \leq a_n \leq \infty$ = $0 \leq a_n$

⇒ C가 ∞인 soft margin은 hard margin과 같다.

Dual Problem of Linearly Non-Separable SVM (Soft Margin)

② $$L(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{m=1}^{N} \sum_{n=1}^{N} \alpha_m \alpha_n y_m y_n \mathbf{k}(\mathbf{x}_m, \mathbf{x}_n)$$

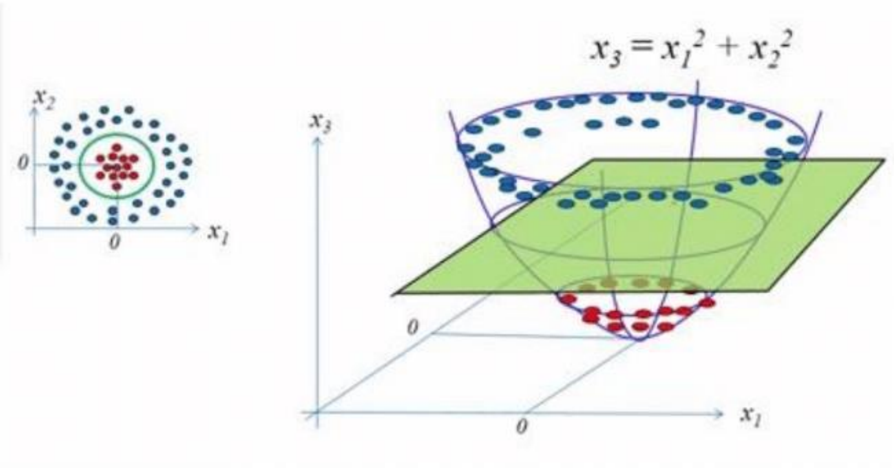
$$\text{subject to} \quad 0 \leq \alpha_n \leq C, \quad \sum_{n=1}^{N} \alpha_n y_n = 0$$

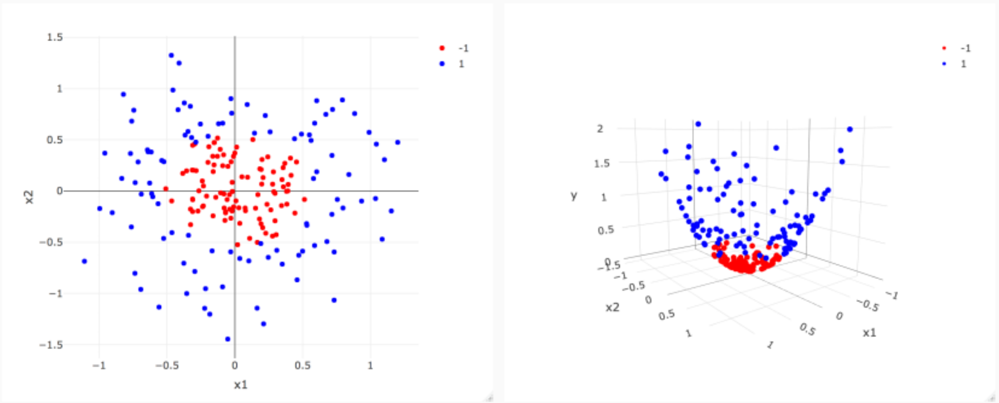
Equation 7. Dual problem for the soft-margin SVM

Kernel-based Learning

❖ Support Vector Machine

- SVM의 아이디어
  - ✓ 원래 공간이 아닌 선형 분류가 가능한 더 고차원의 공간으로 데이터를 보내서(mapping) 모델을 학습하자!





$(X_1, X_2) \in \mathbb{R}^2 \longrightarrow (X_1, X_2, X_1^2 + X_2^2) \in \mathbb{R}^3$

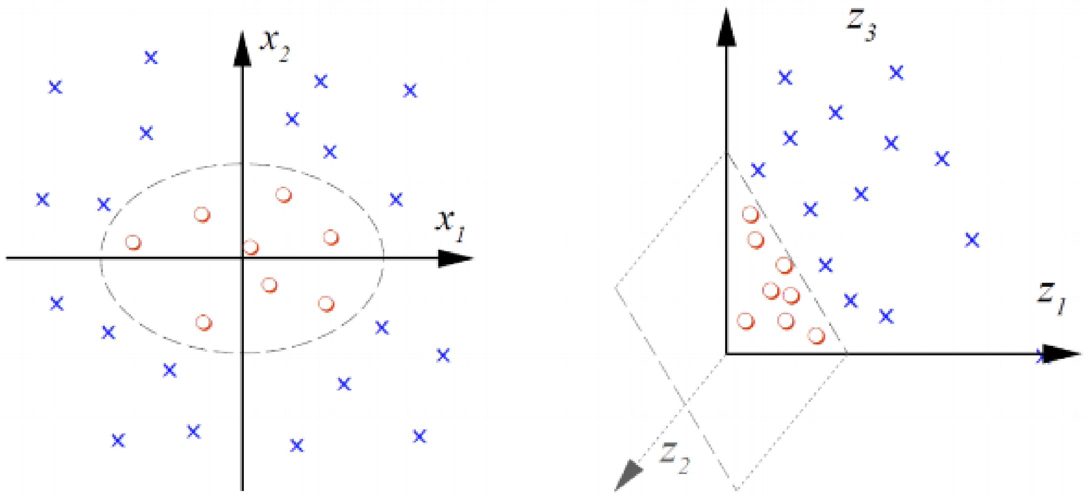
no separating hyperplane　　　separating hyperplane

Kernel-based Learning

❖ Support Vector Machine

- 목적
  - ✓ **마진을 최대화**: 일반화 성능 확보
  - ✓ **유연성 확보**: 고차원의 매핑을 통해 비선형 분류 경계면 생성

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$
$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$$

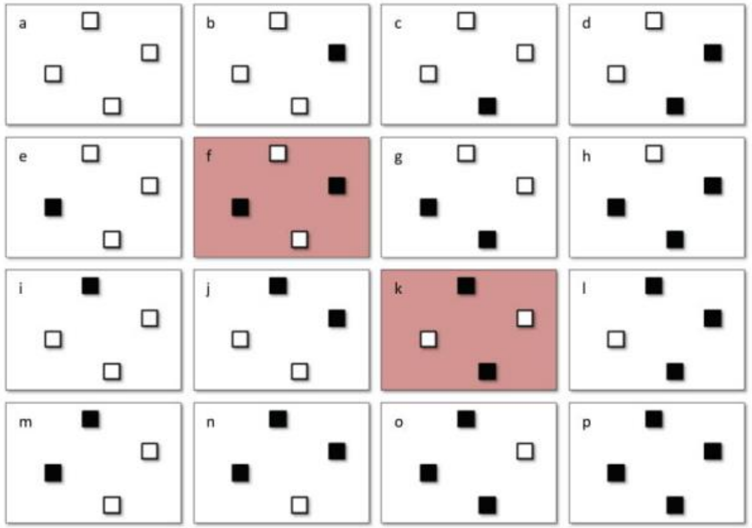


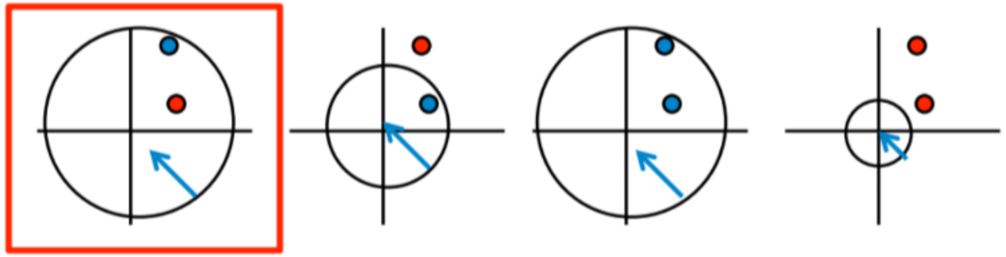
Kernel-based Learning

❖ Overview

- 질문: 선형 분류기는 d 차원에서 (d+2)개의 점을 shatter 하는 것이 가능한가?
- 원형 분류기 (원 안에 있으면 파란색, 원 밖에 있으면 빨간색으로 분류)는 2차원에서 2개의 점을 shatter 할 수 있는가?

- can $\mathcal{H}_1$ shatter four points in $\mathbb{R}^2$?

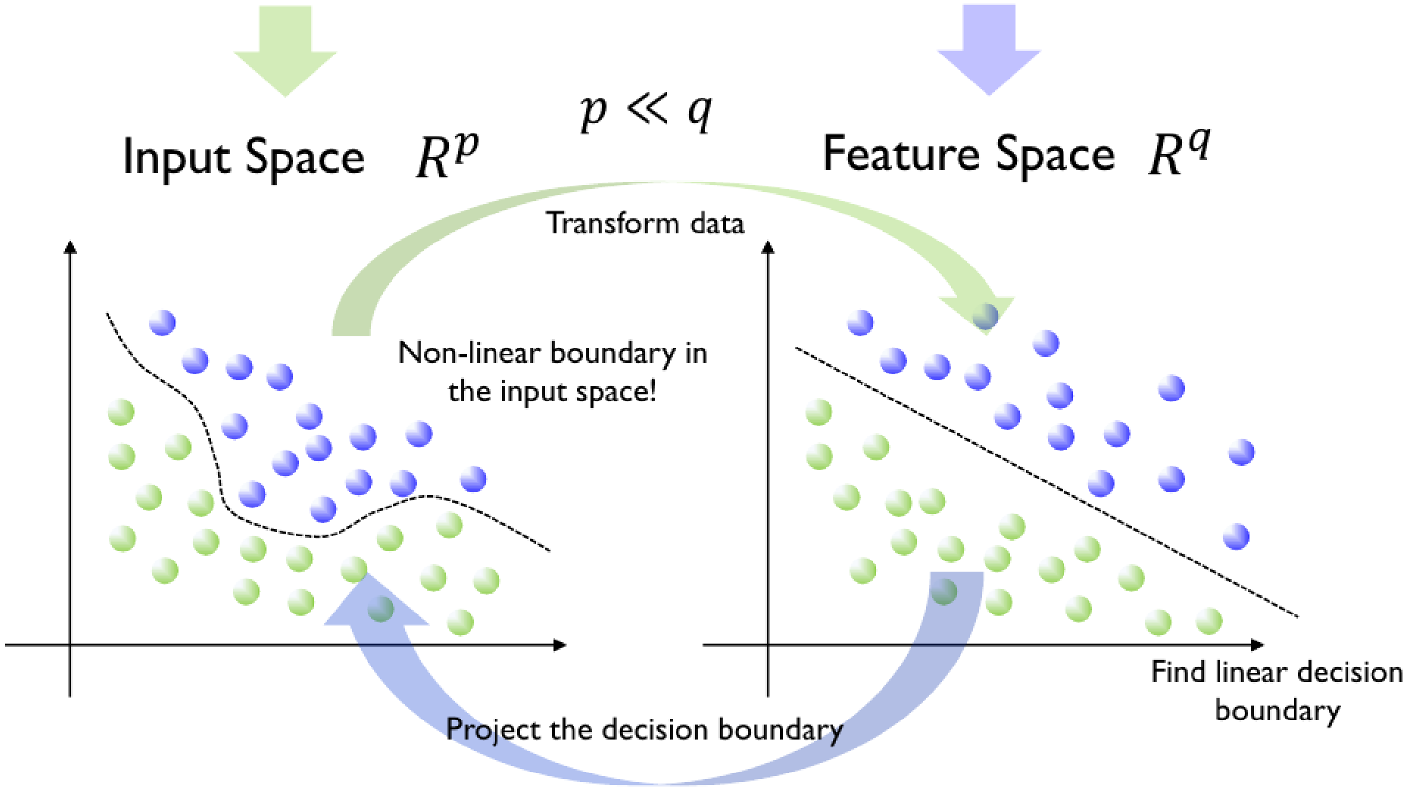




Kernel-based Learning

❖ Support Vector Machine

$$x = (x_1, x_2, \dots \dots, x_n) \rightarrow \phi(x) = z = (z_1, z_2, \dots \dots z_n)$$



$$\phi : x \longmapsto z = \phi(x)$$

Example

$$\phi : (x_1, x_2) \longmapsto (x_1, x_2, x_1^2, x_2^2, x_1 x_2)$$

2D (Original Space) → 5D (Feature Space)

Kernel-based Learning

❖ Support Vector Machine

SVM Lagrangian dual formulation

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

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

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

$x_i \rightarrow \phi(x_i)$

$$\underset{\alpha}{maximize} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j)$$

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

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

$$\underset{\alpha}{maximize} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j \phi(x_i)^T \phi(x_j)$$

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

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

Inner product of $\phi(x_i)^T \phi(x_j)$
$< \phi(x_i), \phi(x_j) >$

$x_i \rightarrow \phi(x_i)$

$$\underset{\alpha}{maximize} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

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

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

ϕ를 이용해서 직접 데이터를 변환할 필요 없이 inner product에 해당하는 $< \phi(x_i), \phi(x_j) >$ 만 정의해도 같은 효과를 얻을 수 있음

Kernel-based Learning

❖ Support Vector Machine

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

3차원으로 변환시키는 함수 (현실은 더 복잡)

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

$$\langle \emptyset(X), \emptyset(Y) \rangle = x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2$$

Question: Can we compute $x_1^2y_1^2+x_2^2y_2^2 + 2x_1x_2y_1y_2$ without knowing the explicit functional form of $\emptyset(X)$ and $\emptyset(Y)$?

$$(X, Y)^2 = \langle (x_1, x_2), (y_1, y_2) \rangle^2$$
$$= \langle x_1y_1 + x_2y_2 \rangle^2$$
$$= x_1^2y_1^2+x_2^2y_2^2 + 2x_1x_2y_1y_2$$
$$= \langle \emptyset(X), \emptyset(Y) \rangle$$

→ This can be obtained without knowing the explicit functional form of $\emptyset(X)$ and $\emptyset(Y)$

without knowing the explicit = implicit

$(X, Y)^2 = \langle (x_1, x_2), (y_1, y_2) \rangle^2 = \mathrm{K}(X, Y)$ (Kernel function)

Kernel-based Learning

❖ Support Vector Machine

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

3차원으로 변환시키는 함수 (현실은 더 복잡)

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

$$\langle\emptyset(X), \emptyset(Y)\rangle = x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2$$

Question: Can we compute $x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2$ without knowing the explicit functional form of $\emptyset(X)$ and $\emptyset(Y)$?

Yes!!!!

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

→ This can be obtained without knowing the explicit functional form of $\emptyset(X)$ and $\emptyset(Y)$

without knowing the explicit = implicit

$(X, Y)^2 = \langle(x_1, x_2), (y_1, y_2)\rangle^2 = \mathrm{K}(X, Y)$ (Kernel function)

Kernel-based Learning

❖ Support Vector Machine

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

3차원으로 변환시키는 함수 (현실은 더 복잡)

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

$$\langle\emptyset(X),\emptyset(Y)\rangle = x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2$$

Question: Can we compute $x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2$ without knowing the explicit functional form of $\emptyset(X)$ and $\emptyset(Y)$?

Yes!!!!

$$(X,Y)^2 = \langle(x_1,x_2),(y_1,y_2)\rangle^2$$
$$= \langle x_1y_1 + x_2y_2\rangle^2$$
$$= x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2$$
$$= \langle\emptyset(X),\emptyset(Y)\rangle$$

→ This can be obtained without knowing the explicit functional form of $\emptyset(X)$ and $\emptyset(Y)$

without knowing the explicit = implicit

$(X,Y)^2 = \langle(x_1,x_2),(y_1,y_2)\rangle^2 = \mathrm{K}(X,Y)$ (Kernel function)

Kernel-based Learning

❖ Support Vector Machine

$\emptyset$ 내적의 식!!!!

- Linear kernel

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

- Polynomial kernel

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

- Sigmoid kernel (Hyperbolic tangent kernel)

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

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

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