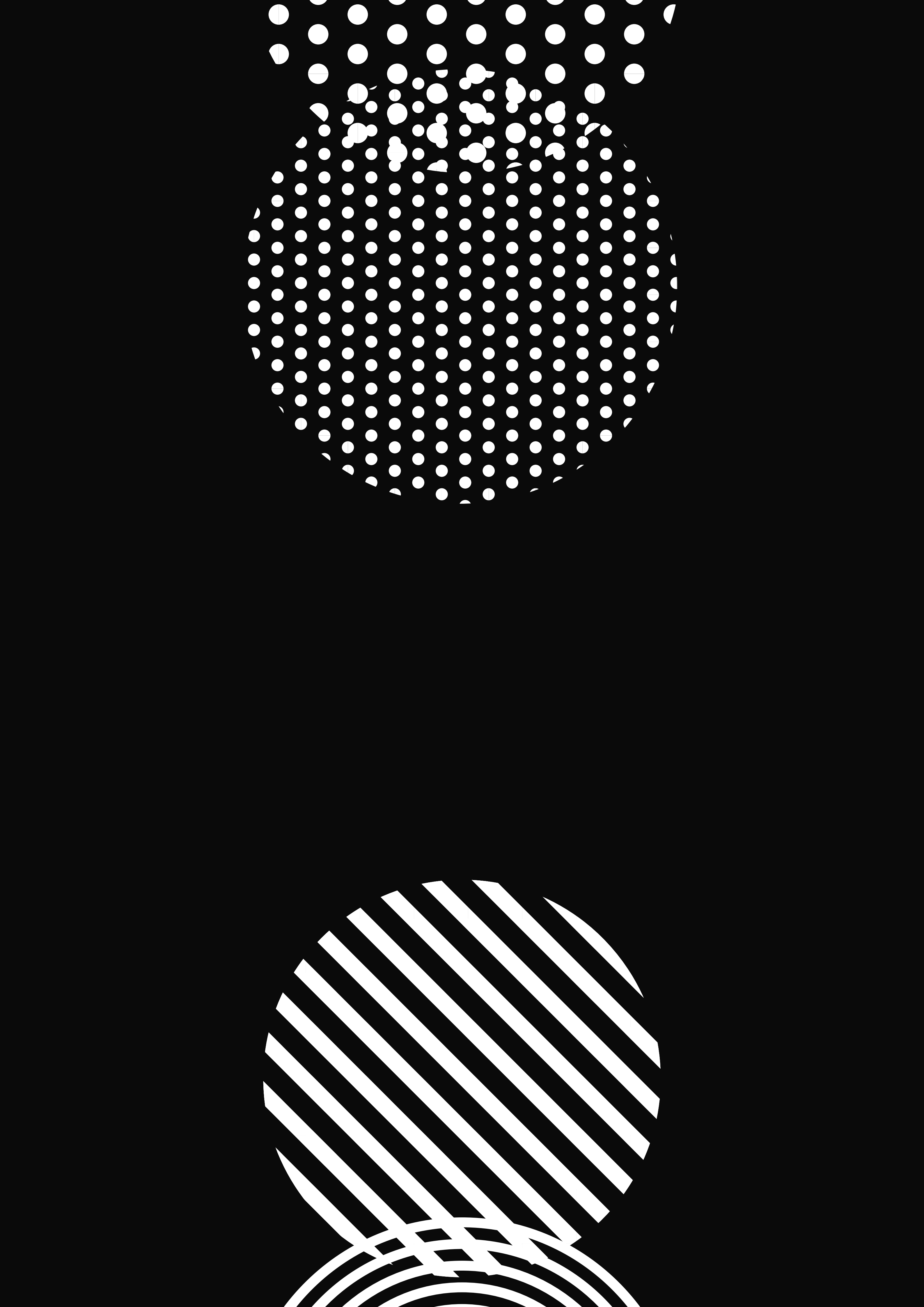
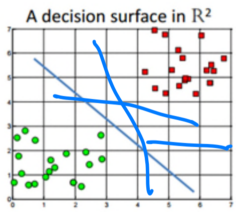
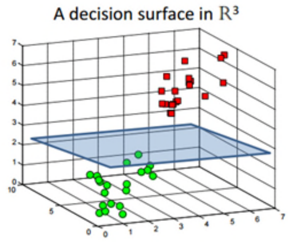


# SVM!! (Support Vector Machine 서포트 벡터 머신)

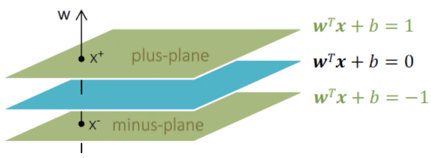
- 딥러닝 이전 가장 성과가 좋은 Classifier
- Linear, NonLinear




뭐가 최적선




Hyperplane : n차원 공간에서 n-1 Subspace (초평면)




$w=(w_1,w_2)^T$

$w^Tx+b = w_1x_1+w_2x_2+b=0$

직선의기울기 $-w_1/w_2$

법선의기울기 $w_2/w_1$

w방향 평행이동

$x^+ = x^- + \lambda w$

$w^Tx^+ + b = 1$

$w^T(x^- + \lambda w)+b = 1$

$w^Tx^- + b + \lambda w^Tw = 1$

$-1+\lambda w^Tw = 1$

$\therefore \lambda = \frac{2}{w^Tw}$

관계식과 값을 알면

Margin = distance $(x^+, x^-)$

$= \|x^+ - x^-\|_2$

$= \|(x^- + \lambda w) - x^-\|_2$

$= \|\lambda w\|_2$

$= \lambda\sqrt{w^Tw}$

$= \frac{2}{w^Tw}\sqrt{w^Tw}$ → $\frac{2}{\sqrt{w^Tw}} = \frac{2}{\|w\|_2}$

마진거리

서포트벡터에서의 거리

$= \frac{\vec{w}}{\|\vec{w}\|} \cdot (\vec{x_1} - \vec{x_2})$

$\vec{w}\cdot\vec{x} + b - 1 = 0$

$c = \frac{1}{\|\vec{w}\|}(1-b+1+b)$

$= \frac{2}{\|w\|}$

마진 최대화

---

내적 $x \cdot w^T = 0$

$= \|a\|\|b\|\cos\theta$

길이

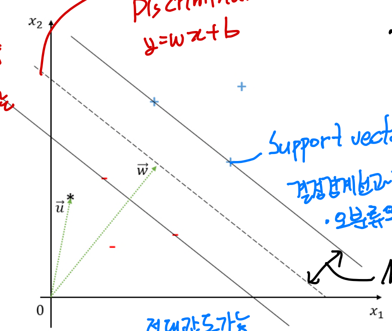
$\therefore \theta = 90°, -90°$

내적으로 $w^T$는 수직이다.

Decision Boundary

Discriminant function

$y = wx+b$




Support vector

결정경계선과 가장가까운

오분류와 가장가까워

Margin M

절대값도가능

- Maximum margin : 최고 정확도, 수렴 속도
- Decision Boundary 도출해보자

선형식 $y = wx+b$ $<0 \Rightarrow +$

$>0 \Rightarrow -$

$w x = \sum_i w_i x_i = w^Tx$

(Schaler)

Schaler

ex) $\binom{m}{n} \binom{}{m} = n\binom{}{}$

- $w^Tx+b \geq M \Rightarrow +$
- $w^Tx+b \leq -M \Rightarrow -$
- $w^Tx+b = M \Rightarrow$ Support Vector

---

## 라그랑주 승수법과 Constraints 유도

$\alpha$는 연립방정식으로!

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

목적함수 : $\frac{1}{2}\|\vec{w}\|^2$

보조방정식 $L = \frac{1}{2}\|\vec{w}\|^2 - \sum_i \alpha_i[y_i(\vec{w}\cdot\vec{x_i}+b)-1]$

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

같음 $= -\sum_i \alpha_i y_i = 0$

최초 boundary

$\vec{w} = \sum_i \alpha_i y_i \vec{x_i}$ → $\vec{x_i}$의 선형합

$\sum_i \alpha_i y_i \vec{x_i} \cdot \vec{u} + b = +1$ … for positive

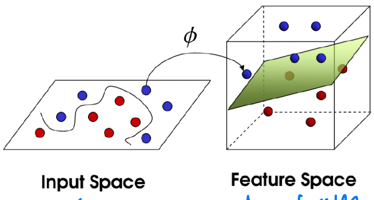
$$L=\frac{1}{2}\|\vec{w}\|^2-\sum_i \alpha_i[y_i(\vec{w}\cdot\vec{x}+b)-1]$$
$$=\frac{1}{2}(\sum_i \alpha_i y_i \vec{x_i})(\sum_j \alpha_j y_j \vec{x_j})-\sum_i(\alpha_i y_i \vec{w}\cdot\vec{x_i}+\alpha_i y_i b-\alpha_i)$$
$$=\frac{1}{2}(\sum_i \alpha_i y_i \vec{x_i})(\sum_j \alpha_j y_j \vec{x_j})-\sum_i(\alpha_i y_i \vec{x_i}(\sum_j \alpha_j y_j \vec{x_j})+\alpha_i y_i b-\alpha_i)$$
$$=\frac{1}{2}(\sum_i \alpha_i y_i \vec{x_i})(\sum_j \alpha_j y_j \vec{x_j})-(\sum_i \alpha_i y_i \vec{x_i})(\sum_j \alpha_j y_j \vec{x_j})-\sum_i \alpha_i y_i b+\sum_i \alpha_i$$
$$=\sum_i \alpha_i-\frac{1}{2}(\sum_i \alpha_i y_i \vec{x_i})(\sum_j \alpha_j y_j \vec{x_j})$$
$$=\sum_i \alpha_i-\frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j \vec{x_i}\cdot\vec{x_j}\cdots(h)$$

$$\max L_D(a_i)=\sum_{i=1}^{n} a_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j x_i^T x_j$$

$\Phi$를 적용하여 고차원공간으로

maxLD(ai)=∑i=1nai−12∑i=1n∑j=1naiajyiyj
Φ(xi)TΦ(xj)

# Kernel -SVM



svm
kernel svm

SVM은 마진최대 초평면 ⇒ 선형분류

⇒ 데이터를 선형분류가 가능한 고차원 공간으로 매핑 후 초평면 찾기

But Kernel의 연산 복잡

1. 매핑, 내적

kernel 방법

$k(x_i,x_j)=\Phi(x_i)^T\phi(x_j)$ 매핑함수 $\phi$는 선형변환 이며 그 표준행렬=A면

$\phi(x)=Ax$ $\quad k(x_i,x_j)=\phi(x_i)^T\phi(x_j)=\underset{1\times n}{x_i^TA^T}\,\underset{n\times 1}{Ax_j}=(1\times 1)$

$\phi$가 m차원 → n차원함수면

$x_i^T:(1\times m)$ $\quad A^T:(m\times n)$ $\quad A:(n\times m)$ $\quad x_j:(m\times 1)$

$\therefore k(x_i,x_i)$는 0이상, $k(x_i,x_j)=k(x_j,x_i)$

$k(x_i,x_j)$로 구성 행렬은 positive semi-definite matrix 양의준정부호행렬

대칭행렬 이어야한다

$A=A^T$

모든 고유값이 양수 $x^*Mx>0$

이 조건 만족 함수는 모두 커널함수

linear

polynomical $\quad (x_1^T x_2+b)^d$ ← 차원수

sigmoid $\quad \tanh\{a(x_1^Tx_2)+b\}$

gaussian $\quad \cancel{k}\ \exp\{-\frac{\|x_1-x_2\|^2}{2\sigma^2}\}$

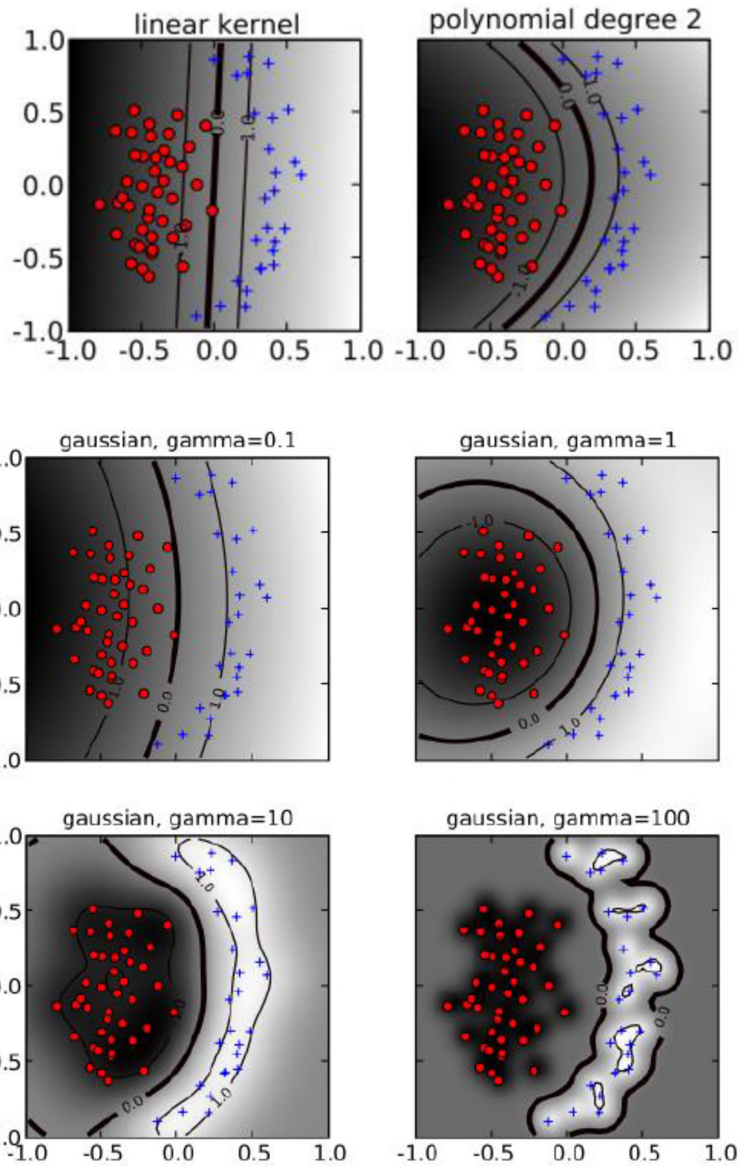
# Gaussian kernel

$2\sigma^2=1$ $\quad k(x_1, x_2) = \exp\{-(x_1-x_2)^2\}$

$= \exp(-x_1)\exp(-x_2)\exp(2x_1x_2)$

· 테일러급수 $\exp(2x_1x_2) = \sum_{k=0}^{\infty} \frac{2^k x_1^k x_2^k}{k!}$

∴ InputSpace가 몇차원이든 무한대차원의 매핑

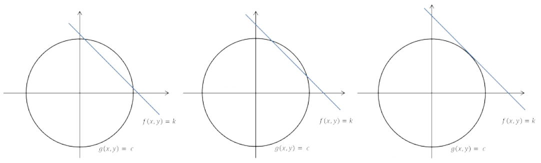


# 라그랑주 승수법

- 최적점 조건 찾는 방법, 최적해 필요조건

가장 제약조건 g를 만족하는 f의 최솟값, 최댓값는 f와 g가 접하는 지점

최대, 최소 찾는 것, 최적화
보조방정식 모든 변수에 대해
편미분값이 0




f, g 접점 (gradient vector) 구배벡터 $\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$

· 접선과 구배 벡터 내적은 0으로 수직

· f와 g가 접하는 것은 구배벡터가 서로 상수배

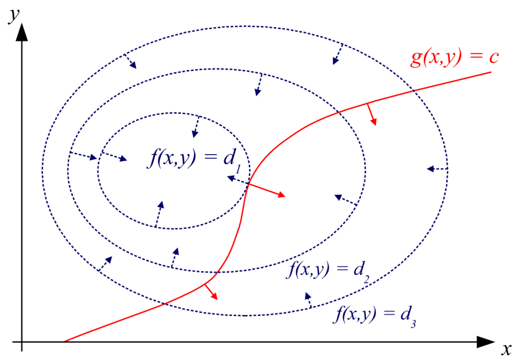
$$\nabla f = \lambda \nabla g$$
상수

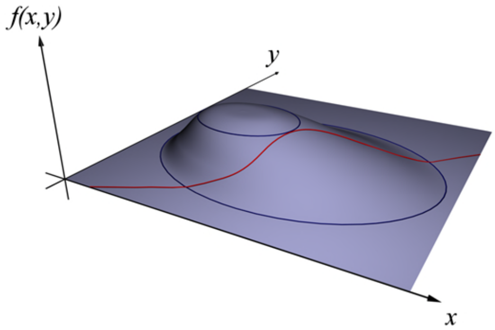
$L(x,y,\lambda) = f(x,y) - \lambda(g(x,y) - c)$

∴ L의 구배벡터가 영벡터가 되면 f, g가 접하는 점 찾을 수 있다.

g(x,y) = c 까지 이용하면 3개의 식으로 해구함

$$L(x,y,\lambda_1,\lambda_2 \cdots \lambda_n) = f(x,y) - \sum_{i=1}^{N} \lambda (g_i(x,y) - c_i)$$




·최대, 최소지점 목적함수 f, 제약조건 g gradient가 평행

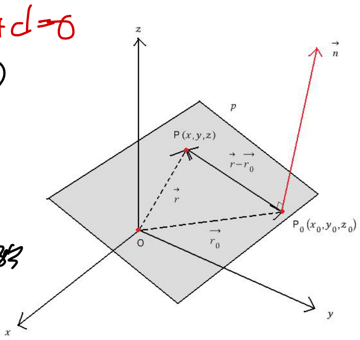
## ·법선 벡터 (Normal Vector)

직선 $ax+by+c=0$
평면 $ax+by+cz+d=0$

· 평면의 방정식 $\vec{r_0} = \vec{OP_0} = (x_0, y_0, z_0)$ $\vec{r} = \vec{OP} = (x,y,z)$

$\vec{n}$은 평면 P와 수직인 벡터

$\vec{P_0P} = \vec{OP} - \vec{OP_0} = \vec{r} - \vec{r_0}$ 는 수직 / $\vec{n}\cdot(\vec{r}-\vec{r_0}) = 0$




·평면의 벡터방정식 $(x_0, y_0, z_0)$ 지나고 법선벡터 $\vec{n}$인 평면 벡터방정식

$\vec{r_0} = (x_0, y_0, z_0)$, $\vec{r} = (x,y,z)$ 이면 $\vec{n}\cdot(\vec{r}-\vec{r_0}) = 0$

$\vec{n} = (a,b,c)$ 라면 $(a,b,c)\cdot(x-x_0, y-y_0, z-z_0) = 0$

$d = -ax_0 - by_0 - cz_0$ 이면 $ax+by+cz+d=0$

$\frac{x}{a}+\frac{y}{b}+\frac{z}{c}=1$ $(a,0,0)$ $(0,b,0)$ $(0,0,c)$ 지남

$2(x-2)+3(y-4)+4(z+1)=0$

$2x+3y+4z=12$

$w^T x^+ + b = 1$

$w^T(x^- + \lambda w) + b = 1 \quad (x^+ = x^- + \lambda w$

$w^T x^- + b + \lambda w^T w = 1 \quad \therefore w$는 찾고자하는 hyperplane 기울기

$\lambda w^T w = 2$

$\lambda = \frac{2}{w^T w}$

vector norm $\|w\|_p$ for $p = 1, 2, 3$

$\|w\|_p = \left(\sum_i |w_i|^p\right)^{\frac{1}{p}}$

$L_2$ norm $\|w\|_2 = \left(\sum_i |w_i|^2\right)^{\frac{1}{2}} = \sqrt{w_1^2 + w_2^2 \cdots w_n^2} = \sqrt{w^T w}$

최적화 문제

$w, b$

$\text{Margin} = \text{distance}(x^+, x^-)$

$= \|x^+ - x^-\|_2$

$= \|(x^- + \lambda w) - x^-\|_2$

$= \|\lambda w\|_2$

$= \lambda \sqrt{w^T w}$

$= \frac{2}{w^T w} \cdot \sqrt{w^T w}$

$= \frac{2}{\sqrt{w^T w}} = \frac{2}{\|w\|_2}$

$\therefore$ Margin은 $w$의 $L_2$ Norm $2$

Margin 최대화

$\max \frac{2}{\|w\|_2} \Leftrightarrow \min \frac{1}{2}\|w\|_2^2$

역수

$L_2$ norm은 제곱할 때 계산어려워 그래서 제곱

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

제약식 subject to $y_i(w^T x_i + b) \geq 1$ 선형

QP (Quadratic programming)

convex optimization 전역최적해존재

linear 제약이 있을때는

Lagrangian Primal

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

Convex continuous 이기 때문에 미분=0 최소값

1. $\frac{\partial L(w,b,\alpha)}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n} a_i y_i x_i$

$w - \sum a_i y_i x_i = 0$

2. $\frac{\partial L(w,b,\alpha)}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} a_i y_i = 0$

$\alpha \geq 0 \;/\; \underline{w, b, a}$

$$\underbrace{\frac{1}{2}\|w\|_2^2}_{1.} - \underbrace{\sum_{i=1}^n a_i(y_i(w^T x_i+b)-1)}_{2.}$$

$$w^{(T)} = \sum_{i=1}^n a_i y_i x_i^{T}$$

1. $\frac{1}{2}\|w\|_2^2 = \frac{1}{2} w^T w$

$$= \frac{1}{2} w^T \sum_{j=1}^n a_j y_j x_j$$

$$= \frac{1}{2} \sum_{j=1}^n a_j y_j (w^T x_j)$$

$$= \frac{1}{2} \sum_{j=1}^n a_j y_j \left(\sum_{i=1}^n a_i y_i x_i^T x_j\right)$$

$$= \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j x_i^T x_j$$

2. $-\sum_{i=1}^n a_i(y_i(w^T x_i+b)-1)$

$$= -\sum_{i=1}^n a_i y_i(w^T x_i+b) + \sum_{i=1}^n a_i$$

$$= -\sum_{i=1}^n a_i y_i w^T x_i - b \underbrace{\sum_{i=1}^n a_i y_i}_{=0} + \sum_{i=1}^n a_i$$

$$= -\sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j x_i^T x_j + \sum_{i=1}^n a_i$$

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

where $\sum_{i=1}^n a_i y_i = 0$

---

maximize $\sum_{i=1}^n \alpha_i - \sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j x_i^T x_j$

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

$\alpha \ge 0$

innerproduct

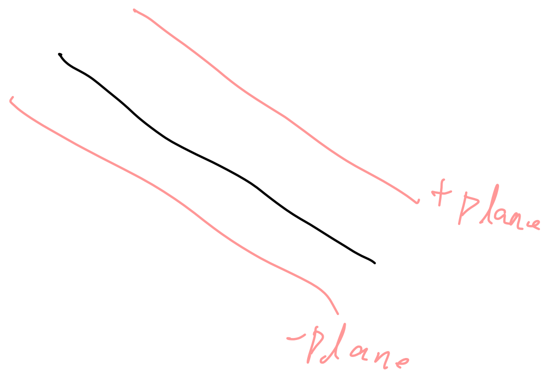
- α를 찾아
- 풀기쉬운형태
- 전역최적해 Convex optimization

---

KKT condition

1. Primal $y_i(w^T x_i + b) \ge 1$
2. Dual $a_i \ge 0$
3. complementary slackness $a_i(y_i(w^T x_i+b)-1) = 0$

보완    여유, 느슨




1. $a_i > 0$ and $\underbrace{y_i(w^T x_i+b)-1=0}_{+\text{plane}}$ / plus, minus 위에 support vector

2. $a_i = 0$ and $y_i(w^T x_i+b)-1 \neq 0$ > ||||

α, w 찾음

⇒ b 찾을수있음

support vector면 $a_i \ge 0$

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

∴ support vector만 이용

데이터 조금사용

# Nonlinear SVM

linear C↑ 마진좁고 overfit
마진넓고 underfit
Softmargin

1. Softmargin ·선형인데 좀 틀려도 돼

minimize $\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$ → error

---

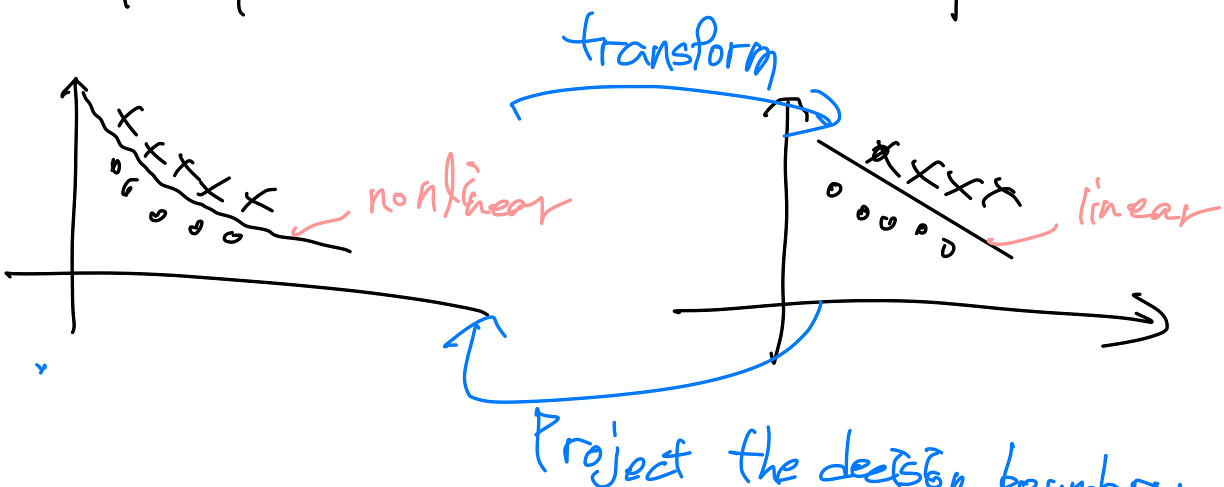
# Kernel

관측치 x들을 보다 높은 차원

Phi x변환

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

Input space $R^p$  $p << q$  Feature space $R^q$ 10




∴ 차원 높여서 선형 hyperplane 찾고 원래차원 Project하면 비선형

---

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

2D  featurespace 5D

그러면 효율적 고차원

# SVM Lagrangian dual formulation

$$\underset{\alpha}{\text{maximize}} \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i,j}^{n} a_i a_j y_i y_j \underbrace{x_i^T x_j}_{\text{inner product}}$$

$$\Downarrow$$

$$\sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i,j}^{n} a_i a_j y_i y_j \underbrace{\phi(x_i)^T\phi(x_j)}_{k(x_i, x_j)}$$

ex) $x=(x_1, x_2)$ $\quad \phi(x) = \underline{(x_1^2, x_2^2, \sqrt{2}x_1x_2)}$

$y=(y_1, y_2)$ $\quad \phi(y)$ $\quad \langle\phi(x), \phi(y)\rangle = x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2$

$\phi$ 형태 알아야 함

But $(x, y)^2 = ((x_1, x_2), (y_1, y_2))^2$ $\therefore$ kernel function 파이를 몰라도 구할 수 있어

$= x_1^2y_1^2 + x_2^2y_2^2 + 2x_1x_2y_1y_2$ $\quad$ trick $\quad$ 파이의 내적

---

kernel 종류

- Sigmoid
- Gaussian
- Polynomial