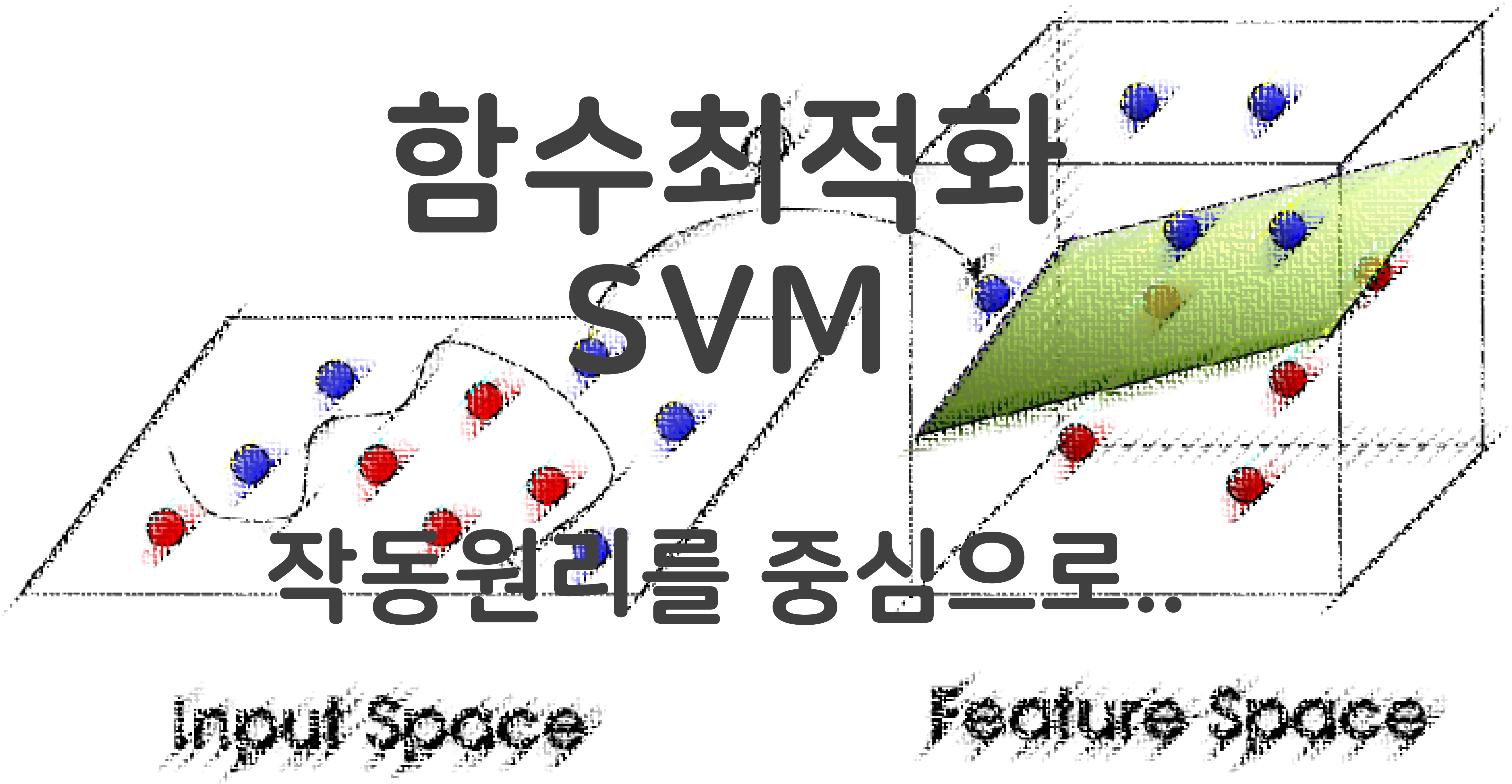
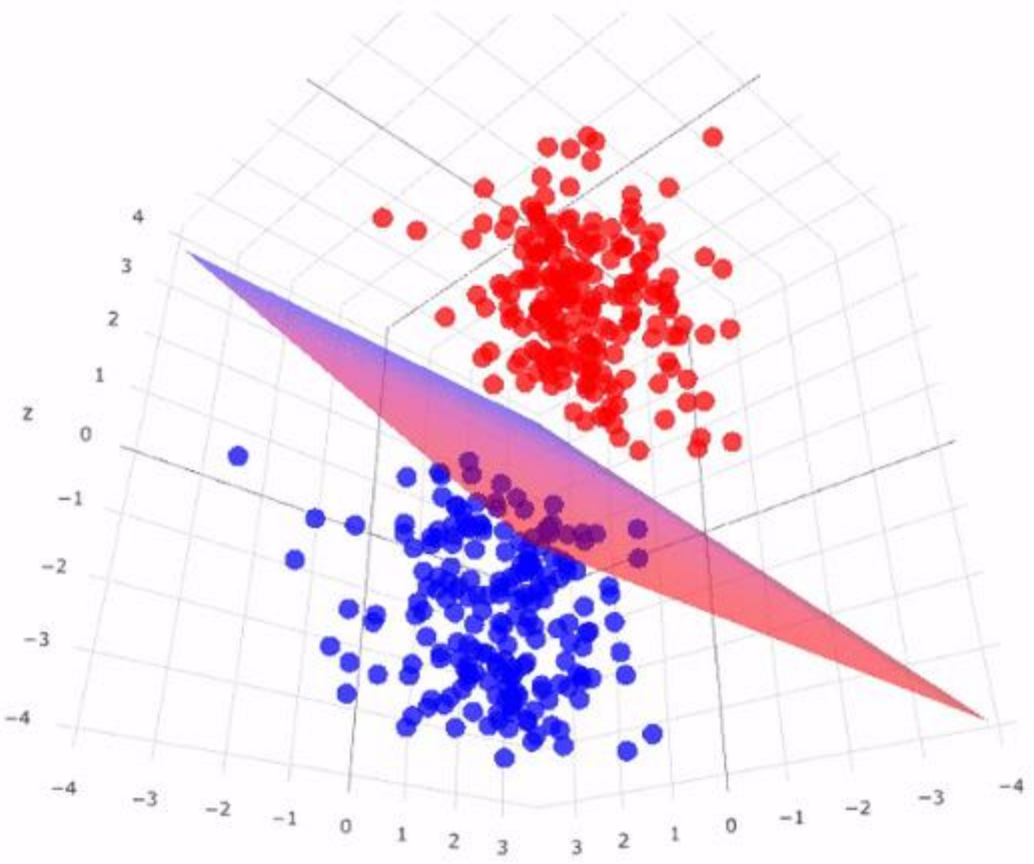
함수최적화 SVM

작동원리를 중심으로..



Toobigs 10기 박규리

SVM 너란녀석.. 왜 좋은거니?

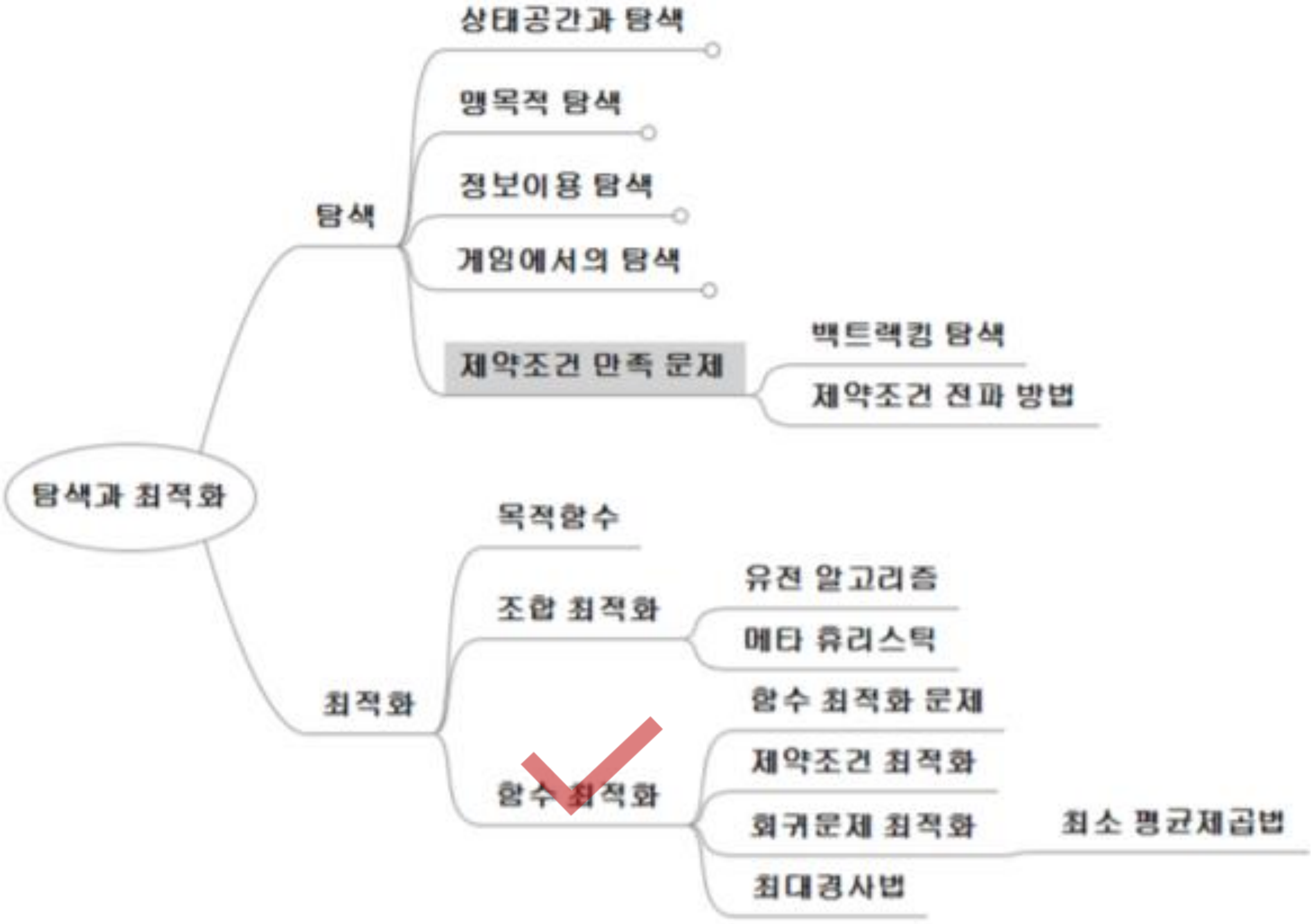


함수 최적화
(function optimization)

탐색과 최적화


탐색과 최적화
탐색
상태공간과 탐색
맹목적 탐색
정보이용 탐색
게임에서의 탐색
제약조건 만족 문제
백트랙킹 탐색
제약조건 전파 방법
최적화
목적함수
조합 최적화
유전 알고리즘
메타 휴리스틱
함수 최적화
함수 최적화 문제
제약조건 최적화
회귀문제 최적화
최소 평균제곱법
최대경사법

탐색과 최적화


탐색과 최적화
탐색
상태공간과 탐색
맹목적 탐색
정보이용 탐색
게임에서의 탐색
제약조건 만족 문제
백트랙킹 탐색
제약조건 전파 방법
최적화
목적함수
조합 최적화
유전 알고리즘
메타 휴리스틱
함수 최적화
함수 최적화 문제
제약조건 최적화
회귀문제 최적화
최소 평균제곱법
최대경사법

함수 최적화
(function optimization)

어떤 목적함수(objective function)가 있을 때,
이 함수를 최대로 하거나 최소로 하는 변수 값을 찾는 최적화 문제

함수 최적화
(function optimization)

어떤 목적함수(objective function)가 있을 때,
이 함수를 최대로 하거나 최소로 하는 변수 값을 찾는 최적화 문제

Find x_1, x_2
which minimizes $f(x_1, x_2) = (x_1 - 1)^2 + {x_2}^2$

목적함수 (objective function)

어떻게 최적화?
(이 함수에서 최소로 하는 값을 어떻게 찾나)

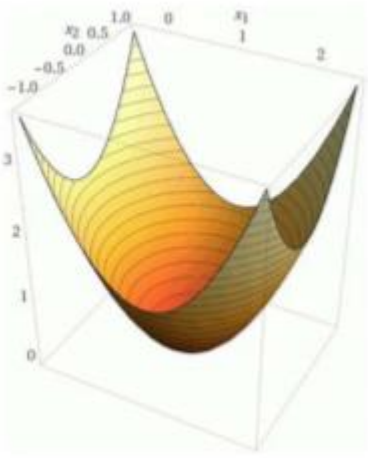
함수 최적화
(function optimization)

어떤 목적함수(objective function)가 있을 때,
이 함수를 최대로 하거나 최소로 하는 변수 값을 찾는 최적화 문제

Find x_1, x_2
which minimizes $f(x_1, x_2) = (x_1 - 1)^2 + {x_2}^2$

목적함수 (objective function)

어떻게 최적화?
(이 함수에서 최소로 하는 값을 어떻게 찾나)



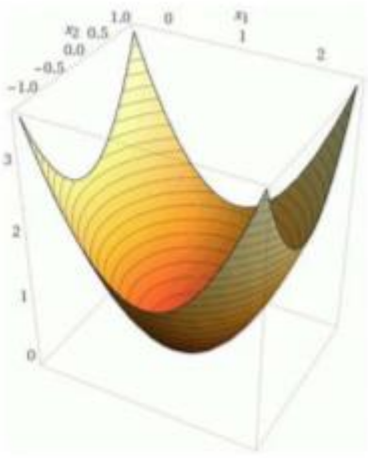
함수 최적화
(function optimization)

어떤 목적함수(objective function)가 있을 때,
이 함수를 최대로 하거나 최소로 하는 변수 값을 찾는 최적화 문제

Find x_1, x_2
which minimizes $f(x_1, x_2) = (x_1 - 1)^2 + {x_2}^2$

목적함수 (objective function)

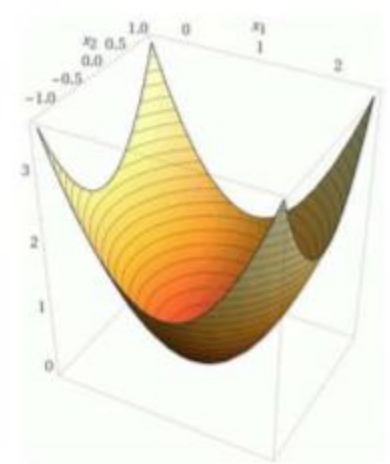
어떻게 최적화?
(이 함수에서 최소로 하는 값을 어떻게 찾나)



아래로 볼록?
볼록한 곳 가운데가 최적점이겠군!

함수 최적화
(function optimization)

어떤 목적함수(objective function)가 있을 때,
ㅐ로 하거나 최소로 하는 변수 값을 찾는 최적화 문제



아래로 볼록?
볼록한 곳 가운데가 최적점이겠군!

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = 2x_1 - 2 = 0 \qquad x_1 = 1$$

$$\frac{\partial f(x_1, x_2)}{\partial x_2} = 2x_2 = 0 \qquad x_2 = 0$$

$$(x_1{}^*, x_2{}^*) = (1,0)$$

늘 하던대로 미분(편미분) 해서
0이 되는 값을 찾는다

제약조건 최적화 (constrained optimization)

하지만 현실의 문제는 늘 제약조건이 있죠!
제약조건(constraints)을 만족시키면서 목적함수를 최적화시키는 변수를 찾아보자구요!

목적함수(최소화)

등식제약조건

부등식제약조건

Find x_1, x_2

which minimizes $f(x_1,x_2)=\frac{1}{2}(x_1^2+x_2^2)$

subject to $h(x_1,x_2)=1-x_1-x_2=0$

$g(x_1,x_2)=\frac{3}{4}-x_2 \le 0$

제약조건 (constraintx)

제약조건이랍니다!

제약조건 최적화
(constrained optimization)

제약조건(constraints)을 만족시키면서 목적함수를 최적화시키는 변수를 찾아보자구요!

목적함수(최소화)

등식제약조건

부등식제약조건

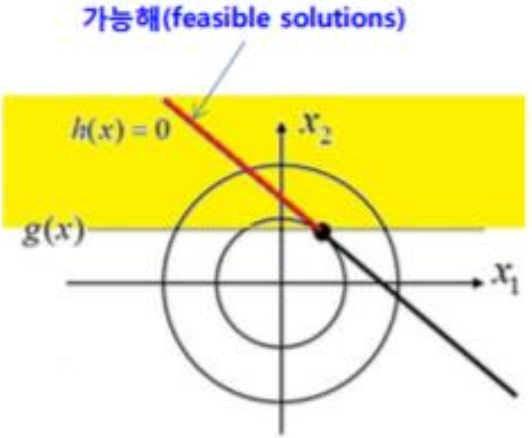
Find x_1, x_2

which minimizes $f(x_1,x_2)=\frac{1}{2}(x_1^2+x_2^2)$

subject to $h(x_1,x_2)=1-x_1-x_2=0$

$g(x_1,x_2)=\frac{3}{4}-x_2\leq 0$

제약조건이랍니다!



그림으로 그리면 요로케돼요

가능해(feasible solutions : 답이 될 수 있는 영역)에서
최적해를 찾아야한다!

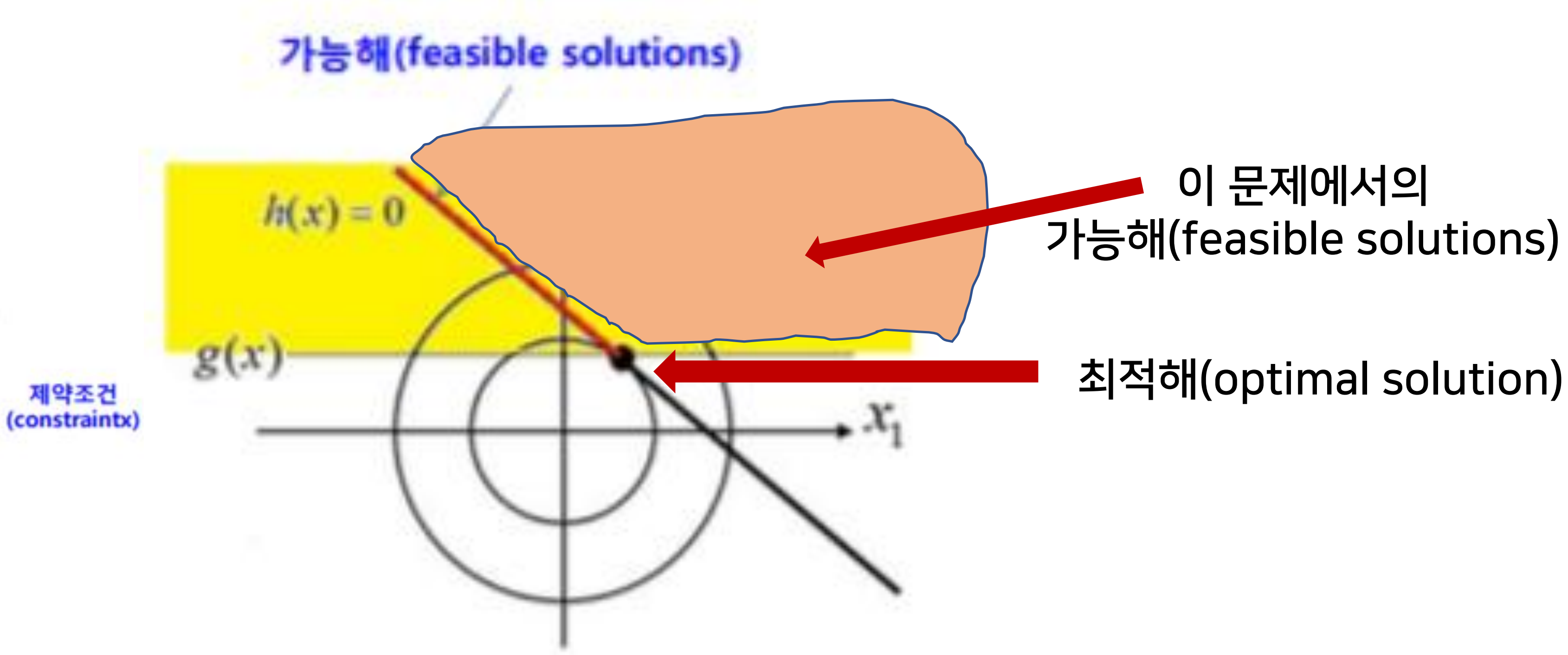
제약조건 최적화
(constrained optimization)

Find x_1, x_2

which minimizes $f(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2)$

subject to $h(x_1, x_2) = 1 - x_1 - x_2 = 0$

$g(x_1, x_2) = \frac{3}{4} - x_2 \leq 0$



가능해(feasible solutions : 답이 될 수 있는 영역)

목적함수를 최대화 하는 변수값(최적해)을 찾아봅시다!

라그랑주 함수
(Lagrange function)

제약조건 내에서 최적화할 때 필요한 우리들의 친구 라그랑주 함수

라그랑주 함수
(Lagrange function)

제약조건 내에서 최적화할 때 필요한 우리들의 친구 라그랑주 함수

: 등식으로 주어지는 제한 조건을 만족하는 영역에서 다변수함수의 극값을 찾을 때 자주 사용하는 방법

라그랑주 함수

위 보기에서 보듯 극점 P는 라그랑주 조건 (1)뿐만 아니라, 제한 조건 $g(a,b)=0$도 만족해야 한다.

변수를 하나 늘린 라그랑주 함수 L을 다음과 같이 정의하자.

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

이때 $\mathbf{grad}L(x,y,\lambda)$는

$$(f_x(x,y)-\lambda g_x(x,y),\ f_y(x,y)-\lambda g_y(x,y),\ -g(x,y))$$

로 주어진다. 이때 f_x는 f를 변수 x로 편미분한 것을 나타내는 기호이며 f_y, g_x, g_y 등도 비슷하게 정의한다. 그러면 라그랑주 승수법은

$$\mathbf{grad}\, L(a,b,\lambda)=\mathbf{0}$$

이라는 하나의 식으로 나타낼 수도 있다.

https://youtu.be/hQ4UNu1P2kw

라그랑주 함수가 이해되질 않는다면
이 링크를 참조하도록 하자.

그래도 이해되지 않는다면
포기하자.

라그랑주 함수
(Lagrange function)

제약조건 최적화할 때 필요한 우리들의 친구 라그랑주 함수

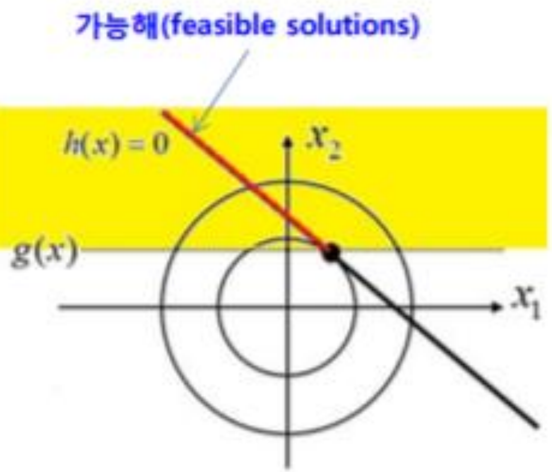
제약조건 최적화(constrained optimization)

Find x_1, x_2

which minimizes $f(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2)$

subject to $h(x_1, x_2) = 1 - x_1 - x_2 = 0$

$$g(x_1, x_2) = \frac{3}{4} - x_2 \le 0$$



라그랑주 함수
(Lagrange function)

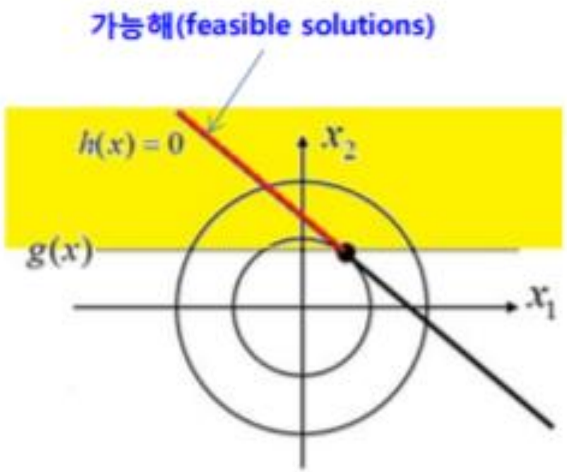
제약조건 최적화할 때 필요한 우리들의 친구 라그랑주 함수

제약조건 최적화(constrained optimization)

Find x_1, x_2

$$\boxed{\begin{aligned} &\text{which minimizes } f(x_1,x_2) = \frac{1}{2}(x_1^2 + x_2^2) \\ &\text{subject to } h(x_1,x_2) = 1 - x_1 - x_2 = 0 \\ &\qquad\qquad g(x_1,x_2) = \frac{3}{4} - x_2 \le 0 \end{aligned}}$$

조건이 너무 많아서 헷갈린다
하나의 함수로 만들자



라그랑주 함수 (Lagrange function)

제약조건 최적화할 때 필요한 우리들의 친구 라그랑주 함수

제약조건 최적화(constrained optimization)

Find x_1, x_2

which minimizes $f(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2)$

subject to $h(x_1, x_2) = 1 - x_1 - x_2 = 0$

$g(x_1, x_2) = \frac{3}{4} - x_2 \le 0$

조건이 너무 많아서 헷갈린다
하나의 함수로 만들자

라그랑주 함수(Lagrange) 함수
: 제약조건들과 목적함수의 결합

$$L(x_1, x_2, \lambda, \alpha) = f(x_1, x_2) + \lambda h(x_1, x_2) + \alpha g(x_1, x_2)$$

$$= \frac{1}{2}(x_1^2 + x_2^2) + \lambda(1 - x_1 - x_2) + \alpha\left(\frac{3}{4} - x_2\right) \quad (\alpha \ge 0)$$

λ, α : 라그랑주 승수(Lagrange multiplier)

라그랑주 함수
(Lagrange function)

제약조건 최적화할 때 필요한 우리들의 친구 라그랑주 함수

라그랑주 함수(Lagrange) 함수
: 제약조건들과 목적함수의 결합

$$L(x_1, x_2, \lambda, \alpha) = f(x_1, x_2) + \lambda h(x_1, x_2) + \alpha g(x_1, x_2)$$

λ, α : 라그랑주 승수(Lagrange multiplier)

$$= \frac{1}{2}(x_1^2 + x_2^2) + \lambda(1 - x_1 - x_2) + \alpha\left(\frac{3}{4} - x_2\right)$$

$(\alpha \geq 0)$

Find x_1, x_2

which minimizes $f(x_1,x_2)=\frac{1}{2}(x_1^2+x_2^2)$

subject to $h(x_1,x_2)=1-x_1-x_2=0$

$g(x_1,x_2)=\frac{3}{4}-x_2 \le 0$

라그랑주 함수 (Lagrange function)

목적함수는 그냥 더해주고

제약조건 두개를 각각 상수를 곱해서 더해준다
그 상수는 라그랑주 승수(람다, 알파)라고 한다

$$L(x_1,x_2,\lambda,\alpha)=\boxed{f(x_1,x_2)}+\boxed{\lambda h(x_1,x_2)}+\boxed{\alpha g(x_1,x_2)}$$

$$=\frac{1}{2}(x_1^2+x_2^2)+\lambda(1-x_1-x_2)+\alpha\left(\frac{3}{4}-x_2\right)$$

λ, α : 라그랑주 승수(Lagrange multiplier)

$(\alpha \ge 0)$

Find x_1, x_2

which minimizes $f(x_1,x_2)=\frac{1}{2}(x_1^2+x_2^2)$

subject to $h(x_1,x_2)=1-x_1-x_2=0$

$g(x_1,x_2)=\frac{3}{4}-x_2 \le 0$

라그랑주 함수 (Lagrange function)

목적함수는 그냥 더해주고

제약조건 두개를 각각 상수를 곱해서 더해준다
그 상수는 라그랑주 승수(람다, 알파)라고 한다

$$L(x_1,x_2,\lambda,\alpha)=\boxed{f(x_1,x_2)}+\boxed{\lambda h(x_1,x_2)}+\boxed{\alpha g(x_1,x_2)}$$

λ, α : 라그랑주 승수(Lagrange multiplier)

$$=\boxed{\frac{1}{2}(x_1^2+x_2^2)}+\boxed{\lambda(1-x_1-x_2)+\alpha\left(\frac{3}{4}-x_2\right)}$$

$\boxed{(\alpha \ge 0)}$

목적함수

제약조건

이 함수에는 조건 하나가 붙음

Find x_1, x_2
which minimizes $f(x_1,x_2)=\frac{1}{2}(x_1^2+x_2^2)$
subject to $h(x_1,x_2)=1-x_1-x_2=0$
$g(x_1,x_2)=\frac{3}{4}-x_2\le 0$

라그랑주 함수 (Lagrange function)

목적함수는 그냥 더해주고

제약조건 두개를 각각 상수를 곱해서 더해준다
그 상수는 라그랑주 승수(람다, 알파)라고 한다

$$L(x_1,x_2,\lambda,\alpha)=\boxed{f(x_1,x_2)}+\boxed{\lambda h(x_1,x_2)}+\boxed{\alpha g(x_1,x_2)}$$

λ, α : 라그랑주 승수(Lagrange multiplier)

$$=\underbrace{\boxed{\frac{1}{2}(x_1^2+x_2^2)}}_{\text{목적함수}}+\underbrace{\boxed{\lambda(1-x_1-x_2)+\alpha\left(\frac{3}{4}-x_2\right)}}_{\text{제약조건}} \qquad \boxed{(\alpha\ge 0)}$$

우리가 하려는 것!

$$\min_{x_1,x_2\in FS} f(x_1,x_2) = \min_{x_1,x_2}\max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha)$$

FS : 가능해(feasible solution)의 집합

$$\min_{x_1,x_2}\max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha) \ge \max_{\lambda,\alpha}\boxed{\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)}$$

$$\ge \max_{\lambda,\alpha} L_d(\lambda,\alpha)$$

$$L_d(\lambda,\alpha)=\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$$

쌍대함수(dual function)

라그랑주 함수 (Lagrange function)

Find x_1, x_2

which minimizes $f(x_1,x_2)=\frac{1}{2}(x_1^2+x_2^2)$

subject to $h(x_1,x_2)=1-x_1-x_2=0$

$g(x_1,x_2)=\frac{3}{4}-x_2 \le 0$

목적함수는 그냥 더해주고

제약조건 두개를 각각 상수를 곱해서 더해준다
그 상수는 라그랑주 승수(람다, 알파)라고 한다

$$L(x_1,x_2,\lambda,\alpha)=\boxed{f(x_1,x_2)}+\boxed{\lambda h(x_1,x_2)}+\boxed{\alpha g(x_1,x_2)}$$

λ, α : 라그랑주 승수(Lagrange multiplier)

$$=\underbrace{\boxed{\frac{1}{2}(x_1^2+x_2^2)}}_{\text{목적함수}}+\underbrace{\boxed{\lambda(1-x_1-x_2)+\alpha\left(\frac{3}{4}-x_2\right)}}_{\text{제약조건}} \qquad \boxed{(\alpha \ge 0)}$$

우리가 하려는 것 : 가능해(feasible solution)을 만족하는 최소의 x1, x2를 찾는 것!

$$\boxed{\min_{x_1,x_2\in FS} f(x_1,x_2)}=\min_{x_1,x_2}\max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha)$$

FS : 가능해(feasible solution)의 집합

$$\min_{x_1,x_2}\max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha) \ge \max_{\lambda,\alpha}\boxed{\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)}$$

$$\ge \max_{\lambda,\alpha} L_d(\lambda,\alpha)$$

$L_d(\lambda,\alpha)=\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$

쌍대함수(dual function)

라그랑주 함수
(Lagrange function)

Find x_1, x_2

which minimizes $f(x_1,x_2) = \frac{1}{2}(x_1^2 + x_2^2)$

subject to $h(x_1,x_2) = 1 - x_1 - x_2 = 0$

$g(x_1,x_2) = \frac{3}{4} - x_2 \le 0$

제약조건 두개를 각각 상수를 곱해서 더해준다
그 상수는 라그랑주 승수(람다, 알파)라고 한다

목적함수는 그냥 더해주고

$$L(x_1,x_2,\lambda,\alpha) = \boxed{f(x_1,x_2)} + \boxed{\lambda h(x_1,x_2)} + \boxed{\alpha g(x_1,x_2)}$$

λ, α : 라그랑주 승수(Lagrange multiplier)

$$= \boxed{\frac{1}{2}(x_1^2 + x_2^2)} + \boxed{\lambda(1 - x_1 - x_2) + \alpha\left(\frac{3}{4} - x_2\right)} \quad \boxed{(\alpha \ge 0)}$$

목적함수 제약조건

우리가 하려는 것 : 바꿔말하면 최대의 알파와 람다값을 갖는 식에서 최소의 x1,x2 찾기!

$$\min_{x_1,x_2 \in FS} f(x_1,x_2) = \boxed{\min_{x_1,x_2} \max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha)}$$

FS : 가능해(feasible solution)의 집합

$$\min_{x_1,x_2} \max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha) \ge \max_{\lambda,\alpha} \min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$$

$$\ge \max_{\lambda,\alpha} L_d(\lambda,\alpha)$$

$$L_d(\lambda,\alpha) = \min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$$

쌍대함수(dual function)

라그랑주 함수
(Lagrange function)

Find x_1, x_2

which minimizes $f(x_1,x_2)=\frac{1}{2}(x_1^2+x_2^2)$

subject to $\boxed{h(x_1,x_2)=1-x_1-x_2=0}$

$g(x_1,x_2)=\frac{3}{4}-x_2 \leq 0$

$$L(x_1,x_2,\lambda,\alpha)=f(x_1,x_2)+\lambda h(x_1,x_2)+\alpha g(x_1,x_2)$$

λ, α : 라그랑주 승수(Lagrange multiplier)

$$=\frac{1}{2}(x_1^2+x_2^2)+\lambda\boxed{\left(1-x_1-x_2\right)}+\alpha\left(\frac{3}{4}-x_2\right)$$

$(\alpha \geq 0)$

제약조건에 의하면 람다에 상관 없이 0이다

우리가 하려는 것 : 바꿔말하면 최대의 알파와 람다값을 갖는 식에서 최소의 x1,x2 찾기!

$$\min_{x_1,x_2 \in FS} f(x_1,x_2) = \boxed{\min_{x_1,x_2}\max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha)}$$

FS : 가능해(feasible solution)의 집합

$$\min_{x_1,x_2}\max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha) \geq \max_{\lambda,\alpha}\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$$

$$\geq \max_{\lambda,\alpha} L_d(\lambda,\alpha)$$

$L_d(\lambda,\alpha)=\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$

쌍대함수(dual function)

라그랑주 함수
(Lagrange function)

Find x_1, x_2

which minimizes $f(x_1,x_2)=\frac{1}{2}(x_1^2+x_2^2)$

subject to $h(x_1,x_2)=1-x_1-x_2=0$

$$\boxed{g(x_1,x_2)=\frac{3}{4}-x_2\le 0}$$

$$L(x_1,x_2,\lambda,\alpha)=f(x_1,x_2)+\lambda h(x_1,x_2)+\alpha g(x_1,x_2)$$

λ, α : 라그랑주 승수(Lagrange multiplier)

$$=\frac{1}{2}(x_1^2+x_2^2)+\lambda(1-x_1-x_2)+\boxed{\alpha\left(\frac{3}{4}-x_2\right)} \quad (\alpha\ge 0)$$

Cf) 음수에다가 0이 아닌값을 곱하면 음수다
0을 곱하면 0이다

Why? (3/4-x2)는 0이거나 음수, 알파는 0이거나 크다
그러므로 알파가 0일 때 최대값!

우리가 하려는 것 : 바꿔말하면 최대의 알파와 람다값을 갖는 식에서 최소의 x1,x2 찾기!

$$\min_{x_1,x_2\in FS} f(x_1,x_2)=\boxed{\min_{x_1,x_2}\max_{\lambda,\alpha}L(x_1,x_2,\lambda,\alpha)}$$

FS : 가능해(feasible solution)의 집합

$$\min_{x_1,x_2}\max_{\lambda,\alpha}L(x_1,x_2,\lambda,\alpha)\ge\max_{\lambda,\alpha}\min_{x_1,x_2}L(x_1,x_2,\lambda,\alpha)$$

$$\ge\max_{\lambda,\alpha}L_d(\lambda,\alpha)$$

$$L_d(\lambda,\alpha)=\min_{x_1,x_2}L(x_1,x_2,\lambda,\alpha)$$

쌍대함수(dual function)

라그랑주 함수
(Lagrange function)

Find x_1, x_2

which minimizes $f(x_1,x_2)=\frac{1}{2}(x_1^2+x_2^2)$

subject to $h(x_1,x_2)=1-x_1-x_2=0$

$\boxed{g(x_1,x_2)=\frac{3}{4}-x_2\leq 0}$

$$L(x_1,x_2,\lambda,\alpha)=f(x_1,x_2)+\lambda h(x_1,x_2)+\alpha g(x_1,x_2)$$

λ, α : 라그랑주 승수(Lagrange multiplier)

$$=\frac{1}{2}(x_1^2+x_2^2)+\boxed{\lambda(1-x_1-x_2)+\alpha\left(\frac{3}{4}-x_2\right)}$$

$(\alpha\geq 0)$

그러므로 이 식은 0이 된다

우리가 하려는 것 : 바꿔말하면 최대의 알파와 람다값을 갖는 식에서 최소의 x1,x2 찾기!

$$\min_{x_1,x_2\in FS} f(x_1,x_2)=\boxed{\min_{x_1,x_2}\max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha)}$$

FS : 가능해(feasible solution)의 집합

$$\min_{x_1,x_2}\max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha)\geq \max_{\lambda,\alpha}\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$$

$$\geq \max_{\lambda,\alpha} L_d(\lambda,\alpha)$$

$L_d(\lambda,\alpha)=\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$

쌍대함수(dual function)

라그랑주 함수 (Lagrange function)

Find x_1, x_2

which minimizes $f(x_1,x_2)=\frac{1}{2}(x_1^2+x_2^2)$

subject to $h(x_1,x_2)=1-x_1-x_2=0$

$$\boxed{g(x_1,x_2)=\frac{3}{4}-x_2 \le 0}$$

$$L(x_1,x_2,\lambda,\alpha)=f(x_1,x_2)+\lambda h(x_1,x_2)+\alpha g(x_1,x_2)$$

λ, α : 라그랑주 승수(Lagrange multiplier)

$$=\frac{1}{2}(x_1^2+x_2^2)+\boxed{\lambda(1-x_1-x_2)+\alpha\left(\frac{3}{4}-x_2\right)} \quad (\alpha \ge 0)$$

그러므로 이 식은 0이 된다 → 제약조건을 만족하면 목적함수만 남음!

그러므로 이 식으로 우리가 하려는 것을 달성할 수 있다!

$$\min_{x_1,x_2 \in FS} f(x_1,x_2) = \boxed{\min_{x_1,x_2}\max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha)}$$

FS : 가능해(feasible solution)의 집합

$$\min_{x_1,x_2}\max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha) \ge \max_{\lambda,\alpha}\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$$

$$\ge \max_{\lambda,\alpha} L_d(\lambda,\alpha)$$

$$L_d(\lambda,\alpha)=\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$$

쌍대함수(dual function)

라그랑주 함수
(Lagrange function)

Find x_1, x_2

which minimizes $f(x_1,x_2)=\frac{1}{2}(x_1^2+x_2^2)$

subject to $h(x_1,x_2)=1-x_1-x_2=0$

$g(x_1,x_2)=\frac{3}{4}-x_2 \le 0$

$$L(x_1,x_2,\lambda,\alpha)=f(x_1,x_2)+\lambda h(x_1,x_2)+\alpha g(x_1,x_2)$$

λ, α : 라그랑주 승수(Lagrange multiplier)

$$=\frac{1}{2}(x_1^2+x_2^2)+\lambda(1-x_1-x_2)+\alpha\left(\frac{3}{4}-x_2\right)$$

$(\alpha \ge 0)$

최적화 하기 위해서 Min과 max위치를 바꿈 : 부등식 관계로 바뀜
(큰 것 중에 작은 것 선택하는 것이 작은 것 중에 큰 것을 선택하는 것보다 크다)

$$\min_{x_1,x_2\in FS} f(x_1,x_2) = \min_{x_1,x_2}\max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha)$$

FS : 가능해(feasible solution)의 집합

$$\boxed{\min_{x_1,x_2}\max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha) \ge \max_{\lambda,\alpha}\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)}$$

$$L_d(\lambda,\alpha)=\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$$

$$\ge \max_{\lambda,\alpha} L_d(\lambda,\alpha)$$

쌍대함수(dual function)

라그랑주 함수 (Lagrange function)

Find x_1, x_2

which minimizes $f(x_1,x_2) = \frac{1}{2}(x_1^2 + x_2^2)$

subject to $h(x_1,x_2) = 1 - x_1 - x_2 = 0$

$g(x_1,x_2) = \frac{3}{4} - x_2 \le 0$

$$L(x_1,x_2,\lambda,\alpha) = f(x_1,x_2) + \lambda h(x_1,x_2) + \alpha g(x_1,x_2)$$

λ, α : 라그랑주 승수(Lagrange multiplier)

$$= \frac{1}{2}(x_1^2 + x_2^2) + \lambda(1 - x_1 - x_2) + \alpha\left(\frac{3}{4} - x_2\right)$$

$(\alpha \ge 0)$

쌍대함수(dual function)

: 라그랑주 함수를 x1,x2에 대해서 최소화(minimize)하도록 만든 식

$$\min_{x_1,x_2 \in FS} f(x_1,x_2) = \min_{x_1,x_2} \max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha)$$

FS : 가능해(feasible solution)의 집합

$$\min_{x_1,x_2} \max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha) \ge \max_{\lambda,\alpha} \boxed{\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)}$$

$$\ge \max_{\lambda,\alpha} L_d(\lambda,\alpha)$$

$$L_d(\lambda,\alpha) = \min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$$

쌍대함수(dual function)

라그랑주 함수 (Lagrange function)

Find x_1, x_2

which minimizes $f(x_1,x_2)=\frac{1}{2}(x_1^2+x_2^2)$

subject to $h(x_1,x_2)=1-x_1-x_2=0$

$g(x_1,x_2)=\frac{3}{4}-x_2\le 0$

$$L(x_1,x_2,\lambda,\alpha)=f(x_1,x_2)+\lambda h(x_1,x_2)+\alpha g(x_1,x_2)$$

λ, α : 라그랑주 승수(Lagrange multiplier)

$$=\frac{1}{2}(x_1^2+x_2^2)+\lambda(1-x_1-x_2)+\alpha\left(\frac{3}{4}-x_2\right)$$

$(\alpha \ge 0)$

쌍대함수(Dual function)을 찾고 그 식의
람다 알파에 대해서 최대화 하는 것을 찾으면
왼쪽식보다 작거나 작게 된다

$$\min_{x_1,x_2\in FS} f(x_1,x_2) = \min_{x_1,x_2}\max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha)$$

FS : 가능해(feasible solution)의 집합

$$\boxed{\min_{x_1,x_2}\max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha)} \ge \max_{\lambda,\alpha}\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$$

$$\ge \boxed{\max_{\lambda,\alpha} L_d(\lambda,\alpha)}$$

$$L_d(\lambda,\alpha)=\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$$

쌍대함수(dual function)

라그랑주 함수
(Lagrange function)

만약에 크거나 같은 것이 아니라 같은 값이 되게 하려면??

$$\min_{x_1,x_2 \in FS} f(x_1,x_2) = \min_{x_1,x_2} \max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha)$$

FS : 가능해(feasible solution)의 집합

$$\boxed{\begin{aligned}\min_{x_1,x_2} \max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha) &\geq \max_{\lambda,\alpha} \boxed{\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)} \\ &\geq \max_{\lambda,\alpha} L_d(\lambda,\alpha)\end{aligned}}$$

$$L_d(\lambda,\alpha) = \min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$$

쌍대함수(dual function)

라그랑주 함수
(Lagrange function)

만약에 크거나 같은 것이 아니라 같은 값이 되게 하려면??

$$\min_{x_1,x_2 \in FS} f(x_1,x_2) = \min_{x_1,x_2}\max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha)$$

FS : 가능해(feasible solution)의 집합

$$\min_{x_1,x_2}\max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha) \ge \max_{\lambda,\alpha} \boxed{\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)}$$

$$\ge \max_{\lambda,\alpha} L_d(\lambda,\alpha)$$

$$L_d(\lambda,\alpha) = \min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$$

쌍대함수(dual function)

쌍대함수(dual function)를 최대화하면서 상보적 여유성
(complementary slackness)를 만족하는 x1,x2를 구한다!!

라그랑주 함수
(Lagrange function)

만약에 크거나 같은 것이 아니라 같은 값이 되게 하려면??

$$\min_{x_1,x_2 \in FS} f(x_1,x_2) = \min_{x_1,x_2} \max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha)$$

FS : 가능해(feasible solution)의 집합

$$\min_{x_1,x_2} \max_{\lambda,\alpha} L(x_1,x_2,\lambda,\alpha) \geq \max_{\lambda,\alpha} \boxed{\min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)}$$
$$\geq \max_{\lambda,\alpha} L_d(\lambda,\alpha)$$

$$L_d(\lambda,\alpha) = \min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$$

쌍대함수(dual function)

쌍대함수(dual function)를 최대화하면서
상보적 여유성(complementary slackness)를 만족하는 x1,x2를 구한다!!

$$\boxed{\alpha g(x_1,x_2) = 0}$$

상보적 여유성(complementary slackness)

$$L(x_1,x_2,\lambda,\alpha) = f(x_1,x_2) + \lambda h(x_1,x_2) + \alpha g(x_1,x_2)$$
$$= \frac{1}{2}(x_1^2+x_2^2) + \lambda(1-x_1-x_2) + \alpha\left(\frac{3}{4}-x_2\right) \quad (\alpha \geq 0)$$

λ, α : 라그랑주 승수(Lagrange multiplier)

부등식의 제약조건을 0으로 만드는 조건!
(상보적 : 알파가 0이 되던지 뒤의 식이 0이되던지 둘중 하나 0)

라그랑주 함수
(Lagrange function)

쌍대함수(dual function)를 최대화하면서
상보적 여유성(complementary slackness)을
만족하는 x1,x2

그것은 바로 최적해!!

(그 이유는 우리보다 똑똑한 분들이 잘 증명해 놓으셨다…ㅎ)

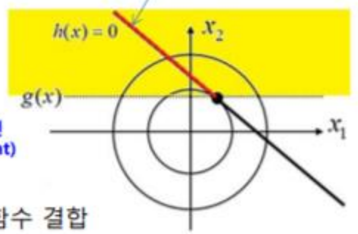
❖ **제약조건 최적화**(constrained optimization)

Find x_1, x_2

which minimizes $f(x_1,x_2)=\frac{1}{2}(x_1^2+x_2^2)$

subject to $h(x_1,x_2)=1-x_1-x_2=0$

$g(x_1,x_2)=\frac{3}{4}-x_2 \le 0$



- **라그랑주(Lagrange) 함수** : 제약조건들과 목적함수 결합

$$L(x_1,x_2,\lambda,\alpha)=f(x_1,x_2)+\lambda h(x_1,x_2)+\alpha g(x_1,x_2)$$

λ, α : 라그랑주 승수(Lagrange multiplier)

$$=\frac{1}{2}(x_1^2+x_2^2)+\lambda(1-x_1-x_2)+\alpha\left(\frac{3}{4}-x_2\right) \quad (\alpha \ge 0)$$

- **최적화 방법**

$$\min_{x_1,x_2\in FS} f(x_1,x_2) = \min_{x_1,x_2}\max_{\lambda,\alpha}L(x_1,x_2,\lambda,\alpha)$$

FS : 가능해(feasible solution)의 집합

$$\min_{x_1,x_2}\max_{\lambda,\alpha}L(x_1,x_2,\lambda,\alpha) \ge \max_{\lambda,\alpha}\boxed{\min_{x_1,x_2}L(x_1,x_2,\lambda,\alpha)}$$

$$\ge \max_{\lambda,\alpha}L_d(\lambda,\alpha)$$

$L_d(\lambda,\alpha)=\min_{x_1,x_2}L(x_1,x_2,\lambda,\alpha)$

쌍대함수(dual function)

- **쌍대함수를 최대화하면서 상보적 여유성을 만족하는 x_1, x_2를 구함**

$\alpha g(x_1,x_2)=0$

상보적 여유성(complementary slackness)

라그랑주 함수
(Lagrange function)

띠용??

예제를 봅시다

라그랑주 함수
(Lagrange function)

라그랑주 함수 예제

$$L(x_1,x_2,\lambda,\alpha) = \frac{1}{2}(x_1^2+x_2^2)+\lambda(1-x_1-x_2)+\alpha\left(\frac{3}{4}-x_2\right)$$

라그랑주 함수(Lagrange funtction)

$$L_d(\lambda,\alpha) = \min_{x_1,x_2} L(x_1,x_2,\lambda,\alpha)$$

쌍대함수(dual function)
최소값을 계산해야한다!

라그랑주 함수 (Lagrange function)

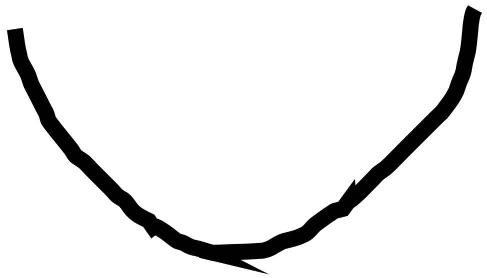
라그랑주 함수 예제

$$L(x_1, x_2, \lambda, \alpha) = \frac{1}{2}(x_1^2 + x_2^2) + \lambda(1 - x_1 - x_2) + \alpha\left(\frac{3}{4} - x_2\right)$$

라그랑주 함수(Lagrange funtction)

$$L_d(\lambda, \alpha) = \min_{x_1, x_2} L(x_1, x_2, \lambda, \alpha)$$

쌍대함수(dual function)
최소값을 계산해야한다!



볼록 함수라고 함

늘 하던대로 미분!

$$\frac{\partial L(x_1, x_2, \lambda, \alpha)}{\partial x_1} = x_1 - \lambda = 0 \quad x_1 = \lambda$$

$$\frac{\partial L(x_1, x_2, \lambda, \alpha)}{\partial x_2} = x_2 - \lambda - \alpha = 0 \quad x_2 = \lambda + \alpha$$

$$L_d(\lambda, \alpha) = -\lambda^2 - \frac{1}{2}\alpha^2 - \lambda\alpha + \lambda + \frac{3}{4}\alpha$$

라그랑주 함수
(Lagrange function)

라그랑주 함수 예제

$$\boxed{L(x_1,x_2,\lambda,\alpha)=\frac{1}{2}(x_1^2+x_2^2)+\lambda(1-x_1-x_2)+\alpha\left(\frac{3}{4}-x_2\right)}$$

라그랑주 함수(Lagrange funtction)

X1과 x2를 구했으니
이제 집어 넣읍시다

$$\frac{\partial L(x_1,x_2,\lambda,\alpha)}{\partial x_1}=x_1-\lambda=0 \quad \boxed{x_1=\lambda}$$

$$\frac{\partial L(x_1,x_2,\lambda,\alpha)}{\partial x_2}=x_2-\lambda-\alpha=0 \quad \boxed{x_2=\lambda+\alpha}$$

$$L_d(\lambda,\alpha)=-\lambda^2-\frac{1}{2}\alpha^2-\lambda\alpha+\lambda+\frac{3}{4}\alpha$$

라그랑주 함수
(Lagrange function)

라그랑주 함수 예제

$$\boxed{L(x_1,x_2,\lambda,\alpha)=\frac{1}{2}(x_1^2+x_2^2)+\lambda(1-x_1-x_2)+\alpha\left(\frac{3}{4}-x_2\right)}$$

라그랑주 함수(Lagrange funtction)

집어 넣으면 이렇게 되지롱

$$\frac{\partial L(x_1,x_2,\lambda,\alpha)}{\partial x_1}=x_1-\lambda=0 \quad \boxed{x_1=\lambda}$$

$$\frac{\partial L(x_1,x_2,\lambda,\alpha)}{\partial x_2}=x_2-\lambda-\alpha=0 \quad \boxed{x_2=\lambda+\alpha}$$

$$L_d(\lambda,\alpha)=-\lambda^2-\frac{1}{2}\alpha^2-\lambda\alpha+\lambda+\frac{3}{4}\alpha$$

$$\boxed{L_d(\lambda,\alpha)=-\lambda^2-\frac{1}{2}\alpha^2-\lambda\alpha+\lambda+\frac{3}{4}\alpha}$$

이렇게 구한 것을 쌍대함수(dual function)이라고 한다

라그랑주 함수
(Lagrange function)

라그랑주 함수 예제

$$\boxed{L_d(\lambda,\alpha) = -\lambda^2 - \frac{1}{2}\alpha^2 - \lambda\alpha + \lambda + \frac{3}{4}\alpha}$$

이렇게 구한 쌍대함수(dual function)를 최대화해야 한다

$$\max_{\lambda,\alpha} L_d(\lambda,\alpha)$$

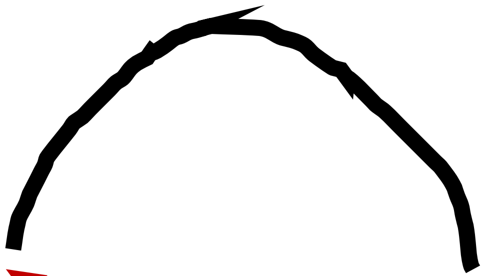
라그랑주 함수
(Lagrange function)

라그랑주 함수 예제

$$L_d(\lambda,\alpha) = -\lambda^2 - \frac{1}{2}\alpha^2 - \lambda\alpha + \lambda + \frac{3}{4}\alpha$$

이렇게 구한 쌍대함수(dual function)를 최대화해야 한다

$$\max_{\lambda,\alpha} L_d(\lambda,\alpha)$$



이거슨 위로 볼록하네?
또 미분!

$$\frac{\partial L_d(\lambda,\alpha)}{\partial \lambda} = -2\lambda - \alpha + 1 = 0 \quad \frac{\partial L_d(\lambda,\alpha)}{\partial \alpha} = -\alpha - \lambda + \frac{3}{4} = 0$$

$$\lambda = \frac{1}{4} \quad \alpha = \frac{1}{2} \qquad \alpha\left(\frac{3}{4} - x_2\right) = 0 \quad x_2 = \frac{3}{4} \quad 1 - x_1 - x_2 = 0 \quad x_1 = \frac{1}{4}$$

$$\therefore\ x_1 = \frac{1}{4} \quad x_2 = \frac{3}{4}$$

라그랑주 함수
(Lagrange function)

라그랑주 함수 예제

$$\frac{\partial L_d(\lambda,\alpha)}{\partial\lambda} = -2\lambda-\alpha+1=0 \qquad \frac{\partial L_d(\lambda,\alpha)}{\partial\alpha} = -\alpha-\lambda+\frac{3}{4}=0$$

$$\boxed{\lambda=\frac{1}{4} \qquad \alpha=\frac{1}{2}} \qquad \alpha\left(\frac{3}{4}-x_2\right)=0 \qquad x_2=\frac{3}{4} \qquad 1-x_1-x_2=0 \qquad x_1=\frac{1}{4}$$

$$\therefore\ x_1=\frac{1}{4} \quad x_2=\frac{3}{4}$$

구한 람다와 베타로

$$\boxed{\alpha\left(\frac{3}{4}-x_2\right)=0} \quad \boxed{x_2=\frac{3}{4}} \quad \boxed{1-x_1-x_2=0} \quad \boxed{x_1=\frac{1}{4}}$$

등식 제약조건 check

상보적 여유성 check

라그랑주 함수 (Lagrange function)

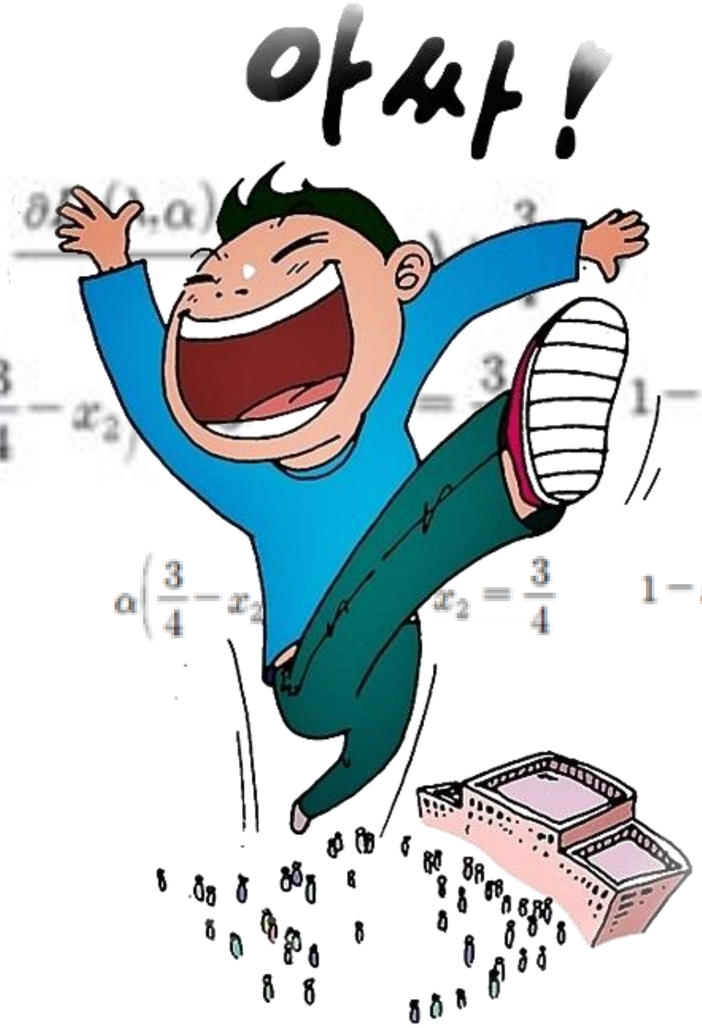
라그랑주 함수 예제

$$\frac{\partial L_d(\lambda,\alpha)}{\partial \lambda} = -2\lambda - \alpha + 1 = 0$$

$$\lambda = \frac{1}{4} \quad \alpha = \frac{1}{2}$$

$$\boxed{\therefore\ x_1 = \frac{1}{4} \quad x_2 = \frac{3}{4}}$$

최적해 구했다 만쉐!!



$$1 - x_1 - x_2 = 0 \quad x_1 = \frac{1}{4}$$

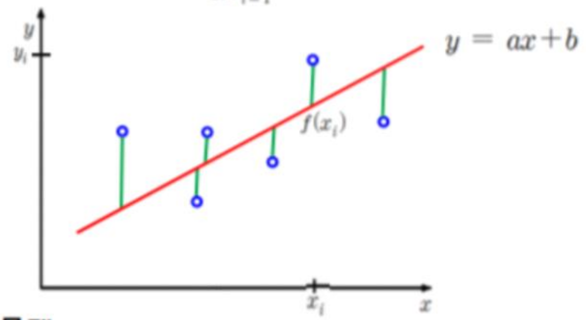
$$1 - x_1 - x_2 = 0 \quad x_1 = \frac{1}{4}$$

참고로 함수 최적화는 이런게 더 있다고 하죠

❖ **회귀(regression) 문제의 최적 함수**

- 주어진 **데이터**를 가장 잘 **근사**(近似, approximation)하는 **함수**
- **최소 평균제곱법**(least mean square method)
  - **오차 함수**(error function) 또는 **에너지 함수**(energy function)를 최소로 하는 함수를 찾는 방법

$$E = \frac{1}{2N}\sum_{i=1}^{N}(y_i - f(x_i))^2$$



  - **최적화 문제**

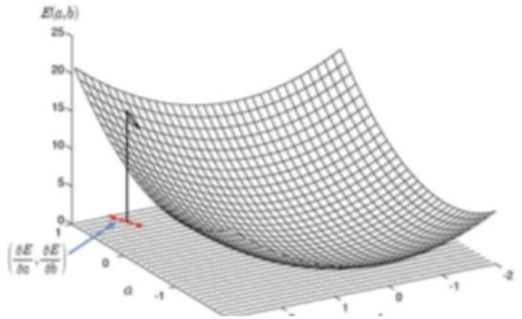
Find a, b which minimizes $\min_{a,b} \frac{1}{2N}\sum_{i=1}^{N}(y_i - ax_i - b)^2$

❖ **최대 경사법**(gradient descent method, **경사 하강법**)

- 함수의 **최소값 위치**를 찾는 문제에서 오차 함수의 **그레디언트**(gradient) 반대 방향으로 조금씩 움직여 가며 최적의 파라미터를 찾으려는 방법
- **그레디언트**
  - 각 파라미터에 대해 편미분한 벡터 $\left(\frac{\partial E}{\partial a}, \frac{\partial E}{\partial b}\right)$
- 데이터의 입력과 출력을 이용하여 각 파라미터에 대한 그레디언트를 계산하여 **파라미터**를 **반복적**으로 **조금씩 조정**

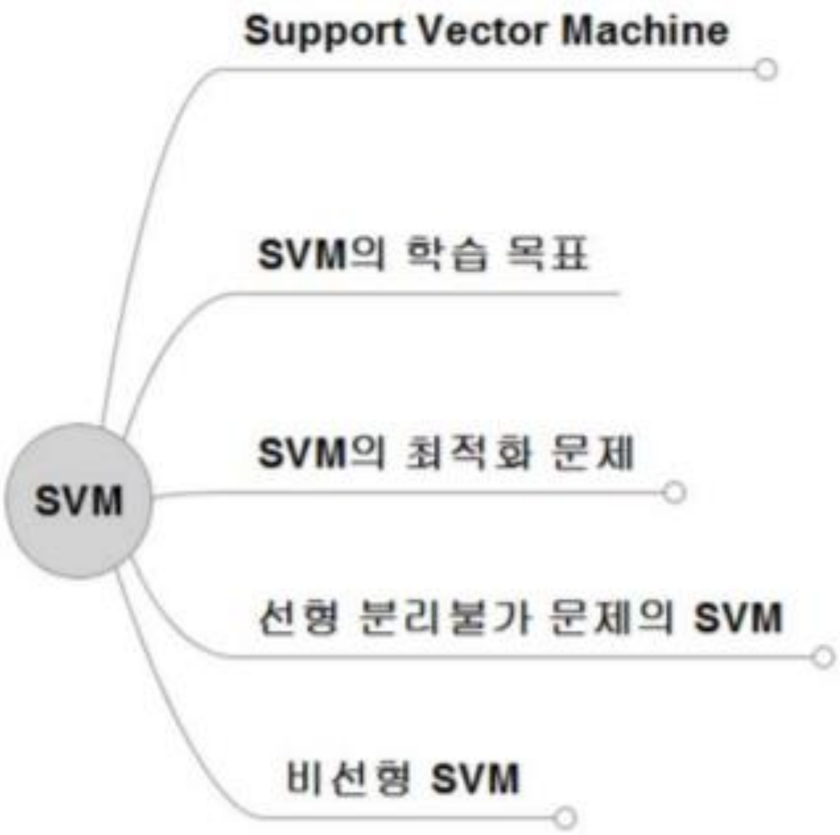
$$a^{(t+1)} = a^{(t)} - \eta\frac{\partial E}{\partial a}$$

$a^{(t)}$: 현 시점에서 파라미터 a의 값
η : 학습율 (0 < η < 1)



서포트 벡터 머신
(SVM : Support Vector Machine)

서포트 벡터 머신
(SVM : Support Vector Machine)



서포트 벡터 머신
(SVM : Support Vector Machine)

Support Vector Machine

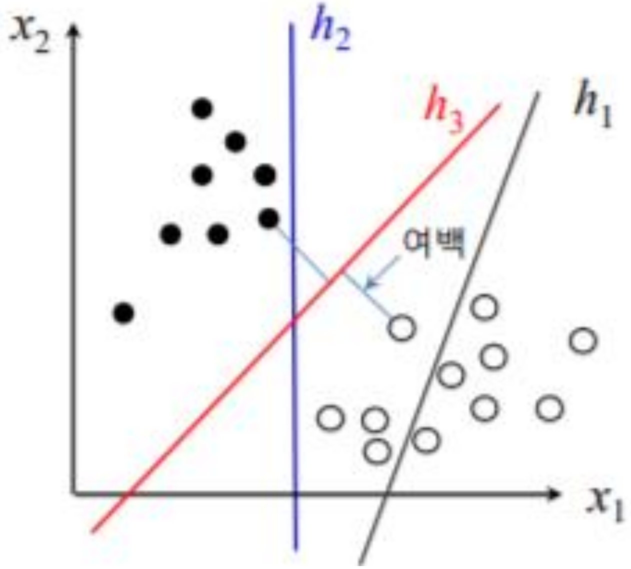
: 분류 오차를 줄이면서
동시에 여백(margin : 결정 경계와 가장 가까이에 있는 학습 데이터가지의 거리)을
최대로 하는 결정 경계(decision boundary)를 찾는 이진분류기(binary classifier)

그림으로 살펴봅시다!

서포트 벡터 머신
(SVM : Support Vector Machine)

Support Vector Machine

: 분류 오차를 줄이면서
동시에 여백(margin : 결정 경계와 가장 가까이에 있는 학습 데이터가지의 거리)을
최대로 하는 결정 경계(decision boundary)를 찾는 이진분류기(binary classifier)
*서포트 벡터(support vector) : 결정경계로부터 가장 가까이에 있는 학습 데이터들



두 데이터를 나누기에
직선 h1과 h2, h3 중에 어떤 게 제일 나은가요?

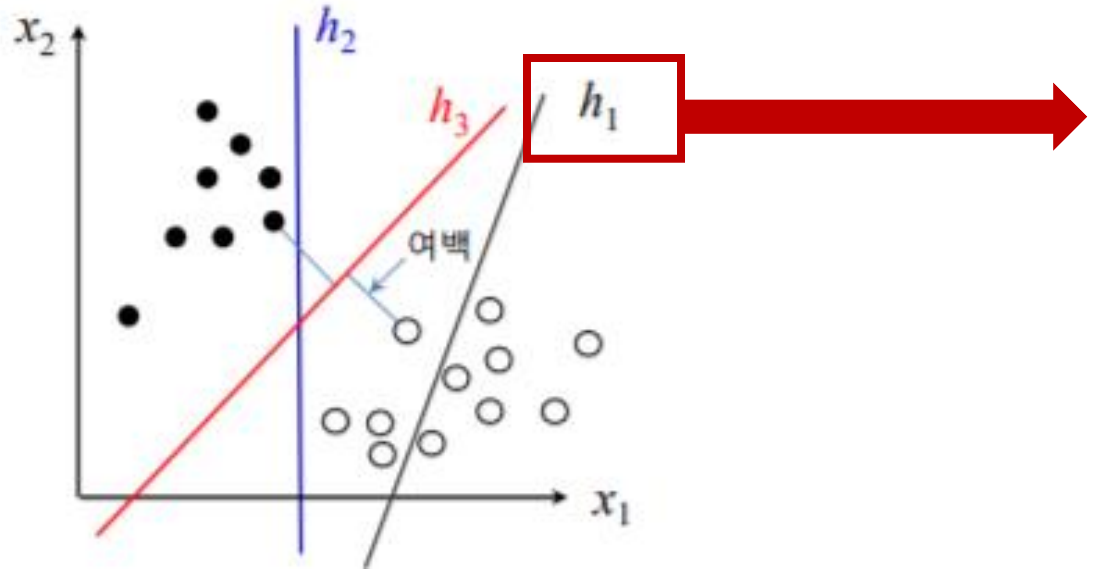
그것을 고른 이유는??

서포트 벡터 머신
(SVM : Support Vector Machine)

Support Vector Machine

: 분류 오차를 줄이면서
동시에 여백(margin : 결정 경계와 가장 가까이에 있는 학습 데이터가지의 거리)을
최대로 하는 결정 경계(decision boundary)를 찾는 이진분류기(binary classifier)

*서포트 벡터(support vector) : 결정경계로부터 가장 가까이에 있는 학습 데이터들



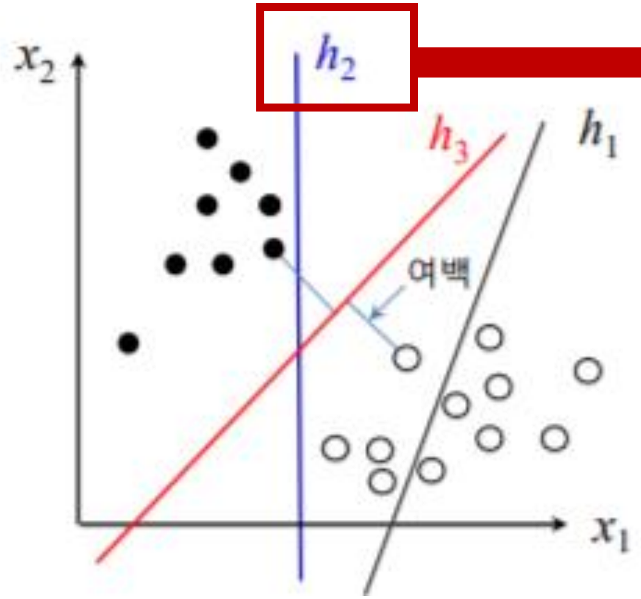
분류 모델로 쓰기에는
오류가 넘나 발생하는군요
넌 탈락

서포트 벡터 머신
(SVM : Support Vector Machine)

Support Vector Machine

: 분류 오차를 줄이면서
동시에 여백(margin : 결정 경계와 가장 가까이에 있는 학습 데이터가지의 거리)을
최대로 하는 결정 경계(decision boundary)를 찾는 이진분류기(binary classifier)

*서포트 벡터(support vector) : 결정경계로부터 가장 가까이에 있는 학습 데이터들



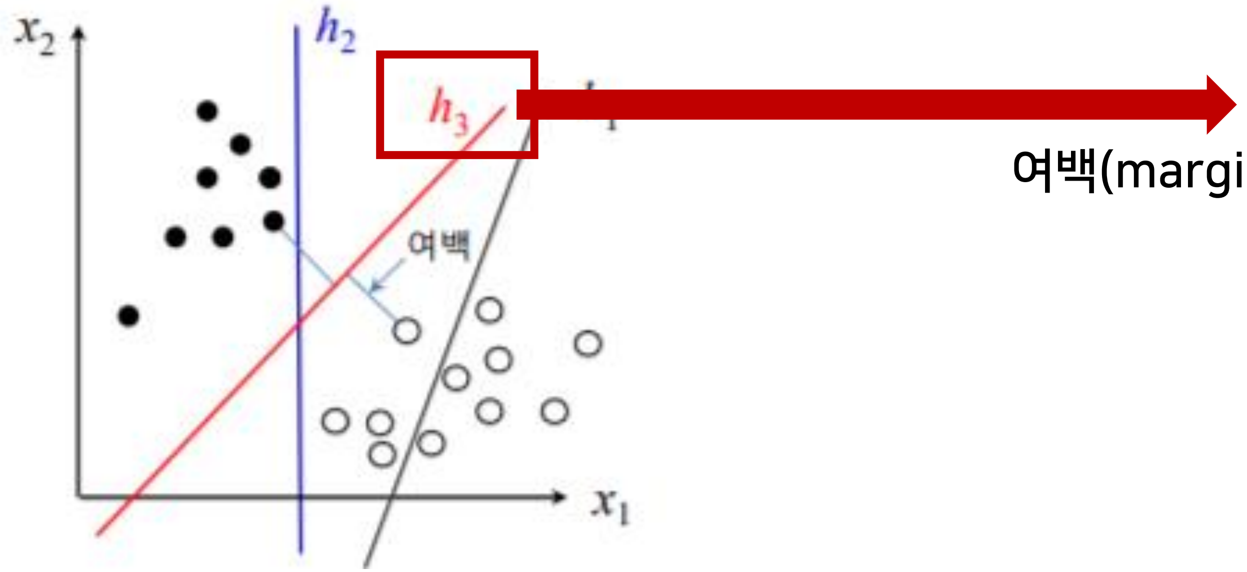
현재 데이터를 잘 분류하지만
General하지 못하다는 느낌이랄까요
직선이 조금만 기울어져도 오류가 발생하겠는걸요
test set에서 모델 성능을 보장할 수 없으므로 탈락시켜드립니다

서포트 벡터 머신
(SVM : Support Vector Machine)

Support Vector Machine

: 분류 오차를 줄이면서
동시에 여백(margin : 결정 경계와 가장 가까이에 있는 학습 데이터가지의 거리)을
최대로 하는 결정 경계(decision boundary)를 찾는 이진분류기(binary classifier)

*서포트 벡터(support vector) : 결정경계로부터 가장 가까이에 있는 학습 데이터들



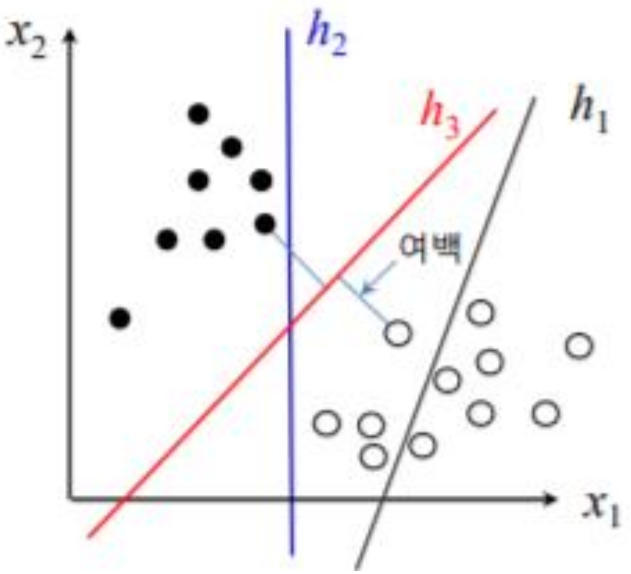
채택되셨습니다
여백(margin)이 커서 분류를 잘 할 것 같네요

서포트 벡터 머신
(SVM : Support Vector Machine)

Support Vector Machine

여백(margin)을 최대화 하는 분류기 찾기

*여백(margine) : 결정 경계와 가장 가까이에 있는 학습데이터까지의 **거리**



서포트 벡터 머신
(SVM : Support Vector Machine)

초평면 기하학

초평면(hyperplane)
: 4차원이상의 공간에서 $\boxed{\text{선형 방정식}}$으로 표현되는 결정경계

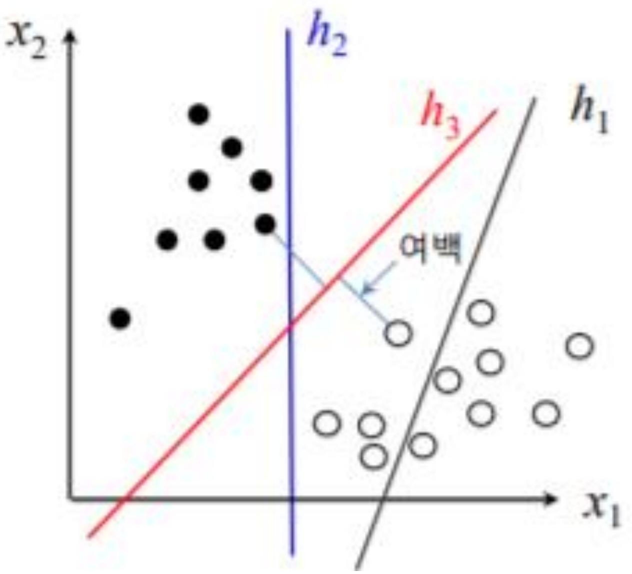
2차원 : 직선
3차원 : 평면
4차원이상 : 초평면(hyperplane)

서포트 벡터 머신
(SVM : Support Vector Machine)

초평면 기하학

초평면(hyperplane)
: 4차원이상의 공간에서 선형 방정식으로 표현되는 결정경계

2차원 : 직선
3차원 : 평면
4차원이상 : 초평면(hyperplane)



여백(margin)을 최대화 하는 hyperplane을 찾는 것이 목표!
: hyperplane을 나타내는 식을 찾아내자!

서포트 벡터 머신
(SVM : Support Vector Machine)

초평면 기하학

초평면(hyperplane)
: 4차원이상의 공간에서 선형 방정식으로 표현되는 결정경계 (decision boundary)

식으로 표현

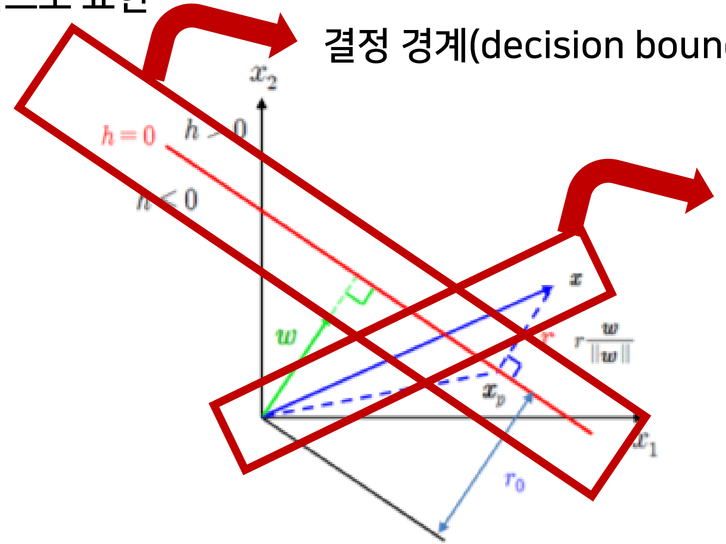
$$\boxed{\text{입력 데이터 } \boldsymbol{x} = [x_1\ x_2\ \cdots\ x_d]^\top}$$

입력(input)

$$\boxed{h(\boldsymbol{x}) = w_1x_1 + w_2x_2 + \cdots + w_dx_d + b = \boldsymbol{w}^\top\boldsymbol{x} + b = 0}$$

결정 경계(decision boundary)

그림으로 표현



결정 경계(decision boundary) : h(x) = 0으로 표현

입력(x) : 벡터로 공간상에 표시

서포트 벡터 머신
(SVM : Support Vector Machine)

초평면 기하학

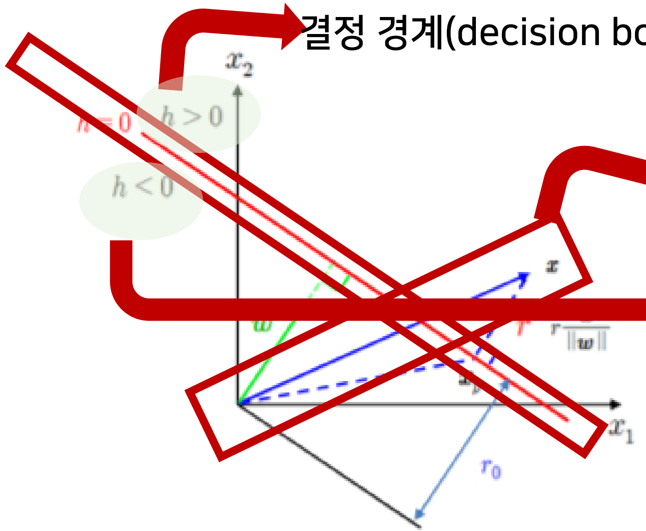
초평면(hyperplane)
: 4차원이상의 공간에서 선형 방정식으로 표현되는 결정경계 (decision boundary)

식으로 표현

입력 데이터 $\boldsymbol{x} = [x_1\ x_2\ \cdots\ x_d]^\top$ 입력(input)

$h(\boldsymbol{x}) = w_1x_1 + w_2x_2 + \cdots + w_dx_d + b = \boldsymbol{w}^\top\boldsymbol{x} + b = 0$ 결정 경계(decision boundary)

그림으로 표현



결정 경계(decision boundary) : h(x) = 0을 기준으로 한쪽은 0보다 커짐

입력(x) : 벡터로 공간상에 표시

결정 경계(decision boundary) : h(x) = 0을 기준으로 한쪽은 0보다 작아짐

서포트 벡터 머신
(SVM : Support Vector Machine)

초평면 기하학

초평면(hyperplane)
: 4차원이상의 공간에서 선형 방정식으로 표현되는 결정경계 (decision boundary)

식으로 표현

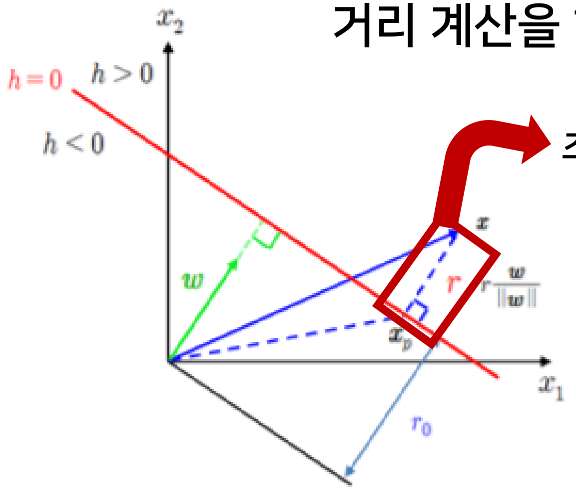
$$\boxed{\text{입력 데이터 } \boldsymbol{x} = [x_1\ x_2\ \cdots\ x_d]^\top}$$

입력(input)

$$\boxed{h(\boldsymbol{x}) = w_1x_1 + w_2x_2 + \cdots + w_dx_d + b = \boldsymbol{w}^\top\boldsymbol{x} + b = 0}$$

결정 경계(decision boundary)

그림으로 표현



거리 계산을 해야하므로 초평면과 x의 어떤 포인트 간의 거리를 계산해야한다!
(평면과 점과의 거리를 계산해야함)

초평면과 x 한 점과의 거리인 r 계산 필수

서포트 벡터 머신
(SVM : Support Vector Machine)

초평면 기하학

초평면(hyperplane)
: 4차원이상의 공간에서 선형 방정식으로 표현되는 결정경계 (decision boundary)

식으로 표현

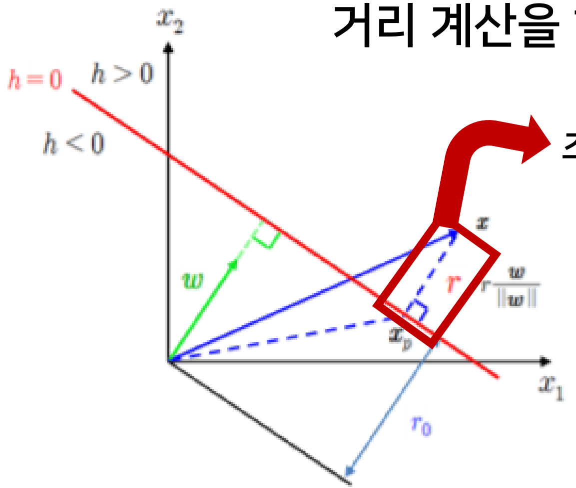
입력 데이터 $\boldsymbol{x} = [x_1\ x_2\ \cdots\ x_d]^\top$

입력(input) : **벡터!**

$$h(\boldsymbol{x}) = w_1x_1 + w_2x_2 + \cdots + w_dx_d + b = \boldsymbol{w}^\top\boldsymbol{x} + b = 0$$

결정 경계(decision boundary)
:평면의 식!, wT(가중치)와 x(입력)와 내적한 값에 b(절편)를 더한 값

그림으로 표현



거리 계산을 해야하므로 초평면과 x의 어떤 포인트 간의 거리를 계산해야한다!
(평면과 점과의 거리를 계산해야함)

초평면과 x 한 점과의 거리인 r 계산 필수

서포트 벡터 머신
(SVM : Support Vector Machine)

초평면 기하학

초평면(hyperplane)
: 4차원이상의 공간에서 선형 방정식으로 표현되는 결정경계 (decision boundary)

식으로 표현

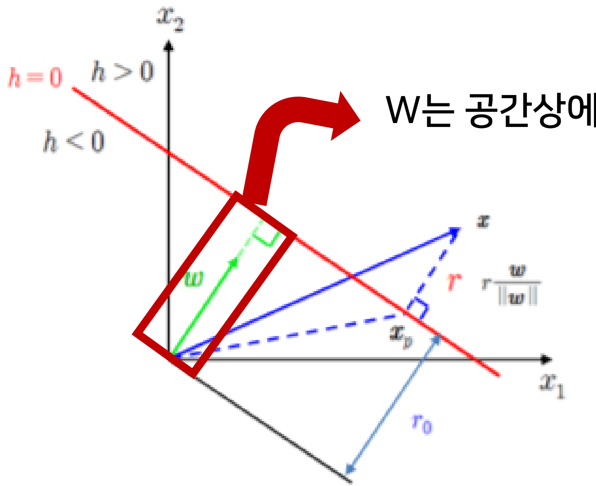
입력 데이터 $\boldsymbol{x} = [x_1\ x_2\ \cdots\ x_d]^\top$

입력(input) : **벡터!**

$$h(\boldsymbol{x}) = w_1x_1 + w_2x_2 + \cdots + w_dx_d + b = \boldsymbol{w}^\top\boldsymbol{x} + b = 0$$

결정 경계(decision boundary)
:평면의 식!, wT(가중치)와 x(입력)와 내적한 값에 b(절편)를 더한 값

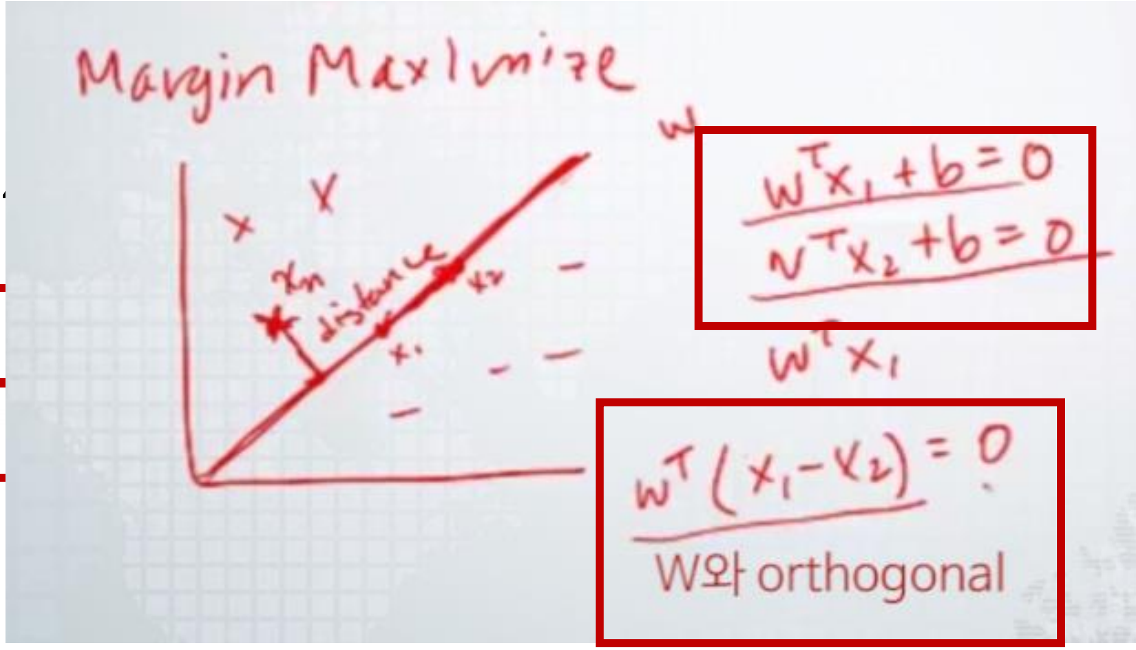
그림으로 표현



W는 공간상에서 h(x)의 법선벡터가 된다

서포트 벡터 머신
(SVM : Support Vector Machine)

초평면 기하학



: ion boundary)

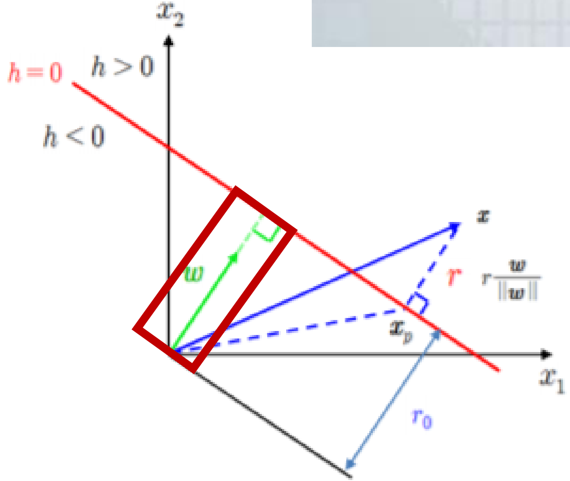
식으로 표현

입력 데이터 $\boldsymbol{x} = [x_1\ x_2$

$h(\boldsymbol{x}) = w_1 x_1 + w_2 x_2 + \cdots$

(입력)와 내적한 값에

그림으로 표현



W와 X는 수직(orthogonal)하다

→(Wt+b = 0) 수식에 의하면

서포트 벡터 머신
(SVM : Support Vector Machine)

초평면 기하학

초평면(hyperplane)
: 4차원이상의 공간에서 선형 방정식으로 표현되는 결정경계 (decision boundary)

식으로 표현

입력 데이터 $\boldsymbol{x} = [x_1\ x_2\ \cdots\ x_d]^\top$ 입력(input)

$h(\boldsymbol{x}) = w_1x_1 + w_2x_2 + \cdots + w_dx_d + b = \boldsymbol{w}^\top\boldsymbol{x} + b = 0$ 결정 경계

$$\boldsymbol{x} = \boldsymbol{x}_p + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$$

법선벡터인 w를
크기로 나눠 단위벡터로
만든 후 r을 곱해준다
단위벡터 w는 크기가 1이고
R과 방향이 동일하므로
이렇게 바꾸는 것 가능

Wt 곱해줌

$$\boldsymbol{w}^\top\boldsymbol{x} = \boldsymbol{w}^\top\boldsymbol{x}_p + r\frac{\boldsymbol{w}^\top\boldsymbol{w}}{\|\boldsymbol{w}\|}$$

$$\boldsymbol{w}^\top\boldsymbol{x} = \boldsymbol{w}^\top\boldsymbol{x}_p + r\|\boldsymbol{w}\|$$

b 더해줌

$$\boldsymbol{w}^\top\boldsymbol{x} + b = \boldsymbol{w}^\top\boldsymbol{x}_p + b + r\|\boldsymbol{w}\|$$

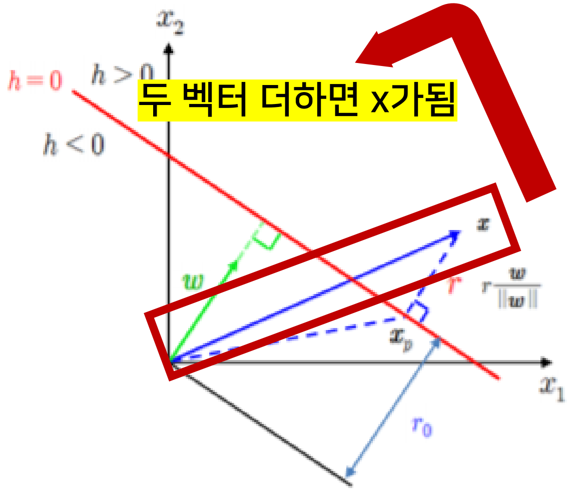
r 구함

$$h(\boldsymbol{x}) = r\|\boldsymbol{w}\| \qquad r = \frac{h(\boldsymbol{x})}{\|\boldsymbol{w}\|}$$

원점에서의 거리

$$r_0 = \frac{h(\boldsymbol{0})}{\|\boldsymbol{w}\|} = \frac{b}{\|\boldsymbol{w}\|}$$

그림으로 표현



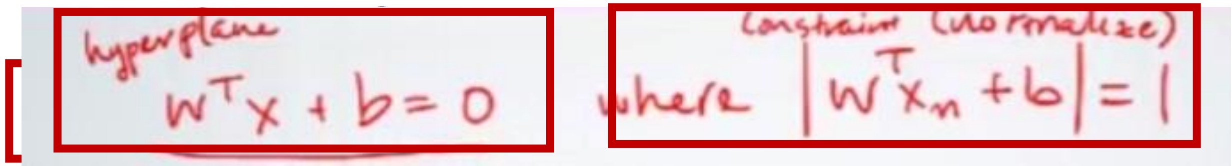
서포트 벡터 머신
(SVM : Support Vector Machine)

초평면 기하학

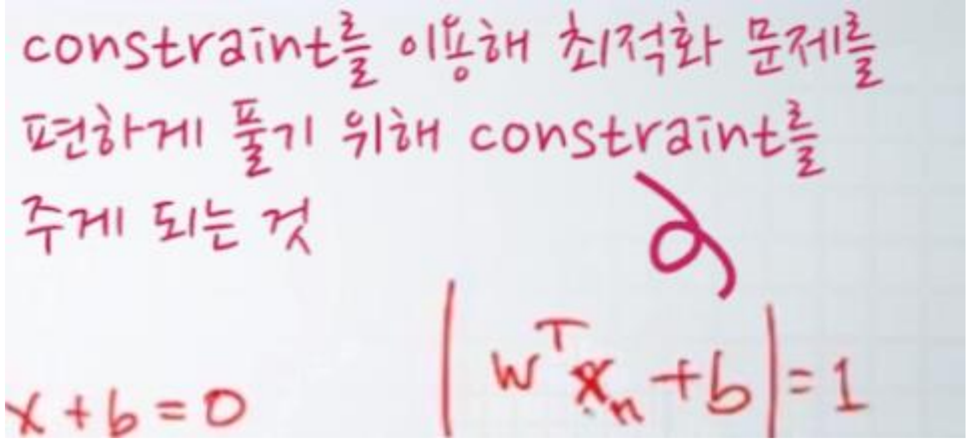
계산을 용이하기 위해 이런 normalize된 식과 constraint를 정의하고 풀어보자!

초평면(hyper plane)

제약조건 constraint



식의 절대값이 1이 넘지않는다는 제약조건

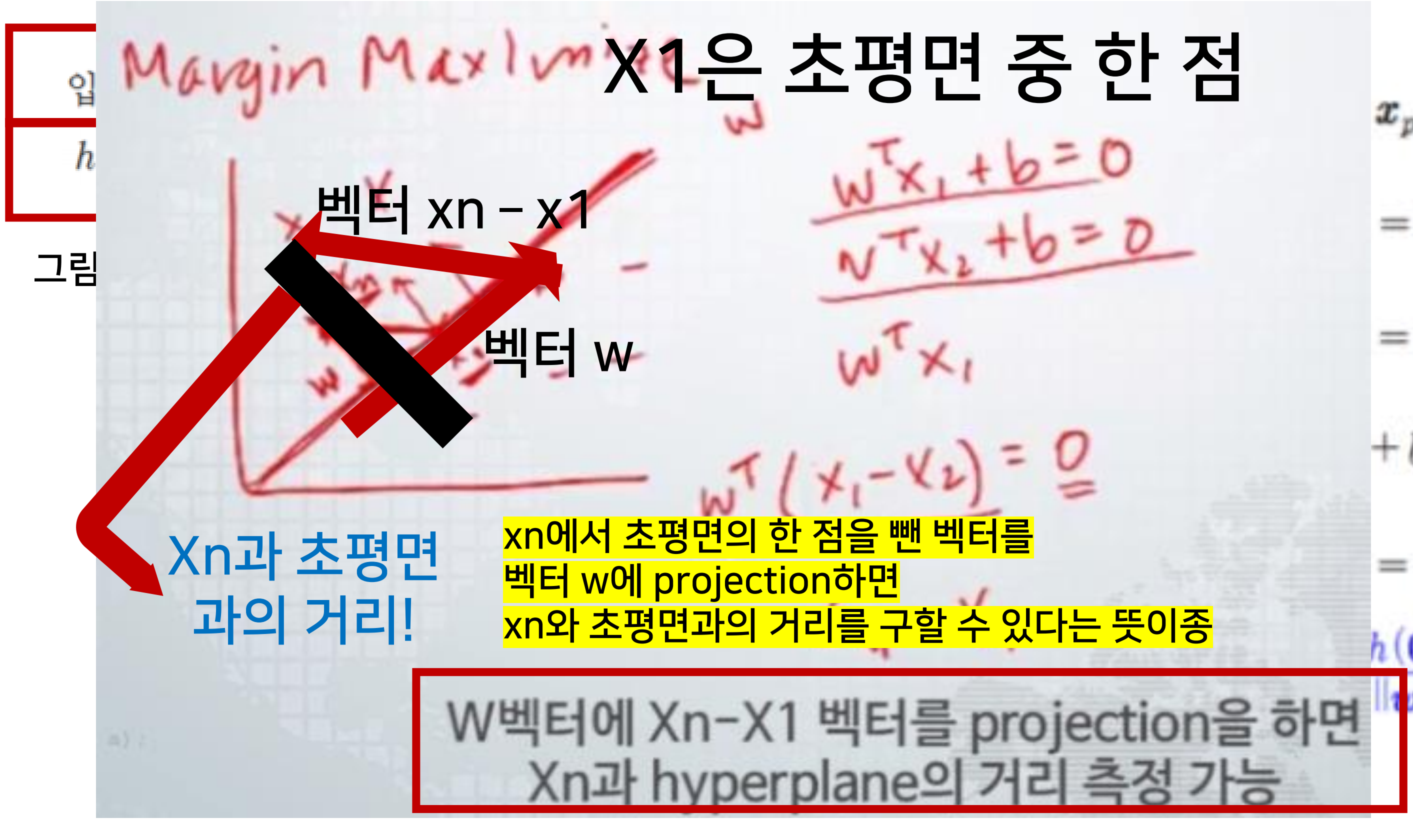


서포트 벡터 머신
(SVM : Support Vector Machine)

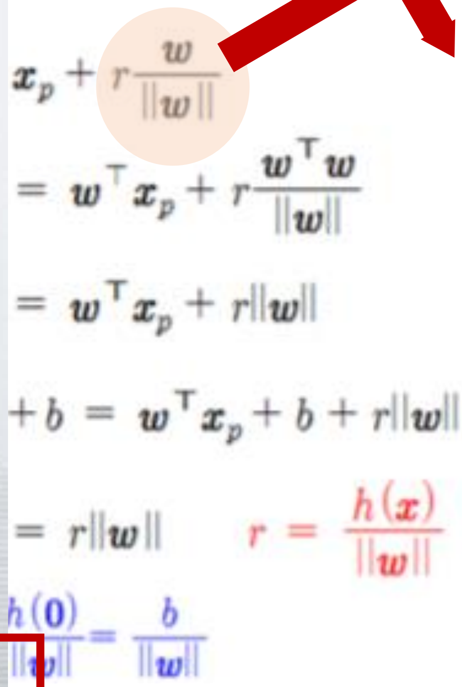
초평면 기하학

초평면(hyperplane)
: 4차원이상의 공간에서 선형 방정식으로 표현되는 결정경계 (decision boundary)

식으로 표현



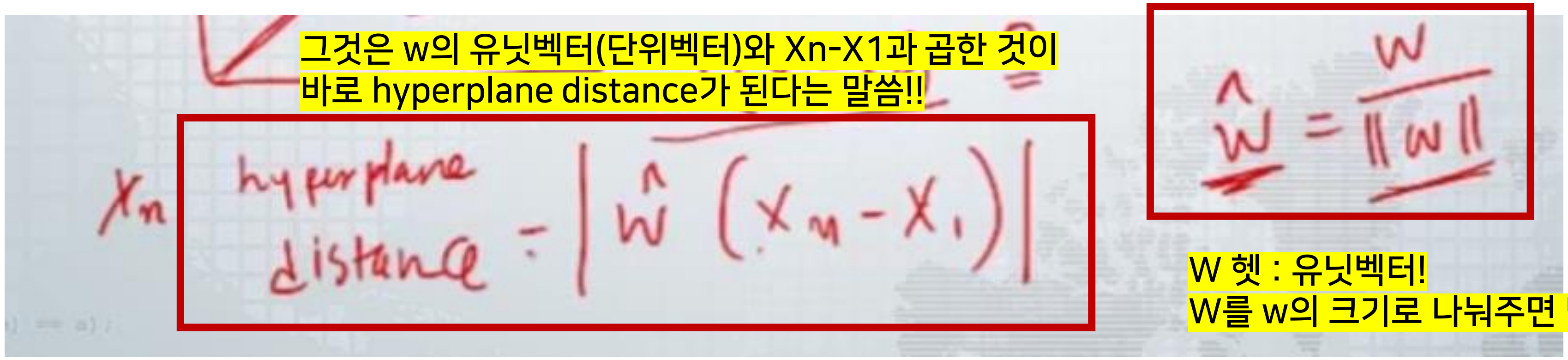
그림



법선벡터인 w를
크기로 나눠 단위벡터로
만든 후 scalar r만큼의 크기를
곱해준다

그거슨 무슨말이냐!

그것은 w의 유닛벡터(단위벡터)와 Xn-X1과 곱한 것이 바로 hyperplane distance가 된다는 말씀!!

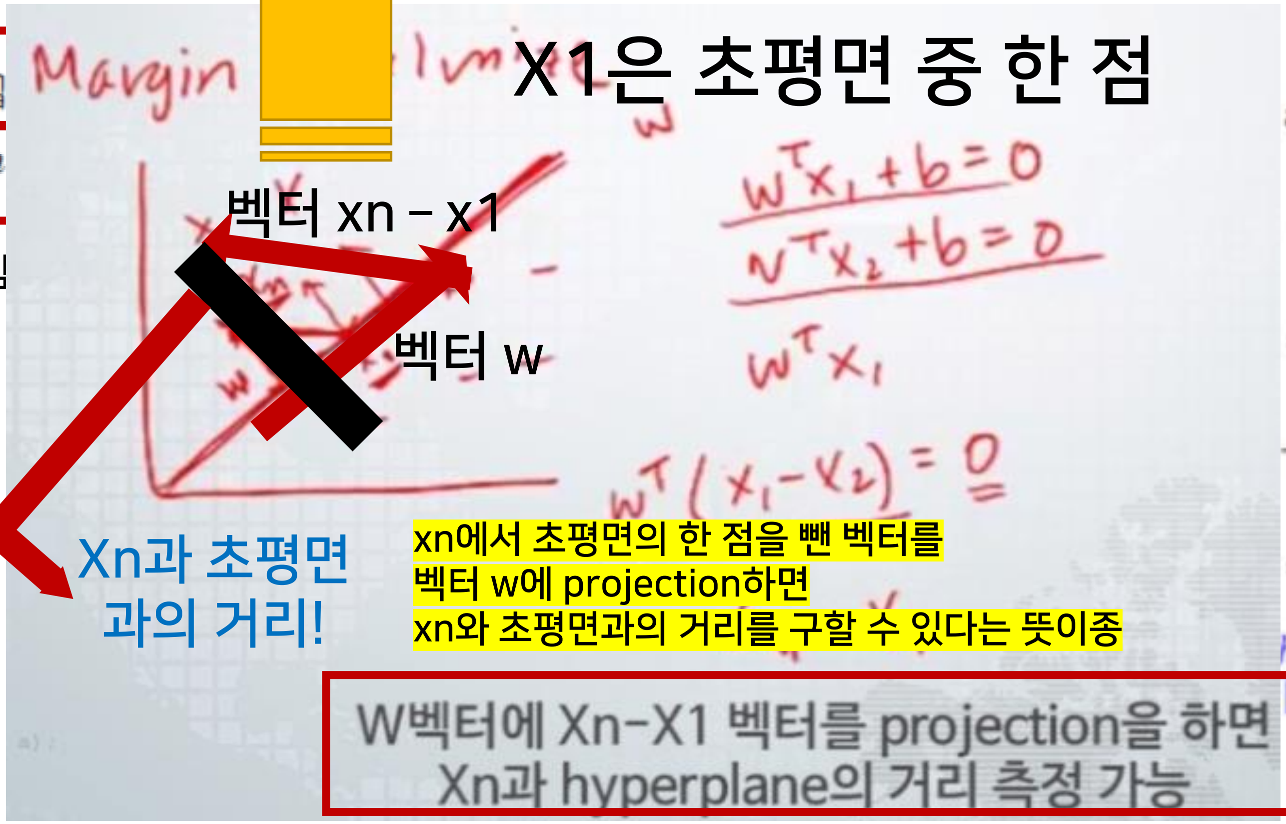


W 헷 : 유닛벡터!
W를 w의 크기로 나눠주면 댐

초평면(hyperplane)
4차원이상의 공간에서 선형 방정식으로 표현되는 결정경계 (decision boundary)

식으로 표현

그림



X1은 초평면 중 한 점

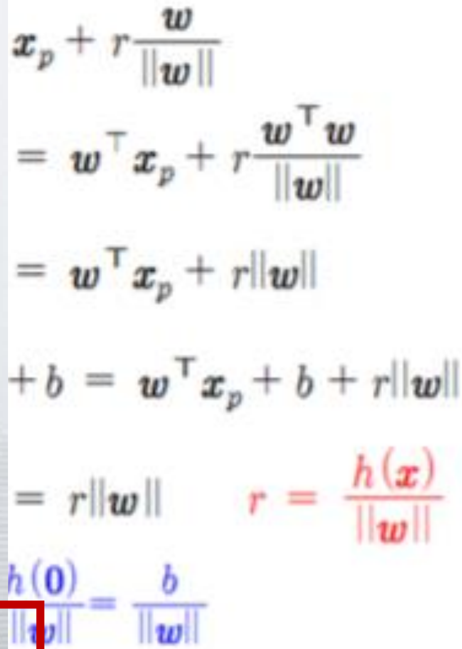
벡터 xn – x1

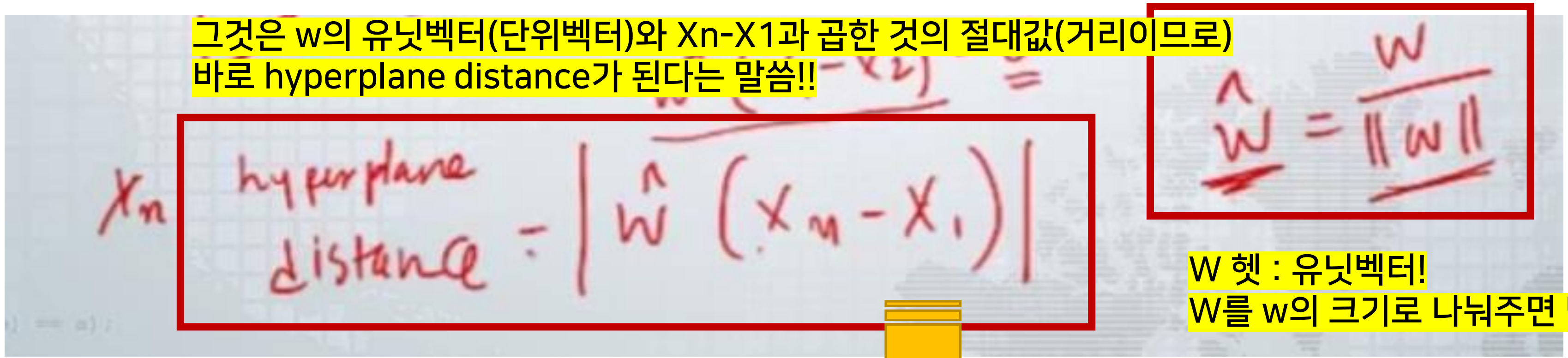
벡터 w

Xn과 초평면과의 거리!

xn에서 초평면의 한 점을 뺀 벡터를
벡터 w에 projection하면
xn와 초평면과의 거리를 구할 수 있다는 뜻이죵

W벡터에 Xn-X1 벡터를 projection을 하면
Xn과 hyperplane의 거리 측정 가능





그것은 w의 유닛벡터(단위벡터)와 Xn-X1과 곱한 것의 절대값(거리이므로)
바로 hyperplane distance가 된다는 말씀!!

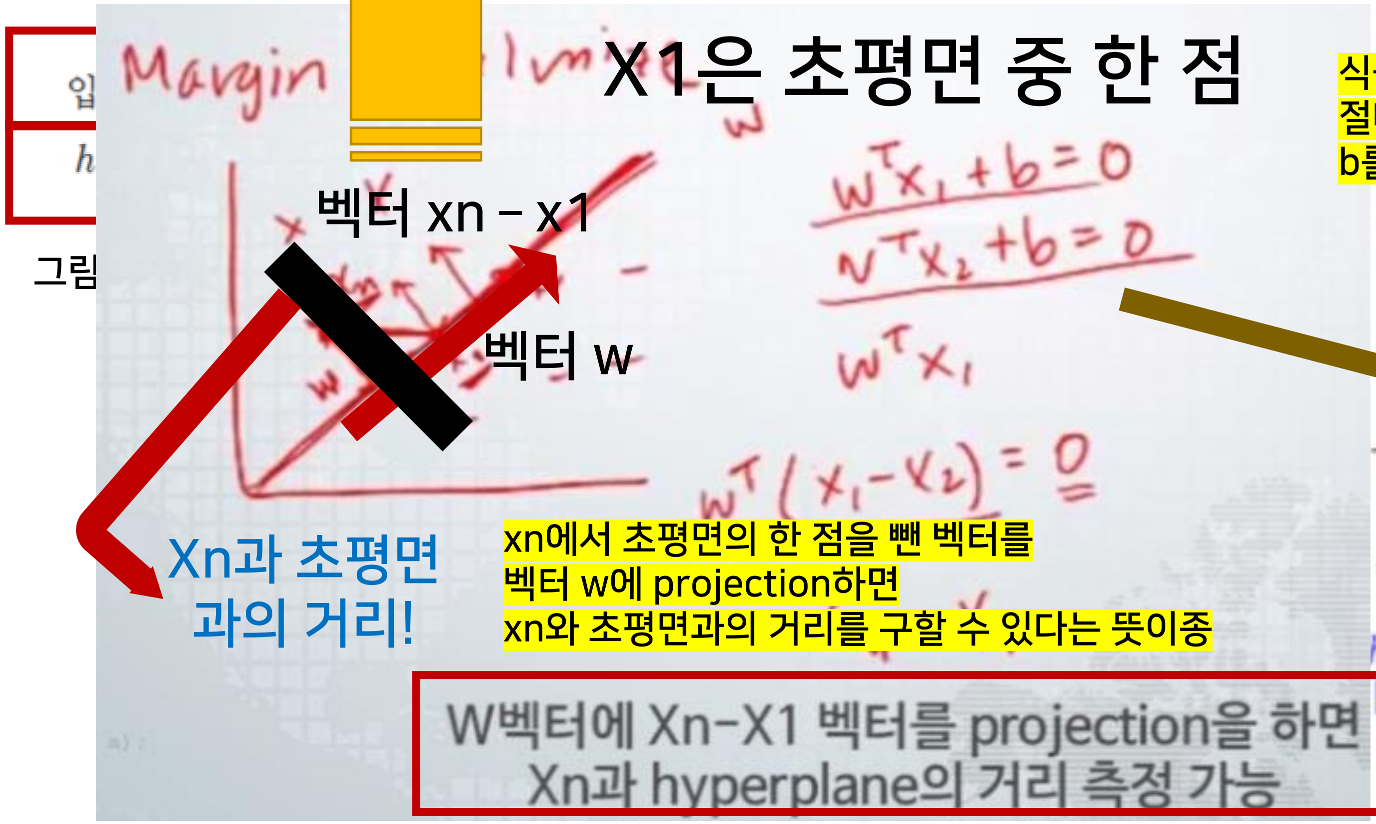
W 헷 : 유닛벡터!
W를 w의 크기로 나눠주면 댐

그거슨 무슨말이냐!

초평면(hyperplane)
4차원이상의 공간에서 선형 방정식으로 표현되는 결정경계 (decision boundary)

식으로 표현

그림



X1은 초평면 중 한 점

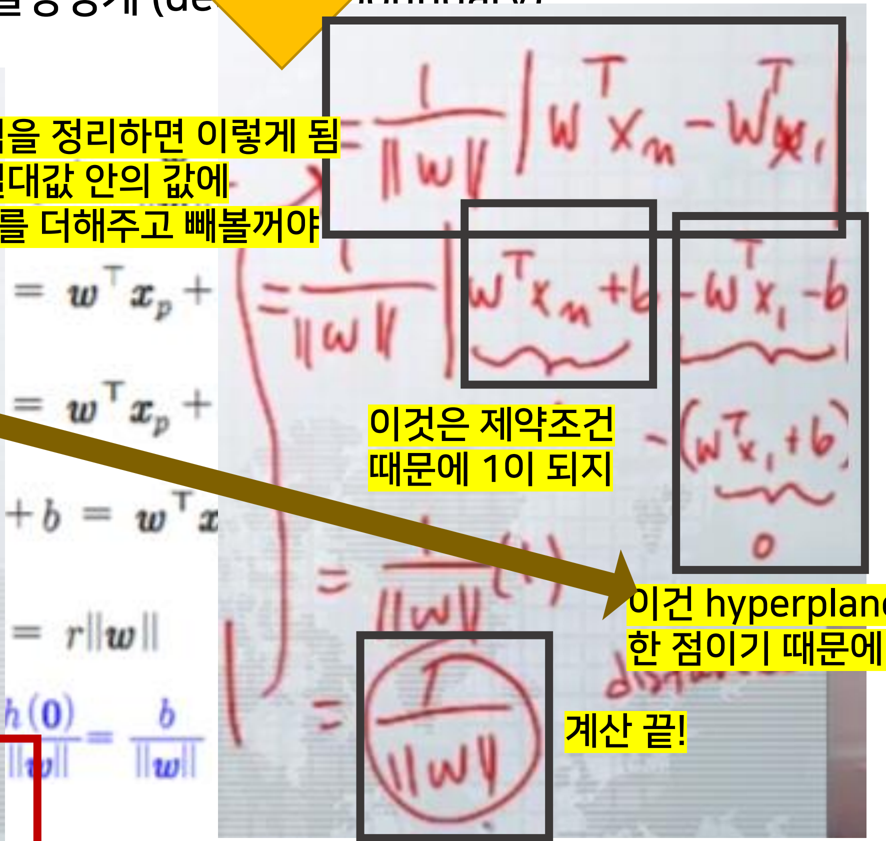
벡터 xn – x1

벡터 w

Xn과 초평면과의 거리!

xn에서 초평면의 한 점을 뺀 벡터를
벡터 w에 projection하면
xn와 초평면과의 거리를 구할 수 있다는 뜻이죵

W벡터에 Xn-X1 벡터를 projection을 하면
Xn과 hyperplane의 거리 측정 가능



식을 정리하면 이렇게 됨
절대값 안의 값에
b를 더해주고 빼볼꺼야

이것은 제약조건
때문에 1이 되지

이건 hyperplane의
한 점이기 때문에 0이야

계산 끝!

서포트 벡터 머신
(SVM : Support Vector Machine)

SVM의 목표

분류를 잘하자

서포트 벡터 머신
(SVM : Support Vector Machine)

SVM의 목표
분류를 잘하자

분류를 위한 초평면의 만족조건

❶ $t_i h(\boldsymbol{x}_i) \geq 1, \quad i = 1, \cdots, N$

분류를 잘하면 서포트벡터는 1이거나 -1이어야함

❷ 서포트 벡터와의 거리, 즉 여백을 최대로 한다.

여백은 당연히 최대

식

조건 ❶ 서포트 벡터 $\boldsymbol{x}'$에서의 $|h(\boldsymbol{x}')| = 1$

$h(\boldsymbol{x}) > 0$인 공간에 $t_i = 1$

$h(\boldsymbol{x}) < 0$인 공간에 $t_i = -1$

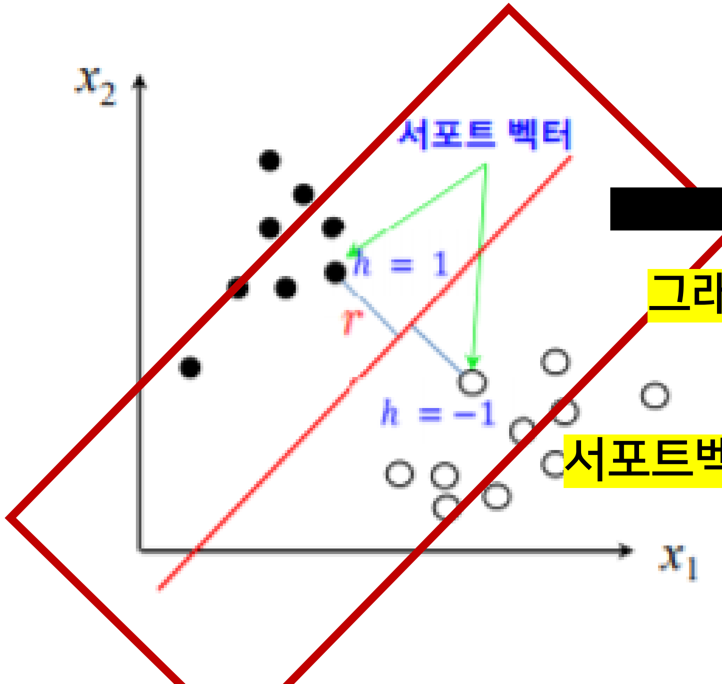
조건 ❷ $r = \dfrac{h(\boldsymbol{x})}{\|\boldsymbol{w}\|}$ 서포트 벡터에 대해서는 $h(\boldsymbol{x}) = 1$

$r = \dfrac{1}{\|\boldsymbol{w}\|}$

서포트 벡터 머신
(SVM : Support Vector Machine)

SVM의 목표
분류를 잘하자

분류를 위한 초평면의 만족조건



❶ $t_i h(\boldsymbol{x}_i) \geq 1, \quad i = 1, \cdots, N$

❷ 서포트 벡터와의 거리, 즉 여백을 최대로 한다.

분류를 잘하면 서포트벡터는 1이거나 -1이어야함

여백은 당연히 최대

식

조건 ❶ 서포트 벡터 $\boldsymbol{x}'$에서의 $|h(\boldsymbol{x}')| = 1$

$h(\boldsymbol{x}) > 0$인 공간에 $t_i = 1$

$h(\boldsymbol{x}) < 0$인 공간에 $t_i = -1$

그래서 제약조건 생김

$r = \frac{h(\boldsymbol{x})}{\|\boldsymbol{w}\|}$ 서포트 벡터에 대해서는 $h(\boldsymbol{x})=1$

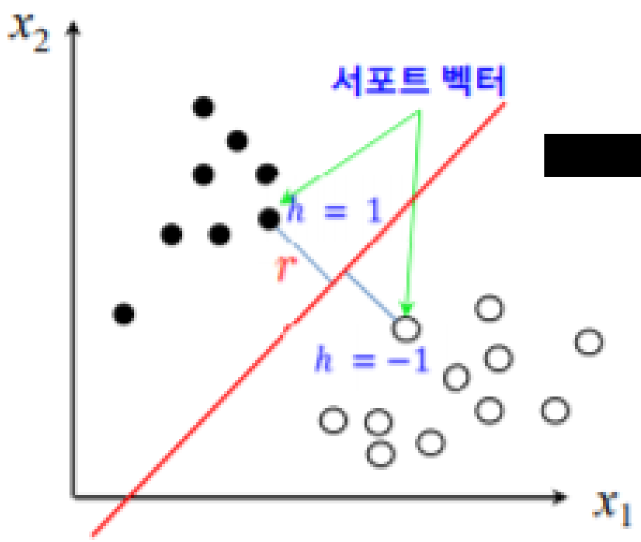
$r = \frac{1}{\|\boldsymbol{w}\|}$

서포트벡터가 h=1이 되도록 하는 함수식(결정경계)를 찾는 것이 목표!

서포트 벡터 머신
(SVM : Support Vector Machine)

SVM의 목표
분류를 잘하자

분류를 위한 초평면의 만족조건



식

❶ $t_i h(\boldsymbol{x}_i) \ge 1, \quad i = 1, \cdots, N$

❷ 서포트 벡터와의 거리, 즉 여백을 최대로 한다.

분류를 잘하면 서포트벡터는 1이거나 -1이어야함

여백은 당연히 최대

조건 ❶ 서포트 벡터 $\boldsymbol{x}'$에서의 $|h(\boldsymbol{x}')| = 1$

$h(\boldsymbol{x}) > 0$인 공간에 $t_i = 1$

$h(\boldsymbol{x}) < 0$인 공간에 $t_i = -1$

조건 ❷ $r = \frac{h(\boldsymbol{x})}{\|\boldsymbol{w}\|}$ 서포트 벡터에 대해서는 $h(\boldsymbol{x}) = 1$

$r = \frac{1}{\|\boldsymbol{w}\|}$

서포트벡터가 h=1이 되면 w의 크기를
minimize(최소화)하는 문제로 바뀐다

서포트 벡터 머신
(SVM : Support Vector Machine)

SVM의 목표
분류를 잘하자

제약조건 최적화

$$\boxed{\text{Find } \boldsymbol{x} \text{ which minimizes } f(\boldsymbol{x})}$$

$$\boxed{\begin{aligned} \text{subject to } g(\boldsymbol{x}) &= 0 \\ h(\boldsymbol{x}) &\le 0 \end{aligned}}$$

목적함수

제약조건

- **라그랑주 함수**

$$L(\boldsymbol{x}, \alpha, \lambda) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{x}) + \alpha h(\boldsymbol{x})$$

$$(\alpha \ge 0)$$

하나의 식으로

서포트 벡터 머신
(SVM : Support Vector Machine)

SVM에서 제약조건 최적화

Find $\boldsymbol{w}, b$ which minimizes $J(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2$

subject to $t_i h(\boldsymbol{x}_i) \geq 1, \quad i = 1, \cdots, N$

일단 최대화문제를 최소화로 변환

Find $\boldsymbol{w}, b$ which minimizes $J(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2$

subject to $1 - t_i h(\boldsymbol{x}_i) \leq 0, \quad i = 1, \cdots, N$

라그랑주 함수로 변환

- **라그랑주 함수**

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \alpha_i (1 - t_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b))$$

$$\alpha_i \geq 0$$

서포트 벡터 머신
(SVM : Support Vector Machine)

SVM에서 제약조건 최적화

$$\text{Find } \boldsymbol{w}, b \text{ which minimizes } J(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\text{subject to } 1 - t_i h(\boldsymbol{x}_i) \le 0, \quad i = 1, \cdots, N$$

우리들의 목적과 제약

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \alpha_i (1 - t_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)) \quad \alpha_i \ge 0$$

라그랑주 함수

- 해에 대한 **KKT(Karush-Kuhn-Tucker) 조건**

① $\alpha_i \ge 0, \quad i = 1, \cdots, N$ — 라그랑주 함수에 딸려있는 조건

② $1 - t_i h(\boldsymbol{x}_i) = 1 - t_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \le 0, \quad i = 1, \cdots, N$ — 부등식 제약조건

③ $\alpha_i (1 - t_i h(\boldsymbol{x}_i)) = 0, \quad i = 1, \cdots, N$ — **상보적 여유성(complementary slackness)**

④ $\frac{\partial L}{\partial \boldsymbol{w}} = 0, \quad \frac{\partial L}{\partial b} = 0$ — 라그랑주 함수를 통해 최적해 구하기

서포트 벡터 머신
(SVM : Support Vector Machine)

SVM에서 제약조건 최적화

Find $\boldsymbol{w}, b$ which minimizes $J(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2$

subject to $1 - t_i h(\boldsymbol{x}_i) \le 0, \quad i = 1, \cdots, N$

우리들의 목적과 제약

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \alpha_i (1 - t_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)) \quad \alpha_i \ge 0$$

라그랑주 함수

- 해에 대한 **KKT(Karush-Kuhn-Tucker) 조건**

① $\alpha_i \ge 0, \quad i = 1, \cdots, N$ 라그랑주 함수에 딸려있는 조건

② $1 - t_i h(\boldsymbol{x}_i) = 1 - t_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \le 0, \quad i = 1, \cdots, N$ 부등식 제약조건

③ $\alpha_i (1 - t_i h(\boldsymbol{x}_i)) = 0, \quad i = 1, \cdots, N$ **상보적 여유성(complementary slackness)**

④ $\frac{\partial L}{\partial \boldsymbol{w}} = 0, \quad \frac{\partial L}{\partial b} = 0$ 라그랑주 함수를 통해 최적해 구하기

- **쌍대 함수(dual function)** w,와 b에 대해서 minimize

$$\tilde{L}(\boldsymbol{\alpha}) = \min_{\boldsymbol{w}, b} L(\boldsymbol{w}, b, \boldsymbol{\alpha})$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0, \quad \frac{\partial L}{\partial b} = 0$$

그러기 위해서 편미분
(아래로 볼록인 convex 함수
그러므로 편미분해서 0일때 최적)

서포트 벡터 머신
(SVM : Support Vector Machine)

SVM에서 제약조건 최적화

Find $\boldsymbol{w}, b$ which minimizes $J(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2$

subject to $1 - t_i h(\boldsymbol{x}_i) \le 0, \quad i = 1, \cdots, N$

우리들의 목적과 제약

$$\boxed{L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N}\alpha_i(1 - t_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)) \quad \alpha_i \ge 0}$$

라그랑주 함수

- 해에 대한 **KKT(Karush-Kuhn-Tucker) 조건**

① $\alpha_i \ge 0, \quad i = 1, \cdots, N$ — 라그랑주 함수에 딸려있는 조건

② $1 - t_i h(\boldsymbol{x}_i) = 1 - t_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \le 0, \quad i = 1, \cdots, N$ — 부등식 제약조건

③ $\alpha_i(1 - t_i h(\boldsymbol{x}_i)) = 0, \quad i = 1, \cdots, N$ — 상보적 여유성(complementary slacknes

④ $\frac{\partial L}{\partial \boldsymbol{w}} = 0, \quad \frac{\partial L}{\partial b} = 0$ — 라그랑주 함수를 통해 최적해 구하기

- **쌍대 함수(dual function)**

w,와 b에 대해서 minimize

$$\boxed{\tilde{L}(\boldsymbol{\alpha}) = \min_{\boldsymbol{w}, b} L(\boldsymbol{w}, b, \boldsymbol{\alpha})}$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0, \quad \frac{\partial L}{\partial b} = 0$$

그러기 위해서 편미분
(아래로 볼록인 convex 함수
그러므로 편미분해서 0일때 최적)

- $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$의 미분을 0으로 하면

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{N}\alpha_i t_i \boldsymbol{x}_i = 0$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{N}\alpha_i t_i = 0$$

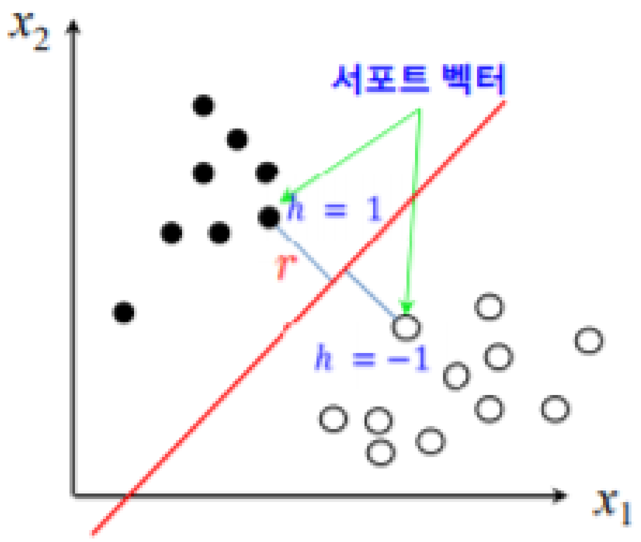
최적해

$$\boxed{\boldsymbol{w} = \sum_{i=1}^{N}\alpha_i t_i \boldsymbol{x}_i}$$

$$\boxed{\sum_{i=1}^{N}\alpha_i t_i = 0}$$

서포트 벡터 머신
(SVM : Support Vector Machine)

SVM에서 제약조건 최적화



- $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$의 미분을 0으로 하면

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{N} \alpha_i t_i \boldsymbol{x}_i = 0$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{N} \alpha_i t_i = 0$$

$$\boxed{\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i t_i \boldsymbol{x}_i}$$

$$\boxed{\sum_{i=1}^{N} \alpha_i t_i = 0}$$

이런 제약조건도 만들어졌네←

t와 x는 이미 주어진 training data

그러므로 알파만 계산하면 함수 계산 가능
(w가 법선이므로..)

서포트 벡터 머신
(SVM : Support Vector Machine)

SVM에서 제약조건 최적화

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i t_i \boldsymbol{x}_i \qquad \sum_{i=1}^{N} \alpha_i t_i = 0$$

- **초평면 함수식**

$$h(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b = \boldsymbol{w} \cdot \boldsymbol{x} + b = \sum_{i=0}^{N} \alpha_i t_i \boldsymbol{x}_i \cdot \boldsymbol{x} + b$$

W와 b가 최소가 되는 함수기 요것이다

서포트 벡터 머신 (SVM : Support Vector Machine)

SVM에서 제약조건 최적화

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i t_i \boldsymbol{x}_i \qquad \sum_{i=1}^{N} \alpha_i t_i = 0$$

- **초평면 함수식**

$$h(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b = \boldsymbol{w} \cdot \boldsymbol{x} + b = \sum_{i=0}^{N} \alpha_i t_i \boldsymbol{x}_i \cdot \boldsymbol{x} + b$$

- $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$에 넣어 전개하면

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \alpha_i (-t_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) + 1)$$

$$\tilde{L}(\boldsymbol{\alpha}) = \frac{1}{2}\left(\sum_{i=1}^{N} \alpha_i t_i \boldsymbol{x}_i\right)\left(\sum_{j=1}^{N} \alpha_j t_j \boldsymbol{x}_j\right)^\top - \sum_{i=1}^{N} \alpha_i t_i \sum_{j=1}^{N} \alpha_j t_j \boldsymbol{x}_j \boldsymbol{x}_i^\top - b \sum_{i=1}^{N} \alpha_i t_i + \sum_{i=1}^{N} \alpha_i$$

$$= -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j t_i t_j \boldsymbol{x}_i \boldsymbol{x}_j^\top + \sum_{i=1}^{N} \alpha_i$$

$$= -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j t_i t_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j + \sum_{i=1}^{N} \alpha_i$$

라그랑주 함수에 적용하면 요렇게 된다고 한다

서포트 벡터 머신 (SVM : Support Vector Machine)

SVM에서 제약조건 최적화

- $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$에 넣어 전개하면

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \alpha_i(-t_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) + 1)$$

$$\tilde{L}(\boldsymbol{\alpha}) = \frac{1}{2}\left(\sum_{i=1}^{N}\alpha_i t_i \boldsymbol{x}_i\right)\left(\sum_{j=1}^{N}\alpha_j t_j \boldsymbol{x}_j\right)^\top - \sum_{i=1}^{N}\alpha_i t_i \sum_{j=1}^{N}\alpha_j t_j \boldsymbol{x}_j \boldsymbol{x}_i^\top - b\sum_{i=1}^{N}\alpha_i t_i + \sum_{i=1}^{N}\alpha_i$$

$$= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boldsymbol{x}_i \boldsymbol{x}_j^\top + \sum_{i=1}^{N}\alpha_i$$

Dual function구해보니까 내적을 가지고 있네

$$= -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j + \sum_{i=1}^{N}\alpha_i$$

서포트 벡터 머신
(SVM : Support Vector Machine)

SVM에서 제약조건 최적화

문제가 요렇게 바꼈넴

- **라그랑주 함수**로 표현한 **본 문제**(**primal problem**)

Find $\boldsymbol{w}, b$ which minimizes $L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{N} \alpha_i (1 - t_i(\boldsymbol{w}^T \boldsymbol{x}_i + b))$

subject to $\alpha_i \ge 0, \quad i = 1, \cdots, N$

⬇

T와 x는 입력데이터이므로 알파만 결정하면 됨

W,b가 사라짐

- w와 b가 없는 **쌍대 문제**(**dual problem**)

Find $\boldsymbol{\alpha}$ which maximizes $\widetilde{L}(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j t_i t_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j + \sum_{i=1}^{N} \alpha_i$

subject to $\sum_{i=1}^{N} \alpha_i t_i = 0, \quad i = 1, \cdots, N$

$\alpha_i \ge 0, \quad i = 1, \cdots, N$

제약조건

- **문제 복잡도**
  - 데이터 개수 N에 관계
  - 데이터 차원과 무관

데이터 차원이 아닌 개수의 문제로 바뀌었다

서포트 벡터 머신
(SVM : Support Vector Machine)

SVM에서 제약조건 최적화

- **쌍대 문제**(dual problem)의 **최소화 문제 변환**

Find $\boldsymbol{\alpha}$ which maximizes $\widetilde{L}(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j + \sum_{i=1}^{N}\alpha_i$

subject to $\sum_{i=1}^{N}\alpha_i t_i = 0, \quad i = 1, \cdots, N$

$\alpha_i \geq 0, \quad i = 1, \cdots, N$

다른건 다 주어짐 알파만 찾으면 됨

알파가 두번 곱해져서 2차식이 됨

그래서 이차식 계획법(quadratic programming) 문제로 변환됨

Find $\boldsymbol{\alpha}$ which minimizes $\widetilde{L}(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j - \sum_{i=1}^{N}\alpha_i$

subject to $\sum_{i=1}^{N}\alpha_i t_i = 0, \quad i = 1, \cdots, N$

$\alpha_i \geq 0, \quad i = 1, \cdots, N$

이차식 계획법(quadratic programming) 문제

서포트 벡터 머신
(SVM : Support Vector Machine)

SVM에서 제약조건 최적화

Find $\boldsymbol{\alpha}$ which minimizes $\tilde{L}(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j - \sum_{i=1}^{N}\alpha_i$

subject to $\sum_{i=1}^{N}\alpha_i t_i = 0, \quad i = 1, \cdots, N$

$\alpha_i \geq 0, \quad i = 1, \cdots, N$

이렇게 표현 가능

$$\tilde{L}(\boldsymbol{\alpha}) = \frac{1}{2}\begin{bmatrix}\alpha_1 & \alpha_2 & \cdots & \alpha_N\end{bmatrix}\begin{bmatrix} t_1 t_1 \boldsymbol{x}_1 \cdot \boldsymbol{x}_1 & t_1 t_2 \boldsymbol{x}_1 \cdot \boldsymbol{x}_2 & \cdots & t_1 t_N \boldsymbol{x}_1 \cdot \boldsymbol{x}_N \\ t_2 t_1 \boldsymbol{x}_2 \cdot \boldsymbol{x}_1 & t_2 t_2 \boldsymbol{x}_2 \cdot \boldsymbol{x}_2 & \cdots & t_2 t_N \boldsymbol{x}_2 \cdot \boldsymbol{x}_N \\ \vdots & \vdots & \ddots & \vdots \\ t_N t_1 \boldsymbol{x}_N \cdot \boldsymbol{x}_1 & t_N t_2 \boldsymbol{x}_N \cdot \boldsymbol{x}_2 & \cdots & t_N t_N \boldsymbol{x}_N \cdot \boldsymbol{x}_N \end{bmatrix}\begin{bmatrix}\alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N\end{bmatrix} - \sum_{i=1}^{N}\alpha_i$$

$$= \frac{1}{2}\boldsymbol{x}^\top H\boldsymbol{x} - \sum_{i=1}^{N}\alpha_i$$

- 선형대수학의 quadratic problem solver 라이브러리 이용
  - MatLab/Octave의 **quadprog( )**
  - Python의 **solvers.qp( )**

이거 풀면 되는데 머리아프니까 저희는 컴퓨터 쓰죠뭐

서포트 벡터 머신
(SVM : Support Vector Machine)

SVM에서 제약조건 최적화

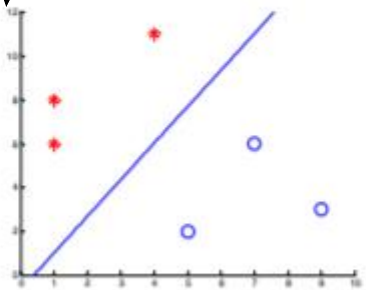
- 선형대수학의 **quadratic problem solver** 라이브러리를 이용
  - 최적화 문제의 해인 $\alpha_1, \alpha_2, \ldots, \alpha_N$ 계산
- KKT 조건의 $\alpha_i(1 - h(\boldsymbol{x}_i)) = 0$ 때문에, $\alpha_i \neq 0$이라면 $h(\boldsymbol{x}_i) = 1$
- $\alpha_i \neq 0$인 $\boldsymbol{x}_i$가 서포트 벡터
- $\boldsymbol{w}$ 의 계산

다 더하면 w구할 수 있음

$$\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i t_i \boldsymbol{x}_i = \begin{bmatrix} -0.333 \\ 0.200 \end{bmatrix}$$

서포트벡터 만이 알파를 결정!

$$X = \begin{bmatrix} 1 & 6 \\ 1 & 8 \\ 4 & 11 \\ 5 & 2 \\ 7 & 6 \\ 9 & 3 \end{bmatrix} \qquad \boldsymbol{\alpha} = \begin{bmatrix} 0.0356 \\ 0.0000 \\ 0.0400 \\ 0.0000 \\ 0.0756 \\ 0.0000 \end{bmatrix}$$



$$h(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b = \sum_{i=1}^{N} a_i t_i \boldsymbol{x}_i \cdot \boldsymbol{x} + b$$

$$h(x_1, x_2) = -0.333x_1 + 0.2x_2 + 0.13$$

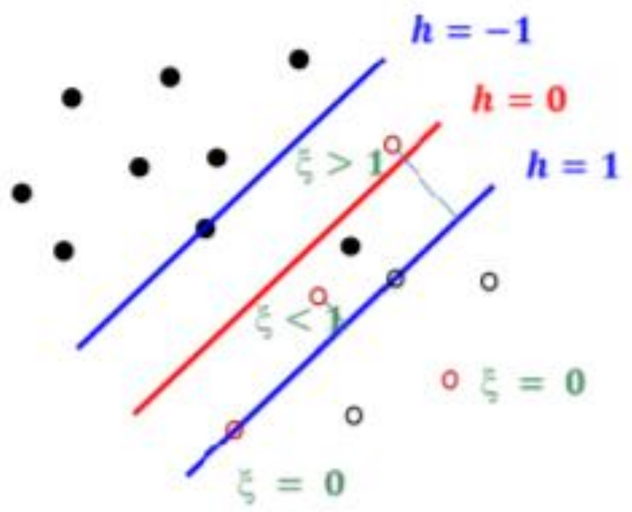
- b의 계산
  - 서포트 벡터 하나를 $t_i(\boldsymbol{w}^\top \boldsymbol{x}_i - b) = 1$ 에 넣어 계산

$$\begin{bmatrix} -0.333 \\ 0.200 \end{bmatrix}^\top \begin{bmatrix} 1 \\ 6 \end{bmatrix} + b = 1 \qquad b = 0.13$$

B는 나머지 구한다음에 대입해보면 되죠
여기서는 x에 (1,6)을 넣었네요

서포트 벡터 머신
(SVM : Support Vector Machine)

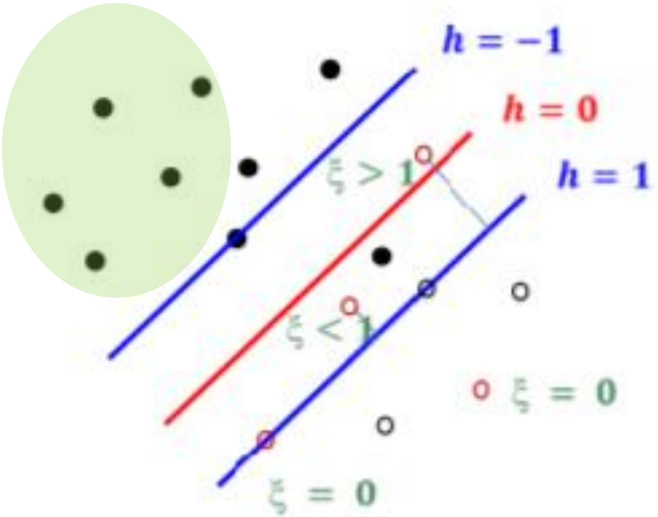
그러나 세상일이 그렇게 쉽나요
선형 분리가 불가능한 SVM도 있답니다



어차피 모든 것들은 완벽한게 없다고 하죠
조금 오류를 허용하여 선형분류를 해봅시다

서포트 벡터 머신
(SVM : Support Vector Machine)

그러나 세상일이 그렇게 쉽나요
선형 분리가 불가능한 SVM도 있답니다



어차피 모든 것들은 완벽한게 없다고 하죠
조금 오류를 허용하여 선형분류를 해봅시다

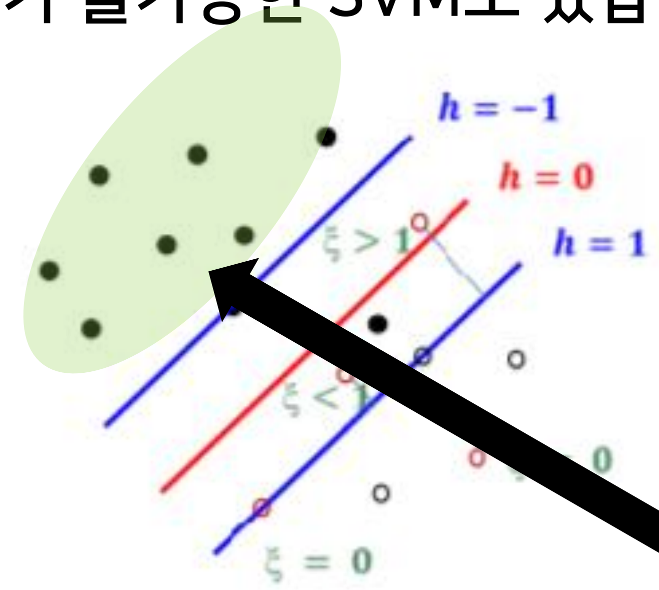
여기서 슬랙변수(slack variable)이라는 말이 등장합니다

- 학습 데이이터별로 하나씩 생성
- SVM의 분할 초평면에서 서포트벡터보다 멀리 위치하면, $\xi_i = 0$
- 이외의 경우, $\xi_i = |t_i - h(\boldsymbol{x}_i)|$

$$\boxed{t_i h(x_i) \geq 1 - \xi_i, \quad i = 1, \cdots, N}$$

서포트 벡터 머신
(SVM : Support Vector Machine)

그러나 세상일이 그렇게 쉽나요
선형 분리가 불가능한 SVM도 있답니다



어차피 모든 것들은 완벽한게 없다고 하죠
조금 오류를 허용하여 선형분류를 해봅시다

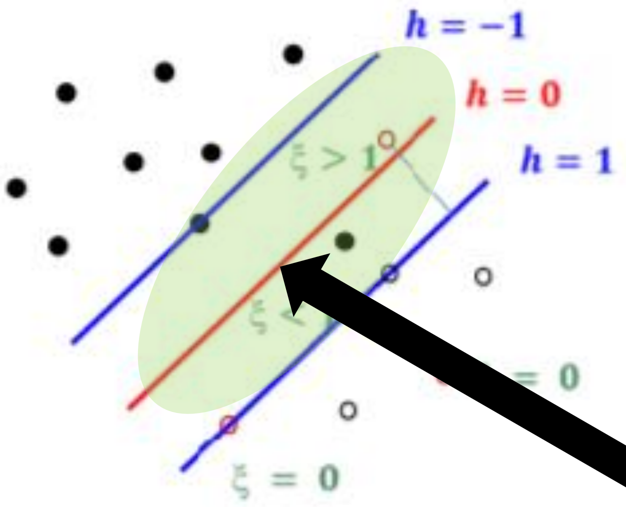
여기서 슬랙변수(slack variable)이라는 말이 등장합니다

- 학습 데이터별로 하나씩 생성
- SVM의 분할 초평면에서 서포트벡터보다 멀리 위치하면, $\xi_i = 0$
- 이외의 경우, $\xi_i = |t_i - h(\boldsymbol{x}_i)|$

$$t_i h(x_i) \geq 1 - \xi_i, \quad i = 1, \cdots, N$$

서포트 벡터 머신
(SVM : Support Vector Machine)

그러나 세상일이 그렇게 쉽나요
선형 분리가 불가능한 SVM도 있답니다



어차피 모든 것들은 완벽한게 없다고 하죠
조금 오류를 허용하여 선형분류를 해봅시다

여기서 슬랙변수(slack variable)이라는 말이 등장합니다

- 학습 데이이터별로 하나씩 생성
- SVM의 분할 초평면에서 서포트벡터보다 멀리 위치하면, $\xi_i = 0$
- 이외의 경우, $\xi_i = |t_i - h(\boldsymbol{x}_i)|$

$$\boxed{t_i h(x_i) \geq 1 - \xi_i, \quad i = 1, \cdots, N}$$

흠 이정도는 틀려도 봐주지 뭐 이런거죠

서포트 벡터 머신
(SVM : Support Vector Machine)

그러나 세상일이 그렇게 쉽나요
선형 분리가 불가능한 SVM도 있답니다

슬랙변수 허용하는 제약조건

- **최적화 문제**
  - 슬랙 변수를 허용하는 **제약조건**

    $t_i h(\boldsymbol{x}_i) \geq 1 - \xi_i, \quad i = 1, \cdots, N$

슬랙값의 합이 최소가 되도록 하려구요
C는 슬랙을 얼마만큼 허용할껀지 나타내는 거에요
크면 클수록 어떻게 될까요?

  - **슬랙 변수 값의 합을 최소화**

$$\text{Find } \boldsymbol{w}, b \text{ which minimizes } J(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$
$$\text{subject to } t_i h(\boldsymbol{x}_i) \geq 1 - \xi_i, \quad i = 1, \cdots, N$$
$$\xi_i \geq 0, \quad i = 1, \cdots, N$$
$$C > 0$$

  - **라그랑주 함수**

$$L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i + \sum_{i=1}^{N}\alpha_i(1 - t_i h(\boldsymbol{x}_i) - \xi_i) - \sum_{i=1}^{N}\beta_i\xi_i$$

왜 라그랑주 함수가 우리들의 친구인지 아시겠죠?
계속 나오거든요
또 라그랑주 함수를 구합니다

서포트 벡터 머신
(SVM : Support Vector Machine)

그러나 세상일이 그렇게 쉽나요
선형 분리가 불가능한 SVM도 있답니다

- **쌍대 문제(dual problem)**

$$\tilde{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j$$

헐 정리했더니 슬랙 사라짐

$$\sum_{i=1}^{N}\alpha_i t_i = 0$$

$0 \le \alpha_i \le C$ ← 상한(upper bound)에 대한 제약조건 추가

그대신 조건이 붙어여

복잡한 문제가 간단해지는 마법

서포트 벡터 머신
(SVM : Support Vector Machine)

하지만 우리가 아무리 관용적으로 슬랙을 도입하더라도 한계가 있어요
약간의 오차는 가능하지만 무지막지하게 꼬여있으면 힘들거든요

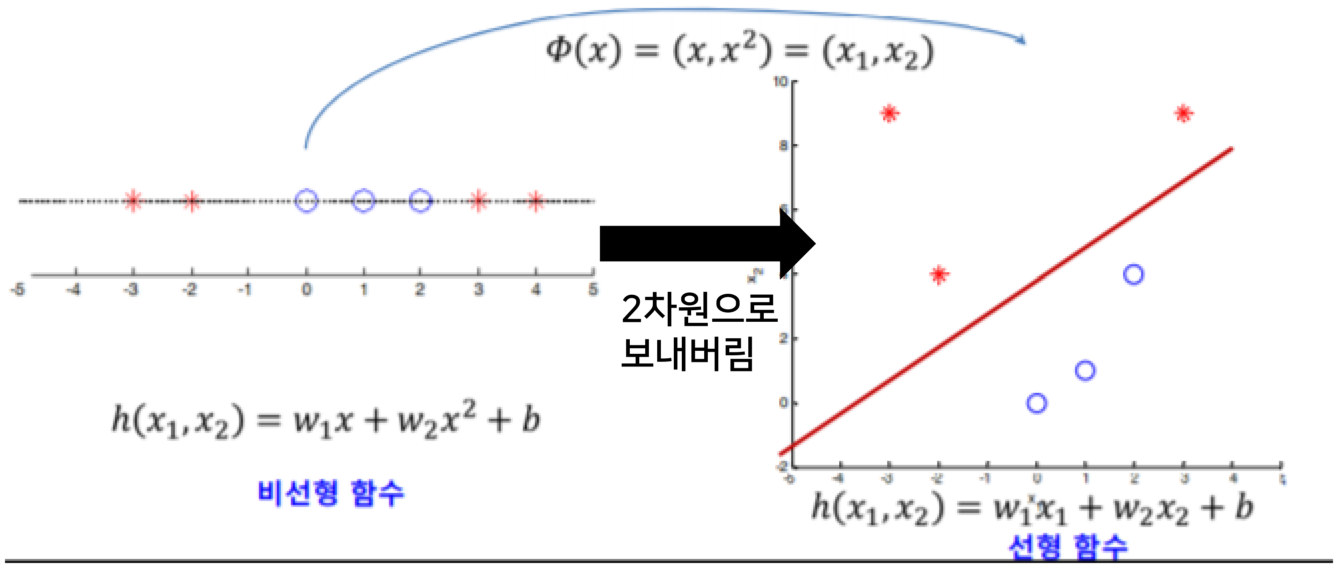
그래서 똑똑한 분들이 재밌는 아이디어를 내죠
데이터를 고차원으로 보내버리면 비선형으로 분류되는 데이터를
선형으로 보낼 수 있다는 거예요

서포트 벡터 머신
(SVM : Support Vector Machine)

하지만 우리가 아무리 관용적으로 슬랙을 도입하더라도 한계가 있어요
약간의 오차는 가능하지만 무지막지하게 꼬여있으면 힘들거든요

그래서 똑똑한 분들이 재밌는 아이디어를 내죠
데이터를 고차원으로 보내버리면 비선형으로 분류되는 데이터를
선형으로 보낼 수 있다는 거예요

쉬운 1차식과 2차식의 예로 알아볼까요?



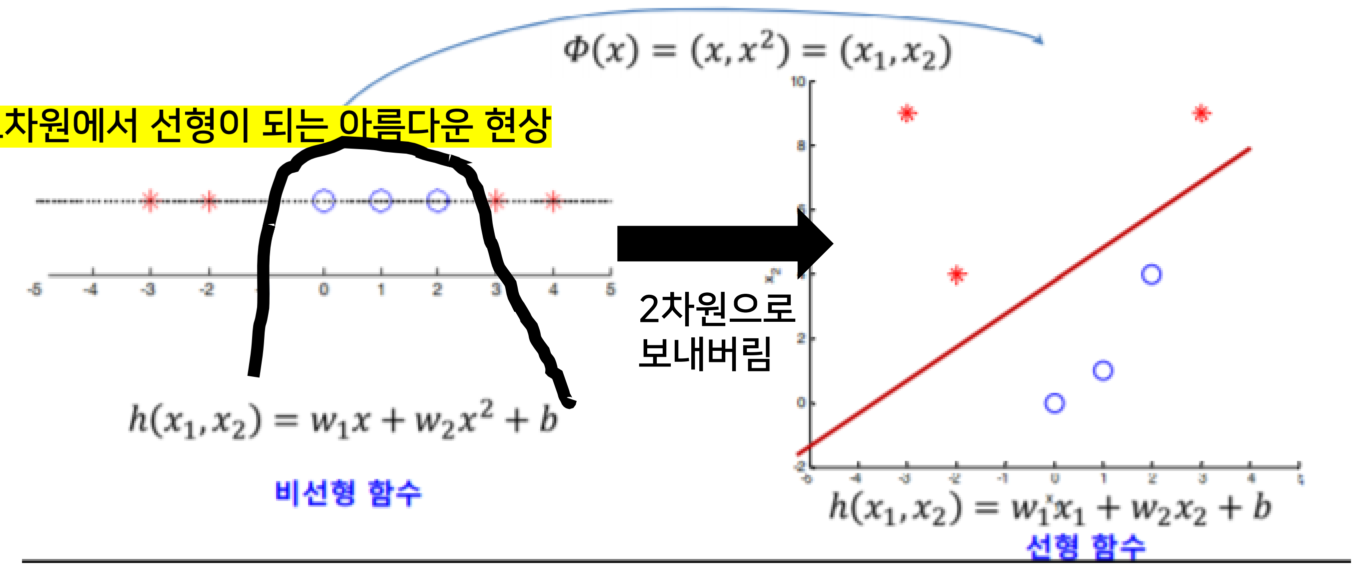
서포트 벡터 머신
(SVM : Support Vector Machine)

하지만 우리가 아무리 관용적으로 슬랙을 도입하더라도 한계가 있어요
약간의 오차는 가능하지만 무지막지하게 꼬여있으면 힘들거든요

그래서 똑똑한 분들이 재밌는 아이디어를 내죠
데이터를 고차원으로 보내버리면 비선형으로 분류되는 데이터를
선형으로 보낼 수 있다는 거예요

쉬운 1차식과 2차식의 예로 알아볼까요?

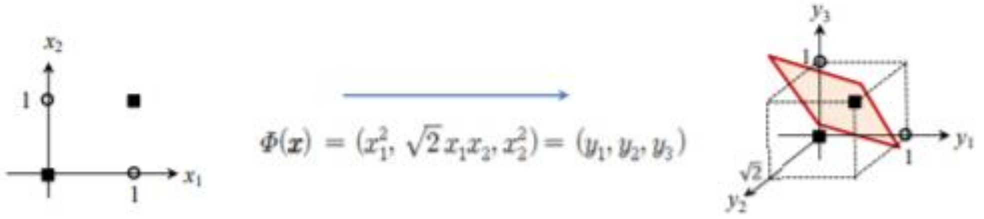
저차원에서 비선형이었던게 고차원에서 선형이 되는 아름다운 현상



서포트 벡터 머신
(SVM : Support Vector Machine)

❖ 데이터의 고차원 사상 – cont.

- XOR 문제



그러나 고차원 변환은 차원의 저주가 발생하기 쉽죠
테스트 데이터에 대한 일반화 능력도 저하될 수 있고요
물론 그건 여백(margin)을 최대화 해서 일반화 능력을 유지할 수 있어요

그러나 고차원으로 보내버리면 계산비용이 어마어마하게
늘어난다는건 사실이에요
그래서 저희는 커널트릭(kernel trict)을 사용합니다

서포트 벡터 머신
(SVM : Support Vector Machine)

❖ SVM의 최적화 문제

- 선형 SVM

Find $\boldsymbol{\alpha}$ which minimizes $\widetilde{L}(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boxed{\boldsymbol{x}_i \cdot \boldsymbol{x}_j} - \sum_{i=1}^{N}\alpha_i$

subject to $\sum_{i=1}^{N}\alpha_i t_i = 0, \quad i = 1, \cdots, N$

$\alpha_i \geq 0, \quad i = 1, \cdots, N$

$$h(\boldsymbol{x}) = \sum_{i=0}^{N}\alpha_i t_i \boxed{\boldsymbol{x}_i \cdot \boldsymbol{x}} + b$$

- 데이터의 고차원 변환

$$\boldsymbol{x}_i \rightarrow \Phi(\boldsymbol{x}_i)$$

차원 변환한 걸 내적

- 비선형 SVM

$$\widetilde{L}(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boxed{\Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)} + \sum_{i=1}^{N}\alpha_i$$

$$h(\boldsymbol{x}) = \sum_{i=1}^{N}\alpha_i t_i \boxed{\Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x})} + b$$

원래 고차원변환은 이렇게 해주어야 하는데요
문제는 내적할 때 무지무지무지 시간이 많이 든다는 거예요

서포트 벡터 머신
(SVM : Support Vector Machine)

그러나! 정말 세상 참 편해졌죠
데이터를 고차원으로 변환하지 않고도 원래 데이터에서
계산 가능한 커널함수(kernel function) K가 있다고 합니다!

커널 함수의 예

- $K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^T \boldsymbol{y})^2$
  - $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$
  - $(\boldsymbol{x}^T \boldsymbol{y})^2 = \boxed{(x_1 y_1 + x_2 y_2)^2 = (x_1 y_1)^2 + 2(x_1 y_1)(x_2 y_2) + (x_2 y_2)^2}$ 원래 이 계산을 해야되는데요
    $= \boxed{[(x_1)^2 \quad \sqrt{2}x_1 x_2 \quad (x_2)^2][(y_1)^2 \quad \sqrt{2}y_1 y_2 \quad (y_2)^2]^T}$ 요로케 하면 똑같이 나오죠
    $= \boldsymbol{\Phi}(\boldsymbol{x}) \cdot \boldsymbol{\Phi}(\boldsymbol{y})$
  - $\boldsymbol{\Phi}(\boldsymbol{x}) = \begin{bmatrix} (x_1)^2 \\ \sqrt{2}x_1 x_2 \\ (x_2)^2 \end{bmatrix} \quad \boldsymbol{\Phi}(\boldsymbol{y}) = \begin{bmatrix} (y_1)^2 \\ \sqrt{2}y_1 y_2 \\ (y_2)^2 \end{bmatrix}$

서포트 벡터 머신
(SVM : Support Vector Machine)

대표적인 커널은 요론게 있다고 해요

- **다항식 커널**(polynomial kernel)

  $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i \cdot \boldsymbol{x}_j + 1)^p$, p는 양의정수

- **RBF**(radial basis function) **커널**

  $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/(2\sigma^2)}$

- **하이퍼볼릭 탄젠트**(hyperbolic tangent) **커널**

  $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\alpha \boldsymbol{x}_i \cdot \boldsymbol{x}_j + \beta)$

서포트 벡터 머신
(SVM : Support Vector Machine)

$$\widetilde{L}(\alpha) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boxed{\Phi(x_i)\cdot\Phi(x_j)} + \sum_{i=1}^{N}\alpha_i$$

$$h(x) = \sum_{i=1}^{N}\alpha_i t_i \boxed{\Phi(x_i)\cdot\Phi(x)} + b$$

원래 이렇게 고차원으로 보내서 계산을 했는데

- **커널 함수 적용**

$$\widetilde{L}(\alpha) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j t_i t_j \boxed{K(x_i, x_j)} + \sum_{i=1}^{N}\alpha_i$$

$$h(x) = \sum_{i=1}^{N}\alpha_i t_i \boxed{K(x_i, x)} + b$$

이렇게 바꿔주고요

❖ **초평면 결정**

- **선형 SVM**

$$\widetilde{L}(\alpha) = \frac{1}{2}[\alpha_1\ \alpha_2\ \cdots\ \alpha_N]\begin{bmatrix} t_1t_1 x_1\cdot x_1 & t_1t_2 x_1\cdot x_2 & \cdots & t_1t_N x_1\cdot x_N \\ t_2t_1 x_2\cdot x_1 & t_2t_2 x_2\cdot x_2 & \cdots & t_2t_N x_2\cdot x_N \\ \vdots & \vdots & \ddots & \vdots \\ t_Nt_1 x_N\cdot x_1 & t_Nt_2 x_N\cdot x_2 & \cdots & t_Nt_N x_N\cdot x_N \end{bmatrix}\begin{bmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_N\end{bmatrix} - \sum_{i=1}^{N}\alpha_i$$

선형 계산은 이렇구요

- **비선형 SVM**

$$\widetilde{L}(\alpha) = \frac{1}{2}[\alpha_1\ \alpha_2\ \cdots\ \alpha_N]\begin{bmatrix} t_1t_1 K(x_1,x_1) & t_1t_2 K(x_1,x_2) & \cdots & t_1t_N K(x_1,x_N) \\ t_2t_1 K(x_2,x_1) & t_2t_2 K(x_2,x_2) & \cdots & t_2t_N K(x_2,x_N) \\ \vdots & \vdots & \ddots & \vdots \\ t_Nt_1 K(x_N,x_1) & t_Nt_2 K(x_N,x_2) & \cdots & t_Nt_N K(x_N,x_N) \end{bmatrix}\begin{bmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_N\end{bmatrix} - \sum_{i=1}^{N}\alpha$$

- **이차식 계획법 라이브러리 사용**

비선형 계산은 이렇구요

  - quadprog( ), solver.qp()
  - $\alpha_1, \alpha_2, \ldots, \alpha_N$ 계산
  - 고차원의 초평면 함수 결정

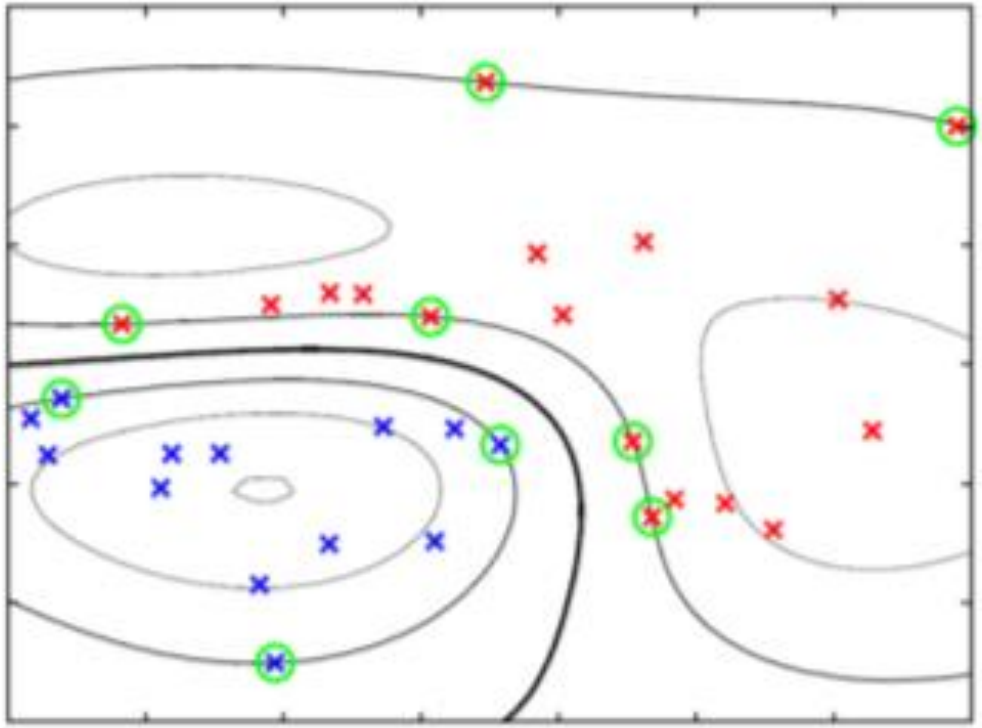
$$h(x) = \sum_{i=1}^{N}\alpha_i t_i \Phi(x_i)\cdot\Phi(x) + b$$

$$h(x) = \sum_{i=1}^{N}\alpha_i t_i K(x_i, x) + b$$

나머지는 또 계산하면 되요

서포트 벡터 머신
(SVM : Support Vector Machine)

비선형 SVM 시각화

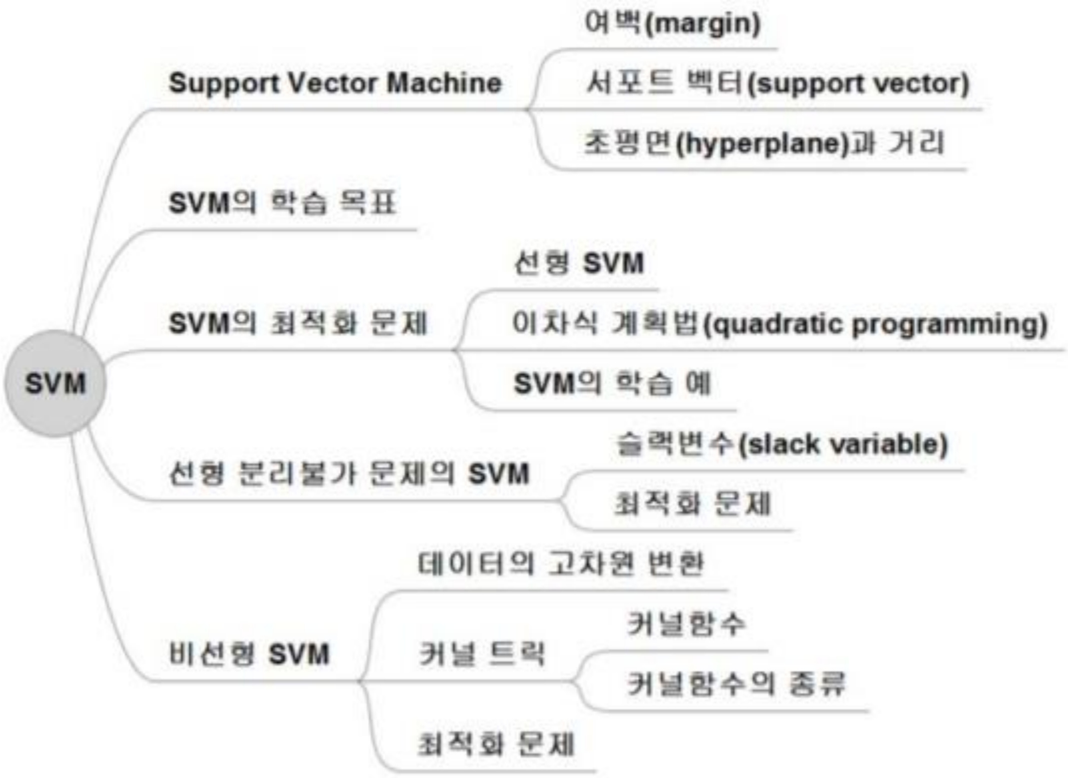


연두색 칠해있는게 서포트벡터구요

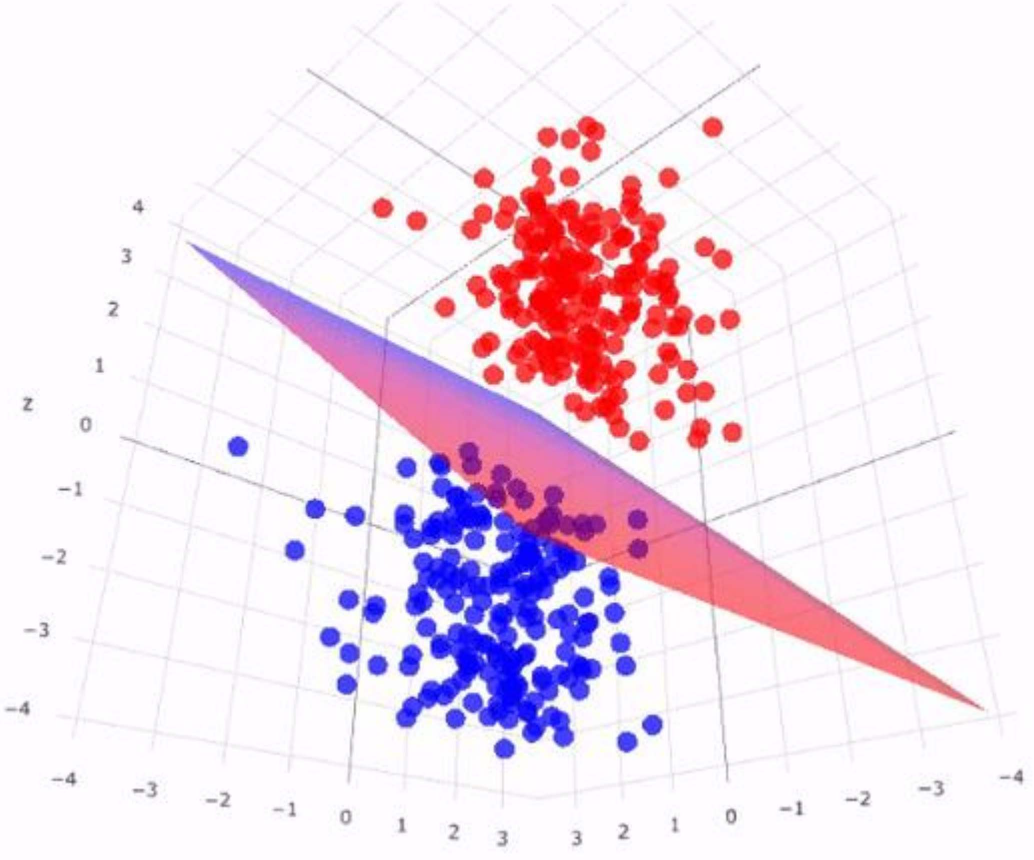
곡선 모양이 결정경계에요

시각화 하니까 예쁘네요

서포트 벡터 머신
(SVM : Support Vector Machine)



SVM 너란녀석.. 그래서 왜 좋은거니?



참고자료

충북대학교 인공지능 수업

http://www.kocw.net/home/search/kemView.do?kemId=1170523

*이 ppt에서 많이 참고(한거 아니고 베낌) 하였다고 합니다

카이스트 인공지능과 기계학습 수업

http://www.kmooc.kr/courses/course-v1:KAISTk+KCS470+2017_K0203/about

*굉장히 유명한 강의이니 들어보시기 바랍니다

앤드류 교수님 강의 리뷰

https://www.slideshare.net/freepsw/svm-77055058

*이것도 넣으려고 했는데 시간관계상 넣지 못했습니다

*꼭 보시기 바랍니다 조금 더 깊은 내용도 있으며 이 내용보다 조금 더 실용적인 내용도 있습니다(커널 선택 팁은 깨알같죠)