



ML and Optimization Intro

Hyerim Bae

Quiz

- What is minimum value of the following function?

$$f(x) = x^2 - 2x + 2$$

$$f(x) = 3 \sin(x) \cos(x) (6x^2 + 3x^3 + x) \tan(x)$$

- How to get the answer?
 - Mathematically
 - Geographically

What is optimization?

- Optimization means ‘minimization’ or ‘maximization’ of a (objective) function

$$\min f(x)$$

Optimization form

- Unconstrained optimization

- $\arg \max$

$$\operatorname{argmax}_x f(x)$$

- Constrained Optimization

$$\text{minimize } f(x)$$

$$\text{subject to } c_j(x) = 0$$

$$c_k(x) \geq 0$$

- Black-box optimization

- Objective function is unknown

$$y = f(\lambda)$$

How to solve optimization problems

- Mathematical methods
 - Simplex
 - Lagrangian method
- Geographically
- Meta Heuristics
- Heuristics
- Differentiation
- Gradient descent
- Optimization by search

1. Mathematical methods for constrained optimization

- Linear Programming
- Integer Programming

Linear Programming

■ 선형계획 모형

- 목적함수와 제약식이 모두 선형으로 수식화될 수 있는 경우
 - 일정한 제약조건 하에서 목적하고자 하는 값을 최대화(최소화)하고자 하는 수리적 방법
 - ✓ 제품배합문제
 - ✓ 작업배정문제
 - ✓ 수송문제
 - ✓

A,B 두 상품을 생산하는데 상품 A는 개당 2원의 이익이 나고, B는 개당 5원의 이익이 발생한다. 상품 A를 생산하는 데 9개의 재료와 3시간 동안 기계를 사용해야 하며, B는 5개의 재료와 4시간의 기계를 사용해야 한다. 이때 재료는 총 300개를 사용할 수 있으며, 기계 가동 시간은 최대 200시간이라고 한다. 또 상품 A는 최소 5개 이상을 생산해야만 한다고 한다. 이때 최대의 이익을 산출해 내는 상품 A와 B의 생산량을 결정하라.

결정 변수: 제품 A의 생산량 $\Rightarrow x_1$
제품 B의 생산량 $\Rightarrow x_2$

목적함수 제약식	{	$Maximize \quad 2x_1 + 5x_2$	← 이익의 최대화
		$s.t. \quad 3x_1 + 4x_2 \leq 200$	← 기계 가동시간 제약
		$9x_1 + 5x_2 \leq 300$	← 재료 사용량 제약
		$x_1 \geq 5$	← A의 최소 생산량 제약
		$x_2 \geq 0$	← 비음인 해만을 구함

Geometric way

$$\text{Maximize } 2x_1 + 5x_2$$

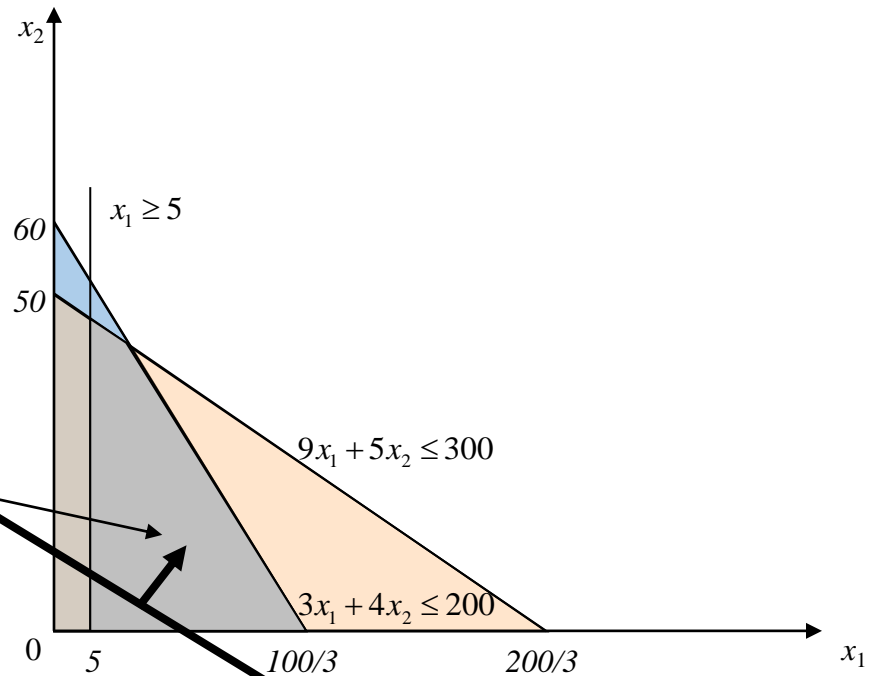
$$\text{s.t. } 3x_1 + 4x_2 \leq 200$$

$$9x_1 + 5x_2 \leq 300$$

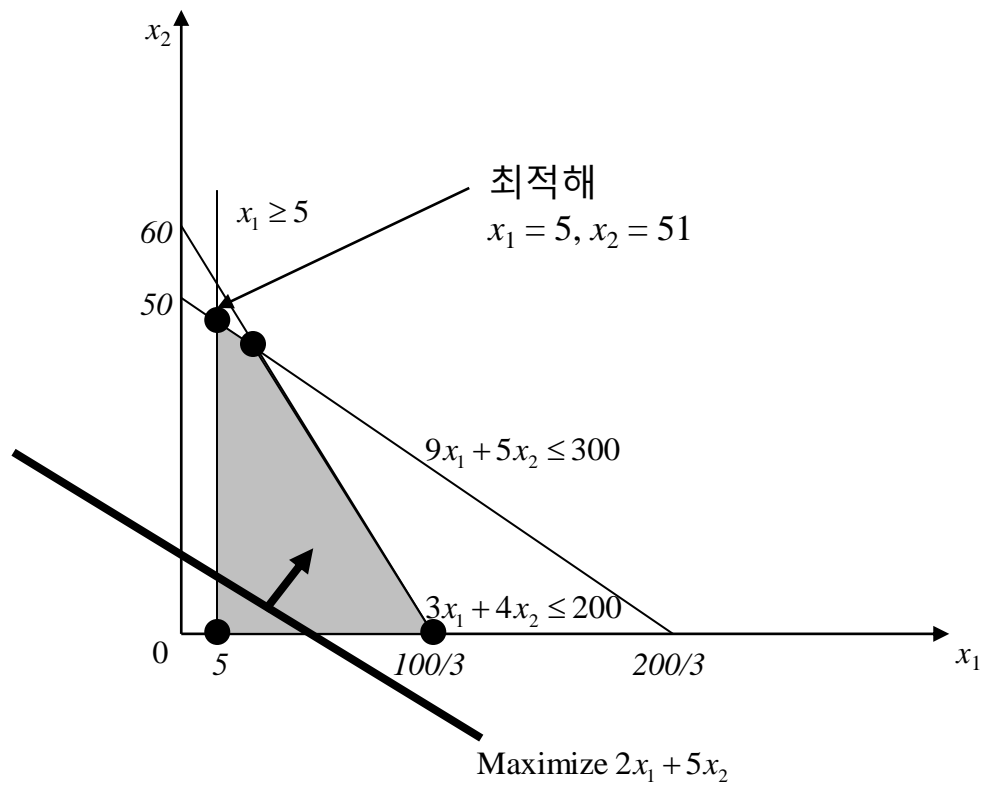
$$x_1 \geq 5$$

$$x_2 \geq 0$$

가능해 공간



Maximize $2x_1 + 5x_2$



[예] 3개 제품의 배합 문제

A,B,C 세 상품을 생산하는데 상품 A의 1단위는 압연시간이 2.4분, 조립 공정에 5.0분이 필요하다. 이익은 600원이 발생한다. 상품 B의 1단위는 압연시간이 3.0분, 용접 공정에 2.5분이 필요하고, 이익은 700원이 발생한다. 상품 C의 1단위는 2.0분의 압연시간과 1.5분의 용접 시간, 2.5분의 조립 시간이 필요하고, 500원의 이익이 발생한다.

압연 공정의 생산 시간은 일주일에 1,200분이고, 용접 공정은 일주일에 600분, 조립 공정은 일주일에 1,500분이 가동될 수 있다.

최대의 이익을 발생시킬 수 있는 제품 A, B, C의 생산량은?

$$\begin{array}{ll} \text{Maximize } 600x_1 + 700x_2 + 500x_3 & \longleftarrow \text{이익의 최대화} \\ \text{s.t. } 2.4x_1 + 3.0x_2 + 2.0x_3 \leq 1200 & \longleftarrow \text{압연 시간 제약} \\ 0.0x_1 + 2.5x_2 + 1.5x_3 \leq 600 & \longleftarrow \text{용접 시간 제약} \\ 5.0x_1 + 0.0x_2 + 2.5x_3 \leq 1500 & \longleftarrow \text{조립 시간 제약} \\ x_1, x_2, x_3 \geq 0 & \longleftarrow \text{비음인 해만을 구함} \end{array}$$

■ 기하학적 접근 ?

- 해가 제약공간상의 Vertex에서 얻어진다.
- 2,3개 이내의 문제에서 가시적인 풀이 가능
 - 3차원 이상의 문제에 적용이 어렵다.
 - 제약식의 수가 많아도 해결이 어렵다.

=> 알고리즘적인 해법이 필요

- Simplex Method
 - Dantzig
 - 제약식의 교점 중에서 최적해를 탐색
- Karmarkar Method

Simplex Method

- 단체법(單體法)
 - 1차 연립방정식 이론을 바탕으로 함
 - 행렬 연산: 가우스-조단 소거법
 - 이해가 쉽고 실용성도 높다.
 - 가능해 집합의 Vertex 중 하나를 최적해로 찾는다.
 - 초기 기저가능해 \Rightarrow [해의 개선] \Rightarrow 최적해

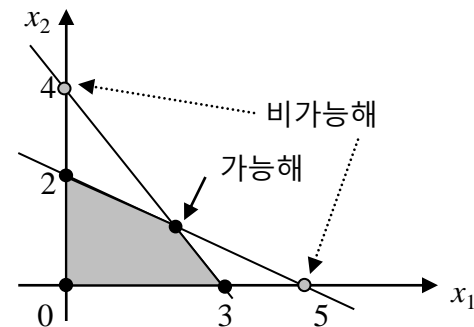
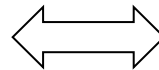
$$\begin{array}{l}
 \left\{ \begin{array}{l} \text{Maximize } 12x_1 + 15x_2 \\ \text{s.t. } 4x_1 + 3x_2 \leq 12 \\ 2x_1 + 5x_2 \leq 10 \\ x_1, x_2 \geq 0 \end{array} \right\} \xrightarrow{\text{제약식을 등식화 함}} \left\{ \begin{array}{l} \text{Maximize } 12x_1 + 15x_2 \\ \text{s.t. } 4x_1 + 3x_2 + x_3 = 12 \\ 2x_1 + 5x_2 + x_4 = 10 \\ x_1, x_2, x_3, x_4 \geq 0 \end{array} \right\}
 \end{array}$$

제약식만을 이용한 연립방정식

$$\left\{ \begin{array}{l} 4x_1 + 3x_2 + x_3 = 12 \\ 2x_1 + 5x_2 + x_4 = 10 \end{array} \right\} \text{ 무수히 많은 해가 존재}$$

두개의 변수 값만으로 연립방정식의 해를 찾는다면, 6개의 해가 얻어짐

기저	해 (x_1, x_2, x_3, x_4)
(x_1, x_2)	$(15/7, 8/7, 0, 0)$
(x_1, x_3)	$(5, 0, -8, 0)$
(x_1, x_4)	$(3, 0, 0, 4)$
(x_2, x_3)	$(0, 2, 6, 0)$
(x_2, x_4)	$(0, 4, 0, -10)$
(x_3, x_4)	$(0, 0, 12, 10)$



목적함수 추가 $\text{Maximize } 12x_1 + 15x_2$

목적함수를 최대화하는 가능해는 1개 뿐 \Rightarrow 최적해

기저	해 (x_1, x_2, x_3, x_4)	목적함수의 값	
(x_1, x_2)	$(15/7, 8/7, 0, 0)$	$300/7$	비가능해
(x_1, x_3)	$(5, 0, -8, 0)$	60	←
(x_1, x_4)	$(3, 0, 0, 4)$	36	
(x_2, x_3)	$(0, 2, 6, 0)$	30	
(x_2, x_4)	$(0, 4, 0, -10)$	60	←
(x_3, x_4)	$(0, 0, 12, 10)$	0	비가능해

최적해

단순히 제약식의 연립방정식만을 이용한 해법은 만일 n 개의 결정변수로 이루어진 m 개의 제약식을 갖는 문제라면 모두 ${}_nC_m$ 개의 해를 찾아야 한다. n 과 m 의 값에 따라 탐색 공간이 지수적으로 증가한다.

\Rightarrow 비가능해는 탐색하지 않고, 가능해 내에서만 탐색을 하되, 목적함수 값을 꾸준히 증가 시킬 수 있도록 하는 방법이 필요.

[예] 단체법을 이용한 풀이

$$\text{Maximize } 4x_1 + 5x_2 + 8x_3 + 11x_4$$

$$\text{s.t. } x_1 + x_2 + x_3 + x_4 \leq 15$$

$$3x_1 + 5x_2 + 10x_3 + 15x_4 \leq 105$$

$$7x_1 + 5x_2 + 3x_3 + 2x_4 \leq 120$$

$$x_j \geq 0, \forall j$$

최소의 목적함수 계수 = -11

계산형으로
변형

$$z - 4x_1 - 5x_2 - 8x_3 - 11x_4 = 0$$

$$x_1 + x_2 + x_3 + x_4 + (x_5) = 15 \quad \leftarrow 15/1 = 15$$

$$3x_1 + 5x_2 + 10x_3 + 15x_4 + (x_6) = 105 \quad \leftarrow 105/15 = 7$$

$$7x_1 + 5x_2 + 3x_3 + 2x_4 + (x_7) = 120 \quad \leftarrow 120/2 = 60$$

$$x_j \geq 0, \forall j$$

최소의 비율을 갖는 2행이 선택.

2행 4열의 변수를 중심으로 pivot을 실시

$$z - \frac{9}{5}x_1 - \frac{4}{3}x_2 - \frac{2}{3}x_3 + \frac{11}{15}x_6 = 77$$

$$\frac{4}{5}x_1 + \frac{2}{3}x_2 + \frac{1}{3}x_3 - \frac{1}{15}x_6 = 8$$

$$\frac{1}{5}x_1 + \frac{1}{3}x_2 + \frac{2}{3}x_3 + \frac{1}{15}x_6 = 7$$

$$\frac{33}{5}x_1 + \frac{13}{3}x_2 + \frac{5}{3}x_3 - \frac{2}{15}x_6 = 106$$

$$x_j \geq 0, \forall j$$

새로운 해 (0,0,0,7,8,0,106)와
개선된 목적함수값 77을 얻게 된다.

$$\begin{array}{rcl}
 z & \frac{9}{5}x_1 - \frac{4}{3}x_2 - \frac{2}{3}x_3 & + \frac{11}{15}x_6 = 77 \\
 & \frac{4}{5}x_1 + \frac{2}{3}x_2 + \frac{1}{3}x_3 & + x_5 - \frac{1}{15}x_6 = 8 \\
 & \frac{1}{5}x_1 + \frac{1}{3}x_2 + \frac{2}{3}x_3 + x_4 & + \frac{1}{15}x_6 = 7 \\
 & \frac{33}{5}x_1 + \frac{13}{3}x_2 + \frac{5}{3}x_3 & - \frac{2}{15}x_6 + x_7 = 106 \\
 & x_j \geq 0, \forall j
 \end{array}$$

Pivoting

목적함수의 모든 계수값이 0이상이므로
현재의 해가 최적해가 된다.

$$\begin{array}{rcl}
 z & + \frac{1}{6}x_2 + \frac{1}{12}x_3 & + \frac{9}{4}x_5 + \frac{7}{12}x_6 = 95 \\
 \textcircled{x_1} & + \frac{5}{6}x_2 + \frac{5}{12}x_3 & + \frac{5}{4}x_5 - \frac{1}{12}x_6 = 10 \\
 & \frac{1}{6}x_2 + \frac{7}{12}x_3 + \textcircled{x_4} - \frac{1}{4}x_5 + \frac{1}{12}x_6 & = 5 \\
 & -\frac{7}{6}x_2 - \frac{13}{12}x_3 - \frac{33}{4}x_5 + \frac{5}{12}x_6 + \textcircled{x_7} & = 40 \\
 & x_j \geq 0, \forall j
 \end{array}$$

최적해 $x^* = (10, 0, 0, 5, 0, 0, 40)$

목적함수값 $z^* = 95$

단체표의 이용 [요약정리]

z	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
1	-4	-5	-8	-11	0	0	0	0
0	1	1	1	1	1	0	0	15
0	3	5	10	15	0	1	0	105
0	7	5	3	2	0	0	1	120
1	-9/5	-4/3	-2/3	0	0	11/15	0	77
0	4/5	2/3	1/3	0	1	-1/15	0	8
0	1/5	1/3	2/3	1	0	1/15	0	7
0	33/5	13/3	5/3	0	0	-2/15	1	106
1	0	1/6	1/12	0	9/4	7/12	0	95
0	1	5/6	5/12	0	5/4	-1/12	0	10
0	0	1/6	7/12	1	-1/4	1/12	0	5
0	0	-7/6	-13/12	0	-33/4	5/12	1	40

제 4 열의 목적함수 계수가 -11로 최소.
따라서 $s = 4$ 가 된다.

$\min\{15/1, 105/15, 120/2\} = 105/15 = 7$.
최소의 비율을 갖는 2행이 선택.
따라서 $r = 2$ 가 된다.

최소의 목적함수 계수는 -9/5. 따라서 $s = 1$ 가 된다.

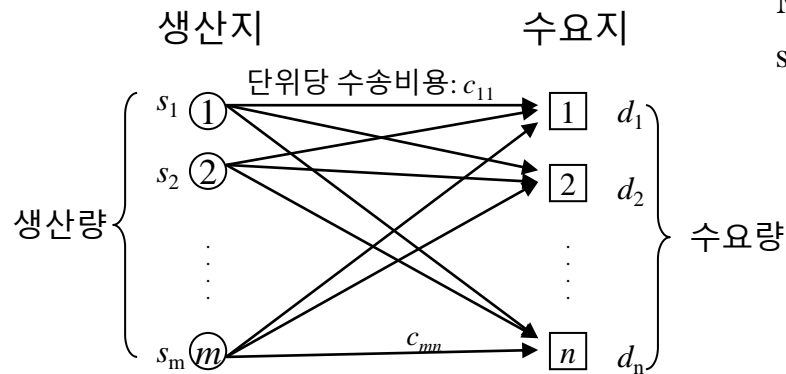
$\min\{8/(4/5), 7/(1/5), 106/(33/5)\} = 8/(4/5) = 10$.
따라서 $r = 1$ 가 된다.

목적함수의 모든 계수값이 0이상이므로
현재의 해가 최적해가 된다.

대표적 선형계획법 문제들

■ 수송 문제(Transportation Problem)

m 개의 생산지로부터 n 개의 수요지로 수요량을 만족시키면서, 최소의 비용으로 전달하는 문제



$$\text{Minimize } c_{11}x_{11} + c_{12}x_{12} + \dots + c_{mn}x_{mn}$$

$$\begin{aligned} \text{s.t. } & x_{11} + x_{12} + \dots + x_{1n} \leq s_1 \\ & \dots \\ & x_{m1} + x_{m2} + \dots + x_{mn} \leq s_m \\ & x_{11} + x_{21} + \dots + x_{m1} \geq d_1 \\ & \dots \\ & x_{1n} + x_{2n} + \dots + x_{mn} \geq d_n \\ & x_{ij} \geq 0, \quad \forall i, j \end{aligned}$$

Optimizing transportation

A 건설회사에서 3곳의 야산으로부터 모래를 운반하여 4곳의 아파트 부지에 공급한다. 모래의 운반과 관련한 비용 및 생산량과 수요량이 다음의 행렬에 정리되어 있다. 최소의 운반 비용을 얻을 수 있는 수송 경로를 구하여라.

아파트 부지 야산	d ₁	d ₂	d ₃	d ₄	공급량
s ₁	2	3	11	7	6
s ₂	1	0	6	1	1
s ₃	5	8	15	9	10
수요량	7	5	3	2	합: 17

단위: 100만원/톤

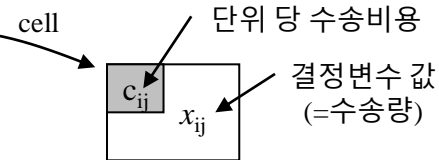
$$\text{Minimize } 2x_{11} + 3x_{12} + 11x_{13} + 7x_{14} + x_{21} + 6x_{23} + x_{24} + 5x_{31} + 8x_{32} + 15x_{33} + 9x_{34}$$

$$\begin{aligned} \text{s.t. } & x_{11} + x_{12} + x_{13} + x_{14} \leq 6 \\ & x_{21} + x_{22} + x_{23} + x_{24} \leq 1 \\ & x_{31} + x_{32} + x_{33} + x_{34} \leq 10 \\ & x_{11} + x_{21} + x_{31} \geq 7 \\ & x_{12} + x_{22} + x_{32} \geq 5 \\ & x_{13} + x_{23} + x_{33} \geq 3 \\ & x_{14} + x_{24} + x_{34} \geq 2 \\ & x_{ij} \geq 0, \forall i, j \end{aligned}$$

- Step 1) 초기해를 여러 가지 방법을 활용하여 설정한다.
 Step 2) 초기해를 개선시킬 수 있는 수송비용의 음환(negative cycle)을 찾는다.
 Step 3) 더 이상 해를 개선시킬 수 있는 음환을 찾을 수 없을 때가 최적이다.

1) 최소가법을 이용하여 설정된 초기해

공급지 \ 수요지	d ₁	d ₂	d ₃	d ₄	공급량
s ₁	2	3	11	7	6
s ₂	1	0	6	1	1
s ₃	5	8	15	9	10
수요량	7	5	3	2	총합: 17



$$\text{총비용} = 2 \cdot 6 + 0 \cdot 1 + 5 \cdot 1 + 8 \cdot 4 + 15 \cdot 3 + 9 \cdot 2 = 112$$

2) 음환을 찾는다.

$$\begin{aligned} x_{12} \uparrow &= +3 - 2 + 5 - 8 = -2 < 0 \\ x_{13} \uparrow &= +11 - 2 + 5 - 15 = -1 < 0 \\ x_{14} \uparrow &= +7 - 2 + 5 - 9 = 1 > 0 \\ x_{21} \uparrow &= +1 - 5 + 8 - 0 = 4 > 0 \\ x_{23} \uparrow &= +6 - 0 + 8 - 15 = -1 < 0 \\ x_{24} \uparrow &= +1 - 0 + 8 - 9 = 0 \end{aligned}$$

x_{12} 진입

3) 개선된 해

공급지 \ 수요지	d ₁	d ₂	d ₃	d ₄	공급량
s ₁	2	3	11	7	6
s ₂	1	0	6	1	1
s ₃	5	8	15	9	10
수요량	7	5	3	2	총합: 17

$$\text{총비용} = 2 \cdot 2 + 3 \cdot 4 + 0 \cdot 1 + 5 \cdot 5 + 15 \cdot 3 + 9 \cdot 2 = 104$$

■ 식단 문제

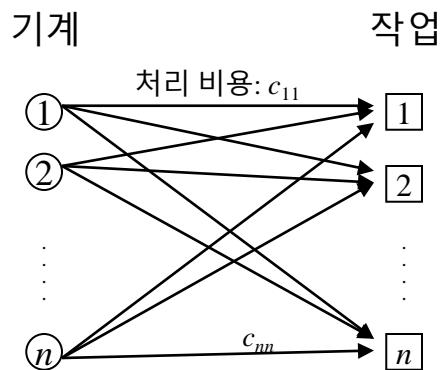
여러 가지 영양분을 지닌 음식들로부터 필수 영양분을 최소의 음식값으로 섭취하는 문제

[예] 주위에서 흔히 볼 수 있는 음식 재료에 포함된 영양분이 다음 표에 정리되어 있다. 최소의 비용으로 식단을 마련해 보고자 한다. 단 하루의 식단에서 쌀은 20포, 쇠고기는 1근, 우유는 2통, 계란은 3개, 배추는 3단을 넘지 않기로 한다.

영양 \ 재료	쌀(포)	쇠고기(근)	우유(통)	계란(12 개)	배추(단)	1 일 필요량
열량(Kcal)	340	1080	362	1040	17	2200
단백질(g)	6.5	167	19	78	1.3	70
비타민(I.U)	0	97	758	7080	255	5000
철분(mg)	0.4	11	0.3	13	0.3	12.5
탄수화물(g)	52	30	25	0	5	
콜레스테롤(u)	0	22	11	120	0	
값(원)	75	1640	370	550	110	

$$\begin{aligned}
 &\text{Minimize } 75x_1 + 1640x_2 + 370x_3 + 550x_4 + 110x_5 \\
 &\text{s.t. } 340x_1 + 1080x_2 + 362x_3 + 1040x_4 + 17x_5 \geq 2200 \\
 &\quad 6.5x_1 + 167x_2 + 19x_3 + 78x_4 + 1.3x_5 \geq 70 \\
 &\quad 97x_2 + 758x_3 + 7080x_4 + 255x_5 \geq 5000 \\
 &\quad 0.4x_1 + 11x_2 + 0.3x_3 + 13x_4 + 0.3x_5 \geq 12.5 \\
 &\quad x_1 \leq 20, \quad x_2 \leq 1, \quad x_3 \leq 2, \quad x_4 \leq 0.25, \quad x_5 \leq 3 \\
 &\quad x_{ij} \geq 0, \quad \forall i, j
 \end{aligned}$$

배정 문제(Assignment Problem): n 개의 작업을 n 개의 기계에 각기 하나씩 최소의 비용이 되도록 할당하는 문제
공급량과 수요량이 각기 1씩 발생하는 특별한 경우의 수송문제



만약 기계 i 가 작업 j 에 할당되면 $x_{ij} = 1$
아니면 $x_{ij} = 0$

$$\text{Minimize } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

$$\text{s.t. } \sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n$$

$$x_{ij} \geq 0, \quad \forall i, j$$

- 배낭 문제(Knapsack Problem)

- 한정된 배낭의 용량에 맞게 각각의 용량을 가지는 물건을 최대의 효용을 얻도록 채우는 문제
 - 대표적인 정수계획법 문제 중의 하나

[예] 갑돌이는 등산을 계획하고 있는데, 가면서 먹을 음식을 결정해야만 한다. 배당에는 총 1.6 kg까지만 음식을 담기로 결정했다고 한다. 각각의 음식의 무게와 그 음식을 가져 감으로써 얻을 수 있는 만족도가 다음과 같을 때, 가장 큰 만족도를 얻을 수 있는 음식의 조합을 결정하시오.

물건	고기	쌀	라면	과일	빵	
만족도	20	48	14	18	20	배낭의 무게
무게(100g)	8	6	2	3	2	16

$$\begin{aligned}
 &\text{Maximize } 20x_1 + 48x_2 + 14x_3 + 18x_4 + 20x_5 \\
 &\text{s.t.} \quad 8x_1 + 6x_2 + 2x_3 + 3x_4 + 2x_5 \leq 16 \\
 &\quad x_i = 1 \text{ or } 0, \quad \forall i
 \end{aligned}$$

ORTools `!pip install ortools`

```
from ortools.linear_solver import pywraplp

def LinearProgrammingExample():
    """Linear programming sample."""
    # Instantiate a Glop solver, naming it LinearExample.
    solver = pywraplp.Solver.CreateSolver('GLOP')

    # Create the two variables and let them take on any non-negative value.
    x1 = solver.NumVar(0, solver.infinity(), 'x1')
    x2 = solver.NumVar(0, solver.infinity(), 'x2')
    print('Number of variables =', solver.NumVariables())

    # Constraint 0: 4x1 + 3x2 <= 12.
    solver.Add(4*x1 + 3 * x2 <= 12)

    # Constraint 1: 2x1 + 5x2 <= 10.
    solver.Add(2*x1 + 5*x2 <= 10)

    print('Number of constraints =', solver.NumConstraints())

    # Objective function: 12x1 + 15x2.
    solver.Maximize(12 * x1 + 15 * x2)

    # Solve the system.
    status = solver.Solve()

    # print_solution
    if status == pywraplp.Solver.OPTIMAL:
        print('Solution:')
        print('Objective value =', solver.Objective().Value())
        print('x1 =', x1.solution_value())
        print('x2 =', x2.solution_value())
    else:
        print('The problem does not have an optimal solution.')

LinearProgrammingExample()
```

- Minimax 문제

Minimize maximum $\{12x_1 - 21x_2, 17x_1 - 10x_2\}$

$$\begin{aligned} \text{s.t.} \quad & 2x_1 - 7x_2 \geq 12 \\ & 6x_1 + 11x_2 \geq 41 \\ & 9x_1 + 17x_2 \leq 102 \\ & x_i \geq 0, \quad \forall i \end{aligned}$$



Minimize Z

$$\begin{aligned} \text{s.t.} \quad & Z \geq 12x_1 - 21x_2 \\ & Z \geq 17x_1 - 10x_2 \\ & 2x_1 - 7x_2 \geq 12 \\ & 6x_1 + 11x_2 \geq 41 \\ & 9x_1 + 17x_2 \leq 102 \\ & x_i \geq 0, \quad \forall i \end{aligned}$$

- Minimum Absolute Value
- Goal Programming
-

Convex optimization

- In the constrained optimization,

$$\text{minimize } f(x)$$

$$\text{subject to } c_j(x) = 0$$

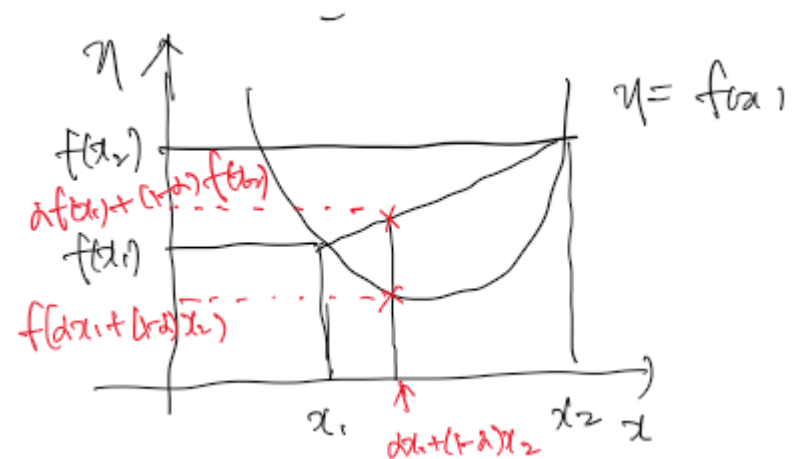
$$c_k(x) \geq 0$$

- ① Objective function is convex and ② feasible set is convex!

Convex function

- A function $f(x)$ is convex if

$$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2), \quad 0 \leq \alpha \leq 1$$



Convex set

- A set F is convex, if

$$x_1, x_2 \in F, \quad 0 \leq \alpha \leq 1$$

$$\alpha x_1 + (1 - \alpha)x_2 \in F$$

Convex optimality

- Local optimum (x^*) is global optimum

① Objective function is convex and ② feasible set is convex!

만약에

$f(x') < f(x^*)$ 라면, 즉 로컬옵티мум이 존재하는 데 그보다 더 작은 optimal이 있다면...

$$\alpha x^* + (1-\alpha)x' \in F \quad \text{"F is convex"}$$

$$\underline{f(\alpha x^* + (1-\alpha)x')} \leq \alpha f(x^*) + (1-\alpha)f(x') \quad \text{"f is convex"}$$

$$\begin{aligned} &= f(x^*) + (1-\alpha) \underbrace{(f(x') - f(x^*))}_{< 0} \\ &< f(x^*) \end{aligned}$$

if $\alpha \rightarrow 1$

x^* 주위에 $f(x^*)$ 보다 작은 값이 존재!

Feasible set 안!

Convex Optimization in ML:SVM

- A separating hyperplane can be written as

$$\mathbf{W} \cdot \mathbf{X} + b = 0$$

where $\mathbf{W} = \{w_1, w_2, \dots, w_n\}$ is a weight vector and b a scalar (bias)

- For 2-D it can be written as

$$w_0 + w_1 x_1 + w_2 x_2 = 0$$

- The hyperplane defining the sides of the margin:

$$H_1: w_0 + w_1 x_1 + w_2 x_2 \geq 1 \quad \text{for } y_i = +1, \text{ and}$$

$$H_2: w_0 + w_1 x_1 + w_2 x_2 \leq -1 \quad \text{for } y_i = -1$$

- Any training tuples that fall on hyperplanes H_1 or H_2 (i.e., the sides defining the margin) are **support vectors**
- This becomes a **constrained (convex) quadratic optimization** problem: Quadratic objective function and linear constraints \rightarrow *Quadratic Programming (QP)* \rightarrow Lagrangian multipliers

- Objective function

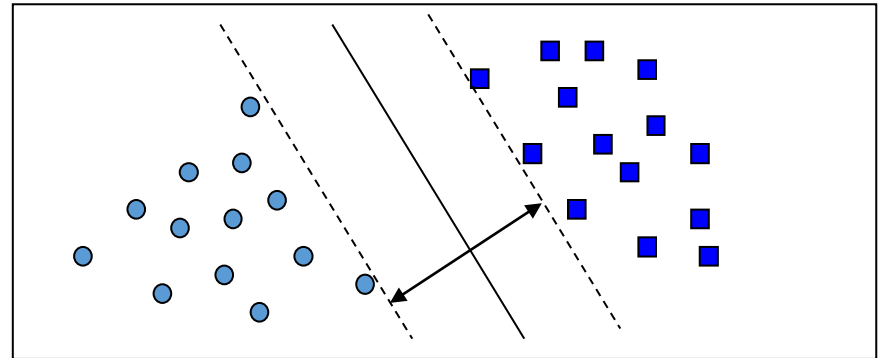
$$b_{i1}: \mathbf{w} \cdot \mathbf{x} + b = 1$$

$$b_{i2}: \mathbf{w} \cdot \mathbf{x} + b = -1$$

$$\mathbf{w} \cdot (\mathbf{x}_1 - \mathbf{x}_2) = 2$$

$$\|\mathbf{w}\| \times d = 2$$

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



- Constraints

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq 1 \quad (y_i = 1)$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 \quad (y_i = -1)$$

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, i = 1, 2, \dots, N$$

- SVM Learning

$$\min_{\mathbf{w}} f(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2}$$

$$\text{s.t.} \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, i = 1, 2, \dots, N$$

- Convex optimization => Lagrangian

$$L_P = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N \lambda_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1)$$

$$\frac{\partial L_P}{\partial \mathbf{w}} = 0 \implies \mathbf{w} = \sum_{i=1}^N \lambda_i y_i \mathbf{x}_i$$

$$\frac{\partial L_P}{\partial b} = 0 \implies \sum_{i=1}^N \lambda_i y_i = 0$$

- Dual Lagrangian

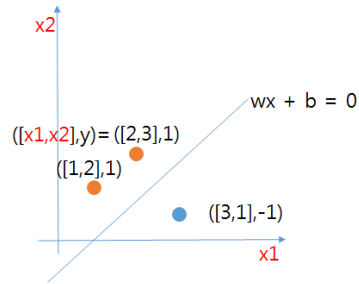
$$L_D = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

- KKT (Karush-Khun-Tucker) condition for inequality in the constraints: [necessary condition](#)

최적해가 되기 위한 조건

$$\lambda_i \geq 0$$

$$\lambda_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1) = 0$$



$$L = \frac{1}{2}w^2 + \sum \alpha_i [y_i(wx + b) - 1]$$

Then differentiate the above equations in terms of w and b

$$\frac{\partial L}{\partial w} = w - \sum \alpha_i y_i x_i = 0$$

$$w = \sum \alpha_i y_i x_i \dots (1)$$

$$\frac{\partial L}{\partial b} = \sum \alpha_i y_i = 0 \dots (2)$$

Apply (1) and (2) into L

$$\begin{aligned} L &= \frac{1}{2}(\sum \alpha_i y_i x_i)^2 - \sum [\alpha_i y_i (\sum \alpha_j y_j x_j) + \alpha_i y_i b - \alpha_i] \\ &= \frac{1}{2}(\sum \alpha_i y_i x_i)^2 - \sum \alpha_i y_i x_i (\sum \alpha_j y_j x_j) - \sum \alpha_i y_i b + \sum \alpha_i \end{aligned}$$

According to equation(2), $\sum \alpha_i y_i b = 0$

$$\begin{aligned} &= \frac{1}{2}(\sum \alpha_i y_i x_i)^2 - (\sum \alpha_i y_i x_i)(\sum \alpha_j y_j x_j) + \sum \alpha_i \\ &= -\frac{1}{2}(\sum \alpha_i y_i x_i \sum \alpha_j y_j x_j) + \sum \alpha_i \\ &= \sum \alpha_i - \frac{1}{2}(\sum \alpha_i y_i x_i \sum \alpha_j y_j x_j) \\ &= \sum \alpha_i - \frac{1}{2}(\sum \sum \alpha_i \alpha_j y_i y_j x_i x_j) \end{aligned}$$

$$\begin{aligned} x_1 &= (1, 2), y_1 = 1 \\ x_2 &= (2, 3), y_2 = 1 \\ x_3 &= (3, 1), y_3 = -1 \end{aligned}$$

By applying into the equations,

$$\begin{aligned} L &= \sum \alpha_i - \frac{1}{2} \sum \alpha_i y_i x_i (\alpha_1 y_1 x_1 + \alpha_2 y_2 x_2 + \alpha_3 y_3 x_3) \\ &= \sum \alpha_i - \frac{1}{2} \sum \alpha_i y_i x_i (\alpha_1 x_1 + \alpha_2 x_2 - \alpha_3 x_3) \\ &= \sum \alpha_i - \frac{1}{2} (\alpha_1 y_1 x_1 (\alpha_1 x_1 + \alpha_2 x_2 - \alpha_3 x_3) - \frac{1}{2} (\alpha_2 y_2 x_2 (\alpha_1 x_1 + \alpha_2 x_2 - \alpha_3 x_3)) - \frac{1}{2} (\alpha_3 y_3 x_3 (\alpha_1 x_1 + \alpha_2 x_2 - \alpha_3 x_3))) \\ &= \sum \alpha_i - \frac{1}{2} (\alpha_1 x_1 (\alpha_1 x_1 + \alpha_2 x_2 - \alpha_3 x_3) - \frac{1}{2} (\alpha_2 x_2 (\alpha_1 x_1 + \alpha_2 x_2 - \alpha_3 x_3)) + \frac{1}{2} (\alpha_3 x_3 (\alpha_1 x_1 + \alpha_2 x_2 - \alpha_3 x_3))) \\ &= \sum \alpha_i - \frac{1}{2} (5\alpha_1^2 + 13\alpha_2^2 + 10\alpha_3^2 + 16\alpha_1\alpha_2 - 10\alpha_1\alpha_3 - 18\alpha_2\alpha_3) \\ &= \alpha_1 + \alpha_2 + \alpha_3 - \frac{1}{2} (5\alpha_1^2 + 13\alpha_2^2 + 10\alpha_3^2 + 16\alpha_1\alpha_2 - 10\alpha_1\alpha_3 - 18\alpha_2\alpha_3) \\ \frac{\partial L}{\partial \alpha_1} &= 1 + \frac{1}{2} (-10\alpha_1 - 16\alpha_2 + 10\alpha_3) = 0 \\ \text{Remember we know that } \sum \alpha_i y_i &= 0, \text{ so } \alpha_1 + \alpha_2 = \alpha_3 \\ 1 + \frac{1}{2} (-10\alpha_1 - 16\alpha_2 + 10(\alpha_1 + \alpha_2)) &= 0 \\ \alpha_2 &= \frac{1}{3} \\ \frac{\partial L}{\partial \alpha_3} &= 1 - \frac{1}{2} (26\alpha_2 + 16\alpha_1 - 18\alpha_3) = 0 \\ \frac{\partial L}{\partial \alpha_3} &= 1 - (13\alpha_2 + 8\alpha_1 - 9\alpha_3) = 0 \\ 8\alpha_1 - 9\alpha_3 &= -\frac{10}{3} \end{aligned}$$

We know that $\alpha_1 + \alpha_2 = \alpha_3$, that is, $\alpha_1 + \frac{1}{3} = \alpha_3$

Finally we can find $(\alpha_1, \alpha_2, \alpha_3) = (\frac{1}{3}, \frac{1}{3}, \frac{2}{3})$

By applying α into $w = \sum \alpha_i y_i x_i$, we can get w

$$\begin{aligned} w &= (-1, 1) \\ \text{Our plane is } y &= wx + b, \text{ therefore } y = (-a, a)(x_1, x_2) + b = -ax_1 + ax_2 + b \end{aligned}$$

Apply $x_1 = (1, 2), y_1 = 1, x_2 = (3, 1), y_3 = -1$ and get $a = \frac{2}{3}, b = \frac{1}{3}$

- For test case \mathbf{z} ,

$$f(\mathbf{z}) = \text{sign}(\mathbf{w} \cdot \mathbf{z} + b) = \text{sign}\left(\sum_{i=1}^N \lambda_i y_i x_i \cdot \mathbf{z} + b\right)$$

- If $f=1$, \mathbf{z} will be classified as positive

Why Is SVM Effective on High Dimensional Data?

- The **complexity** of trained classifier is characterized by the # of support vectors rather than the dimensionality of the data
- The **support vectors** are the essential or critical training examples —they lie closest to the decision boundary (MMH)
- If all other training examples are removed and the training is repeated, the same separating hyperplane would be found
- The number of support vectors found can be used to compute an (upper) bound on the expected error rate of the SVM classifier, which is independent of the data dimensionality
- Thus, an SVM with a small number of support vectors can have good generalization, even when the dimensionality of the data is high

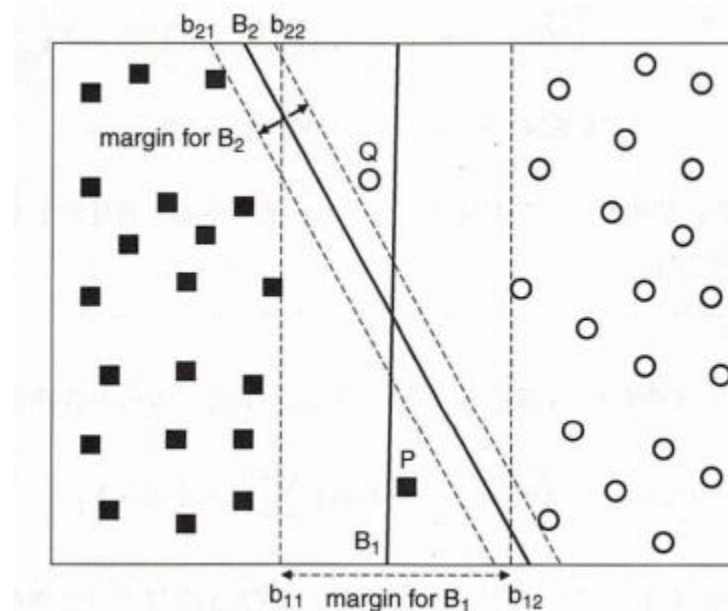
Soft Margin

$$\mathbf{w} \cdot \mathbf{x}_i + b \geq 1 - \xi_i \quad (y_i = 1)$$

$$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1 + \xi_i \quad (y_i = -1)$$

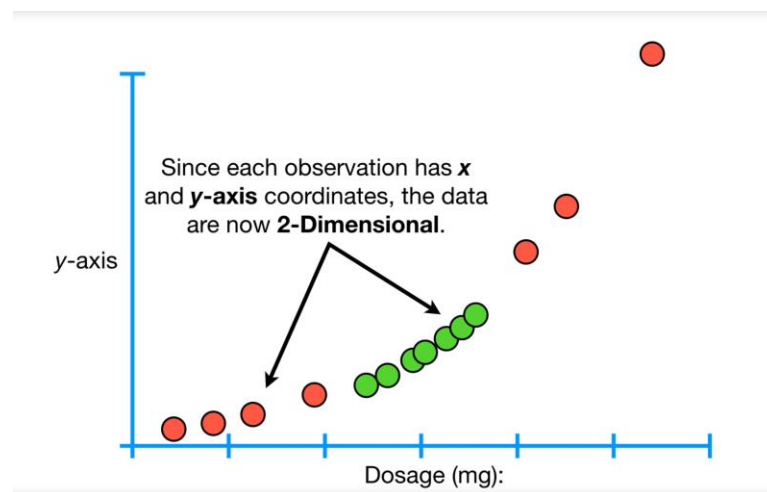
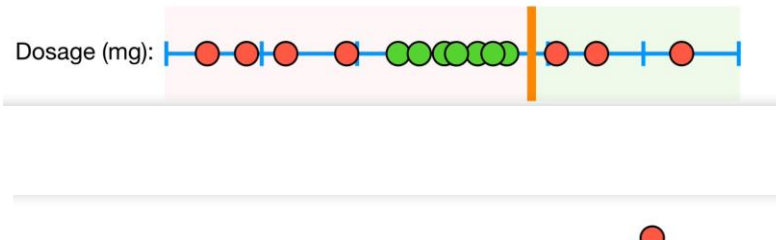
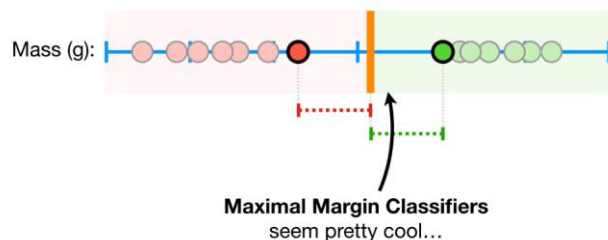
$$\min_{\mathbf{w}} f(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2} + C \left(\sum_{i=1}^N \xi_i \right)^k$$

$$L_p = \frac{1}{2} \|\mathbf{w}\|^2 - C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \lambda_i \{ \mathbf{y}_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i \} - \sum_{i=1}^N \mu_i \xi_i$$



SVM—Linearly Inseparable

- Transform the original input data into a higher dimensional space



Example 6.8 Nonlinear transformation of original input data into a higher dimensional space. Consider the following example. A 3D input vector $\mathbf{X} = (x_1, x_2, x_3)$ is mapped into a 6D space \mathbf{Z} using the mappings $\phi_1(\mathbf{X}) = x_1, \phi_2(\mathbf{X}) = x_2, \phi_3(\mathbf{X}) = x_3, \phi_4(\mathbf{X}) = (x_1)^2, \phi_5(\mathbf{X}) = x_1x_2$, and $\phi_6(\mathbf{X}) = x_1x_3$. A decision hyperplane in the new space is $d(\mathbf{Z}) = \mathbf{WZ} + b$, where \mathbf{W} and \mathbf{Z} are vectors. This is linear. We solve for \mathbf{W} and b and then substitute back so that we see that the linear decision hyperplane in the new (\mathbf{Z}) space corresponds to a nonlinear second order polynomial in the original 3-D input space,

$$\begin{aligned} d(\mathbf{Z}) &= w_1x_1 + w_2x_2 + w_3x_3 + w_4(x_1)^2 + w_5x_1x_2 + w_6x_1x_3 + b \\ &= w_1z_1 + w_2z_2 + w_3z_3 + w_4z_4 + w_5z_5 + w_6z_6 + b \end{aligned}$$

- Search for a linear separating hyperplane in the new space

SVM: Different Kernel functions

- Instead of computing the dot product on the transformed data, it is math. equivalent to applying a kernel function $K(\mathbf{X}_i, \mathbf{X}_j)$ to the original data, i.e., $K(\mathbf{X}_i, \mathbf{X}_j) = \Phi(\mathbf{X}_i) \cdot \Phi(\mathbf{X}_j)$
- Typical Kernel Functions

Polynomial kernel of degree h : $K(\mathbf{X}_i, \mathbf{X}_j) = (\mathbf{X}_i \cdot \mathbf{X}_j + 1)^h$

Gaussian radial basis function kernel : $K(\mathbf{X}_i, \mathbf{X}_j) = e^{-\|\mathbf{X}_i - \mathbf{X}_j\|^2 / 2\sigma^2}$

Sigmoid kernel : $K(\mathbf{X}_i, \mathbf{X}_j) = \tanh(\kappa \mathbf{X}_i \cdot \mathbf{X}_j - \delta)$

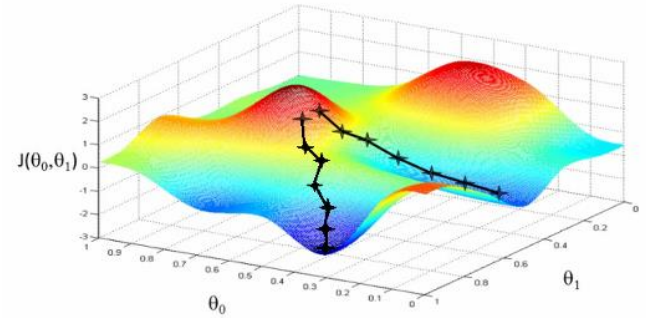
- SVM can also be used for classifying multiple (> 2) classes and for regression analysis (with additional parameters)

Using differentiation

- Minimum (Maximum) value is on the point that

$$f'(x) = 0$$

- Finding x such that $f'(x) = 0$

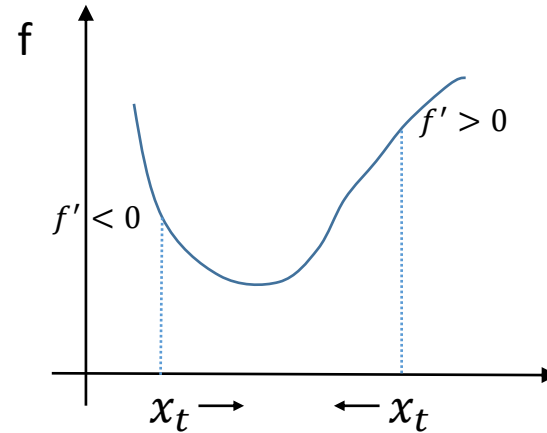


출처: <http://blog.datumbox.com/tuning-the-learning-rate-in-gradient-descent/>

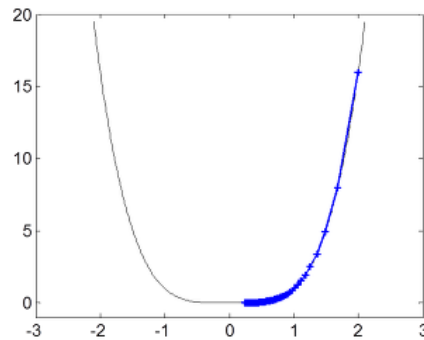
Gradient descent

- Update x value starting from x_0

$$x_{t+1} = x_t - \lambda f'(x_t)$$



$$f(x)=x^4$$



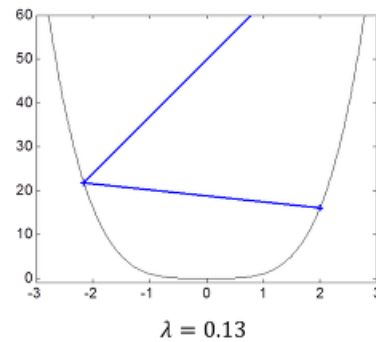
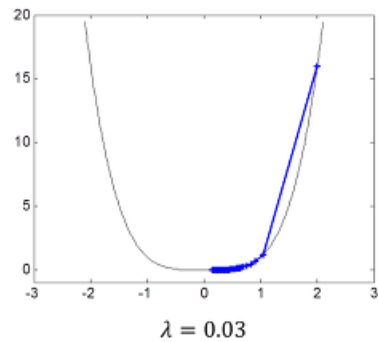
$x_0 = 2.0$
 $x_1 = x_0 - \lambda f'(x_0) = 1.6800$
 $x_2 = x_1 - \lambda f'(x_1) = 1.4903$
 $x_3 = x_2 - \lambda f'(x_2) = 1.3579$
 $x_4 = x_3 - \lambda f'(x_3) = 1.2578$
 $x_5 = x_4 - \lambda f'(x_4) = 1.1782$
 $x_6 = x_5 - \lambda f'(x_5) = 1.1128$
 $x_7 = x_6 - \lambda f'(x_6) = 1.0576$
 $x_8 = x_7 - \lambda f'(x_7) = 1.0103$
 $x_9 = x_8 - \lambda f'(x_8) = 0.9691$
...

<https://darkpgmr.tistory.com/149>

- Problems

- Even after 200 times of updating, it is not close to $x=0$

- If we use larger value of λ



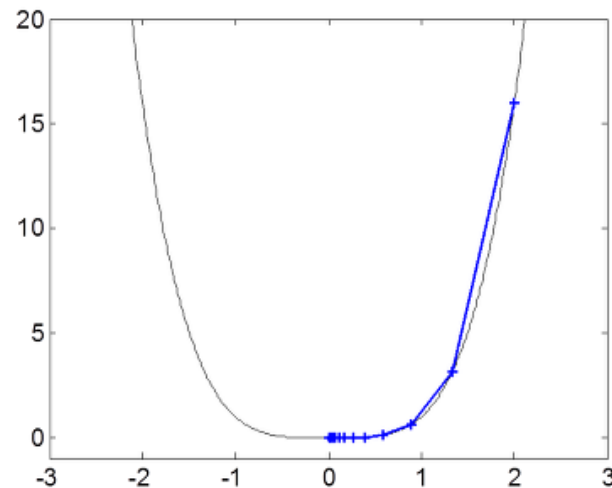
Using secondary derivatives

- Update using the equation

$$x_{t+1} = x_t - \frac{f'(x_t)}{f''(x_t)}$$

- Problems

$$f''(x_t)=0$$



$x_0 = 2.0$
 $x_1 = 1.3333$
 $x_2 = 0.8889$
 $x_3 = 0.5926$
 $x_4 = 0.3951$
 $x_5 = 0.2634$
 $x_6 = 0.1756$
 $x_7 = 0.1171$
 $x_8 = 0.0780$
 $x_9 = 0.0520$
...

Using random sampling

- Sampling base inference

Black-box optimization

- Bayesian optimization

references

- https://developers.google.com/optimization/lp/lp_example#python_7