AI Overview

Hyerim Bae
Department of Industrial Engineering, Pusan National University
hrbae@pusan.ac.kr

Contents

01 AI의 정의

02 AI의 역사

03 AI의 간단한 원리

Topics in the class

Logic

Propositional calculus
Predicate calculus
Reasoning

Search

Problem solving
Agent
Heuristic search
Planning

KBS

Expert system

**Learnin
g**

Deep learning
Unsupervised learning
NN
Genetic Algorithm

**Uncerta
inty**

Bayesian network
Statistical learning

What is AI?

- What is AI
  - Intelligent behavior in artifacts

•Perception
•Reasoning
•Learning
•Communicating
•Acting

  - "the study of ideas that enable computers to be intelligent" "To make computers more useful, to understand the principles that make intelligence possible" (Patrick Winston, 1984)
  - "the study of how to make computers do things at which, at the moment, people are better" (Elaine Rich, 1983)
  - "the part of science concerned with designing intelligent computer system, that is, systems that exhibit the characteristics we associate with intelligence in human behavior-understanding language, learning, reasoning, solving problems, and so on" (The AI Handbook, 1981)

- AI's goal
  - Long term goal: To develop machines that can do the things as well as human can, or possibly even better (Engineering goal)
  - To understand the behavior whether it occurs in machines or in humans or other animals (Scientific goal)

- Scopes of AI

Humanly thinking system[1]	Rationally thinking system[3]
Humanly acting system[2]	Rationally acting system[4]

1. HTS: Cognitive Science
   - "The exciting new effort to make computers think.. *Machines with minds*, in the full and literal sense." (Haugeland 1985)
   - Introspection & psychological experiment
   - GPS [Newell&Simon]
2. HAS
   - "The art of creating machines that perform functions that require intelligence when performed by people."(Kurzweil 1990)
   - Turing Test
     - NLP, Knowledge Representation, Automated Reasoning, Machine Learning
3. RTS: Laws of Thought
   - The study of mental faculties through the use of computational models." (Charniak and McDermott 1985)
   - Logics (Syllogism)
4. RAS: Doing the right things
   - "Computational Intelligence is the study of the design of intelligent agents." (Poole *et al.*, 1998)
   - Agent

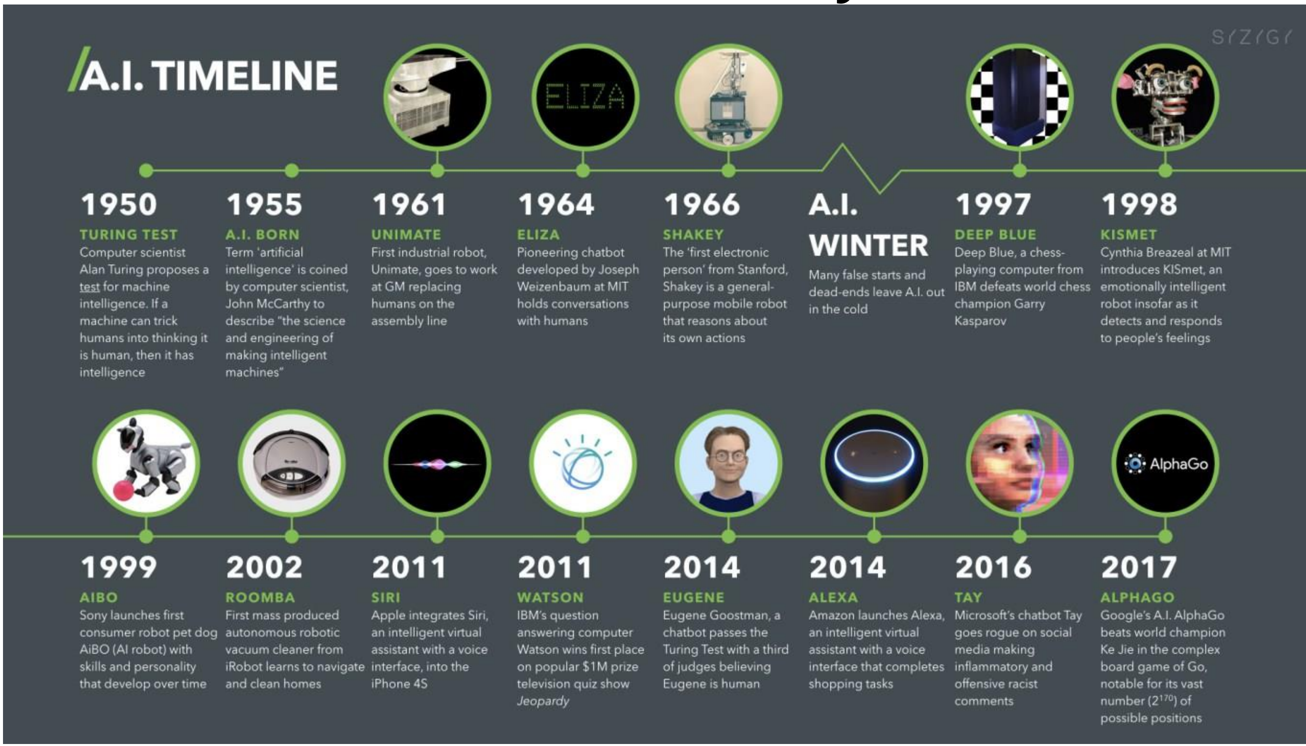
Old categorization of AI area

[Zhang 98]

Symbolic AI	Biological AI
1943: Production rules	1943: McCulloch-Pitt's neurons
1956: "Artificial Intelligence"	1959: Perceptron
1958: LISP AI language	1965: Cybernetics
1965: Resolution theorem proving	1966: Simulated evolution
	1966: Self-reproducing automata
1970: PROLOG language	1975: Genetic algorithm
1971: STRIPS planner	
1973: MYCIN expert system	1982: Neural networks
1982-92: Fifth generation computer systems project	1986: Connectionism
1986: Society of mind	1987: Artificial life
	1992: Genetic programming
1994: Intelligent agents	1994: DNA computing

Deep learning

AI history



Source: digitalwellbeing.org

추론을 통한 탐색 -> 머신 러닝 -> 딥러닝

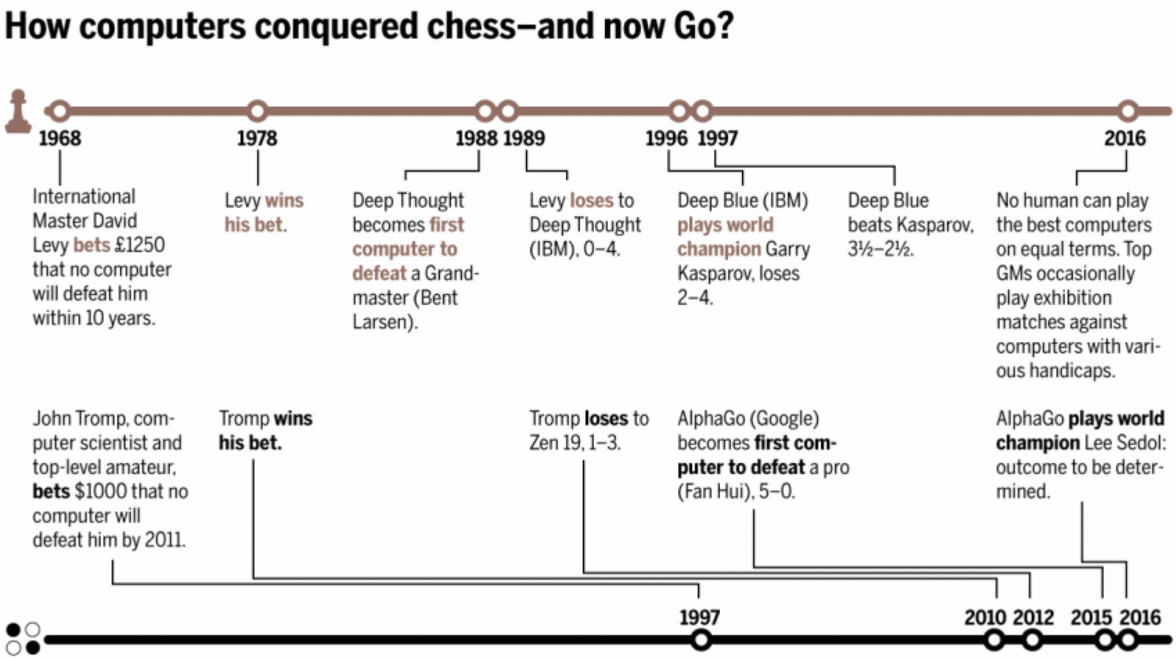
Brain vs. Computer

Attributes	Human	AI
Mass information handling		
Sensing		
Creativeness and imagination		
Learning from past experience		
Memory		
Computation		
Adaptiveness		
Various information source		
Information transfer		
Cost required to obtain knowledge		

컴퓨터가 못 하는 일

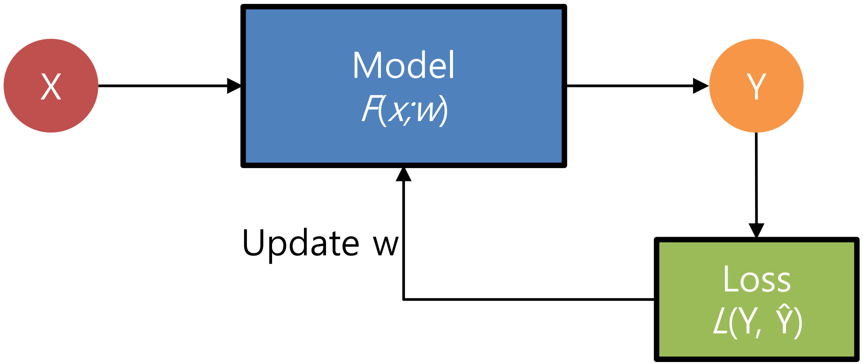
1. 컴퓨터는 생명이 없다. Machines do not have life, as they are mechanical. On the other hand, humans are made of flesh and blood; life is not mechanical for humans.
2. 컴퓨터는 감정이 없다. Humans have feelings and emotions, a can express these emotions. Machines have no feelings and emotions. They st work as per the details fed into their mechanical brain
3. 컴퓨터는 원천
machines cannot
4. 컴퓨터는 상황
situations, and b
capability.
5. 컴퓨터는 양심
machines just pe
6. 컴퓨터는 자신
activities as per
artificial intellige

AI and Go


How computers conquered chess—and now Go?
1968
1978
1988 1989
1996 1997
2016
International Master David Levy bets £1250 that no computer will defeat him within 10 years.
Levy wins his bet.
Deep Thought becomes first computer to defeat a Grandmaster (Bent Larsen).
Levy loses to Deep Thought (IBM), 0–4.
Deep Blue (IBM) plays world champion Garry Kasparov, loses 2–4.
Deep Blue beats Kasparov, 3½–2½.
No human can play the best computers on equal terms. Top GMs occasionally play exhibition matches against computers with various handicaps.
John Tromp, computer scientist and top-level amateur, bets $1000 that no computer will defeat him by 2011.
Tromp wins his bet.
Tromp loses to Zen 19, 1–3.
AlphaGo (Google) becomes first computer to defeat a pro (Fan Hui), 5–0.
AlphaGo plays world champion Lee Sedol: outcome to be determined.
1997
2010 2012
2015 2016

- Can robot love?
- Can human being love robot?

How ML works



- Statistical view
  - 우도(likelyhood)를 최대화하는 파라미터 탐색

$$arg \max_{w} P(Y|X;w)$$

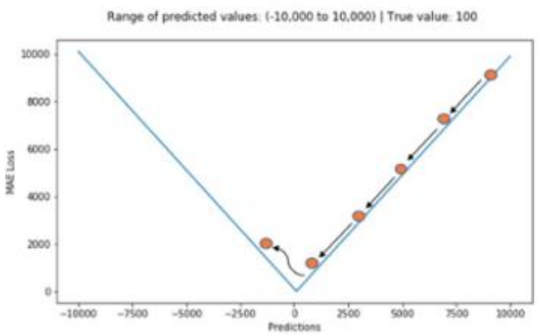
Loss function

- MAE (Mean Absolute Error)
- MSE (Mean Squared Error)
- RMSE (Rooted Mean Squared Error)
- Cross Entropy
- KLD

MAE

- Mean Absolute Error

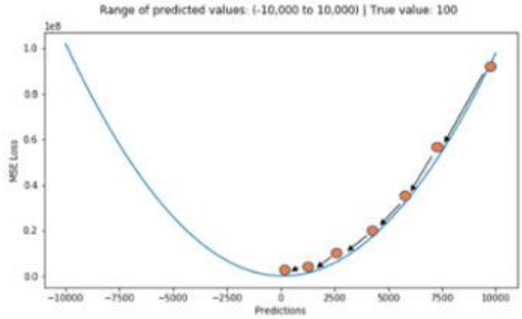
$$L = |y - f(x)|$$



MSE and RMSE

- Mean Squared Error

$$L = (y - f(x))^2$$



- Rooted Mean Squared Error

$$L = \sqrt{(y - f(x))^2}$$

Huber Loss

- Combination of MSE and MAE

$$L = \begin{cases} \frac{1}{2}(y - f(x))^2, if \ |y - f(x)| \leq \delta \\ \delta|y - f(x)| - \frac{1}{2}\delta^2, otherwise \end{cases}$$

  - Quadratic for smaller error
  - Linear otherwise

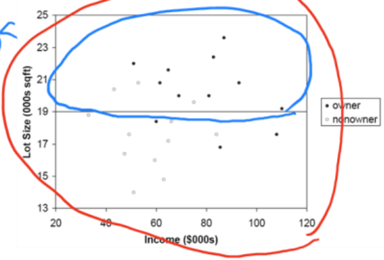
Cross Entropy

- Entropy: (최소)정보량의 평균
  - $entropy = -\sum_{k=1}^{m} p_k \log_2 p_k$
  - 정보량 $I(x) = -\log p(x)$

$Entrp(A) = -(\frac{1}{2}\cdot\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2})$
$= 1$

순도증가. 무질서도 감소

$Entrp(A) = -(\frac{3}{4}\cdot\log\frac{3}{4} + \frac{1}{4}\log\frac{1}{4}) <$
$= 0.811278$

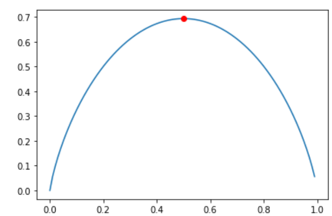


- Cross Entropy:
  - $H(P,Q) = \sum_i p_i \log_2 \frac{1}{q_i} = -\sum_i p_i \log_2 q_i$

- Binary CE

$$L = -y \cdot \log f(x) - (1-y) \cdot \log(1 - f(x))$$

p: usually sigmoid function is used



- Multi-class CE

$$L = -\sum_i y_i \cdot \log p_i$$

Cross entropy vs. MSE

Cross entropy	MSE
-범주형 데이터 예측 시 주로 사용 -각 클래스에 속할 확률을 의미하는 확률분포를 이용해 실제분포와의 엔트로피의 차이를 구한 것.	-수치형 데이터 예측 시 주로 사용 -정답값과 예측값의 차이를 제곱하고 평균을 내준 것.
딥러닝에서 쓰이는 대표적인 손실 함수	

KL-Divergence

- The difference between two distributions
- 어떤 이상적인 분포에 대해, 그 분포를 근사하는 다른 분포를 사용해 샘플링을 한다면 발생할 수 있는 정보 엔트로피 차이를 계산

$$D_{KL}(P||Q) = -\sum_{x}\big(P(x).logQ(x) - P(x).logP(x)\big) = H(P,Q) - H(P,P)$$

$H(P,P)\ is\ the\ entropy\ of\ P$

$H(P,Q)\ is\ the\ cross-entropy\ of\ P\ and\ Q$

KLD(Kullback Leibler Divergence)

- In mathematical statistics, the Kullback–Leibler divergence,(also called relative entropy), is a measure of how one probability distribution is different from a second, reference probability distribution.

P: Distribution P represents the data, the observations, or a probability distribution precisely measured.

Q: Distribution Q represents instead a theory, a model, a description or an approximation of P

- The Kullback–Leibler divergence is then interpreted as the **average difference of the number of bits** required for encoding samples of P using a code optimized for Q rather than one optimized for P.

$$
\begin{aligned}
H(p,q) &= -\sum_i p_i \log q_i \\
&= -\sum_i p_i \log q_i \overbrace{\underbrace{- \sum_i p_i \log p_i}_{=H(p)} + \sum_i p_i \log p_i}^{=0} \\
&= H(p) + \sum_i p_i \log p_i - \sum_i p_i \log q_i \\
&= H(p) + \sum_i p_i \log \frac{p_i}{q_i}
\end{aligned}
$$

p의 엔트로피에 $\boxed{\text{이만큼}}$ 더해진 것이 cross entropy 가 됩니다

이만큼 더해지는 것이 무엇일까요?

→ 바로 분포 p와 분포 q의 정보량 차이 입니다

→ 이것이 바로 KL-divergence 입니다

- 출처: https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence

KLD(Kullback Leibler Divergence)

- $KL(p||q)$

$= H(p,q) - H(p)$

$= -\int p(x) lnq(x) dx - (-\int p(x) lnp(x) dx) = -\int p(x) \ln\left\{\frac{q(x)}{p(x)}\right\} dx$

- $H(p,q) = -\int p(x) lnq(x) dx$
- $H(p) = -\int p(x) lnp(x) dx$
  - p : $true\ distribution$
  - q : $predicted\ distribution$
- $KL(p||q) \neq KL(q||p)$
- p와 q가 같다면 $KL(p||q) = 0$

example

- 실제 사면체 주사위의 확률분포 p= $(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4})$

  추정한 사면체 주사위의 확률 분포 q = $(\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{8})$라 가정.

  - 추정한 확률분포 q를 통해 코딩했을 경우
    - 코딩결과: (0, 10, 110, 111)
    - Entropy = Average code length = 2.25 => **Cross entropy!**
  - 실제 분포 p를 통한 최적의 코딩을 했을 경우
    - 코딩결과: (00,01,10,11)
    - Entropy = Average code length = 2

모델링한 p와 q가 달라서 엔트로피 차이가 발생 => **KL - divergence**

$$(-\sum_x p(x)\log_2 q(x)) - (-\sum_x p(x)\log_2 p(x))) = -\sum_x p(x)\log_2 \frac{q(x)}{p(x)}$$

= 2.25 – 2 = 0.25

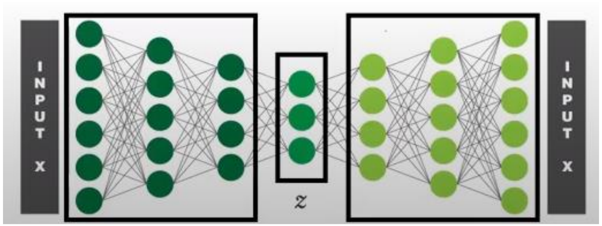
- VAE
  - The structure is the same as autoencoder
  - Finding the distribution of z

$$x = g_\theta(z)$$

$$p(x) = p(x|z)$$

$$p(x) = E_{z\sim p_\theta(x)}[p(x|z)]$$

$$p(x) = E_{z\sim p_\theta(z|x)}[p(x|z)] \cong E_{z\sim q_\emptyset(z|x)}[p(x|z)]$$



$q_\emptyset(.)$ $p_\theta(.)$

ELBO

$$\log(p(x)) = \log\left(\int p(x,z)dz\right) = \log\left(\int p(x|z)p(z)dz\right)$$

$$\log(p(x)) = \log\left(\int p(x|z)\frac{p(z)}{q_\phi(z|x)}q_\phi(z|x)dz\right)$$

$$\log(p(x)) \geq \int \log\left(p(x|z)\frac{p(z)}{q_\phi(z|x)}\right)q_\phi(z|x)dz$$

By Jensen's Inequality

$$\log(p(x)) \geq \int \log(p(x|z))q_\phi(z|x)dz - \int \log\left(\frac{q_\phi(z|x)}{p(z)}\right)q_\phi(z|x)dz$$

$$\boxed{ELBO(\phi) = \mathbb{E}_{q_\phi(z|x)}[\log(p(x|z))] - KL(q_\phi(z|x)\|p(z))}$$

$$\int \log(p(x|z))q_\phi(z|x)dz = \mathbb{E}_{q_\phi(z|x)}[\log(p(x|z))]$$

$$\int \log\left(\frac{q_\phi(z|x)}{p(z)}\right)q_\phi(z|x)dz = KL(q_\phi(z|x)\|p(z))$$

KLD

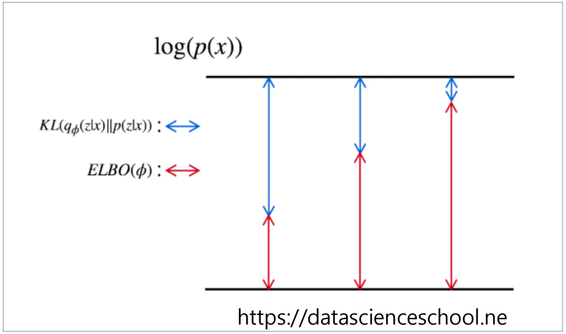
$$\log(p(x)) = \int \log(p(x)) q_\phi(z|x) dz$$

$$\begin{aligned}\log(p(x)) &= \int \log\left(\frac{p(x,z)}{p(z|x)}\right) q_\phi(z|x) dz \\ &= \int \log\left(\frac{p(x,z)}{q_\phi(z|x)} \cdot \frac{q_\phi(z|x)}{p(z|x)}\right) q_\phi(z|x) dz \\ &= \underbrace{\int \log\left(\frac{p(x,z)}{q_\phi(z|x)}\right) q_\phi(z|x) dz}_{\text{ELBO}} + \underbrace{\int \log\left(\frac{q_\phi(z|x)}{p(z|x)}\right) q_\phi(z|x) dz}_{D_{KL}}\end{aligned}$$

우리가 minimize하고 싶은 것

$$\begin{aligned}ELBO &= \int_z \log\left\{\frac{p(z,x)}{q_\phi(z|x)}\right\} q_\phi(z|x)\, dz = \int_z \log\left\{\frac{p(z)\,p(x|z)}{q_\phi(z|x)}\right\} q_\phi(z|x)\, dz \\ &= \int_z \log\{p(x|z)\} q_\phi(z|x)\, dz - \int_z \log\left\{\frac{q_\phi(z|x)}{p(z)}\right\} q_\phi(z|x)\, dz = \underbrace{E_{q_\phi(z|x)}[\log\{p(x|z)\}] - D_{KL}(q_\phi(z|x) \,\|\, p(z))}_{\text{ELBO; Derivation1 의 것과 동일}}\end{aligned}$$

$$\therefore \quad \begin{aligned}\log p(x) &= E_{z\sim q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x)\,||\,p(z)) + D_{KL}(q(z|x)\,||\,p(z|x)) \\ &= ELBO + D_{KL}(q(z|x)\,||\,p(z|x))\end{aligned}$$



https://datascienceschool.net

$$L = -E_{z\sim q(z|x)}[\log p(x|z)] + D_{KL}(q(z|x)\,||\,p(z))$$

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

$$\min \quad f(x)$$

Optimization form

- Unconstrained optimization
  - arg max

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

- Constrained Optimization

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } \; c_j(x) = 0 \\ &\qquad\qquad\quad\;\; c_k(x) \geq 0 \end{aligned}$$

- 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.

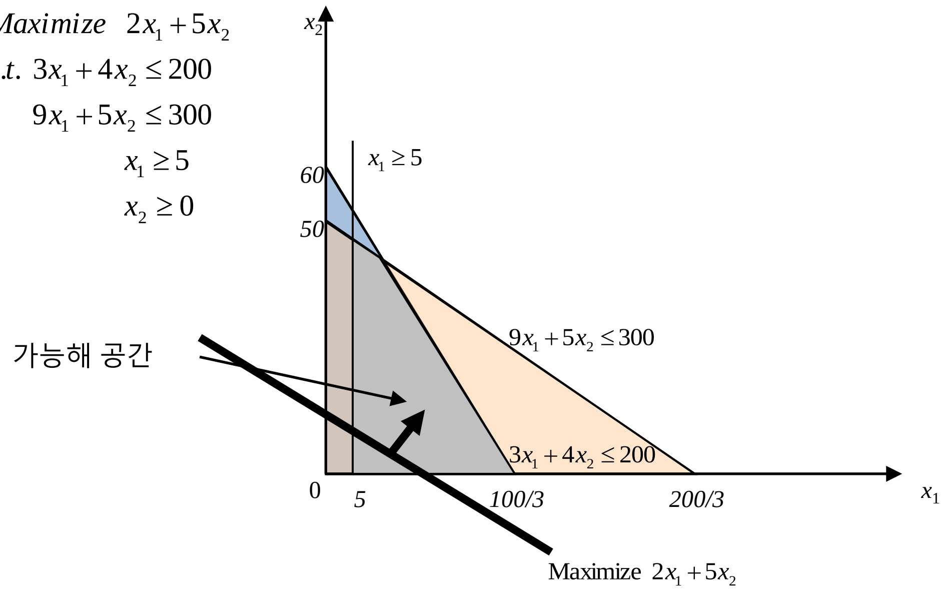
결정 변수: 제품 A의 생산량 => x_1
제품 B의 생산량 => x_2

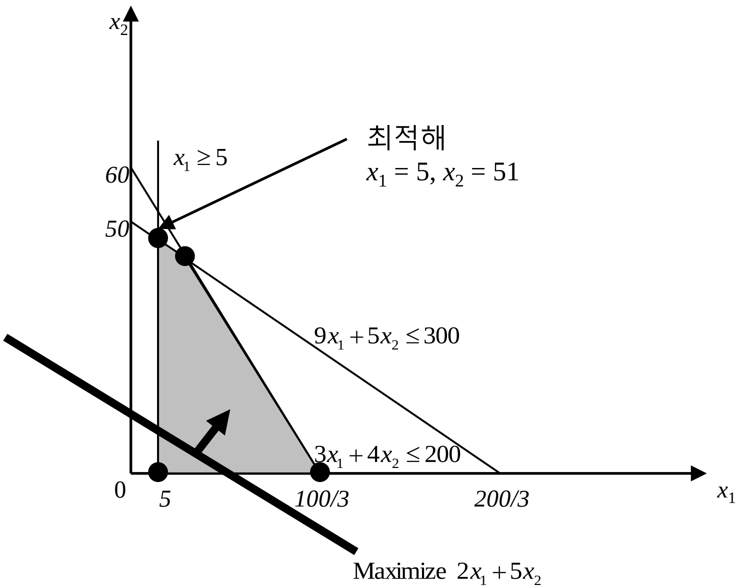
목적함수 → $Maximize \quad 2x_1 + 5x_2$ ← 이익의 최대화

제약식
- $s.t. \quad 3x_1 + 4x_2 \le 200$ ← 기계 가동시간 제약
- $9x_1 + 5x_2 \le 300$ ← 재료 사용량 제약
- $x_1 \ge 5$ ← A의 최소 생산량 제약
- $x_2 \ge 0$ ← 비음인 해만을 구함

Geometric way

$$
\begin{aligned}
&\text{Maximize} \quad 2x_1 + 5x_2 \\
&s.t. \quad 3x_1 + 4x_2 \le 200 \\
&\qquad 9x_1 + 5x_2 \le 300 \\
&\qquad\qquad x_1 \ge 5 \\
&\qquad\qquad x_2 \ge 0
\end{aligned}
$$





x_2
최적해
$x_1 = 5, x_2 = 51$
$x_1 \geq 5$
60
50
$9x_1 + 5x_2 \leq 300$
$3x_1 + 4x_2 \leq 200$
0
5
100/3
200/3
x_1
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?

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

- 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 => [해의 개선] => 최적해

$$\begin{cases} \text{Maximize} & 12x_1 + 15x_2 \\ \text{s.t.} & 4x_1 + 3x_2 \le 12 \\ & 2x_1 + 5x_2 \le 10 \\ & x_1, x_2 \ge 0 \end{cases} \xrightarrow{\text{제약식을 등식화 함}} \begin{cases} \text{Maximize} & 12x_1 + 15x_2 \\ \text{s.t.} & 4x_1 + 3x_2 + x_3 = 12 \\ & 2x_1 + 5x_2 + x_4 = 10 \\ & x_1, x_2, x_3, x_4 \ge 0 \end{cases}$$

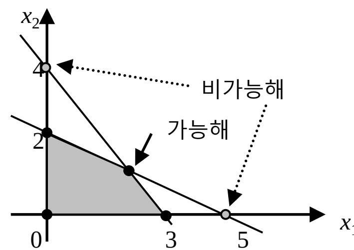
제약식만을 이용한 연립방정식

$$\begin{cases} 4x_1 + 3x_2 + x_3 = 12 \\ 2x_1 + 5x_2 + x_4 = 10 \end{cases}$$ 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)

$\Leftrightarrow$



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

목적함수를 최대화하는 가능해는 1개 뿐 =>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
(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

비가능해

단순히 제약식의 연립방정식만을 이용한 해법은 만일 n개의 결정변수로 이루어진 m개의 제약식을 갖는 문제라면 모두 ${}_nC_m$개의 해를 찾아야 한다. n과 m의 값에 따라 탐색 공간이 지수적으로 증가한다.

=> **비가능해는 탐색하지 않고, 가능해 내에서만 탐색을 하되, 목적함수 값을 꾸준히 증가 시킬 수 있도록 하는 방법이 필요**.

[예] 단체법을 이용한 풀이

$$\begin{aligned}
\text{Maximize} \quad & 4x_1 + 5x_2 + 8x_3 + 11x_4 \\
\text{s.t.} \quad & x_1 + x_2 + x_3 + x_4 \le 15 \\
& 3x_1 + 5x_2 + 10x_3 + 15x_4 \le 105 \\
& 7x_1 + 5x_2 + 3x_3 + 2x_4 \le 120 \\
& x_j \ge 0, \ \forall j
\end{aligned}$$

계산형으로 변형

최소의 목적함수 계수 = -11

초기 기저해

$$\begin{aligned}
z - 4x_1 - 5x_2 - 8x_3 - 11x_4 & & &= 0 \\
x_1 + x_2 + x_3 + x_4 + x_5 & & &= 15 \\
3x_1 + 5x_2 + 10x_3 + 15x_4 & + x_6 & &= 105 \\
7x_1 + 5x_2 + 3x_3 + 2x_4 & & + x_7 &= 120 \\
x_j \ge 0, \ \forall j
\end{aligned}$$

15/1 = 15

105/15 = 7

120/2 = 60

최소의 비율을 갖는 2행이 선택.

2행 4열의 변수를 중심으로 pivot을 실시

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

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

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

Pivoting

목적함수의 모든 계수값이 0이상이므로
현재의 해가 최적해가 된다.

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

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.

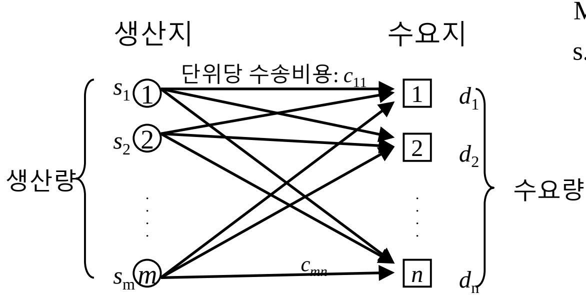
min{ 8/(4/5), 7/(1/5), 106/(33/5) } = 8/(4/5) = 10.
따라서 $r = 1$가 된다.

목적함수의 모든 계수값이 0이상이므로
현재의 해가 최적해가 된다.

대표적 선형계획법 문제들

- 수송 문제(Transportation Problem)

  m개의 생산지로부터 n개의 수요지로 수요량을 만족시키면서, 최소의 비용으로 전달하는 문제



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

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.

야산 \ 아파트 부지	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

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

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

- 식단 문제

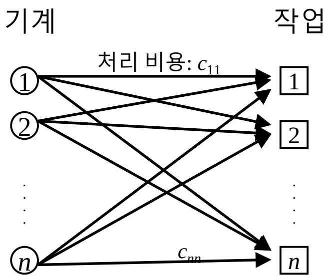
여러 가지 영양분을 지닌 음식들로부터 필수 영양분을 최소의 음식값으로 섭취하는 문제

[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.

재료 / 영양	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	

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

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



만약 기계 i가 작업 j에 할당되면 $x_{ij} = 1$

아니면 $x_{ij} = 0$

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

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

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

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

- 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

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

ORTools

```
!pip install ortools
```

```python
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

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

$\longrightarrow$

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

- Minimum Absolute Value
- Goal Programming
- ….

Convex optimization

- In the constrained optimization,

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

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

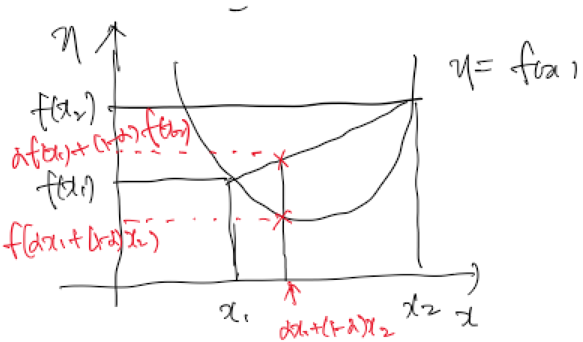
$$c_k(x) \geq 0$$

- 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 로컬옵티멈이 존재하는 데 그보다 더 작은 optimal이 있다면…

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

$f(\underbrace{\alpha x^* + (1-\alpha)x'}) \le \alpha f(x^*) + (1-\alpha) f(x')$ *"f is convex"*

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

$< f(x^*)$

if $\alpha \to 1$ x^* 주변에 $f(x^*)$ 보다 작은 값이 존재 !

Feasible set 안에

Convex Optimization in ML:SVM

- A separating hyperplane can be written as

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

  where $\boldsymbol{W}=\{w_1, w_2, \ldots, w_n\}$ 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:

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

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

- 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 → *Quadratic Programming (QP)* → Lagrangian multipliers

- Objective function

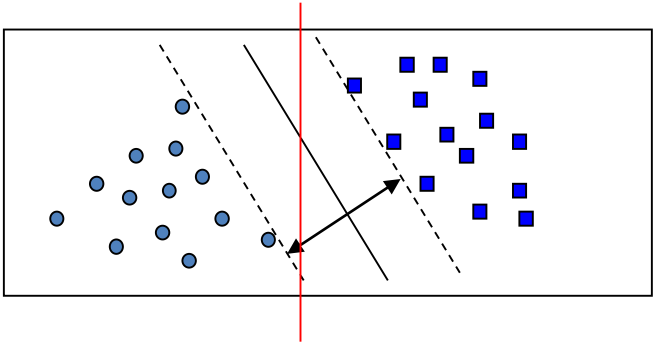
$$b_{i1}: \boldsymbol{w} \cdot \boldsymbol{x} + b = 1$$

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

$$w \cdot (x_1 - x_2) = 2$$

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

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



- Constraints

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

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

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

- SVM Learning

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

s.t. $$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1, i = 1,2, \ldots, N$$

- Convex optimization => Lagrangian

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

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

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

- Dual Lagrangian

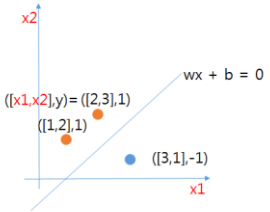
$$L_D = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j}^{N} \lambda_i \lambda_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j$$

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

$$\lambda_i \geq 0$$

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

최적해가 되기 위한 조건



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

Then differentiate the above equations in terms of w and b

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

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

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

Apply (1) and (2) into L

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

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

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

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

$= -\frac{1}{2}(\sum \alpha_i y_i x_i \sum \alpha_j y_j x_j) + \sum \alpha_i$

$= \sum \alpha_i - \frac{1}{2}(\sum \alpha_i y_i x_i \sum \alpha_j y_j x_j)$

$= \sum \alpha_i - \frac{1}{2}(\sum \sum \alpha_i \alpha_j y_i y_j x_i x_j)$

$x_1 = (1,2), y_1 = 1$
$x_2 = (2,3), y_2 = 1$
$x_3 = (3,1), y_3 = -1$

By applying into the equations,

$\mathrm{L} = \sum \alpha_i - \frac{1}{2}\sum \alpha_i y_i x_i(\alpha_1 y_1 x_1 + \alpha_2 y_2 x_2 + \alpha_3 y_3 x_3)$

$= \sum \alpha_i - \frac{1}{2}\sum \alpha_i y_i x_i(\alpha_1 x_1 + \alpha_2 x_2 - \alpha_3 x_3)$

$= \sum \alpha_i - \frac{1}{2}(\alpha_1 y_1 x_1(\alpha_1 x_1 + \alpha_2 x_2 - \alpha_3 x_3) - \frac{1}{2}(\alpha_2 y_2 x_2(\alpha_1 x_1 + \alpha_2 x_2 - \alpha_3 x_3)) - \frac{1}{2}(\alpha_3 y_3 x_3(\alpha_1 x_1 + \alpha_2 x_2 - \alpha_3 x_3))$

$= \sum \alpha_i - \frac{1}{2}(\alpha_1 x_1(\alpha_1 x_1 + \alpha_2 x_2 - \alpha_3 x_3) - \frac{1}{2}(\alpha_2 x_2(\alpha_1 x_1 + \alpha_2 x_2 - \alpha_3 x_3)) + \frac{1}{2}(\alpha_3 x_3(\alpha_1 x_1 + \alpha_2 x_2 - \alpha_3 x_3))$

$= \sum \alpha_i - \frac{1}{2}(5\alpha_1^2 + 13\alpha_2^2 + 10\alpha_3^2 + 16\alpha_1\alpha_2 - 10\alpha_1\alpha_2 - 18\alpha_2\alpha_3)$

$= \alpha_1 + \alpha_2 + \alpha_3 - \frac{1}{2}(5\alpha_1^2 + 13\alpha_2^2 + 10\alpha_3^2 + 16\alpha_1\alpha_2 - 10\alpha_1\alpha_2 - 18\alpha_2\alpha_3)$

$\frac{L}{\partial \alpha_1} = 1 + \frac{1}{2}(-10\alpha_1 - 16\alpha_2 + 10\alpha_3) = 0$

Remember we know that $\sum \alpha_i y_i = 0$, so $\alpha_1 + \alpha_2 = \alpha_3$

$1 + \frac{1}{2}(-10\alpha_1 - 16\alpha_2 + 10(\alpha_1 + \alpha_2)) = 0$

$\alpha_2 = \frac{1}{3}$

$\frac{L}{\partial \alpha_3} = 1 - \frac{1}{2}(26\alpha_2 + 16\alpha_1 - 18\alpha_3) = 0$

$\frac{L}{\partial \alpha_3} = 1 - (13\alpha_2 + 8\alpha_1 - 9\alpha_3) = 0$

$8\alpha_1 - 9\alpha_3 = -\frac{10}{3}$

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

$w = (-1, 1)$
Our plane is $y = wx + b$, therefore $y = (-a, a)(x_1, x_2) + b = -ax_1 + ax_2 + b$

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 z,

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

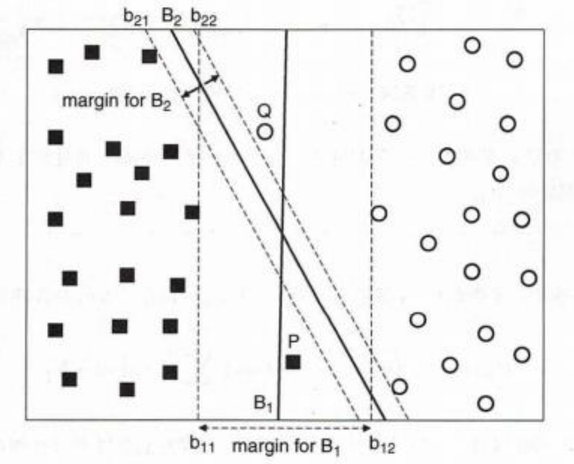
  - If f=1, z will be classified as positive

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

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

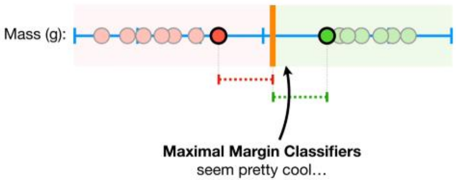


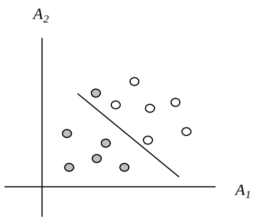
$$\min_{\boldsymbol{w}} f(\boldsymbol{w}) = \frac{\|\boldsymbol{w}\|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i\right)^k$$

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

SVM—Linearly Inseparable

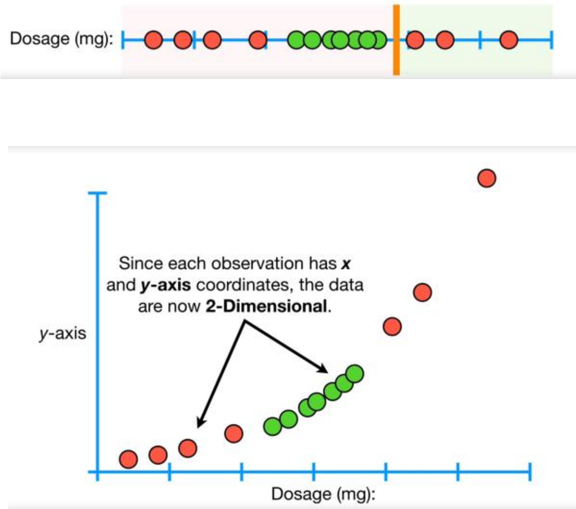
- Transform the original input data into a higher dimensional space





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 Z using the mappings $\phi_1(X) = x_1, \phi_2(X) = x_2, \phi_3(X) = x_3, \phi_4(X) = (x_1)^2, \phi_5(X) = x_1x_2$, and $\phi_6(X) = x_1x_3$. A decision hyperplane in the new space is $d(\mathbf{Z}) = \mathbf{W}\mathbf{Z} + b$, 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,

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



- 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., $K(\mathbf{X}_i, \mathbf{X}_j) = \Phi(\mathbf{X}_i)\,\Phi(\mathbf{X}_j)$
- Typical Kernel Functions

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

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

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

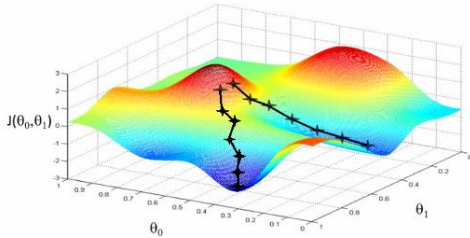
- SVM can also be used for classifying multiple (> 2) classes and for regression analysis (with additional parameters)

Using differentiation

- Minimum (Maximum) value is on the point that

$$f'(x) = 0$$

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

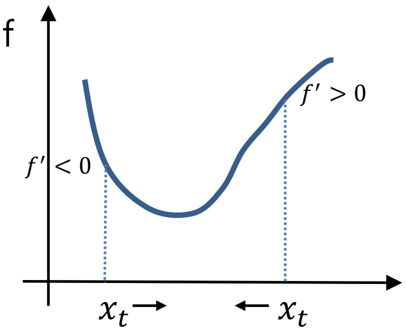


출처: http://blog.datumbox.com/tuning-the-learning-rate-in-gradient-descent/

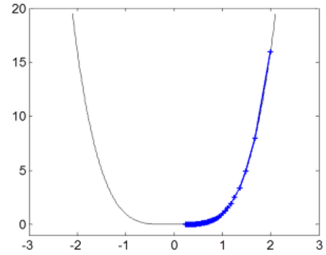
Gradient descent

- Update x value starting from x_0

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



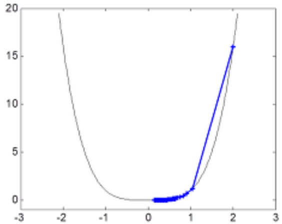
$f(x) = x^4$



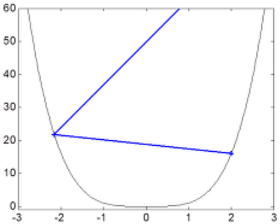
$x_0 = 2.0$
$x_1 = x_0 - \lambda f'(x_0) = 1.6800$
$x_2 = x_1 - \lambda f'(x_1) = 1.4903$
$x_3 = x_2 - \lambda f'(x_2) = 1.3579$
$x_4 = x_3 - \lambda f'(x_3) = 1.2578$
$x_5 = x_4 - \lambda f'(x_4) = 1.1782$
$x_6 = x_5 - \lambda f'(x_5) = 1.1128$
$x_7 = x_6 - \lambda f'(x_6) = 1.0576$
$x_8 = x_7 - \lambda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

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



$\lambda = 0.03$



$\lambda = 0.13$

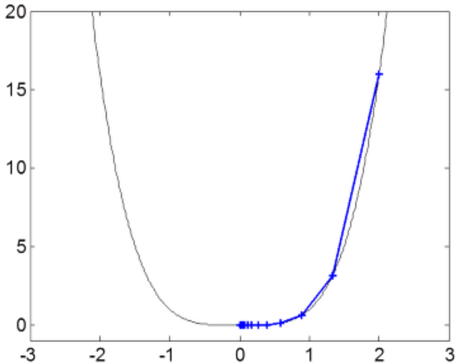
Using secondary derivatives

- Update using the equation: 보다 빠르게 수렴

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

- Problems: 변곡점에서는 정의되지 않음

$f''(x_t)$=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
- https://darkpgmr.tistory.com/149