Optimization Overview
: Convex부터 Bayesian까지

DevFest Pangyo 2018
: AI FOR EVERYONE
2018.11.23

이규영
mineatte@gmail.com

Objective

- 최적화 문제와 용어
- 최적화 문제의 분류와 범주 이해
- 각 최적화 문제에 대한 대표적인 솔루션 소개
- 머신러닝과 최적화의 교집합

최적화 문제 정의

제약 조건 하에서 x에 대한 목적함수 f(x)의
최대치와 최소치를 찾는 것

Objective function : $\min_{x} f(x)$ or $\max_{x} f(x)$

Equality constraints : $h_i(x) = 0$

Inequality constraints : $g_i(x) \geq 0$

Deterministic vs Stochastic

deterministic optimization

최적화 모델을 구성하고 있는 모든
파라미터들이 고정되어 있다고 보는 것

Approach	Key Features
Gradient search	Move locally in most promising direction, according to gradient
Random search	Move randomly to new point, no information used in search
Simulated annealing	Sometimes move in locally worse directions, to avoid being trapped in local extrema
Genetic algorithms and scatter search	Population based, generates new members using (local) operations on attributes of current members
Tabu search	Use memory (search history) to avoid tabu moves
Neural networks	(Nonlinear) function approximation
Math programming	Powerful arsenal of rigorously tested software

stochastic optimization

각 파라미터들은 고정된 값이 아닌
확률분포를 갖는다고 가정하는 것

Approach
Stochastic approximation
Stochastic gradient descent
Scenario optimization
Meta heuristic - Random Search, GA, SA, PSO …

Deterministic Optimization

Deterministic Optimization Classification

Variables	Functions	Classification
All Continuous	All Linear	**Linear Programming**
	One or more nonlinear	**Non-linear Programming**
All discrete	All Linear	**Integer Programming**
	One or more nonlinear	**Integer Non-linear Programming**
Discrete & Continuous	All Linear	**Mixed Integer Programming**
	One or more nonlinear	**Mixed Integer Non-linear Programming**

또한, Non-linear 문제는 convex와 non-convex, constraint와 unconstraint problem 으로 구분되어질 수 있음

Linear Programming

목적 함수와 제약식 모두가 선형인 최적화

목적 : $$\max_{\mathbf{x}} \ \boldsymbol{c}^T\mathbf{x}$$

조건 : $$\boldsymbol{A}\mathbf{x} \le \boldsymbol{b}$$

$$\mathbf{x} \ge 0$$

$\boldsymbol{A}$: $m\mathbf{x}n$ matrix
$\boldsymbol{c}$, $\mathbb{x}$: n-dim vector
$\boldsymbol{b}$: m-dim vector

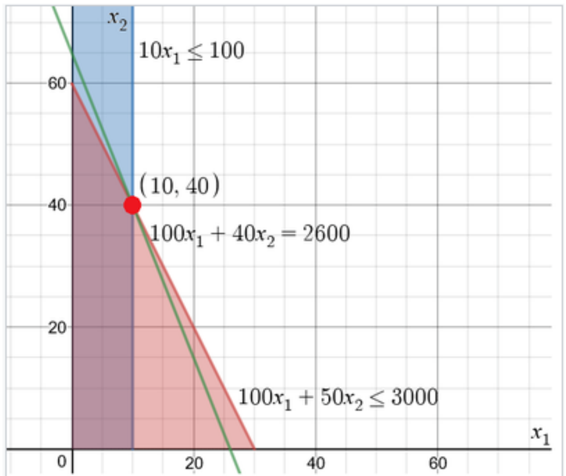
A가 두 가지 종류의 빵을 판매하는데, 초코빵을 만들기 위해서는 밀가루 100g과 초콜릿 10g이 필요하고 밀빵을 만들기 위해서는 밀가루 50g이 필요하다. 재료비를 제하고 초코빵을 팔면 100원이 남고 밀빵를 팔면 40원이 남는다. 오늘 A는 밀가루 3000g과 초콜릿 100g을 재료로 갖고 있다. 만든 빵을 전부 팔 수 있고 더 이상 재료 공급을 받지 않는다고 가정한다면, 이익을 극대화 하기 위해서 어떤 종류의 빵을 얼마나 만들어야 하는가?

목적: $$\max_{\mathbf{x}} 100x_1 + 40x_2$$

조건: $$100x_1 + 50x_2 \le 3000$$

$$10x_1 \le 100,$$

$$x_1, x_2 \ge 0.$$



Linear Programming

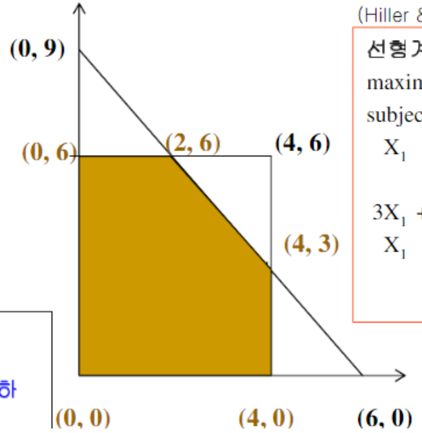
Simplex 방법

- 모든 함수들이 선형이기 때문에 등호 제약 조건이나 부등호 제약 조건에 의해 정의된 가능해 공간(feasible region)은 볼록(convex)집합임
- 따라서 선형 계획 문제는 볼록 계획 문제이고, 만일 하나의 최적해가 존재한다면 그것은 전역 최적해(global optimum)임.

  - 모퉁이점 가능해(Corner-point feasible solution: CPFS)
  - 모퉁이점 불가능해:3개 (0,9), (4, 6), (6,0)

CPF해	이웃하는 CPF해(adjacent BFS)
(0, 0)	(0, 6) 과 (4, 0)
(0, 6)	(2, 6) 과 (0, 0)
(2, 6)	(4, 3) 과 (0, 6)
(4, 3)	(4, 0) 과 (2, 6)
(4, 0)	(0, 0) 과 (4, 3)

최적성 기준
최적해가 최소한 하나 있는 LP에서,
Z의 값으로 보아 현재 CPF해보다 더 나은 이웃하는 CPF해가 없으면 현재 CPF해가 최적해이다.



(Hiller & Lieberman)

선형계획모형

maximize $Z = 3X_1 + 5X_2$

subject to

$X_1 \leq 4$ ----(1)

$2X_2 \leq 12$ ----(2)

$3X_1 + 2X_2 \leq 18$ ----(3)

$X_1 \geq 0$ ----(4)

$X_2 \geq 0$ ----(5)

Linear Programming tools

TOOLS	Descriptions
CPLEX	The IBM ILOG CPLEX Optimizer solves integer programming problems, very large linear programming problems using either primal or dual variants of the **simplex method** or the barrier interior point method, **convex and non-convex quadratic programming problems**, and **convex quadratically constrained problems**
MATLAB	A general-purpose and matrix-oriented programming-language for numerical computing. **Linear programming** in MATLAB requires the Optimization Toolbox in addition to the base MATLAB product
Excel Solver Function	A **nonlinear solver** adjusted to spreadsheets in which function evaluations are based on the recalculating cells. Basic version available as a standard add-on for Excel

이 외에도 python, R, java에서 optimization을 위해 사용할 수 있는 library들이 다양하게 개발되어 있음

Nonlinear Programming

Problem Definition

$$\min \quad f(x) \qquad \leftarrow \textbf{Objective function}$$

$$\text{Subject to} \quad h_i(x) = 0, \quad i = 1,2,\ldots,m \qquad \leftarrow \textbf{Equality constraint}$$

$$g_j(x) \geq 0, \quad j = 1,2,\ldots,p \qquad \leftarrow \textbf{Inequality constraint}$$

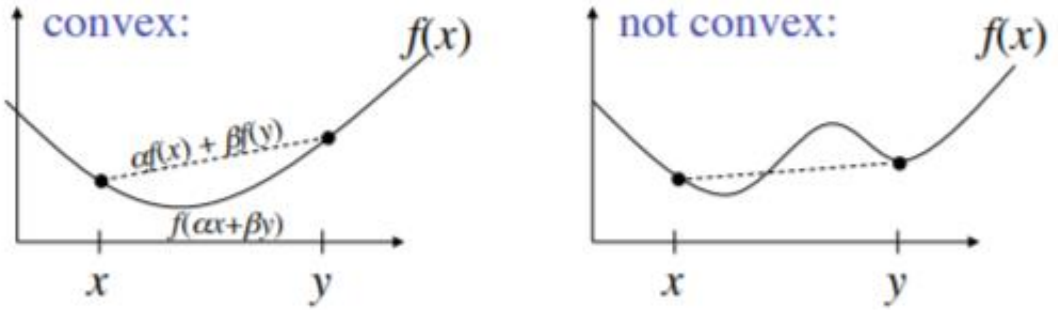
- $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is an n-dim vector with n decision variables.
- $f(\boldsymbol{x})$, $g_i(\boldsymbol{x})$, and $h_j(\boldsymbol{x})$ are real valued twice continuously differentiable functions.

Nonlinear Programming

Convex programming

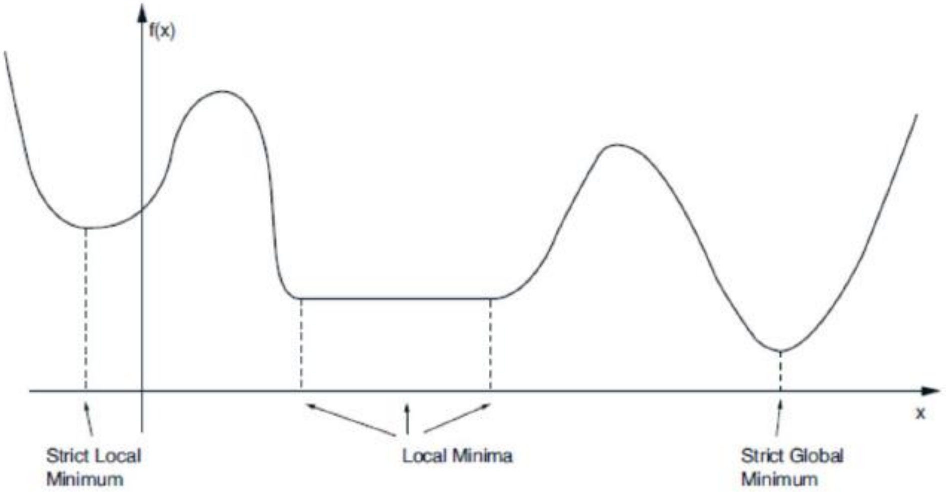
All the functions f_i (i=0…m) are convex:

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y) \quad \text{where} \quad \begin{array}{c} \alpha + \beta = 1 \\ \alpha, \beta \in [0,1] \end{array}$$



Nonlinear Programming

Local vs global



Strictly Convex : $f(\alpha x + \beta y) < \alpha f(x) + \ \beta(y)$

Nonlinear Programming

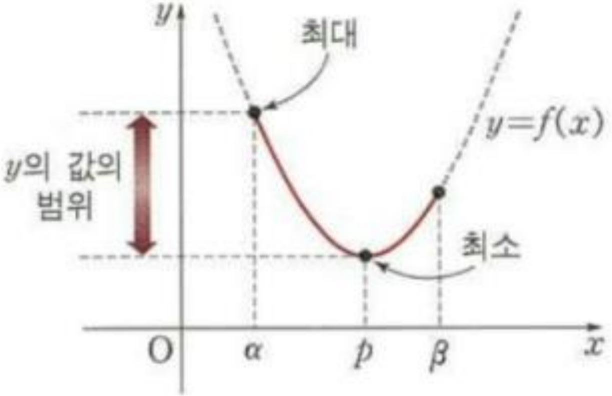
Convex programming

Unconstrained Problem :

$$\nabla f(\boldsymbol{x}^*) = \begin{bmatrix} \partial f/\partial x_1 \\ \vdots \\ \partial f/\partial x_n \end{bmatrix}_{\boldsymbol{x}=\boldsymbol{x}^*} = \mathbf{0}.$$

x^* : Local minimum

함수 최대 최소



2번 미분 가능하며, 이계도함수가 존재할 때,

$f'(x) = 0$ 이고,

$f''(x) = 0$: 변곡점

$f''(x) > 0$: 극소

$f''(x) < 0$: 극대

Nonlinear Programming

Convex programming

1. If the objective function is strictly convex and a global minimum exists, then the global minimum is unique.
2. A local minimum = a global minimum.

Linear Programming
Quadratic Programming
...

→ Karush-Kuhn-Tucker(KKT) Condition

Nonlinear Programming

Convex programming - Quadratic Programming

Quadratic Program(QP)는 목적함수(objective function)가 이차식(convex quadratic)이고, 제약함수(constraint functions)가 모두 affine인 convex optimization problem

the objective of quadratic programming is to find an *n*-dimensional vector **x**

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\mathbf{x}^{\mathrm{T}} Q\mathbf{x} + \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ &\text{subject to} && A\mathbf{x} \leq \mathbf{b}, \end{aligned}$$

- a real-valued, n-dimensional vector c,
- an n × n-dimensional real symmetric matrix Q,
- an m × n-dimensional real matrix A, and
- an m-dimensional real vector b

Nonlinear Programming

Convex programming - KKT condtions

KKT conditions

Solutions can be tested for optimality using Karush-Kuhn-Tucker conditions just as is done for other nonlinear problems:[5]

Condition 1: sum of gradients is zero:

$$\nabla L(x^*, \lambda^*, \mu^*) = q^T + \frac{1}{2}x^T Q + \lambda^{*T} A + \mu^{*T} B$$

Condition 2: all constraints satisfied:

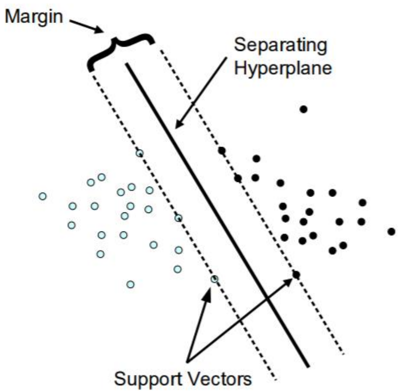
$$Ax^* - a = 0$$
$$Bx^* - b \leq 0$$

Condition 3: complementary conditions:

$$\mu^T Bx - b = 0$$
$$x^T, \lambda^T, \mu^T \geq 0$$

Nonlinear Programming

Convex programming 활용 - Support vector machine



$$\text{minimize } \frac{1}{n}\sum_{i=1}^{n}\zeta_i + \lambda\|w\|^2$$

$$\text{subject to } y_i(w \cdot x_i - b) \geq 1 - \zeta_i \ \text{ and } \ \zeta_i \geq 0, \text{ for all } i.$$

- Quadratic programming
- KKT 조건으로 Optimal을 찾음

Nonlinear Programming

Gradient descent algorithm Applications

신경망 모형은 Cost function을 minimize하는 parameter를 찾는 문제

Model: $H(x) = Wx + b$

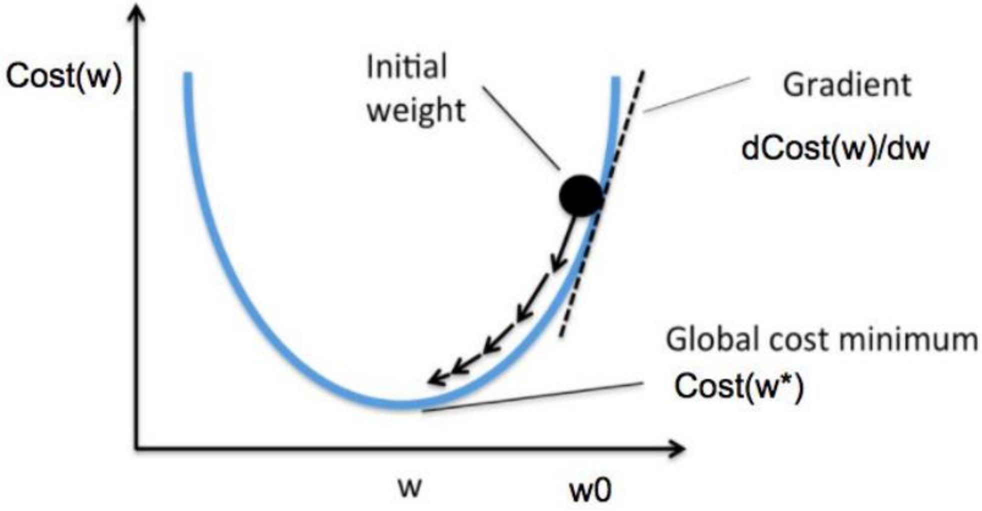
Cost Function: $cost = \frac{1}{m} \sum_{i=1}^{m} (H(x^{(i)}) - y^{(i)})^2$

Minimize Cost: W, b $\underset{W,b}{\text{minimize}}\, cost(W, b)$

Nonlinear Programming

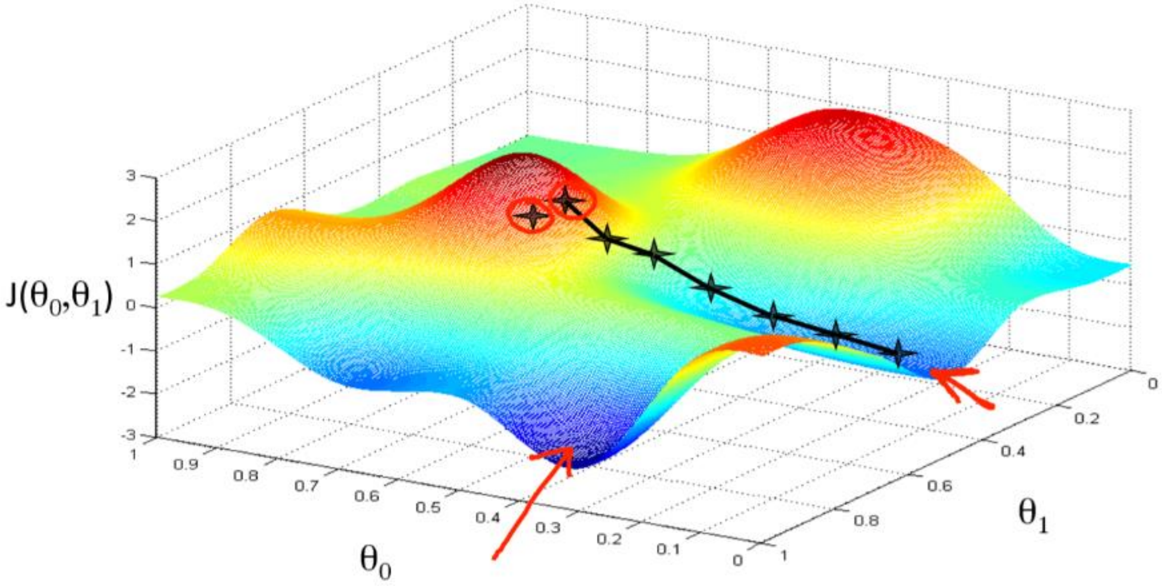
Non-convex : Gradient descent algorithm

$$W := W - \overset{\text{Learning Rate}}{\boxed{\alpha}} \frac{\partial}{\partial W} cost(W)$$



Nonlinear Programming

Non-convex : Gradient descent algorithm



Nonlinear Programming

Gradient descent algorithm

- GD need to calculate the gradients for the whole dataset to perform just one update.
- Very Slow, Big dataset do not fit in memory
- GD does not allow us to update our model online

Stochastic Gradient descent algorithm

- SGD perform a parameter update for each training example
- Much faster than GD
- But High Variance that cause the objective function to fluctuate heavily

Mini-batch Gradient Descent

- This takes the best of both worlds (GD and SGD)
- Fast update, and also stable Convergence

그리고 momentum…

Nonlinear Programming

Non-convex programming

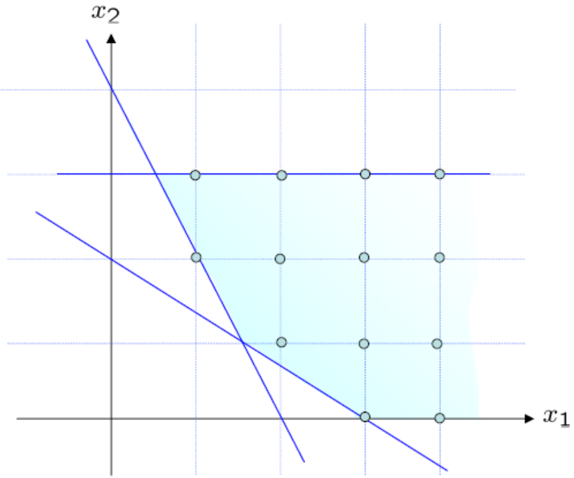
- Gradient Descent Algorithm
- Random search
- Meta Heuristic

…

Integer Programming

Integer programming

$$
\begin{aligned}
\text{min} \quad & x_1 + x_2 \\
\text{s.t.} \quad & \boxed{\begin{aligned} & 2x_1 + x_2 \geq 4 \\ & x_1 + 1.5x_2 \geq 3 \\ & x_2 \leq 3 \\ & x_1,\ x_2 \geq 0 \text{ and integer} \end{aligned}}
\end{aligned}
$$



Stochastic Optimization

Objective function
or
Constraints

← **Random Variables**

Approaches

Stochastic approximation
Stochastic Gradient Descent
Heuristic
- Genetic Algorithm
- Simulated Annealing
- Random Search

...

Heuristic Optimization

휴리스틱의 어원은 라틴어의 'heuristicus' 와 그리스어 'heuriskein' 에서부터 시작되었으며, "찾아내다(find out)" 그리고 발견하다(discover)"라는 의미

최적화에서 메타 휴리스틱 이란
한정된 시간에서 최적의 답을 찾는 것이 아니라
만족할 만한 수준의 해법을 찾는 것

Heuristic Optimization

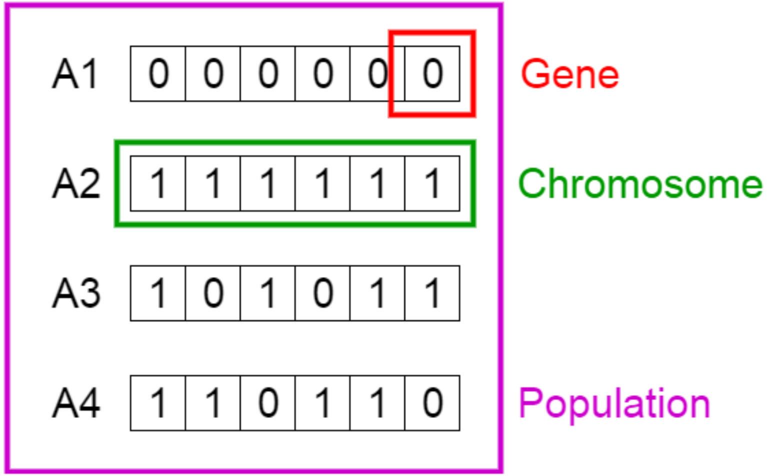
Genetic Algorithm

유전 알고리즘은 자연계의 생물 유전학에 기본 이론을 두며, 병렬적이고 전역적인 탐색 알고리즘으로 5단계로 구성

START
Generate the initial population
Compute fitness
REPEAT
    Selection
    Crossover
    Mutation
    Compute fitness
UNTIL population has converged
STOP

Heuristic Optimization

Genetic Algorithm - Initial population



Hyper parameter : Population

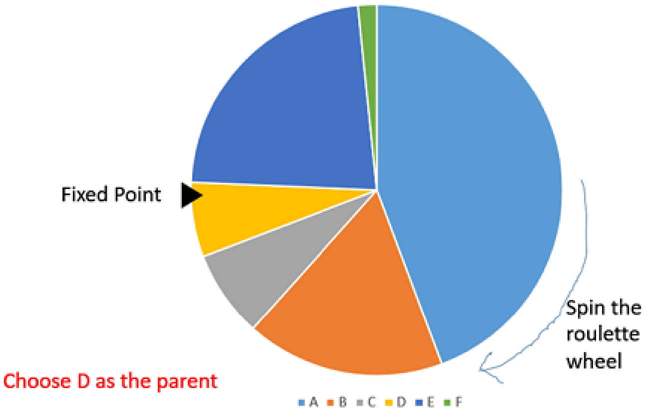
Heuristic Optimization

Genetic Algorithm - Fitness Function & Selection

Fitness Function ≅ Objective Function

: 생존에 얼마나 적합한지를 판단하기 위한 함수

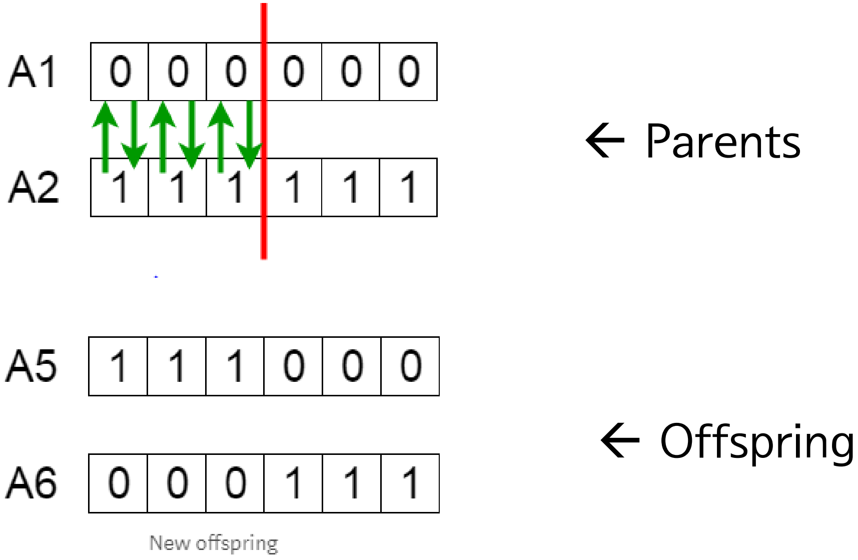
Selection



Chromosome	Fitness Value
A	8.2
B	3.2
C	1.4
D	1.2
E	4.2
F	0.3

Heuristic Optimization

Genetic Algorithm - Crossover



Hyper parameter : crossover 율

Heuristic Optimization

Genetic Algorithm - Mutation

Before Mutation

A5	1	1	1	0	0	0

After Mutation

A5	1	1	0	1	1	0

Hyper parameter : Mutation 율

Heuristic Optimization

Genetic Algorithm - termination

Hyper parameter : 세대수 (repeat iteration 수)

다른 휴리스틱 방법들

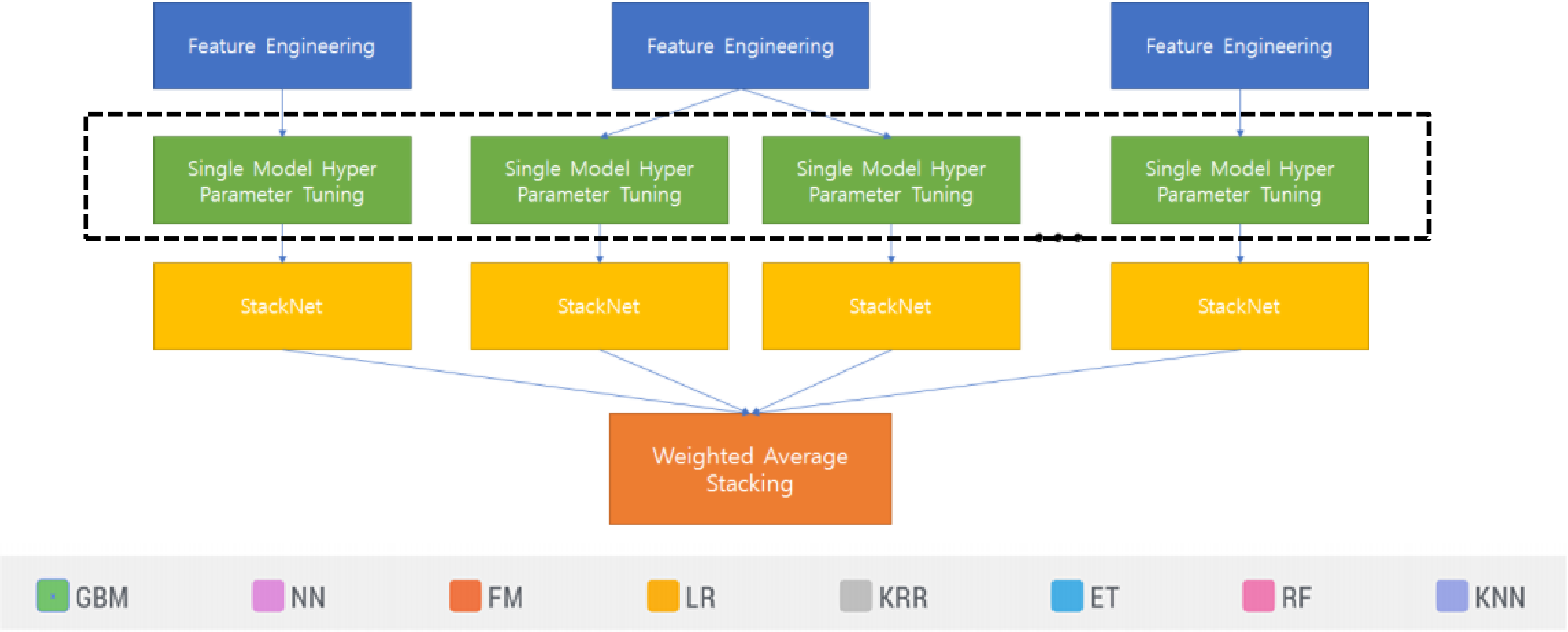
- **Simulated Annealing**
- **Ant colony optimization**
- **Particle swarm optimization**
- **Controlled random search**

휴리스틱의 장점

- Discrete vs. continuous
- Deterministic vs. Stochastic

Bayesian Optimization

Hyper parameter Tuning as Optimization Problem



Xgboost hyper parameter 튜닝 예시

Bayesian Optimization

Hyper parameter Tuning for Optimization Problem

Grid Search	Random Search	Bayesian Optimization
- Range 설정하고 Range 내에서 전체를 탐색	- Range 설정하고 Range 내에서 Random 탐색	- Parameter를 함수로 가정하여 형태를 추정하면서 optimal search

Meta Heuristic

- 해의 보장성 문제
- 해의 보장성을 좋게 하기 위해 Meta Heuristic의 parameter 또한 튜닝해야하는 문제

Bayesian Optimization

일반적인 비선형 최적화 방법론	베이지안 최적화
- 목적함수가 convex거나 연속/미분가능할 때 최적화를 수행하여, 목적함수가 정의되지 않고 구하는데 시간이 오래 걸리는 경우 성능이 저하됨 - 목적함수가 Global mimum을 구하기 어려우며 휴리스틱을 적용할 수 있지만 휴리스틱 적용 시에는 global mimum이 보장되지 않음	- Global mimum을 구하기 위한 방법론 - 목적함수가 Black-box 형태로 간주할 수 있어 불확실성이 포함된 목적함수에도 적용가능함

Bayesian Optimization

논문 요약

- empirical하게 좋은 성능을 보이는 적절한 kernel function과 acquisition function을 제안
- Bayesian optimization의 hyper parameter를 fully Bayesian approach를 통해 전체 optimization에서 한 번에 계산할 수 있는 방법을 제안
- MCMC를 사용해 풀 수 있는 theoretically tractable parallelized Bayesian optimization을 제안

사전 지식

- 베이즈 정리 (Bayes'Theorem)
- Stochastic process (Random process라고도 부른다)
- Gaussian process (GP) & kernel function of GP
- **Bayesian optimization & acquisition function**
- Markov chain Monte Carlo (MCMC)

Bayesian Optimization

Bayesian Optimization

$$x^* = \arg\min_{x \in X} f(x).$$

$f(x)$: input을 넣었을 때 output이 무엇인지만 알 수 있는 black box function이라 가정

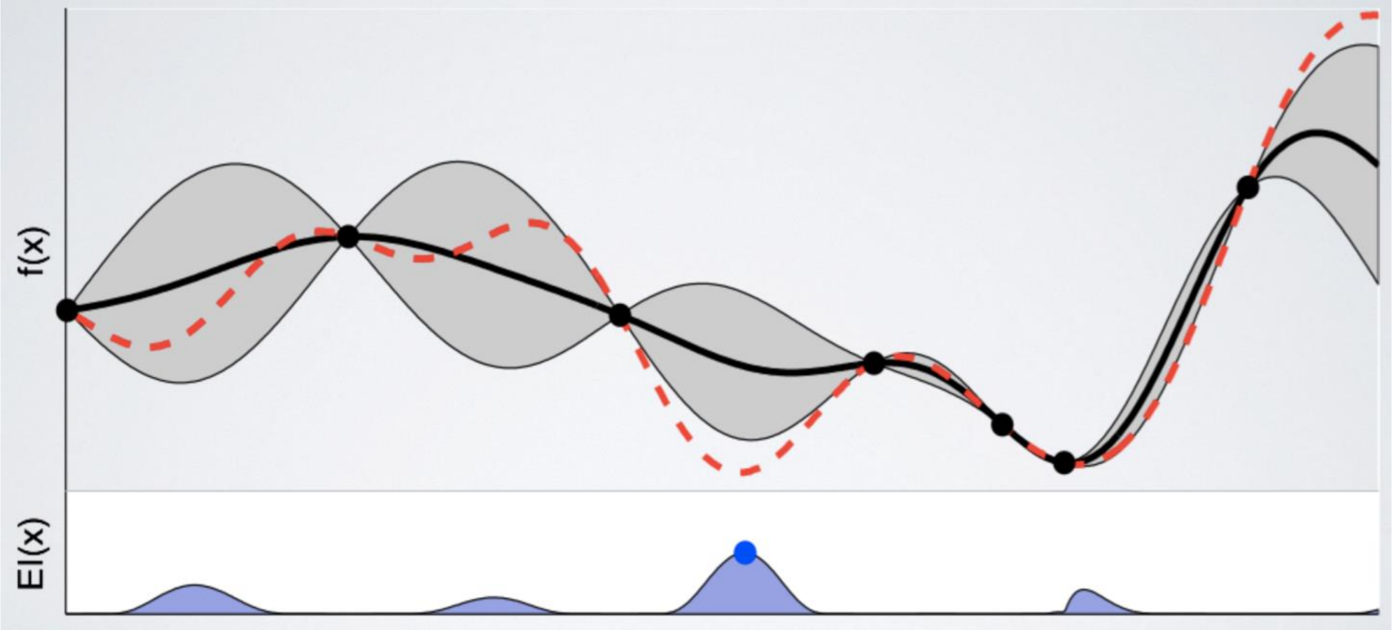
Bayesian Optimization Procedure

1) 관측 데이터들을 통해 $f(x)$를 추정
2) $f(x)$를 더 정밀하게 예측하기 위해 decision rule을 통해 선택
3) 새로 관측한 데이터를 추가하여 적절한 Stopping criteria에 도달할 때가지 1)을 반복

1) 에서는 f(x)가 Gaussian process prior를 가진다고 가정하고 posterior를 계산하여 추정
2) 에서는 다음 샘플포인트를 구하기 위하여,acquisition function을 사용
- Exploration vs. Exploitation
  (certainty of the objective function vs. maximizing the objective function value)
- 궁극적 목표는 목적함수의 evaluation 수를 최소화 하면서 최적화 실행

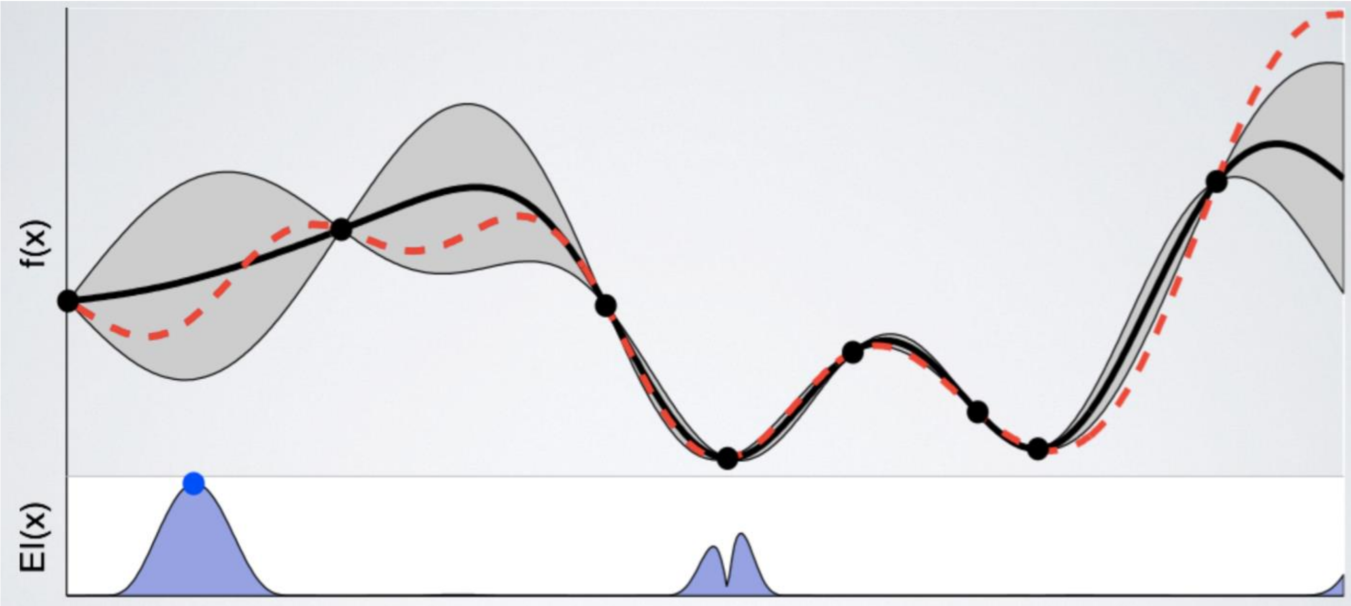
Bayesian Optimization

Bayesian Optimization



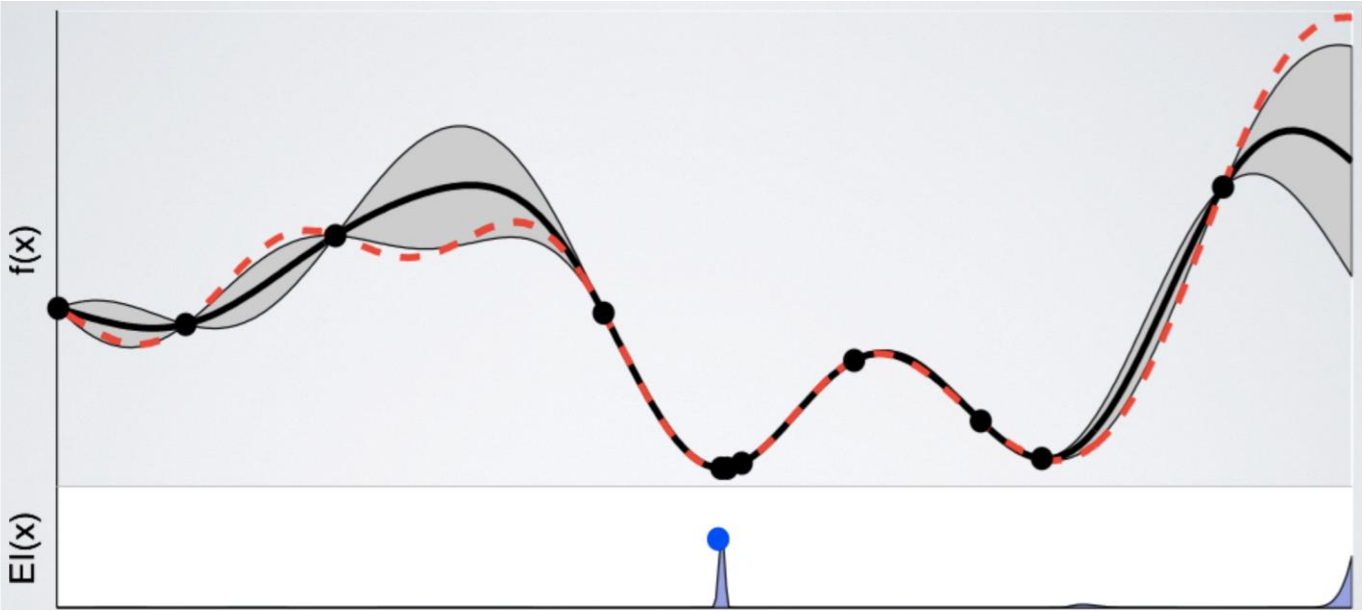
Bayesian Optimization

Bayesian Optimization



Bayesian Optimization

Bayesian Optimization



NIPS 2017 optimization workshop

Black Box Optimization via a Bayesian-Optimized Genetic Algorithm

베이지안 최적화에 대한 대안으로 사용되는 인기있는 CMA-ES 검색 알고리즘과 유사한 유전 알고리즘 및 간단하고 강력한 최적화 알고리즘을 제시

Stochastic Gradient Descent: Going As Fast As Possible But Not Faster

SGD 학습 시 발생하는 gradient explode 문제를 해결하기 위해 SGD 학습률을 제어할 수 있는 SALERA 방법 제시

A Generic Approach for Escaping Saddle points

Non-convex 문제의 first-order method에서 saddle point에 안착되는 문제와 second-order methods에서 Hessian 계산에 오래걸리는 문제를 해결하기 위해 first-order and a second-order subroutine의 프레임워크를 제시

다루지 않은 최적화의 문제들

Multi-objective Program

Multi-level Optimization

Online Optimization

More…

IMGPO, Infinite-Metric GP Optimization

Random Search

감사합니다

mineatte@gmail.com

Q&A