Appendix.02 Optimization Theory

Definition.A.2.1 Optimization Problem with constraint in general programming

Let $f(\mathbf{x})\ :\ \mathbb{R}^n \to \mathbb{R}$ called an object or cost function,
$g_i(\mathbf{x})\ :\ \mathbb{R}^n \to \mathbb{R}$ called an inequality constraint function, and
$h_j(\mathbf{x})\ :\ \mathbb{R}^n \to \mathbb{R}$ called an equality constraint function be.
The following nonlinear programming is called a optimization probelm.

$$\min_{\mathbf{x}}\ f(\mathbf{x})\ s.t.\ g_i(\mathbf{x}) \leq 0,\ \ i = 1, 2, \cdots, m,\ \ h_j(\mathbf{x}) = 0,\ \ j = 1, 2, \cdots, k.$$

$g_i(\mathbf{x}) \leq 0$ is called an inequlity constraint and $h_j(\mathbf{x}) = 0$ is called an equality constraint.

Definition.A.2.2 Lagrangian function

In a optimization problem,

$$\mathscr{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{x}) + \sum_{j=1}^{k} \mu_j h_j(\mathbf{x})\ where\ \boldsymbol{\lambda}\ :\ m \times 1,\ \ \boldsymbol{\mu}\ :\ k \times 1$$

λ_i and μ_i are also called dual variables or weights. $\mathbf{x}$ is called a primal variable.

Theorem.A.2.1 Karush-Kuhn-Tucker(KKT) conditions

If $\mathbf{x}^* \in \mathbb{R}^n$ is a local minimum,
then $\exists\ \boldsymbol{\lambda}^* \in \mathbb{R}^m\ and\ \boldsymbol{\mu}^* \in \mathbb{R}^k\ s.t$
(i) Stationarity :

$$\nabla f(\mathbf{x}) + \sum_i \lambda_i \nabla g_i(\mathbf{x}) + \sum_j \mu_j h_j(\mathbf{x}) = \nabla_{\mathbf{x}} \mathscr{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \mathbf{0}$$

(ii) Primal feasibility :

$$g_i(\mathbf{x}^*) \leq 0,\ \ ^\forall i,\ \ h_j(\mathbf{x}^*) = \mathbf{0},\ \ ^\forall j$$

(iii) Dual feasibility :

$$\lambda_i^* \geq 0,\ \ i = 1, \cdots, m$$

(vi) Complementary slackness :

$$\lambda_i^* g_i(\mathbf{x}^*) = 0,\ \ i = 1, \cdots, m$$

This $\mathbf{x}^*$ is also called a local solution of a optimization problem. KKT Conditions are necessary(not sufficient) for global optimality.

$$\mathbf{x}^*(\text{Satisfying the KKT Conditions})\ \leftarrow \text{Global optimal}$$

Definition.A.2.3 Convex function

A function $f(\mathbf{x})$ is said to be convex if

$$^\forall t \in [0, 1],\ \ f(t\mathbf{x}_1 + (1-t)\mathbf{x}_2) \leq t f(\mathbf{x}_1) + (1-t) f(\mathbf{x}_2),\ \ ^\forall \mathbf{x}_1, \mathbf{x}_2 \in D$$

Definition.A.2.4 Convex set

A set S is said to be convex if

$$\mathbf{x},\,\mathbf{y} \in S \;\rightarrow\; t\mathbf{x} + (1-t)\mathbf{y} \in S,\;\; t \in [0,\,1]$$

Definition.A.2.5 Convex optimization

Let $f(\mathbf{x})\;:\;\mathbb{R}^n \rightarrow \mathbb{R}$ and $\mathbf{g}(\mathbf{x})\;:\;\mathbb{R}^n \rightarrow \mathbb{R}^m$ be convex functions and $\mathbf{h}(\mathbf{x})\;:\;\mathbb{R}^n \rightarrow \mathbb{R}^k$ be an affine function.
An optimization problem is called convex if

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad s.t. \quad g_i(\mathbf{x}) \le 0,\;\; i = 1,\,\cdots,\,m, \qquad h_j(\mathbf{x}) = 0,\;\; j = 1,\,\cdots,\,k.$$

Theorem.A.2.2 KKT Conditions in convex problems

For convex problem, KKT conditions becomes necessary and also sufficient for global optimality.
(i) Stationarity :

$$\bigtriangledown\; f(\mathbf{x}^*) + \bigtriangledown\;(\boldsymbol{\lambda}^{*T}\mathbf{h}(\mathbf{x}^*)) + \;\bigtriangledown\;(\boldsymbol{\mu}^{*T}\mathbf{g}(\mathbf{x}^*)) = 0$$

(ii) Primal feasibility :

$$\mathbf{g}(x^*) \le \mathbf{0},\;\; \mathbf{h}(\mathbf{x}^*) = 0$$

(iii) Dual feasibility :

$$\boldsymbol{\lambda}^* \ge 0$$

(vi) Complementary slackness :

$$\boldsymbol{\lambda}^{*T}\mathbf{g}(\mathbf{x}^*) = 0$$

Definition.A.2.6 Lagrange dual function

$$\begin{aligned}\mathscr{D}(\boldsymbol{\lambda},\,\boldsymbol{\mu}) &= \min_{\mathbf{x}\,\in\,\mathscr{X}} \mathscr{L}(\mathbf{x},\,\boldsymbol{\lambda},\,\boldsymbol{\mu}) \\ &= \min_{\mathbf{x}\,\in\,\mathscr{X}} \{f(\mathbf{x}) + \boldsymbol{\lambda}^T\mathbf{h}(\mathbf{x}) + \boldsymbol{\mu}^T\mathbf{g}(\mathbf{x})\}\end{aligned}$$

$$\text{where} \quad \mathscr{X} = \{\mathbf{x} : \mathbf{g}(\mathbf{x}) \le \mathbf{0},\;\; \mathbf{h}(\mathbf{x}) = \mathbf{0}\}$$

Definition.A.2.7 Lagrange dual problem

$$\begin{aligned}\max_{\lambda\,\ge\,0,\;\mu} \mathscr{D}(\boldsymbol{\lambda},\,\boldsymbol{\mu}) &= \max_{\lambda\,\ge\,0,\;\mu}\; \min_{\mathbf{x}\,\in\,\mathscr{X}} \mathscr{L}(\mathbf{x},\,\boldsymbol{\lambda},\,\boldsymbol{\mu}) \\ &= \max_{\lambda\,\ge\,0,\;\mu}\; \min_{\mathbf{x}\,\in\,\mathscr{X}} \{f(\mathbf{x}) + \boldsymbol{\lambda}^T\mathbf{g}(\mathbf{x}) + \boldsymbol{\mu}^T\mathbf{h}(\mathbf{x})\}\end{aligned}$$

Theorem.A.2.3 Optimization-theoretic solution of distance between a vector and a hyperplane

Let a vector $\mathbf{x}_0$ and a hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$ be.

$$\min_{\mathbf{x}} \parallel \mathbf{x}_0 - \mathbf{x} \parallel^2 \ s.t. \ \mathbf{w}^t\mathbf{x} + b = 0$$

$$\therefore \ \parallel \mathbf{x}_0 - \mathbf{x}^* \parallel^2 = \frac{(\mathbf{w}^T\mathbf{x}_0 + b)^2}{\parallel \mathbf{w} \parallel}$$

Proof)
Let the function $\mathscr{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \parallel \mathbf{x}_0 - \mathbf{x} \parallel^2 + \mu(\mathbf{w}^T\mathbf{x} + b)$
(i)

$$\nabla_{\mathbf{x}} \mathscr{L} = 2(\mathbf{x}_0 - \mathbf{x}^*) + \mu\mathbf{w} = \mathbf{0}$$

$$2\mathbf{w}^T(\mathbf{x}_0 - \mathbf{x}^*) = -\mu \parallel \mathbf{w} \parallel^2 \quad (\because \ inner \ product \ with \ (\mathbf{x}_0 - \mathbf{x}^*))$$

$$2 \parallel \mathbf{x}_0 - \mathbf{x}^* \parallel^2 = \frac{2(\mathbf{w}^T(\mathbf{x}_0 - \mathbf{x}^*))^2}{\parallel \mathbf{w} \parallel^2}$$

$$\therefore \quad \parallel \mathbf{x}_0 - \mathbf{x}^* \parallel^2 = \frac{\{\mathbf{w}^T(\mathbf{x}_0 - \mathbf{x}^*)\}^2}{\parallel \mathbf{w} \parallel^2} \qquad \blacksquare$$

Theorem.A.2.4 Gradient descent method

Gradient descent is an iterative method to find a stationary point of an unconstraint optimization problem :

$$\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

$$L(\boldsymbol{\theta} + \eta\mathbf{d}) \approx L(\boldsymbol{\theta}) + \eta \nabla_{\boldsymbol{\theta}}^T L(\boldsymbol{\theta})\mathbf{d} \quad where \quad \eta > 0, \quad \parallel \mathbf{d} \parallel = 1$$

$$L(\boldsymbol{\theta} + \eta\mathbf{d}) - L(\boldsymbol{\theta}) \approx \eta \nabla_{\boldsymbol{\theta}}^T L(\boldsymbol{\theta})\mathbf{d} = \eta\cos(\phi) \parallel \nabla_{\boldsymbol{\theta}}^T L(\boldsymbol{\theta}) \parallel$$

Find the directional vector $\mathbf{d}$ that minimizes $L(\boldsymbol{\theta} + \eta\mathbf{d}) - L(\boldsymbol{\theta}) \leq 0$

$$\cos(\phi) = -1 \rightarrow \mathbf{d} = -\frac{\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})}{\parallel \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \parallel}$$

$$\therefore \boldsymbol{\theta} + \eta\mathbf{d} = \boldsymbol{\theta} - \eta\frac{\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})}{\parallel \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \parallel} = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

Theorem.A.2.4 Types of gradient descent method

(i) Standard (or steepest) Gradient Descent

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \bigtriangledown \mathbb{E}[J(\mathbf{w})]$$

- Practically infeasible
- Thus, we need distribution about data $\mathbf{x}$ (Contradiction)
- So, We can use sample mean

(ii) Stochastic Gradient Descent

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \bigtriangledown J_i(\mathbf{w})$$

- Simple to implement
- Effective for large-scale problem
- Much less memory
- Unstable : zigzaging
- Purpose : We just consider one of data. Just one.
- It can be convergent. But there is little unstable.

(iii) Batch gradient Descent

$$\mathbf{w} \leftarrow \eta \bigtriangledown \sum_{i=1}^{N} J_i(\mathbf{w})$$

- Accurate estimation of gradients
- Parallelization of learning
- Large memory
- Big time-complexity can be problem in this method.(So slow)
- But, there isn't problem in convergence.
- Purpose : We consider all of data!

(vi) Mini-Batch Gradient Descent

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \bigtriangledown \sum_{i \in \mathfrak{I}}^{N} J_i(\mathbf{w}), \quad 1 \leq |\mathfrak{I}| \leq N$$

- Most generalized version
- Effective to deal with large
- Amount of training data
- Purpose : We just consider seveal datas.

Reference :

Deep Learning - Yosha Benjio
Neural networks and learning machines - Simon Haykin
https://wikipedia.org (https://wikipedia.org)
https://ko.wikipedia.org/wiki/위키백과:TeX_문법
(https://ko.wikipedia.org/wiki/%EC%9C%84%ED%82%A4%EB%B0%B1%EA%B3%BC:TeX_%EB%AC%B8%EB%B2%95)