Indian Institute of Technology Madras

Optimization

Nirav Bhatt
Email: niravbhatt@iitm.ac.in
Office: BT 307 Block II
Biotechnology Department

September 27, 2024

Content

- Unconstrained vs Constrained optimization
- Types of Optimizations problems:
  - Linear programming (LP)
  - Quadratic programming (QP)
  - Nonlinear programming (NLP)
  - Dynamic optimization as an NLP
- Overview of Numerical solution approaches
- Book: Nocedal J, Wright SJ, editors. Numerical optimization. New York, NY: Springer New York; 1999 Aug 27. Reference Chapters 1, 2, 12
- Reference Book: An Introduction to Optimization, EDWIN K. P. CHONG and STANISLAW H. ZAK

Learning outcomes

Introduction to optimization and different forms of optimization

- The students are expected to learn
  - Different types of optimization problems and their application
  - KKT Conditions for finding an optimal solution
  - Overview of Line search and trust region for numerical optimization

Optimization

Elements

- Mathematical Optimization (or Mathematical Programming):
  Select a best option or a set of options from the available set
- Consider an optimization problem with decision variables $\mathbf{x} = [x_1, x_2, ..., x_n]^T$

$$\begin{aligned} \max_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{Ax} \le \mathbf{b} \\ & \mathbf{1}^T\mathbf{x} \le 1 \end{aligned}$$

  or

$$\max_{x_1, x_2, ..., x_n} \quad f(x_1, x_2, ..., x_n)$$

- $f(\mathbf{x})$ is called *cost function* or *objective function* or *loss function*
- $\mathbf{Ax} \le \mathbf{b}$ and $\sum_{i=1}^{n} x_i \le 1$ are *constraints*

Optimization

Some Problems in ML/DL

- Regression Analysis: Given a dataset $(\mathbf{y}, \mathbf{X})$, fit a function form $f(\mathbf{X}, \theta) : \mathbf{R}^n \times \mathbf{R}^p \to \mathbf{R}$

$$\min_{\boldsymbol{\theta}} \quad \|\mathbf{y} - f(\mathbf{X}, \theta)\|_2^2$$

- Classification Problems: Given $(\mathbf{y}, \mathbf{X})$, fit a function form $f(\mathbf{X}, \theta) : \mathbf{R}^n \times \mathbf{R}^p \to \{0, 1\}$

$$\min_{\boldsymbol{\theta}} \quad -\frac{1}{m}\sum_{i=1}^{m}(y_i log(p_i(\mathbf{x}, \theta)) + (1 - y_i) log(1 - p_i(\mathbf{x}, \theta))$$

where $p_i(\cdots)$ is the probability of the class 1.

Optimization

Constraints

- Types of constraints
  - Linear Constraints

$$\mathbf{A}\mathbf{x} \leq \mathbf{b} \quad \text{(Inequality)}$$

$$\mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq} \quad \text{(Equality)}$$

  - Inequality Constraints

$$\mathbf{g}(\mathbf{x}) \leq \mathbf{0} \quad \text{(Inequality)}$$

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \quad \text{(Equality)}$$

  - Bounds: $\mathbf{x}_{lb} \leq \mathbf{x} \leq \mathbf{x}_{ub}$
- $\mathbf{x} \in \mathcal{S}$ For example, $\mathcal{S} = \{-1, 0, 1\}$

Optimization
Types

- Types based constraints: (i) Unconstrained optimization and (ii) Constrained Optimization; (i) Static optimization and (ii) Dynamic optimization
- Types of function or variable set smoothness: (i) Continuous Optimization, and (ii) Integer optimization
- Types of function and constraints
  - Linear Programming (LP)
  - Quadratic Programming (QP)
  - Nonlinear Programming (NLP) or Nonlinear Optimization
  - Mixed Integer LP or NLP

Optimization

- Unconstrained Optimization: x Decision variables

$$\max_{x} \textit{ or min} \quad f(x) \tag{1}$$

- Constrained Optimization

$$\begin{aligned} \max_{x} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq b_i, \quad i = 1, \ldots, p \\ & \mathbf{a}_j^T \mathbf{x} = c_j, \quad j = 1, \ldots, q \\ & \mathbf{x}^{LB} \leq \mathbf{x} \leq \mathbf{x}^{UB} \end{aligned} \tag{2}$$

- $\mathbf{x}^*$ denote an optimal solution

Motivation

Dynamic Optimization

- The current profit is a function of the current production ($x(t)$) and the rate of change of production ($x'(t)$)
- The continuous problem can be defined as:

$$\begin{aligned} \max \quad & J[x] = \int_{t=1}^{T} f(t, x(t), \dot{x}(t))dt \\ \text{s.t.} \quad & x(t) \geq 0, \quad x(0) = x_0 \end{aligned} \tag{3}$$

- Objective: Find the function $x(t)$ that maximizes the functional $J[x]$
- $x^*(t)$: Optimal trajectory

Optimization

- Linear Programming

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{C}\mathbf{x} = \mathbf{d}, \\ & \mathbf{x}^{LB} \leq \mathbf{x} \leq \mathbf{x}^{UB} \end{aligned} \tag{4}$$

- Quadratic Programming

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ & \mathbf{C}\mathbf{x} = \mathbf{d}, \\ & \mathbf{x}^{LB} \leq \mathbf{x} \leq \mathbf{x}^{UB} \end{aligned} \tag{5}$$

$\mathbf{H}$: Symmetric matrix

Optimization

- Nonlinear Programming

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & f(\mathbf{x}) \\
\text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, M, \\
& h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, N \\
& \mathbf{x}^{LB} \leq \mathbf{x} \leq \mathbf{x}^{UB}
\end{aligned}
\tag{6}
$$

- Integer Quadratic Programming

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{c}^T \mathbf{x} \\
\text{s.t.} \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\
& \mathbf{C}\mathbf{x} = \mathbf{d}, \\
& \text{some } \mathbf{x} \text{ are integer}
\end{aligned}
\tag{7}
$$

Motivation

Static Optimization

- Static Optimization: An optimal number or finite set of numbers
- Static Optimization:

$$\max_{x} \quad f(x) \tag{8}$$

- Assumptions: $f(x)$ is continuously differentiable
- First order necessary condition: $\frac{\partial f}{\partial x}(x^*) = 0$
- Second order necessary condition: $\frac{\partial^2 f}{\partial x^2} \leq 0$
- Example: The operating point x^* that maximizes the profit $f(x)$, where x: $\#$ of units

Motivation

Static Optimization

- Example

$$\max_{x} \quad 1000000 + 4000x - x^2 \tag{9}$$

- $\frac{\partial f}{\partial x}(x^*) = 0 \Rightarrow 4000 - 2x = 0$
  $x^* = 2000$
- $\frac{\partial^2 f}{\partial x^2} = -2 < 0$

Optimization

Static Optimization

- Static Optimization with several variables

$$\max_{x_1,\ldots,x_n} \quad f(x_1,\ldots x_n) \tag{10}$$

- Example: A plant can produce n items. Find the operating point $x_1,\ldots x_n$ that maximizes the profit $f(x_1,\ldots x_n)$

Motivation

Static Optimization: Detour to Some Problems in Linear Algebra

- Recall: $\mathbf{Ax} = \mathbf{b}$ when $\mathbf{b}$ is not in the column space spanned by $\mathbf{A}$.
- Projection of $\mathbf{b}$ on the plane spanned by $\mathbf{Ax}$
- Error $\mathbf{e} = \mathbf{Ax} - \mathbf{b}$ or the closest point on the plane spanned by the columns of $\mathbf{A}$ from $\mathbf{b}$
- Optimization Problem

$$\min_{x_1,\ldots,x_n} \quad \|\mathbf{Ax} - \mathbf{b}\|^p$$

$p = 1, 2, 3 \ldots, \infty$

Motivation

Static Optimization

- Example

$$\max_{x} \quad 1000000 + 300x_1 + 500x_2 - x_1^2 - x_2^2 \qquad (11)$$

- $\frac{\partial f}{\partial x}(x_1^*) = 0 \Rightarrow 300 - 2x_1 = 0 \Rightarrow x_1^* = 150$
- $\frac{\partial f}{\partial x}(x_2^*) = 0 \Rightarrow 500 - 2x_2 = 0 \Rightarrow x_2^* = 250$
- $\frac{\partial^2 f}{\partial x_1^2} = \frac{\partial^2 f}{\partial x_2^2} = -2 < 0$
- These conditions are correct??

Static Optimization

Local vs Global SOlutions

- Local Minimizer
  A point $\mathbf{x}^*$ is a local minimizer if there is a neighborhood $\mathcal{N}$ of $\mathbf{x}^*$ such that $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{N}$.

  A point $\mathbf{x}^*$ is a *strict* local minimizer (also called a strong local minimizer) if there exists a neighborhood $\mathcal{N}$ of $\mathbf{x}^*$ such that $f(\mathbf{x}^*) < f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{N}$ with $\mathbf{x} \neq \mathbf{x}^*$

- Global Minimizer
  A point $\mathbf{x}^*$ is a global minimizer if $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x}$
- $\mathbf{x}^*$ is a stationary point if $\nabla f(\mathbf{x}^*) = 0$.

Static Optimization

Unconstrained Optimization

- First-order Necessary Conditions
  If $\mathbf{x}^*$ is a local minimizer and f is continuously differentiable in an open neighborhood of $\mathbf{x}^*$, then $\nabla f(\mathbf{x}^*) = 0$.

- Second-order Necessary Conditions
  If $\mathbf{x}^*$ is a local minimizer of f then $\nabla^2 f(\mathbf{x})$ exists and is continuous in an open neighborhood of $\mathbf{x}^*$, then $\nabla f(\mathbf{x}^*) = 0$ and $\nabla^2 f(\mathbf{x}^*)$ is positive semidefinite.

- Second-order Sufficient Conditions
  Suppose that $\nabla^2 f(\mathbf{x}^*)$ is continuous in an open neighborhood of $\mathbf{x}^*$ and that $\nabla f(\mathbf{x}^*) = 0$ and $\nabla^2 f(\mathbf{x}^*)$ is *positive definite*.
  Then, $\mathbf{x}^*$ is a strict local minimizer of f.

Constrained Optimization

Equality Constraints

- Consider the following optimization problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & h_i(x) = 0, \ i = 1, 2, \dots, m \end{aligned} \tag{12}$$

- Constraint optimization to Unconstrained optimization
- Lagrange function with Lagrange multipliers $\lambda_1, \dots, \lambda_m$

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x)$$

Constrained Optimization

- First-order necessary condition

$$\nabla f(x^*) + \lambda_1^* \nabla h_1(x^*) + \ldots + \lambda_m^* \nabla h_m(x^*) = 0$$

$$h_i(x^*) = 0, \; i = 1, 2, \ldots, m$$

- Second-order necessary condition

$w^T L_{xx}(x^*, \lambda^*) w \geq 0, \forall w$ such that $\nabla h_i(x^*).w = 0, \; i = 1, 2, \ldots, m$

where

$$L_{xx}(x^*, \lambda^*) = \nabla^2 f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla^2 h_i(x^*)$$

$L_{xx}(x^*, \lambda^*)$ is positive definite on the tangent space defined by $\nabla h_i(x^*).w = 0$ the tangent space

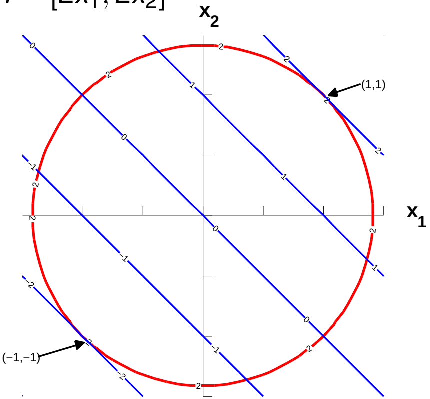
Example: Constrained Optimization

- Problem:

$$\min \quad x_1 + x_2$$

$$\text{s.t.} \quad x_1^2 + x_2^2 - 2 = 0$$

- $\nabla f = [1, 1]^T \;\; \nabla h = [2x_1, 2x_2]^T$

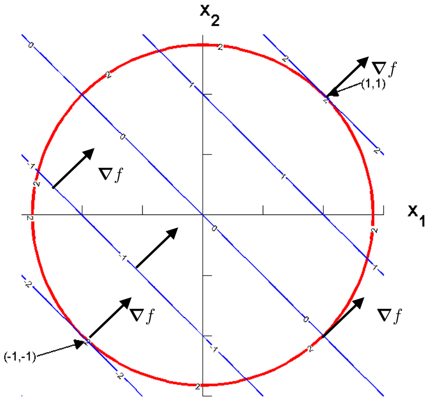


Example: Constrained Optimization

- Problem:

$$\begin{aligned} \min \quad & x_1 + x_2 \\ \text{s.t.} \quad & x_1^2 + x_2^2 - 2 = 0 \end{aligned}$$

- $\nabla f = [1, 1]^T \;\; \nabla h = [2x_1, 2x_2]^T$

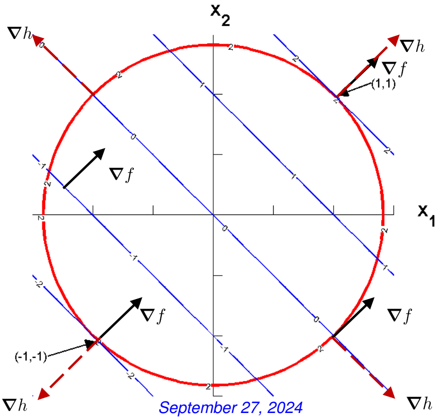


Example: Constrained Optimization

- Problem:

$$\begin{aligned} \min \quad & x_1 + x_2 \\ \text{s.t.} \quad & x_1^2 + x_2^2 - 2 = 0 \end{aligned}$$

- $\nabla f = [1,1]^T \;\; \nabla h = [2x_1, 2x_2]^T$



Constrained Optimization: Example

KKT Conditions

- Constrained optimization problem:

$$
\begin{aligned}
\min \quad & f(x) \\
\text{s.t.} \quad & h_i(x) = 0,\ i = 1, \ldots, m \\
\text{s.t.} \quad & g_j(x) \leq 0,\ j = 1, \ldots, n
\end{aligned}
$$

  where f, h_i, and g_j: smooth, and real-valued functions on a subset of $\mathbb{R}^n$
- x is a feasible point, if it satisfies the equality and inequality constraints
- A constraint j is said to be active if $g_j(x) = 0$ at any point x
- $A(x)$: A set of all active constraints at any point x
- For a feasible point x and the active set $A(x)$, if the gradients ∇h_i, and $\nabla g_j, \forall j \in A(x)$ are linearly independent,they satisfy the Linear Independence Constraint Qualification (LICQ).

Constrained Optimization

KKT Conditions

- Lagrangian for the problem with the Lagrange multipliers λ and μ:

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \sum_{j=1}^{n} \mu_j g_j(x)$$

- First order Necessary condtions (Karush-Kuhn-Tucker conditions): For x^* a local solution

$$\begin{aligned}
\nabla_x L(x^*, \lambda^*, \mu^*) &= 0 \\
h_i(x^*) &= 0, i = 1, \ldots, m \\
g_j(x^*) &\leq 0, i = 1, \ldots, n \\
\lambda_i &> 0,\ \mu_j \geq 0 \\
\mu_j(x^*) g_j(x^*) &= 0 \text{ (Complementarity condition)}
\end{aligned}$$

- Complementarity condition: Ensures $\mu_j = 0, \forall j \notin A(x)$

Constrained Optimization

KKT Conditions

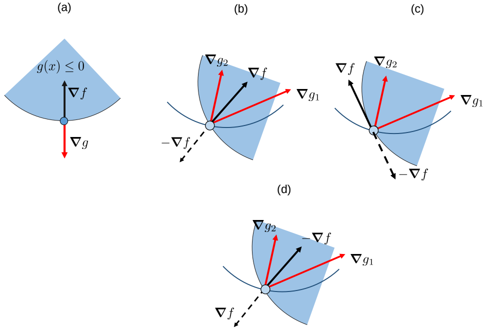
- Complementarity condition: Ensures $\mu_j = 0, \forall j \notin A(x)$
- Second order necessary condition

$$w^T \nabla^2 L(x^* \lambda^*, \mu^*)\, w \geq 0,\ \forall w \in T(x^*)$$

$$T(x^*) = \{w | \nabla h_i(x^*)^T w = 0,\ \forall i \text{ and } \nabla g_j(x^*)^T w = 0,\ \forall j \in A(x^*)\}$$

- (a)$\mu_j > 0, \forall j \in A(x^*)$: ∇f inwards: Otherwise maximization and ∇g outward: Otherwise increase inward

- $-\nabla f(x^*) = \sum_{j=1}^{m} \mu_j \nabla g_j(x^*)$
  (b) $\mu_1, \mu_2 < 0$,
  (c) $\mu_1 > 0, \mu_2 < 0$,
  (d) $\mu_1, \mu_2 > 0$



Constrained Optimization

KKT Conditions

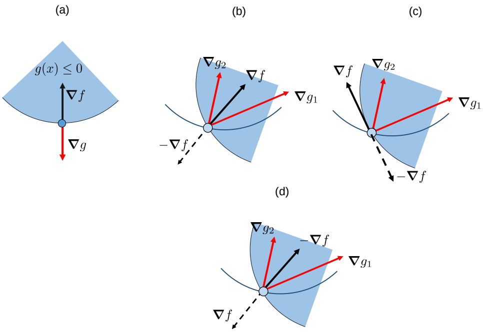
- Complementarity condition: Ensures $\mu_j = 0, \forall j \notin A(x)$
- Second order necessary condition

$$w^T \nabla^2 L(x^* \lambda^*, \mu^*)\, w \geq 0,\ \forall w \in T(x^*)$$

$$T(x^*) = \{w | \nabla h_i(x^*)^T w = 0,\ \forall i \text{ and } \nabla g_j(x^*)^T w = 0,\ \forall j \in A(x^*)\}$$

- (a) $\mu_j > 0, \forall j \in A(x^*)$: ∇f inwards: Otherwise maximization and ∇g outward: Otherwise increase inward
- $-\nabla f(x^*) = \sum_{j=1}^{m} \mu_j \nabla g_j(x^*)$

  (b) $\mu_1, \mu_2 < 0$,
  (c) $\mu_1 > 0, \mu_2 < 0$,
  (d) $\mu_1, \mu_2 > 0$



Constrained Optimization

KKT Conditions: Problem

- Optimization problem

$$\begin{aligned} \min \quad & x_1^2 + 2x_2^2 \\ \text{s.t.} \quad & x_1 + x_2 \geq 3 \\ \text{s.t.} \quad & x_2 - x_1^2 \geq 1 \end{aligned}$$

- The Lagrangian function:

$$l(x, \mu_1, \mu_2) = x_1^2 + 2x_2^2 - \mu_1(x_1 + x_2 - 3) - \mu_2(x_2 - x_1^2 - 1), \ \mu_1, \mu_2 \geq 0$$

- KKT Conditions
  1. $2x_1 - \mu_1 + 2\mu_2 x_1 = 0$
  2. $4x_2 - \mu_1 - \mu_2 = 0$
  3. $x_1 + x_2 \geq 3$
  4. $x_2 - x_1^2 \geq 1$
  5. $\mu_1(x_1 + x_2 - 3) = 0$
  6. $\mu_2(x_2 - x_1^2 - 1) = 0$
  7. $\mu_1, \mu_2 \geq 0$

Constrained Optimization

KKT Conditions: Problem

KKT Conditions

1. $2x_1 - \mu_1 + 2\mu_2 x_1 = 0$
2. $4x_2 - \mu_1 - \mu_2 = 0$
3. $x_1 + x_2 \geq 3$
4. $x_2 - x_1^2 \geq 1$
5. $\mu_1(x_1 + x_2 - 3) = 0$
6. $\mu_2(x_2 - x_1^2 - 1) = 0$
7. $\mu_1, \mu_2 \geq 0$

- Case 1: (No active constraint) , then $(x_1^*, x_2^*) = (0, 0)$
- Case 2: $\mu_1 > 0, \mu_2 = 0$ (First constraint is active), then $(x_1^*, x_2^*) = (2, 1)$ and $\mu_1 = 4 > 0$, Does not satisfy Condition 4
- Case 3: $\mu_1 = 0, \mu_2 > 0$ (Second constraint is active), then $(x_1^*, x_2^*) = (0, 1)$ and $\mu_2 = 4 > 0$ Does not satisfy Condition 3
- Case 4: $\mu_1 > 0, \mu_2 > 0$ (Both constraints are active), then $(x_1^*, x_2^*) = (-2, 5)$ and $(x_1^*, x_2^*) = (1, 2)$
  - $(x_1^*, x_2^*) = (-2, 5) \implies \mu_1 + 4\mu_2 = -4$
  - $(x_1^*, x_2^*) = (1, 2) \implies \mu_1 = 6$, and $\mu_2 = 2$,

Constrained Optimization

KKT Conditions: Problem

KKT Conditions

1. $2x_1 - \mu_1 + 2\mu_2 x_1 = 0$
2. $4x_2 - \mu_1 - \mu_2 = 0$
3. $x_1 + x_2 \geq 3$
4. $x_2 - x_1^2 \geq 1$
5. $\mu_1(x_1 + x_2 - 3) = 0$
6. $\mu_2(x_2 - x_1^2 - 1) = 0$
7. $\mu_1, \mu_2 \geq 0$

- Case 1: (No active constraint) , then $(x_1^*, x_2^*) = (0, 0)$
- Case 2: $\mu_1 > 0, \mu_2 = 0$ (First constraint is active), then $(x_1^*, x_2^*) = (2, 1)$ and $\mu_1 = 4 > 0$, Does not satisfy Condition 4
- Case 3: $\mu_1 = 0, \mu_2 > 0$ (Second constraint is active), then $(x_1^*, x_2^*) = (0, 1)$ and $\mu_2 = 4 > 0$ Does not satisfy Condition 3
- Case 4: $\mu_1 > 0, \mu_2 > 0$ (Both constraints are active), then $(x_1^*, x_2^*) = (-2, 5)$ and $(x_1^*, x_2^*) = (1, 2)$
  - $(x_1^*, x_2^*) = (-2, 5) \implies \mu_1 + 4\mu_2 = -4$
  - $(x_1^*, x_2^*) = (1, 2) \implies \mu_1 = 6$, and $\mu_2 = 2$,

Constrained Optimization

KKT Conditions: Problem

KKT Conditions

1. $2x_1 - \mu_1 + 2\mu_2 x_1 = 0$
2. $4x_2 - \mu_1 - \mu_2 = 0$
3. $x_1 + x_2 \geq 3$
4. $x_2 - x_1^2 \geq 1$
5. $\mu_1(x_1 + x_2 - 3) = 0$
6. $\mu_2(x_2 - x_1^2 - 1) = 0$
7. $\mu_1, \mu_2 \geq 0$

- Case 1: (No active constraint) , then $(x_1^*, x_2^*) = (0, 0)$
- Case 2: $\mu_1 > 0, \mu_2 = 0$ (First constraint is active), then $(x_1^*, x_2^*) = (2, 1)$ and $\mu_1 = 4 > 0$, Does not satisfy Condition 4
- Case 3: $\mu_1 = 0, \mu_2 > 0$ (Second constraint is active), then $(x_1^*, x_2^*) = (0, 1)$ and $\mu_2 = 4 > 0$ Does not satisfy Condition 3
- Case 4: $\mu_1 > 0, \mu_2 > 0$ (Both constraints are active), then $(x_1^*, x_2^*) = (-2, 5)$ and $(x_1^*, x_2^*) = (1, 2)$
  - $(x_1^*, x_2^*) = (-2, 5) \implies \mu_1 + 4\mu_2 = -4$
  - $(x_1^*, x_2^*) = (1, 2) \implies \mu_1 = 6$, and $\mu_2 = 2$,

Constrained Optimization

KKT Conditions: Problem

KKT Conditions

1. $2x_1 - \mu_1 + 2\mu_2 x_1 = 0$
2. $4x_2 - \mu_1 - \mu_2 = 0$
3. $x_1 + x_2 \geq 3$
4. $x_2 - x_1^2 \geq 1$
5. $\mu_1(x_1 + x_2 - 3) = 0$
6. $\mu_2(x_2 - x_1^2 - 1) = 0$
7. $\mu_1, \mu_2 \geq 0$

- Case 1: (No active constraint) , then $(x_1^*, x_2^*) = (0, 0)$
- Case 2: $\mu_1 > 0, \mu_2 = 0$ (First constraint is active), then $(x_1^*, x_2^*) = (2, 1)$ and $\mu_1 = 4 > 0$, Does not satisfy Condition 4
- Case 3: $\mu_1 = 0, \mu_2 > 0$ (Second constraint is active), then $(x_1^*, x_2^*) = (0, 1)$ and $\mu_2 = 4 > 0$ Does not satisfy Condition 3
- Case 4: $\mu_1 > 0, \mu_2 > 0$ (Both constraints are active), then $(x_1^*, x_2^*) = (-2, 5)$ and $(x_1^*, x_2^*) = (1, 2)$
  - $(x_1^*, x_2^*) = (-2, 5) \implies \mu_1 + 4\mu_2 = -4$
  - $(x_1^*, x_2^*) = (1, 2) \implies \mu_1 = 6$, and $\mu_2 = 2$,

Constrained Optimization

KKT Conditions: Problem

KKT Conditions

1. $2x_1 - \mu_1 + 2\mu_2 x_1 = 0$
2. $4x_2 - \mu_1 - \mu_2 = 0$
3. $x_1 + x_2 \geq 3$
4. $x_2 - x_1^2 \geq 1$
5. $\mu_1(x_1 + x_2 - 3) = 0$
6. $\mu_2(x_2 - x_1^2 - 1) = 0$
7. $\mu_1, \mu_2 \geq 0$

- Case 1: (No active constraint) , then $(x_1^*, x_2^*) = (0, 0)$
- Case 2: $\mu_1 > 0, \mu_2 = 0$ (First constraint is active), then $(x_1^*, x_2^*) = (2, 1)$ and $\mu_1 = 4 > 0$, Does not satisfy Condition 4
- Case 3: $\mu_1 = 0, \mu_2 > 0$ (Second constraint is active), then $(x_1^*, x_2^*) = (0, 1)$ and $\mu_2 = 4 > 0$ Does not satisfy Condition 3
- Case 4: $\mu_1 > 0, \mu_2 > 0$ (Both constraints are active), then $(x_1^*, x_2^*) = (-2, 5)$ and $(x_1^*, x_2^*) = (1, 2)$
  - $(x_1^*, x_2^*) = (-2, 5) \implies \mu_1 + 4\mu_2 = -4$
  - $(x_1^*, x_2^*) = (1, 2) \implies \mu_1 = 6$, and $\mu_2 = 2$,

Constrained Optimization

KKT Conditions: Problem

KKT Conditions

1. $2x_1 - \mu_1 + 2\mu_2 x_1 = 0$
2. $4x_2 - \mu_1 - \mu_2 = 0$
3. $x_1 + x_2 \geq 3$
4. $x_2 - x_1^2 \geq 1$
5. $\mu_1(x_1 + x_2 - 3) = 0$
6. $\mu_2(x_2 - x_1^2 - 1) = 0$
7. $\mu_1, \mu_2 \geq 0$

- Case 1: (No active constraint) , then $(x_1^*, x_2^*) = (0, 0)$
- Case 2: $\mu_1 > 0, \mu_2 = 0$ (First constraint is active), then $(x_1^*, x_2^*) = (2, 1)$ and $\mu_1 = 4 > 0$, Does not satisfy Condition 4
- Case 3: $\mu_1 = 0, \mu_2 > 0$ (Second constraint is active), then $(x_1^*, x_2^*) = (0, 1)$ and $\mu_2 = 4 > 0$ Does not satisfy Condition 3
- Case 4: $\mu_1 > 0, \mu_2 > 0$ (Both constraints are active), then $(x_1^*, x_2^*) = (-2, 5)$ and $(x_1^*, x_2^*) = (1, 2)$
  - $(x_1^*, x_2^*) = (-2, 5) \implies \mu_1 + 4\mu_2 = -4$
  - $(x_1^*, x_2^*) = (1, 2) \implies \mu_1 = 6$, and $\mu_2 = 2$,

Constrained Optimization

KKT Conditions: Problem

KKT Conditions

1. $2x_1 - \mu_1 + 2\mu_2 x_1 = 0$
2. $4x_2 - \mu_1 - \mu_2 = 0$
3. $x_1 + x_2 \geq 3$
4. $x_2 - x_1^2 \geq 1$
5. $\mu_1(x_1 + x_2 - 3) = 0$
6. $\mu_2(x_2 - x_1^2 - 1) = 0$
7. $\mu_1, \mu_2 \geq 0$

- Case 1: (No active constraint) , then $(x_1^*, x_2^*) = (0, 0)$
- Case 2: $\mu_1 > 0, \mu_2 = 0$ (First constraint is active), then $(x_1^*, x_2^*) = (2, 1)$ and $\mu_1 = 4 > 0$, Does not satisfy Condition 4
- Case 3: $\mu_1 = 0, \mu_2 > 0$ (Second constraint is active), then $(x_1^*, x_2^*) = (0, 1)$ and $\mu_2 = 4 > 0$ Does not satisfy Condition 3
- Case 4: $\mu_1 > 0, \mu_2 > 0$ (Both constraints are active), then $(x_1^*, x_2^*) = (-2, 5)$ and $(x_1^*, x_2^*) = (1, 2)$
  - $(x_1^*, x_2^*) = (-2, 5) \implies \mu_1 + 4\mu_2 = -4$
  - $(x_1^*, x_2^*) = (1, 2) \implies \mu_1 = 6$, and $\mu_2 = 2$,

Constrained Optimization

KKT Conditions: Problem

KKT Conditions

1. $2x_1 - \mu_1 + 2\mu_2 x_1 = 0$
2. $4x_2 - \mu_1 - \mu_2 = 0$
3. $x_1 + x_2 \geq 3$
4. $x_2 - x_1^2 \geq 1$
5. $\mu_1(x_1 + x_2 - 3) = 0$
6. $\mu_2(x_2 - x_1^2 - 1) = 0$
7. $\mu_1, \mu_2 \geq 0$

- Case 1: (No active constraint) , then $(x_1^*, x_2^*) = (0, 0)$
- Case 2: $\mu_1 > 0, \mu_2 = 0$ (First constraint is active), then $(x_1^*, x_2^*) = (2, 1)$ and $\mu_1 = 4 > 0$, Does not satisfy Condition 4
- Case 3: $\mu_1 = 0, \mu_2 > 0$ (Second constraint is active), then $(x_1^*, x_2^*) = (0, 1)$ and $\mu_2 = 4 > 0$ Does not satisfy Condition 3
- Case 4: $\mu_1 > 0, \mu_2 > 0$ (Both constraints are active), then $(x_1^*, x_2^*) = (-2, 5)$ and $(x_1^*, x_2^*) = (1, 2)$
  - $(x_1^*, x_2^*) = (-2, 5) \implies \mu_1 + 4\mu_2 = -4$
  - $(x_1^*, x_2^*) = (1, 2) \implies \mu_1 = 6$, and $\mu_2 = 2$,

Constrained Optimization

KKT Conditions: Problem

KKT Conditions

1. $2x_1 - \mu_1 + 2\mu_2 x_1 = 0$
2. $4x_2 - \mu_1 - \mu_2 = 0$
3. $x_1 + x_2 \geq 3$
4. $x_2 - x_1^2 \geq 1$
5. $\mu_1(x_1 + x_2 - 3) = 0$
6. $\mu_2(x_2 - x_1^2 - 1) = 0$
7. $\mu_1, \mu_2 \geq 0$

- Case 1: (No active constraint) , then $(x_1^*, x_2^*) = (0, 0)$
- Case 2: $\mu_1 > 0, \mu_2 = 0$ (First constraint is active), then $(x_1^*, x_2^*) = (2, 1)$ and $\mu_1 = 4 > 0$, Does not satisfy Condition 4
- Case 3: $\mu_1 = 0, \mu_2 > 0$ (Second constraint is active), then $(x_1^*, x_2^*) = (0, 1)$ and $\mu_2 = 4 > 0$ Does not satisfy Condition 3
- Case 4: $\mu_1 > 0, \mu_2 > 0$ (Both constraints are active), then $(x_1^*, x_2^*) = (-2, 5)$ and $(x_1^*, x_2^*) = (1, 2)$
  - $(x_1^*, x_2^*) = (-2, 5) \implies \mu_1 + 4\mu_2 = -4$
  - $(x_1^*, x_2^*) = (1, 2) \implies \mu_1 = 6$, and $\mu_2 = 2$,

Constrained Optimization

KKT Conditions: Problem

KKT Conditions

1. $2x_1 - \mu_1 + 2\mu_2 x_1 = 0$
2. $4x_2 - \mu_1 - \mu_2 = 0$
3. $x_1 + x_2 \geq 3$
4. $x_2 - x_1^2 \geq 1$
5. $\mu_1(x_1 + x_2 - 3) = 0$
6. $\mu_2(x_2 - x_1^2 - 1) = 0$
7. $\mu_1, \mu_2 \geq 0$

- Case 1: (No active constraint), then $(x_1^*, x_2^*) = (0, 0) \implies$ Violates 3 and 4
- Case 2: $\mu_1 > 0, \mu_2 = 0$ (First constraint is active), then $(x_1^*, x_2^*) = (2, 1)$ and $\mu_1 = 3 > 0 \implies$ Violates 4
- Case 3: $\mu_1 = 0, \mu_2 > 0$ (Second constraint is active), then $(x_1^*, x_2^*) = (0, 1)$ and $\mu_2 = -1 < 0$ $\implies$ Violates 3 and 7
- Case 4: $\mu_1 > 0, \mu_2 > 0$ (Both constraints are active), then $(x_1^*, x_2^*) = (-2, 5)$ and $(x_1^*, x_2^*) = (1, 2)$
  - $(x_1^*, x_2^*) = (-2, 5) \implies \mu_1 = +4, \quad \mu_2 = -4$ not possible, Violates 7
  - $(x_1^*, x_2^*) = (1, 2) \implies \mu_1 = 6$, and $\mu_2 = 2$, all KKT conditions satisfied

Example: Constrained Optimization

- Find the semi-major and semi-minor axes of the ellipse defined by

$$(x_1 + x_2)^2 + 2(x_1 - x_2)^2 - 8 = 0$$

- Hint: Calculate the farthest (nearest) point on the ellipse from the origin
- Problem formulation:

$$\begin{aligned} \min \quad & x_1^2 + x_2^2 \\ \text{s.t.} \quad & (x_1 + x_2)^2 + 2(x_1 - x_2)^2 - 8 = 0 \end{aligned} \tag{13}$$

Numerical Optimization

Different Types of Algorithms

- Optimization Algorithms:
  generate a sequence of iterates: $\{x_k\}_0^\infty$
- Termination Criteria:
  - No more progress can be made
  - A solution has been approximated with sufficient accuracy
- How to generate a new point x_{k+1} from x_k
  Use information about f at x_k and/or previous iteration points
- x_{k+1} must be such that $f(x_{k+1}) < f(x_k)$
- Strategies to find a new x_{k+1}:
  - Line search
  - Trust region

Numerical Optimization

Different Types of Algorithms

- Strategies
  - Line search
  - Trust region
- Line search Strategy: Choose a direction p_k from x_k so that $f(x_{k+1}) < f(x_k)$
- $x_{k+1} = x_k + \alpha p_k, \quad \alpha > 0$
- Choose a direction p_k and find a step length α.
- Objective function:

$$\min_{\alpha} \quad f(x_k + \alpha p_k) \tag{14}$$

Numerical Optimization

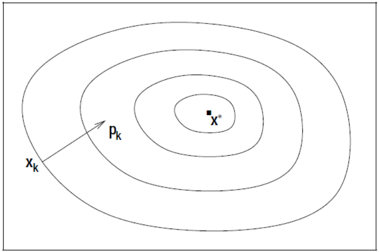
Different Types of Algorithms

- Line Search: Objective function:

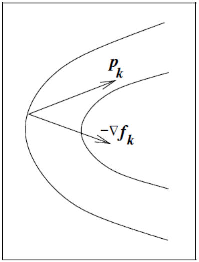
$$\min_{\alpha} \quad f(x_k + \alpha p_k) \tag{15}$$

- The best $p_k = -\nabla f(x_k) = \left(\frac{\partial f}{\partial x}\right)_{x_k}$

Steepest Descent direction



Line search: Descent direction



Numerical Optimization

Different Types of Algorithms

- Line Search: Objective function:

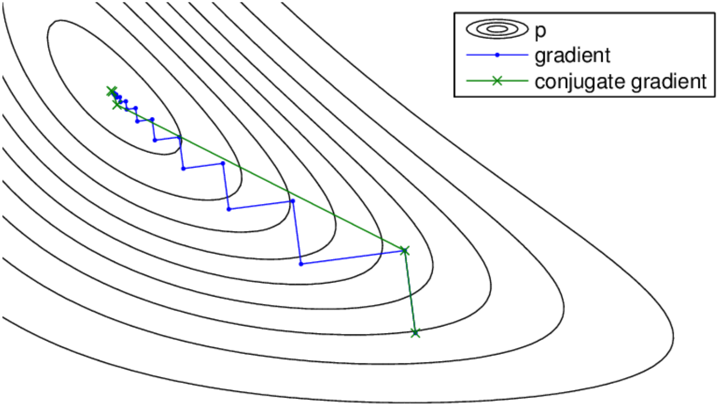
$$\min_{\alpha} \quad f(x_k + \alpha p_k) \tag{16}$$

- The best $p_k = -\nabla f(x_k) = \left(\frac{\partial f}{\partial x}\right)_{x_k}$
- Newton's direction:
  $p_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$
- Quadratic approximation of $f(x)$ is sufficient to represent the $f(x)$:-> Netwon's direction is reliable.
- Conjugate gradient directions:
  $p_k = -\nabla f(x_k) + \beta_k p_{k-1},\ \beta_k$: scalar value
  Conjugate vectors: $p_i \mathbf{A}\, p_j = 0$, for $i \neq j$ and $\mathbf{A}$: Symmetric positive definite matrix

Numerical Optimization

Line Search Strategy

- Why conjugate gradient Directions*

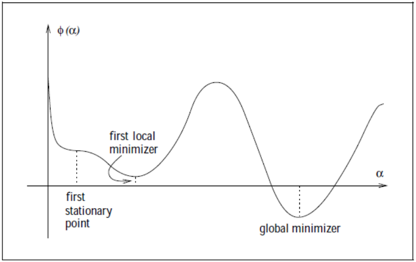


*: Honkela, A. et al., Journal of Machine Learning Research, 11, 2010.

Numerical Optimization

Line Search Strategy

- Step size α
  - p_k: Provide direction
  - α_k : Helps in reducing f value
  - Challenge: Many evaluations of f (some time ∇f)



- Simplest Sufficient condition on α_k, $f(x_k + \alpha_k p_k) < f(x_k)$
- Several Conditions: Wolfe Conditions, Goldstein Conditions.

Numerical Optimization

Trust Region

- Trust region method
- Idea: Construct a model function m_k at x_k using information on f
- m_k may not be a good approximation of f
- Limit search for x_{k+1} in a region: Trust region
- Objective function

$$\min \quad m_k(x_k + p_k) \tag{17}$$

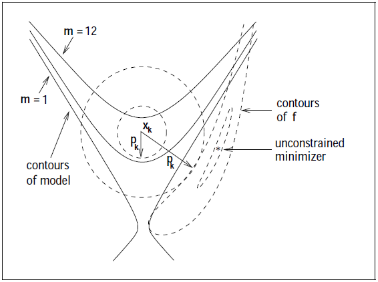
Numerical Optimization

Trust Region

- Not sufficient decrease in f for the candidate solution, change the trust region
- Trust region method

$$\min \quad m_k(x_k + p_k) \tag{18}$$

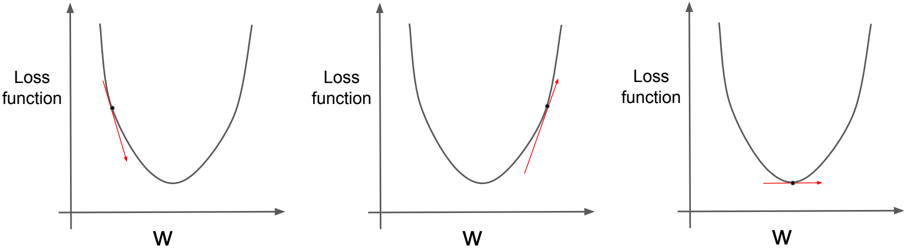
- m_k: Quadratic approximation



Optimization for Machine and Deep Learning: Terminology

- Central Idea to DS/ML/AI: Minimize or maximize an objective function
- Deep learning objective: minimize the loss function or objective function
- Direct solution & Iterative solution
- Arthur Samuel's paradigm:
  "Mechanism" to improve performance by tweaking weights (parameters)
- Loss function can be tweaked by varying the model parameters
  Million-dimensional space!

Gradient Descent



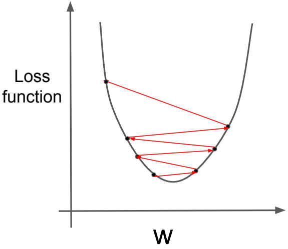
$\nabla Loss w < 0$	$\nabla Loss w > 0$	$\nabla Loss w = 0$
Increment w	Decrement w	Optimum w
$w = w + \Delta w$	$w = w - \Delta w$	$\Delta w = 0$ (stationary point)

What should be $\Delta\, w$?

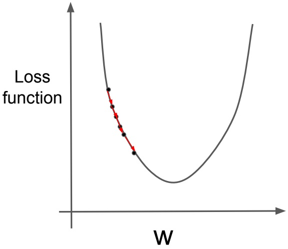
$\Delta w = -\eta \nabla Loss w \Rightarrow\ w - \eta \nabla Loss w$, where η : Learning rate

Gradient Descent
Learning Rate

Learning rate is too big⇒ Oscillation diverges



Learning rate is small⇒ Long time to convergence



- Learning rate should be:
  - Near local solution: proceed quickly with small learning rate
  - Far from local solution: proceed quickly with large learning rate

Gradient Descent

Three types

Batch GD

- Use entire data set for computing gradient
- Update rule

$$w = w - \eta \nabla_w \mathrm{L}(\mathrm{w})$$

- Slow
- large data set
  does not fit in memory

Stochastic GD (online)

- Use data point (x_i, y_i) for computing gradient
- Update rule

$$w = w - \eta \nabla_w \mathrm{L}\left(w, x_i, y_i\right)$$

- Much faster
- Update with high variance
  L fluctuates heavily

Mini-batch GD

- Use a subset of data points for computing gradient
- Update rule
  $w = w - \eta \nabla_w \mathrm{L}(w, x_{i:i+n}, y_{i:i+n})$
- Best of both methods
- Reduces variances in parameter updates

Algorithm of choice: Mini-batch gradient descent for NN and DL
Mini-batch GD is often referred to as SGD
Typical batch size of 50 or 256 are used

Gradient Descent

Momentum SGD

- SGD: Sometimes slow
- How do we accelerate the learning rate?
- Momentum approaches:
  - Use the concept of momentum from physics
  - v: velocity variable
- Update rule

$$w_{k+1} = w_k - \eta \nabla_w \mathsf{L}(w, x_i, y_i) + \beta v$$

- v: accumulates the past gradients
- $\beta \in [0, 1)$: momentum parameter
- Larger β, current direction is affected by the more past gradients

Gradient Descent

Stochastic GD

- Learning rate: An important hyperparameter
- Adaptive learning rate: Direction and magnitude of gradients for different parameters
- AdaGrad Learning rate:
  Scale learning rates of all parameters by the square root of the sum of all the past gradient squares

$$w_{k+1} = w_k - \frac{\eta .}{\delta + \sqrt{v_{k+1} .}} \odot \nabla_w L, \;\; v_{k+1} = v_k + (\nabla_w L(w_k)) \odot (\nabla_w L(w_k))$$

- RMSProp Learning rate: Exponential moving average instead of sum

$$w_{k+1} = w_k - \frac{\eta .}{\sqrt{\delta + v_{k+1} .}} \odot \nabla_w L,$$

$$v_{k+1} = \rho\, v_k + (1-\rho)(\nabla_w L(w_k)) \odot (\nabla_w L(w_k))$$

- Other algorithm: Adaptive moments "Adam": Combination of RMSProp and Momentum SGD

Gradient Descent

Stochastic GD: Conclusions

- Which algorithm should I choose for a particular application?
- Which is the best algorithm to apply?
- Answer: None or all
- Robust performance: RMSProp and AdaDelta family[1]
- Hyper-parameter tuning does not matter much in adaptive learning
- Higher power concludes

  Choice of algorithm depends on user's knowledge of algorithm

[2] Schaul, et al. "No more pesky learning rates." International Conference on Machine Learning. 2013.

Gradient Descent

Stochastic GD: Summary

- The element for applying stochastic GD
  - The loss function form
  - A way to compute gradients wrt parameters
  - Initialization for parameters and learning rate and other hyperparameters

[2] Schaul, et al. "No more pesky learning rates." International Conference on Machine Learning. 2013.