ELEN90026 Introduction to Optimisation

Lecture Notes and Final Revision

Lecturer: Dr. Iman Shames

*** by Shijie Huang ***

August 9, 2018

Contents

1 Lecture 1: An Introduction to Optimization and Some Basics

1.1 Lecture Outline

1. Some history
2. Introduction of a General Optimization Problem
3. Complexity Bounds For a Problem
4. Some Necessary Background to Construct and Analyse Optimization Algorithms
5. Definition of minima and Basic Theorems for Identifying Them

1.2 A General Optimization Problem

A general optimization problem

$$\min_x f(x)$$
$$s.t. \quad x \in X$$

- $f(x)$ is the *object function* or *cost function*
- x is the decision variable (or control in old literature)
- X: feasible set (continuous or discrete)

standard form of optimization problem

$$\min_x f(x)$$
$$s.t. \quad c_i(x) = 0, i \in \varepsilon, \qquad c_i(x) \geq 0, \quad i \in I$$

- f and c_i, $i \in \varepsilon \cup i$ are differentiable functions
- $c_i(x) = 0$, $i \in \varepsilon$: equality constraints
- $c_i(x) \geq 0$, $i \in I$: inequality constraints

Definition: Feasible Set

Define feasible set X to be the set of points x that satisfy the constraints

$$X = \{x | c_i(x) = 0, i \in \varepsilon, \quad c_j(x) \geq 0, \quad j \in I\}$$

1.3 Different Types of Optimization Problem

- Constrained vs. Unconstrained Optimization
- Global vs. Local Optimization
- Stochastic vs. Deterministic Optimization
- Convex vs. Non-convex Optimization

An optimization method M is applied to a problem P of a class C with an oracle O which is just a unit, that answers the successive questions of the method.
Types of Oricle:
1. Zero-order Oracle: the value (derivative-free method)
2. First-order Oracle: the value and the gradient (gradient descent)
3. Second-oder Oracle: the value, the gradient and the hessian (newton, quasi-newton, BFGS)

1.4 Complexity Bounds for Global optimization

Another way of saying is "why global optimization is inefficient?" Consider the following problem:

$$\min_x f(x), \quad s.t. \;\; 0 \leq x_i \leq 1, \;\; i = 1, ..., n$$

where f is lipschitz on the feasible set with Lipschitz constant L, i.e.

$$||f(x) - f(y)|| \leq L||x - y|| \;\; \forall x, y \in X$$

Note that this is equivalent of saying the first derivative (slope) is bounded by L.

Theorem

Let f^* be the global minimum of the cost function

$$0 \leq f(\bar{x}) - f^* \leq L\frac{\sqrt{n}}{2p}$$

Theorem

A uniform grid of at least $(\lfloor L\frac{\sqrt{n}}{2\varepsilon} \rfloor + 2)^n$ points is required to achieve $f(\bar{x}) - f^* \leq \varepsilon$

Therefore, to get a relatively close global optimum, the lower bound is $(\lfloor L\frac{\sqrt{n}}{2\varepsilon} \rfloor)^n$, which can be terrible number (no. of operations and function validations)

1.5 Some Basics Before Learning Optimization

1.5.1 Sequences

Definition (Sequence Convergence)

1. $\bar{x} \in R^n$ is an accumulation point or limit point for $\{x_k\}$ if there is an infinite subsequence $\{t_k\}_{k \in N}$, such that the subsequence $\{t_k\}_{k \in N}$ converges to $\bar{x}$, or:

$$\lim_{k \to \infty} x_{t_k} = \bar{x}$$

2. For any $\varepsilon > 0$ and $K > 0$, we have:

$$||x_k - x|| \leq \varepsilon, \text{ for some } k \geq K$$

Note: A sequence converges iff it has exactly one limit point.

Definition (Cauchy Sequences)

There exists an integer K such that $\forall\ k \geq K$ and $j \geq K$, we have

$$||x_k - x_j|| \leq \varepsilon$$

where $\varepsilon > 0$, then $\{x_i\}$ is a Cauchy Sequence.

Every sequence that converges is a Cauchy sequence (Converse is also true when we are in R^n).

For a scalar sequence $\{x_k\}$

- bounded above if $x_k \leq u$
- bounded below if $x_k \geq l$
- non-decreasing if $x_{k+1} \geq x_k$ for all k
- non-increasing if $x_{k+1} \leq x_k$ for all k

Sequence $\{x_k\}$ **converges** if:
Non-decreasing and bounded above
Non-increasing and bounded below

	Notation	Condition
Supremum	$\sup\{x_k\}$	smallest real number u such that $x_k \leq u$, for all k
Infimum	$\inf\{x_k\}$	largest real number l such that $x_k \geq l$, for all k
Suprema$\{u_i\}$	u_i=$\{x_k \vert k \geq i\}$	if u_i nonincreasing, converges to $\bar{u} = \lim sup x_k$
Infima$\{l_i\}$	$l_i = inf\{x_k \vert k \geq i\}$	if u_i nondecreasing, converges to $\bar{l} = \lim inf x_k$

1.5.2 Rate of Convergence

Let $\{x_k\}$ converges to x^*.

$$\frac{||x_{k+1} - x^*||}{||x_k - x^*||^\rho} \leq r, \text{ for all sufficiently large } k$$

r is the convergence factor, and the convergence rate can be illustrated as follows:

Rate of Convergence	Definition
Q-linear Convergence	$\rho = 1$, $r \in (0, 1)$
Q-superlinear Convergence	$\rho = 1$, $r \to 0$ or $\lim_{k\to\infty} r = 0$
Q-quadratic, cubic... Convergence	$\rho = 2, 3, ...$, bounded constant r (does not have to be < 1)
R (Root) - Convergence	if there is a sequence $\{v_k \vert v_k \geq 0\}$ for all k, and $\{v_k\}$ converges Q-linearly(Q-superlinearly, Q-quadratically)

1.5.3 Topology of Euclidean Space R^n

Definition (Bounded Set)

Set $F \subset R^n$ is bounded if $||x|| \leq M, \quad M > 0, \quad \forall x \in F$

Definition (Open Set)

Set $F \subset R^n$ is open if $B(x, \varepsilon) = \{y \in R \vert \vert\vert x - y|| < \varepsilon\} \subset F$ for some $\varepsilon > 0$

Definition (Closed Set)

Set $F \subset R^n$ is closed if all limit points of all $\{x_k\} \subset F$ are elements of F

Definition (Interior)

The interior of a set F, denoted by $int(F)$, is the largest open set contained in F.

Definition (Closure)

The closure of F denoted by $cl(F)$, is the smallest closed set containing F, or $x \in cl(F)$, if $\lim_{k\to\infty} x_k = x$ for some sequence $\{x_k\}$ in F.

Definition (Neighbourhood)

Given a point $x \in R^n$, we call $N \subset R^n$ a neighbourhood of x if it is an open set containing x.

(A neighbourhood of a point is a set of points containing that point where one can move some amount (ε) away from that point without leaving that set)

Definition (Compact Set)

The set F is compact if every sequence $\{x_k\}$ of points in F has at least one limit point, and all such limit points are in F.
(Proof: $F \subset r^n$ is closed and bounded $\implies F$ is compact.)

Theorem (Weierstrass) existence of minimisers

If $X \subset R^n$ is non-empty and compact (closed and bounded), and f is continuous, there exists a global minimizer of the optimization problem

$$\min_x \ f(x), \quad s.t. \ \ x \in X$$

This theorem is useless in terms of designing an efficient algorithm.

Definition (Local Minimiser)

vector $x^* \in X$, N is the neighbourhood of x^*
x^* is local minimiser if $f(x) \geq f(x^*)$ for all $x \in X \cap N$
(Strong/strict) local minimiser: $f(x) > f(x^*)$ for all $x \in X \cap N$ and $x \neq x^*$
Global Minimiser: when $N = X$

Definition (Isolated Local Minimiser)

vector $x^* \in X$, N is the neighbourhood of x^*
x^* is an isolated local minimiser if there is a neighbourhood N if x^* such that x^*

1.5.4 Useful Theorems for Recognizing a Local Minimum

Theorem (Taylor's Theorem)

If $f : R^n \to R$ is continuously differentiable and $p \in R^n$.
Then we have:

$$f(x+p) = f(x) + \nabla f(x+tp)^T p$$

If f is twice differentiable:

$$\nabla f(x+p) = \nabla f(x) + \int_0^1 \nabla^2 f(x+tp)^T p$$

$$f(x+p) = f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x+tp) p$$

for some $t \in (0,1)$

For unconstrained optimization

Theorem (First order Necessary Optimality condition)

If x^* is the local minimiser, f is differentiable in an open neighbourhood of x^*, then $\nabla f(x^*) = 0$

Theorem (Second-Order Necessary Optimality condition)

If x^* is the local minimiser, f is **twice** differentiable in an open neighbourhood of x^*, then $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \geq 0$

Theorem (Second-Order sufficient Optimality condition)

If f is **twice** differentiable in an open neighbourhood of x^*, and

$$\nabla f(x^*) = 0 \text{ and } \nabla^2 f(x^*) > 0$$

Then x^* is an **isolated** local minimiser.

2 Lecture 2: Line Search Method

2.1 Lecture Outline

1. Line search methods
2. Step length conditions
3. Convergence of line search methods and their rate
4. Newton's method and its convergence
5. Quasi-Newton methods convergence

2.2 Line Search Methods

2.2.1 The Goal

Consider the following:
The goal is to construct a sequence $\{x_k\}$ through an **iterative method** that converges to some x^* at which the optimality conditions are satisfied, i.e.:
First order necessary condition:

$$||\nabla f(x^*)|| = 0$$

Second order **necessary** condition:

$$||\nabla f(x^*)|| = 0, \nabla^2 f(x^*) \geq 0$$

Second order **sufficient** condition:

$$||\nabla f(x^*)|| = 0, \nabla^2 f(x^*) > 0$$

2.2.2 Line Search Iteration

$$x_{k+1} = x_k + \alpha_k p_k$$

> α_k: step length (step size)
> p_k: search direction
> **So the question here is how do we choose α_k and p_k?**
> And remember we want the iterations toward the optimal x^*

2.2.3 A General Choice of the Search Direction

Idea: choose p_k = descent direction, i.e. $p_k^T f_k < 0$, or the angle between two vectors (direction you move towards and the gradient direction) is greater than 90 degrees so that the direction you move towards is the right direction.
often we choose the following direction:

$$p_k = -B_k^{-1}\nabla f_k$$

What is B_k^{-1}? It is a matrix that is symmetric and non-singular (+ve definite)
Gradient Descent Method: $B_k = 1$
Newton Method: $B_k = \nabla^2 f_k$ Hessian matrix
Quasi-Newton Method: B_k approximates the Hessian and is updated with low cost

2.2.4 Step Length

Informally, we want the step length to be:
(1) small enough to result in sufficient decrease
(2) long enough to not get stuck
Introduce the first condition **Armijo condition** :

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T p_k, \quad c \in (0, 1)$$

Note:
(1). $c_1 \alpha \nabla f_k^T p_k$ is negative so the next f_{k+1}, $f(x_k + \alpha p_k)$ is smaller than current f_k [1]
(2). c_1 is chosen to be very small in practice.
The problem with Armijo condition is that for small α, it will always satisfy (yet the step length could be too small that we may get stuck, or results in too long iterations).
The solution to that is to introduce a second condition, **curvature condition**:

$$\nabla f(x_k + \alpha p_k)^T p_k \geq c_2 \nabla f_k^T p_k, \quad c_2 \in (c_1, 1)$$

Note $\nabla f(x_k + \alpha p_k)^T p_k$ is the derivative of $f(x_k + \alpha p_k)$, and in practice we choose $c_2 = 0.9$
Sum up those two conditions, we have:

Wolfe Condition

Guarantee suffcient decrease: $f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T p_k, \quad c_1 \in (0, 1)$

Guarantee not too small step sizes: $\nabla f(x_k + \alpha p_k)^T p_k \geq c_2 \nabla f_k^T p_k, \quad c_2 \in (c_1, 1)$

Strong Wolfe Condition: No longer allow the derivative to be too positive

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T p_k, \quad c_1 \in (0, 1)$$

$$|\nabla f(x_k + \alpha p_k)^T p_k| \geq c_2 |\nabla f_k^T p_k|, \quad c_2 \in (c_1, 1)$$

Note there is no guarantee that Wolfe Condition will have results that is close to a minimizer.

[1] The goal is min f

Theorem (Satisfiability of the Wolfe Condition)

Suppose that $f : R^n \to R$ is continuously differentiable. Let p_k be a descent direction atx_k, and assume that f is bounded below along the ray $\{x_k + \alpha p_k | \alpha > 0\}$. Then if $0 < c_1 < c_2 < 1$, there exist intervals of step lengths satisfying the (Strong) Wolfe condition

Note: other conditions, Goldstein conditions, can also be used (mostly in Newton type, not suited for quasi-newton type).
Note: curvature condition can be dispensed with backtracking approach.

2.2.5 Convergence of Line Search Method

So how do we know that this line search method converges?
Consider a Newton-like method:

$$x_{k+1} = x_k + \alpha_k p_k, \quad p_k = -B_k^{-1} \nabla f_k$$

Gradient descent, $B_k = I$, converges to a **stationary point**[2] if the step length α_k satisfy Wolfe or Goldstein condition.
Newton Like: assume $B_k > 0$ with a uniformly bounded condition number, i.e. $\exists M$ such that:

$$||B_k||||B_k^{-1}|| \leq M, \quad \forall k$$

Then we know that:

$$\cos\theta_k \geq 1/M \quad [3]$$

$$\lim_{k\to\infty} ||\nabla f_k|| = 0$$

Newton or Newton-like methods are **globally convergent** if
(1) B_k has bounded condition number, and +ve definite
(2) Step length α satisfy Wolfe or Goldstein conditions

Some discussion:
(1) Gradient descent globally convergence, but too slow
(2) Pure newton iteration converges rapidly if you start close enough, but could stuck in the local optimum or diverge from the optimal solution.

1. Steepest Descent

[2] No guarantee that the method converges to a minimiser, but only it is attracted by stationary points
[3] proof in assignment 1 question 3, textbook question 3.5

Theorem (Convergence of the Steepest Descent)

Suppose that
(1) $f : R^n \to R$ is **twice continuously differentiable.**
(2) Iteration generated by the gradient descent method with exact line search converge to a point x^* where Hessian matrix $\nabla^2 f(x^*) > 0$ (i.e. the algorithm can converge to an stationary point)
Then we have

$$f_{k+1} - f^* \leq r^2(f_k - f^*) \text{ where } r \in (\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}, 1)$$

$0 \leq \lambda_1 ... \leq \lambda_n$ are eigenvalues of Hessian matrix $\nabla^2 f(x^*)$

2. Newton's Method

1. Newton Raphson method is a root finding method. In optimization it finds the roots to $\nabla f(x) = 0$. Or solve the following system of equation

$$\nabla^2 f_k p_k = -\nabla f_k \quad [4]$$

2. The idea is to linearise around each iteration point x_k to find the next value $x_{k+1} = x_k + p_k$, where p_k is the Newton step/update at step k.
3. There is no guarantee of convergence
4. Each step do not necessarily get closer to answer
5. **One big big problem: Hessian is not always +ve definite**, p_k may not always be a descent direction
6. Therefore, here we only talk about local convergence (that is, assume we have a very good initial guess)
7. Newton Update

$$x_{k+1} = x_k + (-(\nabla f(x_k))^{-1}) f(x_k))$$

$$x_{k+1} = x_k - (f(x_k))(\nabla f(x_k))^{-1}$$

$$F(x_k) + \nabla F(x_k)\Delta x = 0 \quad [5]$$

8. For Newton to work, we really need to start close enough to the solution.

[4] so that $x_{k+1} - x_k = 0$
[5] $\Delta x = x_{k+1-x_k}$, this is a root finding equation, $F(x)$ could be a function, could be a derivative of a function

Theorem (Convergence of the Newton Method)

Suppose that

(1) $f : R^n \to R$ is **twice continuously differentiable.**

(2) Hessian $\nabla^2 f(x)$ is Lipschitz continuous in a neighbourhood of a solution x^* at which sufficient optimality conditions are satisfied [a]

Then we have the following results:

1. if the starting point x_0 is sufficiently close to x^* [b], the sequence of iterations converges to x^*.
2. Rate of convergence is quadratic
3. The sequence $\{||\nabla f_k||\}$ converges quadratically to zero.

[a] i.e. $\nabla f(x^*) = 0$, $\nabla^2 f(x^*) > 0$

[b] Meaning we can find an stationary point

3. Quasi-Newton Method

Theorem (Convergence of the Quasi-Newton Method)

Suppose that

(1) $f : R^n \to R$ is **twice continuously differentiable.**

(2) Iteration is: $x_{k+1} = x_k + p_k$ [a]

(3) $p_k = -B_k^{-1}\nabla f_k$ and B_k is symmetric and +ve definite

(4) $x_k \to x^*$ such that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) > 0$

Then $\{x_k\}$ converges superlinearly [b] iff:

$$\lim_{k\to\infty} \frac{||(B_k - \nabla^2 f(x^*))p_k||}{||p_k||} = 0$$

[a] Here assume $\alpha_k = 1$

[b] Generally speaking, Quasi-Newton methods normally satisfy the condition and are therefore superlinearly convergent

2.2.6 Newton Method with Hessian Modification

In Newton's method, if we pick an initial point that is far away from the solution, then p_k may not be a descent direction.

The solution is to solve the following modified system of equation:

$$(\nabla^2 f_k + E_k)p_k = -\nabla f_k$$

where E_k either a positive diagnoal matrix or a full matrix.

Why modification? To ensure B_k is sufficiently positive definite

Definition (Bounded Modified Factorization Property)

Whenever the sequence of Hessians $\{\nabla^2 f_k\}$ is bounded, the matrices in the sequence $\{B_k\}$ have bounded condition, that is:

$$k(B_k) = ||B_k||||B_k^{-1}|| \leq C, \quad C > 0, \quad \forall k \geq 0$$

Theorem (Convergence of the Quasi-Newton Method)

Suppose that
(1) $f : R^n \to R$ is **twice continuously differentiable.** on an open set N
(2) Starting point x_0 of the Algorithm is such that the sublevel set $L = \{x \in N | f(x) \leq f(x_0)\}$ is compact.
(3) If Bounded Modified Factorization Property holds, we have:

$$\lim_{k \to \infty} ||\nabla f(x_k)|| = 0$$

2.2.7 Line Search Newton with Hessian Modification

Algorithm 1 Line Search Newton with Modification

function F1(A[.], B[.], n, s)
    Initialise x_0
    for k= 0, 1, 2, ... **do**
        $B_k \leftarrow \nabla^2 f_k + E_k$ ▷ E_k is chosen to ensure B_k is sufficiently positive definite;
        Solve $B_k p_k = -\nabla f_k$
        $x_{k+1} \leftarrow x_k + \alpha_k p_k$ ▷ α_k satisfies the Wolfe, Goldstein, or Armijo backtracking conditions
    end for
end function

3 Lecture 3: Convexity and Convex Optimality

3.1 Lecture Outline

1. Convex sets
2. Convex functions
3. Convex optimization Problems
4. Optimality conditions for convex problems

3.2 Affine and Convex Sets

Affine Set

An affine set is a set that contains the line through any two distinct points in it.

How to prove? Pick a linear combination of the other two points, prove that the point is on the same line

A function f is linear if $f(ax + by) = af(x) + b$ for all relevant values of a, b, x and y.

A function g is affine if $g(x) = f(x) + c$ for some linear function f and constant c. Note that we allow $c = 0$, which implies that every linear function is an affine function.

Affine transformation: a point multiply by matrix and add a vector to it.

Convex Set

A convex set contains the line segment between any two points in the set (i.e. connect two points, the line is in the set)

How to prove? Pick a line $z = \theta x_1 + (1 - \theta)x_2$, prove z is in the same convex set

Convex Combination

$$x = \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n, \quad \theta_1 + ...\theta_n = 1, \quad \theta_i \geq 0$$

Convex Hull

Set of all convex combinations of members of a set (i.e. choose all possible convex combinations, and convex hull is the smallest convex set that contains the original set) [a]

[a] Note original set is not necessarily convex, but convex hull is

Convex Cone

A convex cone is a set that contains all conic combinations of the points in the set.
Additionally, a convex cone is a **proper cone** if:
1. it is closed (Contains its boundary);
2. it is solid (has non-empty interior;
3. it is pointed (contains no lines)
e.g. Positive semi-definite cone:

$$\{X \in R^{n\times n} | X = X^T, v^T X v \geq 0. \forall v \neq 0\}$$

3.2.1 Excursion: (Semi) Definite Matrices

Definition ((Semi) Definite Matrices)

A **symmetric matrix** P [a], is positive (semi-) definite, $P \succ 0 (P \succeq 0)$ iff:

$$(1)\ \forall v \neq 0, v^T P v > 0 (v^T P v \geq 0)$$

(2) or equivalently the smallest eigen-value of matrix $\lambda_{\min}(P) > 0 (\lambda_{\min}(P) \geq 0)$

[a] $P = P^T$

Definition (Generalised Matrices Inequality)

$$X \leq Y \Leftrightarrow x_i \leq y_i, \quad i = 1, ..., n$$

3.2.2 Convex Sets (Continued)

Some convex sets and relative proofs

How to prove?
1. pick any two points that satisfy the definition
2. show their linear combination satisfy convex relationship

1. Hyperplane: $\{x | a^T x = b\}$ (affine: a system of linear equations, and all solutions of it consist of an affine set)
2. Halfspace: $\{x | a^T x \leq b\}$

$$\text{Assume } z = \theta x_1 + (1-\theta) x_2 \text{ where } a^T x_1 \leq b \text{ and } a^T x_2 \leq b$$

$$a^T z = a^T(\theta x_1) + a^T(1-\theta) x_2$$

$$= \theta a^T x_1 + (1-\theta) a^T x_2$$

$$\leq \theta b + (1-\theta)b = b$$

3. Norm balls with centre x_c and radius r:

$$B(x_c, r) = \{x | ||x - x_c|| \leq r\}$$

Proof:

Assume x_1, x_2 where

$$||x_1 - x_c|| \leq r$$

$$||x_2 - x_c|| \leq r$$

$$||\theta x_1 + (1-\theta)x_2 - x_c||$$

$$||\theta x_1 - \theta x_c + (1-\theta)x_2 - (1-\theta)x_c||$$

$$||\theta(x_1 - x_c) + (1-\theta)(x_2 - x_c)||$$

By Triangle inequality $||x + y|| \leq ||x|| + ||y||$

$$\leq \theta||x_1 - x_c|| + (1-\theta)||x_2 - x_c||$$

$$\leq \theta r + (1-\theta)r$$

$$\Rightarrow ||\theta x_1 + (1-\theta)x_2 - x_c|| \leq r$$

3.3 Convex Function

Definition (Convex Function)

A function $f : D \to R$ is convex if:

1. D is convex and
2. $\forall x, y \in D,\ \theta \in [0,1] : f((1-\theta)x + \theta y) \leq {}^{a}(1-\theta)f(x) + \theta f(y)$

[a] if **strictly convex (not strong!)**, the sign is $<$

Definition (Epigraph)

$$epi(f) = \{(x, s) | x \in D, s \geq f(x)\}$$

Theorem (epigraph of f)

The epigraph of a function is convex iff the function itself is convex

Definition (Strong Convexity)

A function $f : D \to R$ is strongly covex with coefficient (modulus) σ if:

$$(1)\ D \text{ is convex}$$

$$(2)\ f((1-\theta)x + \theta y) + \frac{\sigma}{2}\theta(1-\theta)||x-y||^2 \leq (1-\theta)f(x) + \theta y$$

for all $\theta \in [0,1]$ and $x, y \in D$, and some $\sigma > 0$

Note that any **strongly convex** function is **strictly convex** but **NOT** vice versa.

Properties and Results

If f is differentiable:

$$(1)\ (\nabla f(x) - \nabla f(y))^T(x-y) \geq \sigma||x-y||^2$$

$$(2)\ f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{\sigma}{2}||x-y||^2$$

If f is twice differentiable:

$$\nabla^2 f(x) - \sigma I \geq 0\ ^a$$

[a] Hessian, curvature never goes to zero (bounded below by a >0 number)

3.4 Convex Optimization

Definition (Convex Optimization Problem)

The optimization problem: $\min_x f(x), \quad s.t. x \in X$
If the following holds:
(1) f is a convex function
(2) X is a convex set
Then the problem is called **convex optimization problem**

Note the problem is equivalent to: $\max -f(x), \quad s.t. \ \ x \in X$ if $-f$ is **concave function**.

Theorem (Local Implies Global Optimality for Convex Problem)

If an optimization problem is a convex optimization problem, then every **local minimum** is also a **global minimum**

Also note that if the function is strictly convex in the neighbourhood of the solution, the minimiser is unique (strict inequality).
For a convex optimization problem either there is a unique minimiser or the minimisers form a convex set ($\Rightarrow$ theorem).

Theorem (Convexity of the Set of Minimisers)

Consider the convex optimization problem described above. Then the set of its minimisers is convex.

Theorem (Convexity for Differentiable Functions)

If $f : D \to R$ is continuously differentiable, and D is a convex set, then f is convex iff the following holds:

$$\forall x, y \in D, \quad f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

Or in other words, tangent below the graph.

Theorem (Convexity for Twice Differentiable Functions)

If $f : D \to R$ is **twice** continuously differentiable, and D is a convex set, then f is convex iff the following holds:

$$\forall x \in D : \nabla^2 f(x) \geq 0$$

Theorem

The function $f : D \to R$ is convex iff:
$g : D_g \to R$ where $g(t) = f(x + tp)$ and $D_g = \{t | x + tp \in D\}$ is convex for any line restriction, i.e. for any $x \in D$ and $p \in R^n$.

3.4.1 Operations That Preserve Convexity

Operations That Preserve Convexity	
Input Transformation	
Extension	
Summation	
Point-wise Maximum	
Point-wise Supremum	
Minimisation	
Composition	
Persepective	

3.5 Standard Form of Convex Optimization Problems

An (constrained) optimization problem:

$$\min_x f(x)$$

$$s.t. \quad c_i(x) = 0, \quad i \in E, \quad c_i(x) \geq 0, \quad i \in I$$

Sufficient condition for the above to be convex problem

(1) f is convex
(2) $c_i(x)$ is affine for $i \in E$ (Or $Ax = b$)
(3) $c_i(x)$ is concave for $i \in I$

3.6 Optimality Condition for Convex Problems

Theorem (First Order Optimality (both sufficient and necessary) Condition for Convex Problems)

If $f(x)$ is continuously differentiable, and we have the following problem

$$\min_x f(x), \quad s.t. \;\; x \in X$$

A point x^* is a global minimiser iff:

$$\forall x \in X, \nabla f(x^*)^T (x - x^*) \geq 0 \quad [a]$$

[a] The value of the function will only increases in the direction of any derivatives (for all x)

Theorem (Unconstrained Convex Problems)

If $f(x)$ is continuously differentiable, and we have the following problem

$$\min_x f(x), \quad s.t. \;\; x \in X$$

A point x^* is a global minimiser iff:

$$\nabla f(x^*) = 0$$

4 Lecture 4: Quasi-Newton Method

4.1 Lecture Outlier

1. Quasi-Newton methods
2. BFGS Method and its properties
3. Limited-memory BFGS (most popular Quasi-Newton method)
4. BFGS for sparse problems and partially separable cost functions

4.2 Review of Newton Method and Why Quasi-Newton

In Newton Method, we can modify the Hessian.
We can look at Newton Method in a different perspective. We want to know why we want the direction to be

$$p = -(\nabla^2 f(x_k))^{-1} \nabla f(x)$$

Note, the goal in Newton Method is to approximate $f(x)$ with a quadratic function (i.e. Taylor's Theorem)
Assume unit step size $\alpha_k = 1$

$$f(x_k + p) = f(x_k) + p^T \nabla f(x_k) + \frac{1}{2} p^T \nabla^2 f(x_k) p + \text{ higher order terms}$$

$$f(x_k + p) = f(x_k) + p^T \nabla f(x_k) + \frac{1}{2} p^T \nabla^2 f(x_k) p$$

Let $m(p; x_k) = f(x_k) + p^T \nabla f(x_k) + \frac{1}{2} p^T \nabla^2 f(x_k) p$,

$$\nabla m(p; x_k) = \nabla f(x_k) + \nabla^2 f(x_k) p$$

If I want to minimize the quadratic function,

$$\nabla m(p; x_k) = 0$$

$$p = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

Note that here we need to inverse the Hessian $\nabla^2 f(x_k)$, which is costly $O(n^3)$, and often the Hessian is ill-conditioned (singularity) so that we are not able to inverse it.

4.3 Quasi-Newton Method

The standard unconstrained optimization problem:

$$\min_x f(x)$$

and the solution can be a *Line Search Iteration*:

$$x_{k+1} = x_k + \alpha_k p_k$$

where α_k is the step size that satisfies the Wolfe Condition.

Now here we concern more about the direction $p_k = -B_k^{-1}\nabla f_k$

In Newton Method:

$$p_k = -(\nabla^2 f_k)^{-1}\nabla f_k$$

However, as we mentioned, may be costly or even hard to calculate Hessian. Therefore, the idea for Quasi-Newton method is that we still use Newton method, however, we estimate B_k so that

$$B_k \to \nabla^2 f(x_k)$$

We do not have to know (and often do not know) Hessian, instead we approximate it [6]

4.4 BFGS Method

4.4.1 The Idea of BFGS

First form a **local** quadratic approximation (i.e. Taylor's theorem) of the cost function at time k:

$$m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p \quad [7]$$

$B_k \in R^{n\times n}$ n by n matrix and $B_k > 0$ positive definite.
(1) The minimiser of $m_k(p)$ is the search direction.
(2) Our object is to construct B_k iteratively so that Hessian is approximated.
So What requirements should be imposed on B_{k+1}?

The gradient of m_{k+1} should match that of f at x_k and x_{k+1} [8]

$$\nabla m_{k+1}(p) = \nabla f_{k+1} + pB_{k+1}$$

$$\nabla m_{k+1}(-\alpha_k p_k) = \nabla f_{k+1} - \alpha_k B_{k+1} p_k$$

We want the requirement stated above to be fulfilled:

$$\nabla m_{k+1}(-\alpha_k p_k) = \nabla f_k$$

[6] i.e. in Newton method, we need both Gradient and Hessian; In BFGS and Gradient descent, only gradient is required

[7] in Newton, $B_k = \nabla^2 f_k$

[8] i.e. keeep the derivative the same between two consecutive approximation

Therefore, we have an important requirement result:

$$\alpha_k B_{k+1} p_k = \nabla f_{k+1} - \nabla f_k$$

We define the following:

$$s_k = x_{k+1} - x_k = \alpha_k p_k \quad [a]$$

$$y_k = \nabla f_{k+1} - \nabla f_k$$

[a] Update rule: $x_{k+1} = x_k + \alpha_k p_k$

Then we have the secant equation:

Secant Equation

$$B_{k+1} s_k = y_k$$

Therefore, B_{k+1} is only positive definite when

$$s_k^T B_k s_k > 0 \Rightarrow s_k^T y_k > 0^a$$

[a] This curvature must be enforced

Curvature condition for Convex function

If f is strongly convex,

$$(x_{k+1} - x_k)^T (\nabla f_{k+1} - \nabla f_k) \geq \sigma ||x_{k+1} - x_k|| \quad [a]$$

The curvature condition will be satisfied for any two distinct points x_k and x_{k+1}

[a] One of the property of strong convexity assuming f is differentiable

Curvature condition under (Strong) Wolfe Condition for Line Search

Recall that we have (Strong) Wolfe Condition:

$$\nabla f_{k+1}^T p_k \geq c_2 \nabla f_k^T p_k, \quad c_2 \in (c_1, 1)$$

$$\nabla f_{k+1}^T \alpha_k p_k \geq c_2 \nabla f_k^T p_k$$

$$\nabla f_k^T s_k - \nabla f_k^T s_k + \nabla f_{k+1}^T s_k \geq c_2 \nabla f_k^T s_k$$

$$(\nabla f_{k+1}^T - \nabla f_k^T) s_k \geq (c_2 - 1) \nabla f_k^T s_k$$

$$\Rightarrow y_k s_k \geq (c_2 - 1) \nabla f_k^T s_k$$

Since $c_2 < 1$ and p_k is the descent direction, $\nabla f_k^T p_k < 0$,

$$y_k^T s_k \geq (c_2 - 1) \alpha_k \nabla f_k^T p_k > 0$$

Therefore, the curvature condition is satisfied when (Strong) Wolfe Condition is satisfied for Line Search method.

However, when curvature condition is satisfied, the secant equation has infinitely many solutions. (> 0)

Therefore, if we want to determine B_{k+1} uniquely, among all symmetric matrices satisfying the secant equation, B_{k+1} is, closest to the current matrix B_k:

$$\min ||B - B_k||$$

$$s.t. \quad B = B^T \ [9], \quad B s_k = y_k$$

To determine the solution, a useful norm is the **weighted Frobenius norm**

$$||A||_W = ||W^{0.5} A W^{0.5}||_F$$

Let $W = {G_k}^{-1}$ where G_k is the average Hessian:

$$G_k = \int_0^1 \nabla^2 f(x_k + r\alpha_k p_k) dr$$

Then we will have the DFP updating formula:

[9] Note that B_k should be both symmetric and postive definite

DFP Updating Rule

$$B_{k+1} = (I - \rho_k y_k s_k^T) B_k (I - \rho_k s_k y_k^T) + \rho_k y_k y_k^T$$

$$\rho_k = \frac{1}{y_k^T s_k}$$

4.4.2 Sherman-Morrison-Woodbury Formula

Suppose

$$\hat{A} = a + UV^T$$

Then $\hat{A}$ is nonsingular if $(i + V^T A^{-1} U)$ is nonsingular and

Sherman-Morrison-Woodbury Formula

$$\hat{A}^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1} U)^{-1} V^T A^{-1} \quad [a]$$

[a] This formula can be used to solve a linear systems of the form $Ax = d$ and $x = A^{-1}d$

Apply this formula on $B_k^{-1} = H_k$

4.4.3 BFGS Method

BFGS Update

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k}$$

B_k and H_k undergoes a rank-two update [a]. Therefore, instead of re-computing the hessian or inverse of hessian every iteration from scratch, we apply a simple modification that combines the most recently observed information about the objective function with the existing knowledge embedded in our current Hessian approximation (i.e. inverse of Hessian cost $O(n^3)$ while arithmetic operations cost only $O(n^2)$.

[a] $\Delta H_k = H_{k+1} - H_k$ is rank two

Similar to DFP method, in BFGS, H_k is directly approximated by solving:

$$\min ||H - H_k||$$

$$s.t. \;\; H = H^T, \;\; Hy_k = s_k$$

The norm is weighted Frobnius with $Wy_k = s_k$
If we let $W = G_k$, the BFGS update rule is obtained as:

BFGS Update

$$H_{k+1} = V_K^T H_k V_k + \rho_k s_k s_k^T$$

$$V_k = I - \rho_k y_k s_k^T, \quad \rho_k = \frac{1}{y_k^T s_k}$$

This requires fewer operations than DFPs

Note that if the matrix H_k incorrectly estimates the curvature in the object function, then the Hessian approximation will tend to correct itself within a few steps. However, the self-corrrecting properties of BFGS hold ONLY when an adequate line search is performed. (i.e. Wolfe condition). The performance of BFGS method can be degrade if the line search is not based on the Wolfe Conditions (i.e. using others like Armijo backtracking)

4.5 BFGS: Convergence

In general, we cannot say anything about the convergence property of BFGS. That is, we cannot prove that the iterates of these quasi-newton methods approach a stationary point of the problem from any starting point and any B_0.

Yet, there may be special cases where BFGS converges (under special condtions)

Theorem (Special Case Convergence of the BFGS Method)

Suppose:
(1) f is twice continuously differentiable.
(2) B_0 is any symmetric positive definite initial matrix
(3) x_0 is a starting point for which the sublevel set $L = \{x | f(x) \leq f(x_0)\}$ is convex.
(4) There exists $0 < m \leq M$ such that:

$$m||z||^2 \leq z^T \Gamma(x) z \leq M||z||^2$$

$$\forall z \in R^n, \quad x \in L, \quad \Gamma(x) = \nabla^2 f(x) \text{ Hessian } \ ^a$$

Then the sequence $\{x_k\}$ generated by the BFGS Method (with $\varepsilon = 0$) converges to the minimiser x^* of f.

[a] i.e. exists a neighbourhood where it is bounded (locally)

Theorem (Superlinear Convergence of the BFGS method)

Suppose:
(1) f is twice continuously differentiable.
(2) The iterates generated by the BFGS algorithm converge to a minimiser x^* at which $\Gamma(x)$ is locally Lipschitz.
(3) and

$$\sum_{k=1}^{\infty} ||x_k - x^*|| < \infty$$

Then x_k converges to x^* at a superlinear rate.

Summary of Converge:
In Newton Method, we can say convergence for the gradient
In BFGS Method, there is no guarantee of convergence for the gradient

Summary of Convergence Rate:
In Newton method: quadratic rate per iteration, complexity $O(n^3)$
In BFGS Method: superlinear rate per iteration, complexity $O(n^2)$
In Gradient Method: linear rate per iteration, complexity $O(n)$

Hence, trade off between speed and get answer per step time.
i.e. No. of steps vs. per step time complexity

4.6 Limited-Memory BFGS

The problem with BFGS: A large problem is problematic because Hessian matrices cannot be computed at a reasonable cost or are not sparse
A solution to this is **Limited-Memory BFGS**
The idea of L-BFGS is to use curvature information from only the most recent iterations to construct the Hessian approximation. Curvature information from earlier iterations is discarded in the interest of saving storage. That is, we look at the immediate past of set of computations. Here in L-BFGS, we no longer store the entire n × n H_k or H_{k+1} matrix at each time $t = i$, instead, based on the update formula, we store only n × 1 $\{s_i\}$ and $\{y_i\}$ vector (normally in pairs, $\{s_i, y_i\}$).
Additionally, we do not store every $\{s_i, y_i\}$ pair, instead, we store the latest m pairs [10]
For this method to make sense,

$$\text{elements to be stored } \frac{n^2+n}{2} > 2mn$$

[10] In practice, $m \in (3, 20)$, problem dependent. As m increases, L-BFGS → BFGS

$$\frac{n+1}{4} > m$$

BFGS Update

$$H_{k+1} = V_K^T H_k V_k + \rho_k s_k s_k^T$$

$$V_k = I - \rho_k y_k s_k^T, \quad \rho_k = \frac{1}{y_k^T s_k}$$

This requires fewer operations than DFPs

We still use the update rule, but each time we have to calculate H_{k+1} using the $\{s_i, y_i\}$ pairs from the very beginning (iteratively, we need $\{s_0, y_0\}$ up to $\{s_i, y_i\}$). Therefore, although we have less storage requirements, more calculations are involved.

$$H_1 = V_0^T H_0 V_0 + \rho_0 s_0 s_0^T$$

$$H_2 = V_1^T (V_0^T H_0 V_0 + \rho_0 s_0 s_0^T) V_1 + \rho_1 s_1 s_1$$

$$H_3 = ...$$

To calculate H_k, we need all information from $\{s_0, y_0\}$ up to $\{s_i, y_i\}$, however, we can use limited fix window $\{s_i, y_i\}$ pairs to approximate H_k.

5 Lecture 5: Least-Square Optimization

5.1 Lecture Outline

1. Least-square problems
2. Linear least-squares Problems
3. Statistical justification for Least-squares
4. Linear least-squares and regularisation
5. Nonlinear least-squares and Gauss-Newton Method

5.2 Least-Square Problems

The model is as follows:

Least Square Problem

$$\min_x f(x) = \frac{1}{2}\sum_{i=1}^{m} r_i^2(x)$$

$r_i : R^n \to R$ is smooth [a] and called **residuals**

Here the idea is that we want to minimize the error of what we expect to happen vs. what we actually observe.

[a] i.e. Differentiable infinitely

Note here a standing assumption is that $m \geq n$, that is there are more residuals than the unknown parameters of the system of equation.

In matrix form:

$$f(x) = \frac{1}{2}||r(x)||_2^2$$

where the residual vector $r(x) = [r_1(x)....r_m(x)]^T$

The derivative of matrix $r(x)$ a $m \times n$ matrix called **Jacobian matrix** $J(x)$:

$$J(x) = \begin{bmatrix} \nabla r_1(x)^T \\ \vdots \\ \nabla r_m(x)^T \end{bmatrix}$$

Consequently, the gradient and hessian of $f(x)$ is given by:

$$\nabla f(x) = \sum_{i=1}^{m} r_i(x)\nabla r_i(x) = J(x)^T r(x)$$

$$\nabla^2 f(x) = \sum_{i=1}^{m} \nabla r_i(x)\nabla r_i(x)^T + \sum_{i=1}^{m} r_i(x)\nabla^2 r_i(x)$$

$$\nabla^2 f(x) = J(x)^T J(x) + \sum_{i=1}^{m} r_i(x) \nabla^2 r_i(x)$$

Note the second term $\sum_{i=1}^{m} r_i(x) \nabla^2 r_i(x)$ does not contribute too much to the Hessian, in a linear least square,$\sum_{i=1}^{m} r_i(x) \nabla^2 r_i(x) = 0$, hence the **Jacobian matrix is a good approximation of Hessian**

5.3 Fixed-Regressor Model

Suppose we have some data points, and we propose a model with some unknown parameters that may potentially explain the data points well.
The goal is to estimate the parameters of a model, say, ϕ, for a system (some data points that generated by the system, and we can observe those true data). That is, we want the model to best fit the true data, i.e.

$$\phi(t; x) = y(t)$$

Here, t is the input (the usual x that we normally have), and the x are the unknown parameters.

Fixed-Regressor Model

The cost function / least square function:

$$f(x) = \sum_{i=1}^{m} ||\phi(t_i; x) - y(t_i)||^2$$

which can be interpreted as, the sum of the squared errors between the true that points and the predicted data point given an input t_i.
The goal is to minimize the cost function to get the best parameters that we can possibly get to fit our proposed model to the true data points.

5.3.1 Maximum Likelihood Estimate

Important assumption: assume $r_i(x) = \phi(t_i; x) - y(t_i)$ are independent and identically distributed (IID) with a certain variance σ_i^2 and probability density function PDF $g_i(.)$
The **likelihood** of observing a particular set of measurements y_i, given the unknown parameter x is actually:

$$P(y_1, ..., y_m|x) = \prod P(y_i|x) \quad ^{11} = \prod g_i(\phi(t_i; x) - y(t_i))$$

We want to maximize the likelihood of the model to fit with the true data, and find a set of parameters $x^* = argmax$ If the distribution of the residuals $r_i(x)$ is Gaussian, i.e.

$$g_i(r) = \frac{1}{\sqrt{2\pi\sigma_i^2}} exp(-\frac{r^2}{2\sigma_i^2})$$

[11] product of all likelihood, this operation requires IID

The likelihood function becomes

$$P(y_1, ..., y_m|x) = c\ exp(-\sum \frac{(\phi(t_i; x) - y(t_i))^2}{2\sigma_i^2})$$

where c is a constant that does not depend on x, $c = \prod_{i=1}^{m}(2\pi\sigma_i^2)^{-0.5}$ Therefore, we want to maximize the likelihood function (so that our model best fits the true data), we need to minimize the follow function:

$$f(x) = \frac{1}{2}\sum \frac{(\phi(t_i; x) - y(t_i))^2}{2\sigma_i^2}$$

Which is a weighted least square problem, it is a **measure of quality**.

Note the assumption of IID and Gaussian distribution is important, if we change the distribution function of the residuals, we will end up with a different optimization problem!

Another way of writing it is,

$$f(x) = \frac{1}{2}r(x)^T S^{-1} r(x) \quad [12]$$

$$S = diag(\sigma_1^2, ..., \sigma_m^2)$$

If $\sigma_1 = ... = \sigma_m$, the likelihood is maximised if the following function is minimised:

$$f(x) = \frac{1}{2}\sum(\phi(t; x) - y(t_i))^2 = \frac{1}{2}\sum r_i^2(x)$$

5.4 Linear Least Square Problem

So in many situations $\phi(t; x)$ is a linear function of x. Then the cost function becomes:

Linear Least Square Problem

$$f(x) = \frac{1}{2}||r(x)||^2 = \frac{1}{2}||Jx - y||^2$$

Gradient: $\nabla f(x) = J^T(Jx - y)$, Hessian: $\nabla^2 f(x) = J^T J$

Here, $f(x)$ is convex

Normal Equations

Any x^* that results $\nabla f(x^*) = 0$ is the global minimiser of $f(x)$ Let $\nabla f(x) = 0$

$$J^T Jx - J^T y = 0$$

$$J^T Jx = J^T y$$

[12] $x^T Mx$ this is a weighted Euclidean norm

Assume $m > n$[a] and hence J is full column rank [b]

Now, our object is to solve the system of equation:

$$J^T J x = J^T y$$

[a] No. of residuals > No. of unknown paramters
[b] Meaning that there is unique solution to the system

There are three ways to solve the system of equations:
(1) Cholesky Factorization
(2) QR Factorisation
(3) Singular Value Decomposition (SVD)

5.4.1 Cholesky-Based Algorithm

The most obvious approach is to solve $J^T J x^* = J^T y$. Let $A = J^T J$ and $b = J^T y$. A is square and symmetric matrix.
Steps:
(1) compute A and b
(2) compute the Cholesky factorization of the symmetric matrix A. It is possible for $m \geq n$ and Rank(J) = n:

$$J^T J = C^T C$$

where C is square upper triangular matrix.
(3) Solve the triangular systems to find x^*:

$$C^T \tau = b$$

$$C x = \tau$$

However, if J is ill conditioned (i.e. singularity problems), the Cholesky factorization process may break down.

5.4.2 QR Factorization

Introduce a orthogonal matrix $Q \in R^{m \times m}$, which have the following property:

$$Q Q^T = I$$

The orthogonal matrix will not change the length of x:

$$||Qx|| = \sqrt{x^T Q^T Q x} = \sqrt{x^T x} = ||x||$$

Given the above property, we perform the Q matrix onto the linear least square

$$||Jx - y|| = ||Q^T(Jx - y)||$$

$$||Jx - y||^2 = || \begin{bmatrix} Q_1^t \\ Q_2^T \end{bmatrix} (J\Pi\Pi^T x - y)||_2^2$$

=

$$|| \begin{bmatrix} R \\ 0 \end{bmatrix} (\Pi^T x) - \begin{bmatrix} Q_1^T y \\ Q_2^T y \end{bmatrix} ||_2^2$$

$$= ||R(\Pi^T x) - Q_1^T y||^2 + ||Q_2^T y||^2$$

$||Jx - y||$ is minimised by solving the following system of equation:

$$R\Pi^T x = Q_1^T y$$

This is more robust than Cholesky Factorization, but for greater robustness or more information about the sensitivity of the solution to perturbations in the data, SVD is used.

5.4.3 Singular Value Decomposition (SVD)

Any matrix, regardless of symmetric or not, can be written in the following way

$$J = U \begin{bmatrix} S \\ 0 \end{bmatrix} V^T = U_1 S V^T$$

$U = [U_1, U_2] \in R^{m \times m}$ is orthogonal, $S \in R^{n \times n} = \text{diag}(\sigma_1, ..., \sigma_n)$ where $\sigma_1 \geq .. \geq \sigma_n > 0$, $V \in R^{n \times n}$ So:

$$J^T J = U S V^T$$

$$J^T J = V S U_1^T U_1^T S V^T$$

$$= V S^2 V^T$$

Note, V are the eigenvectors of $J^T J$ and hence are called **singular-value matrix**, σ_i^2 are eigenvectors of $J^T J$ and are called **singular-values** Like in the QR factorization,

$$||Jx - y|| = ||U^T(Jx - y)||$$

$$||Jx - y||^2 = ||S(V^T x) - U_1^T y||^2 + ||U_2^T y||^2$$

The second norm is independent of x.
$||Jx - y||$ is minimised by minimising $||S(V^T x) - U_1^T y||^2$:

$$x^* = V S^{-1} U_1^T y$$

When J is rank deficient, some σ_i are exactly zero, then we can use the following method to solve the problem:

$$x^* = \sum_{i \in \{j|\sigma_j \neq 0\}} \frac{u_i^T y}{\sigma_i} v_1 + \sum_{i \in \{j|\sigma_j = 0\}} r_i v_i$$

Often the solution with the smallest norm is the most desirable, so we set $r_i = 0$

5.4.4 Linear Least-Squares Problem and Moore-Penrose Pseudoinverse

The problem we are solving:

$$J^T J x = J^T y$$

A silly way to solve it (if $J^T J$ is full column rank [13], or invertible):

$$x^* = (J^T J)^{-1} J^T y$$

$J^\dagger = (J^T J)^{-1} J^T$ is called the Moore-Penrose pseudoinverse

Definition (Moore-Penrose pseudoinverse)

Let $J = USV^T$ be the singular value decomposition (SVD) of $J \in R^{m \times n}$. The the Moore-Pensrose Pseudoinverse, $J^\dagger$, is

$$J^\dagger = VS^\dagger U^T$$

where $S^\dagger$ is obtained by replacing the nonzero entries of the diagonal of matrix S with their inverse and transposing it.

So we can use Moore-Penrose Pseudoinverse if $J^T J$ or JJ^T is invertible.
If $J^T J$ is invertible, $J^\dagger = (J^T J)^{-1} J^T$
If JJ^T is invertible, $J^\dagger = J^T (JJ^T)^{-1}$
What happens if neither of them are invertible? (Ill-posed square problems)

5.4.5 Linear Least-Squares Problem and Regularisation

The solution is to use regularisation method:

[13] A square matrix is full rank iff its determinant is nonzero

Tikhonov Regularisation

$$\min_x ||Jx - y||^2 + \frac{1}{\gamma}||\Gamma x||^2$$

where Γ is a proper scaling matrix (full-rank and often identity)
The additional term is like a penalty on the decision variable
And the derivative:

$$\nabla F(x) = J^T Jx + \frac{1}{\gamma}\Gamma^T \Gamma x - J^T y = 0$$

$$x^* = (J^T J + \frac{1}{\gamma}\Gamma^T \Gamma)^{-1} J^T y$$

Note that if $\gamma \to \infty$, $(J^T J + \frac{1}{\gamma}\Gamma^T \Gamma)^{-1} J^T \to J^\dagger$

5.5 Nolinear Least Square: Gauss - Newton Method

In Non-linear Least-squares, hessians are hard to compute.
The Gauss-Newton method is in essence a modified Newton method with line search. The idea is that instead of finding the search direction via

$$\nabla^2 f_k p = -\nabla f_k$$

We approximate $\nabla^2 f_k$ with $J_k^T J_k$ ($\nabla^2 f_k \approx J_k^T J_k$) and solve the following system of equation:

$$J_k^T J_k p_k^{GN} = -J_K^T r_k$$

That is, we do not have to compute Hessians of residuals, $\nabla^2 r_i$

Assume $\text{Rank}(J_k) = n$ and $\nabla f_k \neq 0$, p_k^{GN} is a descent direction [a], hence, suitable for line-search method.

$$(p_k^{GN})^T \nabla f_k = (p_k^{GN})^T J_k^T r_k$$
$$= -(p_k^{GN})^T J_k^T J_k p_k^{GN} = -||J_k p_k^{GN}||^2 \leq 0$$

The inequality is strict unless $J_k p_k^{GN} = 0$, in which case x_k is a stationary point.

$$J_k^T r_k = \nabla f_k = 0$$

Additionally, note that an optimal solution x^* for a linear least square problem must satisfy the following:

$$J^T J x^* = J^T y$$

Hence, p_k^{GN} is actually the solution to the following linear least-square problem:

$$\min_p \frac{1}{2}||J_k p + r_k||^2$$

Hence, all the methods we mentioend (QR, SVD) can be applied to find search direction, and there is no need to calculate the Hessian directly, instead, an approximation $J_k^T J_k$ can be used.

[a] Note that this is different from Newton method, here we can guarantee search direction

Implementations of the Gauss-Newton method usually perform a line search in the direction p_k^{GN}, requiring the step length to satisfy conditions like Armijo and Wolfe conditions.

Theorem (Convergence of Gauss-Newton Method)

Suppose:
(1) each residual function r_i is Lipschitz continously differentiable in a neighborhood of N of the bounded sublevel set $L = \{x | f(x) \leq f(x_0)\}$ where x_0 is the starting point for the algorithm,
(2) the Jacobians $J(x)$ satisfy the uniform full-rank condition on N, i.e. $\exists \gamma > 0$ such that $||J(x)z|| \geq \gamma ||z||$ [a], $\forall x \in X$
(3) If the iterates x_k are generated by the Gauss-Newton method with step length α_k that satisfy Wolfe conditions,
We have

$$\lim_{k \to \infty} J_k^T r_k = 0$$

Or gradient of r_k is going to converge to zero

[a] $\frac{||J(x)z||}{||z||} \geq \gamma$, or the norm should be bounded below by γ

6 Lecture 6: Derivative Free Method

6.1 Lecture Outline

1. Noise and Finite difference
2. Implicit filtering
3. coordinate descent
4. Nelder-Mead
5. Simulating gradients and gaussian gradient oracles

6.2 Noice and Finite difference

In many applications derivatives are not available. e.g.

1. the result of an experimental measurement
2. a stochastic simulation, with the underlying analytic form of f unknown (every time you ask a question, then a value $f(x)$ will come out but we don't know the underlying objective function $f(x)$
3. really hard to code or calculate the derivative of an objective function.

Methods to resolve the problem

1. Method 1: Automatic/Algorithmic differentiation
2. Method 2: Finite difference method (really the definition of the derivative but without the limit, i.e. we just take some Δ

However, a differential equation solver or some other complex numerical procedure that helps to calculate f can make some errors (noises). Therefore, the finite-difference estimates can be inaccurate when the objective function contains noise. i.e. there are some noise contained in the objective function (ideal function f is perturbed by some noise).

$$\hat{f}(x) = f(x) + \eta(x) \quad [14]$$

Methods to resolve the problem

The centered finite-difference approximation to the gradient of $\hat{f}$

$$g_\epsilon(x) = [\frac{\hat{f}(x + \epsilon e_i) - \hat{f}(x - \epsilon e_i)}{2\epsilon}]$$

where e_i is the i-th element standard basis. e.g. $e_1 = [1, 0, 0, ..., 0]^T$

[14] The value of $\eta(x)$ will differ at each evaluation, even at the same x and does not need to be x dependent

6.2.1 Analysis of Centered Finite-Difference Method

We want to analyse how far our approximation is away from the true derivative.

$$||g_\epsilon(x) - \nabla f(x)||$$

Elementwise:

$$|| \ [g_\epsilon(x)]_i - [\nabla f(x)]_i \ ||$$

For $[g_\epsilon(x)]_i$

$$[g_\epsilon(x)]_i = \frac{\hat{f}(x + \epsilon e_i) - \hat{f}(x - \epsilon e_i)}{2\epsilon}$$

Apply Taylor's expansion:

$$\hat{f}(x + \epsilon e_i) + \eta_1(x)$$

$$= f(x) + \epsilon e_i^T \nabla f(x) + \frac{1}{2}\epsilon^2 e_i^T \nabla^2 f(x + \epsilon t_1 e_i) e_i + \eta_1(x) \tag{1}$$

$$\hat{f}(x - \epsilon e_i) + \eta_2(x)$$

$$= f(x) - \epsilon e_i^T \nabla f(x) + \frac{1}{2}\epsilon^2 e_i^T \nabla^2 f(x - \epsilon t_2 e_i) e_i + \eta_2(x) \tag{2}$$

(1) - (2) yields:

$$[g_\epsilon(x)]_i$$

$$= \frac{2\epsilon \nabla f(x) - \frac{1}{2}\epsilon^2 e_i^T (\nabla^2 f(x + \epsilon t_1 e_i) - \nabla^2 f(x - \epsilon t_2 e_i)) e_i + \eta_1(x) - \eta_2(x)}{2\epsilon}$$

$$= \nabla f(x) - \frac{1}{4}\epsilon e_i^T (\nabla^2 f(+) - \nabla^2 f(-)) e_i + \frac{1}{2\epsilon}(\eta_1(x) - \eta_2(x))$$

Therefore:

$$|| \ [g_\epsilon(x)]_i - [\nabla f(x)]_i \ ||$$

$$= || - \frac{1}{4}\epsilon e_i^T (\nabla^2 f(+) - \nabla^2 f(-)) e_i + \frac{1}{2\epsilon}(\eta_1(x) - \eta_2(x))||$$

Triangle inequality:

$$\leq || - \frac{1}{4}\epsilon e_i^T (\nabla^2 f(+) - \nabla^2 f(-)) e_i|| + ||\frac{1}{2\epsilon}(\eta_1(x) - \eta_2(x))||$$

Here, to further analyse the norm of the deviation, we need to make two important assumptions:

Two Assumptions

Assumption 1: Assume the noise level $\bar{\eta}(x;\epsilon) = sup_{||z-x||\infty \leq \epsilon}|\eta(z)|$, that is, the noise has an upper bound.

Assumption 2: Let $\nabla^2 f(x)$ be Lipschitz continuous in a neighbourhood of $\{z|||z-x||_\infty \leq \varepsilon\}$

For the second norm, utilizing the first assumption:

$$||\frac{1}{2\epsilon}(\eta_1(x) - \eta_2(x))|| \leq ||\frac{2\max \eta_{1,2}(x)}{2\epsilon}|| \leq \frac{\bar{\eta}}{\epsilon}$$

For the first norm, we use Lipschitz condition $||f(x)-f(y)|| \leq L||x-y||$ and utilizing the second assumption:

$$||\nabla^2 f(+) - \nabla^2 f(-)|| \leq L||x + t_1\epsilon e_i - x + t_2\epsilon e_i|| \leq 2L\epsilon$$

Hence,

$$||g_\epsilon(x) - \nabla f(x)|| \leq L\epsilon^2 + \frac{\bar{\eta}(x;\epsilon)}{\epsilon}$$

Fundamental problem of finite-difference method

Note the importance of ϵ. We want ϵ to be small in the original finite-difference approximation $g_\epsilon(x)$ but from the above equation, second term will grow, then there is not even a guarantee that this approximation is in the descent direction. (This is the fundamental problem of finite-difference method)

Hence, we want methods that we can do better.

6.3 Implicit Filtering

Assumption: f is smooth (formally, infinitely differentiable, informally, first and second derivative are continuous)

This method really works if we can have a handle over the noise level. In other words, it is useful if we can make sure that the noise level in functional evaluation can be decreased as we run our algorithms. For example, running trials for multiple times and averaging, or setting higher accuracy level for an ODE or PDE solver (we have to let the solver to run longer and longer overtime for the errors to decrease).

It does not work if your function is generated by some simulation or stochastic functions.

7 Lecture 7: Theory of Constrained Optimization

7.1 Lecture Outline

1. Constrained Optimization Problems
2. Tangent Cone
3. Linearized Feasible Directions
4. Constraint Qualifications
5. First-Order Necessary Conditions - KKT Conditions
6. Second-Order Conditions

7.2 Constrained Optimization Problems

A standard constrained optimization problem:

Standard Constrained Optimization Problem

$$\min_{x \in R^n} f(x)$$

$$s.t. \;\; c_i(x) = 0, \quad i \in E, \quad c_i(x) \geq 0, \quad i \in I$$

(1) f and c_i are smooth (continuously differentiable) and real, E and I are finite sets of indices.
(2) f is the objective function, $c_i, \; i \in E$ are equality constraints, and $c_i, \quad i \in I$ are inequality constraints

Definition (Feasible Set)

Define the feasible set X to be the set of points x that satisfy the constraints,

$$X = \{x | c_i(x) = 0, \quad i \in E, \; c_j(x) \geq 0, \quad j \in I\}$$

Note that now the constrained problem can be modelled as:

$$\min_{x \in X} f(x)$$

Now we focus on two types of optimality conditions:
1. **Necessary conditions**: conditions that must be satisfied by any solution point (under certain assumptions).
2. **Sufficient conditions**: if satisfied at a certain point x^*, guarantee that x^* is in fact a solution.

Definition (Active Set)

The active set $A(x)$ at any feasible point [a] x consists of
(1) the equality constraints indices from E and
(2) the indices of the inequality constraints $i \in I$ for which $c_i(x) = 0$
or i.e.:

$$A = E \cup \{i \in I | c_(x) = 0\}$$

At a feasible point x, the inequality constraint $i \in I$ is called
(1) **active** if $c_i(x) = 0$
(2) **inactive** if the strict inequality $c_i(x) > 0$ is satisfied.

[a] A point x is feasible if $x \in X$

7.3 Tangent Cone and Constraint Qualification

Some notes before talking about
Tangent Cone, Linearized Feasible Direction and Constraint Qualification

The idea is that at some point x that is feasible, we want to move, from that point, a little bit s (vector) on some direction.
However, we cannot go anywhere (because constraints). Instead, there are two conditions that we want to satisfy:
(1) After moving some s, we want the end point to remain feasible (that is, satisfy all the constraints)
(2) Decrease the value of f (since a standard optimization problem requires to minimize f)
To satisfy condition (1):
We use Taylor expansion to approximate the end point constraint function $c_1(x + s)$ [a]

$$c_1(x + s) \approx c_1(x) + \nabla c_1(x)^T s = \nabla c_1(x)^T s$$

We want $c_1(x+s) = 0$ (i.e. remain feasible or satisfy the constraints), and since $c_1(x) = 0$ (by assumption, we start with a point that is feasible)
Therefore, the condition now becomes:

$$\nabla c_1(x)^T s = 0$$

[a] Using equality constraints as example

(continued)
To satisfy condition (2), we employ Taylor expansion again:

$$f(x+s) \approx f(x) + \nabla f(x)^T s$$

$$f(x+s) - f(x) \approx \nabla f(x)^T s$$

Since we want f value to decrease after moving, the second condition can now become:

$$\nabla f(x)^T s < 0$$

Therefore, in a short summary, if we let $d \approx s/||s||$ (direction, unit vector), the two conditions that we want to satisfy:

$$\nabla c_1(x)^T d = 0 \ \ ^a, \ \ \nabla f(x)^T d < 0$$

Be careful, if $\nabla f(x)$ and $\nabla c_1(x)$ are parallel, that there exist some λ such that $\nabla f(x) = \lambda c_1(x)$ satisfies for all x, then such direction d does not exist.

However, it is important to realize that here we use linear approximation (Taylor expansion) to form an approximate problem (both objective and constraints are approximated to be linear). This approximation makes sense ONLY WHEN the linearized approximation captures the essential geometric features of the feasible set near the point x in the question.
If near x, the linearization is fundamentally different from the feasible set (for instance, it is an entire plane, while the feasible set is a single point) then we CANNOT expect the linear approximation to yield useful information about the original problem.
Therefore, it is important to make assumptions about the nature of the feasible set, near x, and this is the constraint qualification, i.e. LICQ and Slater, etc.

[a] inequality constraint: $c_1(x)^T d \geq 0$

Given a feasible point $x \in X$, we call $\{z_k\}$ a **feasible sequence approaching** x if $z_k \in X$ for all k sufficiently large and $z_k \to x$.
A **tangent** is a limiting direction of a feasible sequence.

Definition (Tangent Cone)

The vector d is a tangent (or the tangent vector) to X at a point x if there are a feasible sequence $\{z_k\}$ approaching x and a sequence of positive scalars t_k with $t_k \to 0$ such that

$$\lim_{k\to\infty} \frac{z_k - x}{t_k} = d$$

The set of all tangents to X is called the **tangent cone** and is denoted by $T_X(x^*)$
One way to define $t_k = ||z_k - x||$

Definition (Linearized Feasible Directions (cone))

Given a feasible point x and the active constraint set $A(x)$, the set of linearised feasible directions $F(x)$ is:

$$F(x) = \{d | d^T \nabla c_i(x) = 0, \quad i \in E; \quad d^T \nabla c_j(x) \geq 0, \quad j \in A(x) \cap I\}$$

That is, if we move along those directions d, we remain feasible.

Note the difference:
(1) The definition of tangent cone does not rely on the algebraic specification of the set X, only on its geometry. For Tangent Cone, the idea is effectively using the first-order Taylor series expansion of the objective function and constraint functions about x to form an approximate problem in which both objective and constraints are linear.
(2) The linearised feasible direction set depends on the definition of the constraint function c_i.

7.4 Constraint Qualifications

Constraint qualifications are conditions under which the linearized feasible set $F(x)$ is similar to the tangent cone $T_X(x)$. And most constraint qualification ensure $F(x) = T_X(x)$.
These conditions ensure that the $F(x)$ captures the essential geometric features of the set X in the vicinity of x, as represented by $T_X(x)$
The following constraint qualification is the most often used in the design of algorithms:

Definition (LICQ)

Given the point x and an active set $A(x)$, if the set of active constraint gradients

$$\{\nabla c_i(x), \quad i \in A(x)\}$$

is linearly independent [a], then we say linear independence constraint qualification (LICQ) holds.

In general, if LICQ holds, none of the active constraint gradients can be zero

[a] Full column rank

Theorem

Let $\bar{x}$ be a feasible point. Then the following statements are true

(1) $T_X(\bar{x}) \subset F(\bar{x})$: Tangent cone is a subset of linearized feasible directions set

(2) If LICQ is satisfied at $\bar{x}$, then $T_X(\bar{x}) = F(\bar{x})$

7.5 First Order Optimality Conditions

Theorem (A fundamental Necessary Condition)

If x^* is a local minimiser, then $\nabla f(x^*)^T d \geq 0, \quad \forall d \in T_X(x^*)$

Theorem (A fundamental Necessary Condition)

If LICQ holds at x^*, and it is a local minimiser. then

$$\nabla f(x^*)^T d \geq 0, \quad \forall d \in F(x^*)$$

Note the converse of the above is not necessarily true.

The above two conditions are not tractable (not useful). Therefore, we use Farkars' Lemma to helps interpret the condition of the optimality condition in a tractable way.

Theorem (Farkas' Lemma - Theorem of Alternatives)

For any matrices $B \in R^{n \times m}$, $C \in R^{p \times n}$, and vector $g \in R^n$

We have either

(1) $g \in K = \{By + Cw | y \geq 0\}$ for some $y \in R^m$ and $w \in R^n$, or

(2) there exists $d \in R^n$ such that $g^T d < 0$, $B^T d \geq 0$, $C^T d = 0$

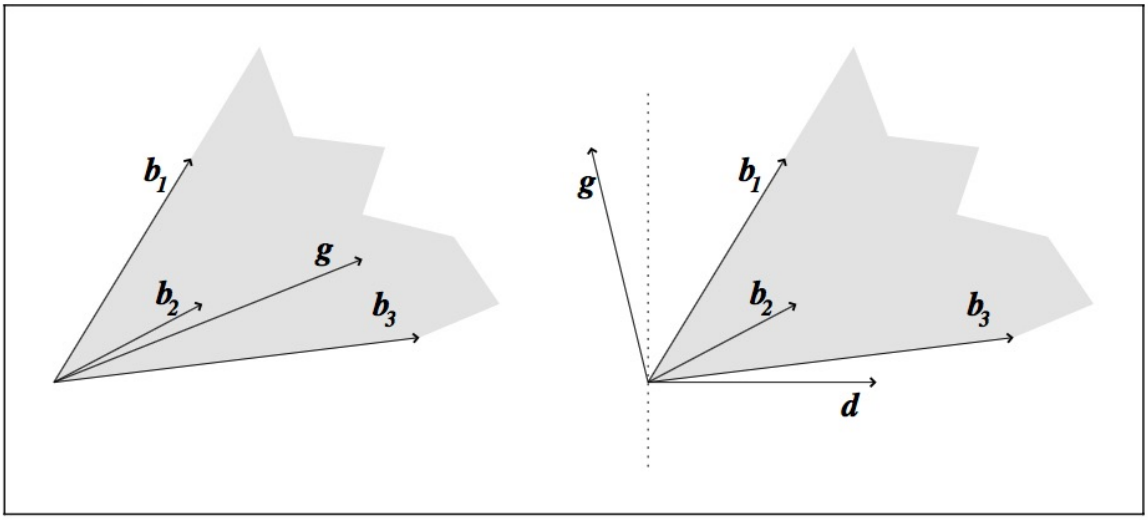

Figure 12.13 Farkas' Lemma: Either $g \in K$ (left) or there is a separating hyperplane (right).

7.6 First Order Necessary Condition

Definition (Lagrangian Function)

$$L(x, \lambda) = f(x) - \sum_{i \in E \cup I} \lambda_i c_i(x)$$

λ indicates how hard f is "pushing" or "pulling" the solution against the particular constraint $c_i(x)$. Or the cost that you want to push to constraint if you are at the boundary, you have to pay extra to push it further.

Theorem (First Order Necessary Condition - KKT Condition)

Suppose:
(1) x^* is local minimiser of function f with constraints c_i
(2) Function f and all constraints c_i are continuously differentiable
(3) LICQ holds at x^*
Then there is a Lagrange multiplier vector λ^*, with components λ_i^*, $i \in E \cup I$, such that the following conditions are satisfied at (x^*, λ^*)

$$\nabla_x L(x^*, \lambda^*) = 0$$

$$c_i(x^*) = 0, \forall i \in E$$

$$c_i(x^*) \geq 0, \ \forall i \in I, \ \lambda_i^* \geq 0, \ \forall i \in I$$

Complementary Slackness/Condition: $\lambda_i^* c_i(x^*) = 0, \quad i \in E \cap I$

Note that for a local minimiser x^*, there may be many vectors λ^* for which the KKT conditions are satisfied. When the LICQ holds, however, the optimal λ^* is unique.

Definition (Complementary Conditions)

The last condition are complementary conditions. They imply that either constraint i is active or $\lambda_i^* = 0$, or possibly both.

If $c_i(x)$ is inequality constraint. Then:
(1) $c_i(x) = 0$, is active $\Rightarrow$ corresponding λ can be anything
(2) $c_i(x) > 0$, is inactive $\Rightarrow$ corresponding λ must be zero
The Lagrange multipliers corresponding to inactive inequality constraints are zero. Therefore, the first KKT condition can be re-write as follows:

$$\nabla_x L(x^*, \lambda^*) = \nabla f(x^*) - \sum_{i \in A(x^*)} \lambda_i^* \nabla c_i(x^*) = 0$$

7.7 Second-Order Conditions

7.7.1 Motivations

1. If $d^T \nabla f(x^*) = 0$ for $d \in F(x^*)$, from the first derivative information alone, it is not known whether a move along this direction will increase or decrease the objective function f.
2. Therefore, the Second-Order conditions examine the second derivative terms in the Taylor series expansions of f and c_i, to see whether this extra information resolves the issue of increase or decrease in f.

Definition (Critical Cone)

Given $F(x^*)$ and some Lagrange multiplier vector λ^* satisfying the KKT conditions, we define the critical cone $C(x^*, \lambda^*)$ as follows:

$$C = \{d \in F(x^*) | d^T \nabla c_i(x^*) = 0,\ \forall i \in A(x^*)\ with\ \lambda_i^* > 0\}$$

The critical cone contains those directions d that "adhere" to the active inequality constraints even when we were to make small changes to the objective and equality, as well as to the equality constraints.
It is the subset of the tangent space, where the objective function does not vary to first order

Apply Taylor's expansion on Lagrange function:

$$L[(x^*, \lambda^*) + d] \approx L(x^*, \lambda^*) + d^T \nabla L + \frac{1}{2} d^T \nabla^2 L d$$

Theorem (Second-Order Necessary Conditions)

Suppose:
(1) x^* is a local minimiser
(2) LICQ satisfies (i.e. $\nabla c_i(x)$ active constraints are linearly independent)
(3) λ^* is the Lagrange multiplier vector for which KKT conditions are satisfied.

$$d^T \nabla^2_{xx} L(x^*, \lambda^*) d \geq 0, \forall d \in C(x^*, \lambda^*)$$

Theorem (Second-Order Sufficient Conditions)

Suppose:
(1) $\bar{x}$ is a feasible point
(2) $\bar{\lambda}$ is the Lagrange multiplier vector for which KKT conditions are satisfied (3) and

$$d^T \nabla^2_{xx} L(x^*, \lambda^*) d > 0, \forall \text{ non-zero } d \in C(x^*, \lambda^*)$$

Then $\bar{x}$ is a strict local minimiser. Note only those points that lie in the critical cone. Sufficient conditions are conditions on f and $c_i, i \in E \cup I$, that ensures that x^* is the local solution of the problem.
Constraint qualification is not required
$\Leftrightarrow$ Check if Hessian is positive definite.

7.8 Other Constraint Qualification

Theorem

Suppose that at some $x^* \in X$, all active constraints $c_i,\ i \in A(x^*)$, are linear. Then $F(x^*) = T_X(x^*)$

Note: it is neither weaker nor stronger than the LICQ condition; there are situations in which one condition is satisfied but not the other.

Definition (Slater's Condition)

If:
(1) $c_i, \forall i \in E$ is linear,
(2) $-c_i, \forall i \in I$ is convex,
(3) there exists $\bar{x} \in X$ such that $c_i(\bar{x}) = 0,\ \forall i \in E,\ c_i(\bar{x}) > 0,\ \forall i \in I$
Then Slater's condition holds.

Theorem

If Slater's condition holds, then $F(\bar{x}) = T_X(\bar{x}),\ \bar{x} \in X$

One cannot compare LICQ with Slater's or other constraint qualifications.

8 Lecture 8: Lagrangian and Duality

8.1 Lecture Outline

1. The Lagrangian and Duality
2. Perturbation Analysis of Constrained Problems
3. Dual Decomposition of a Resource Allocation Problem
4. sub-gradient Methods for Dual Optimisation
5. Termination Conditions Based on the Duality Gap
6. Distributed Optimization Via Dual Decomposition and Subgradient Methods

8.2 The Lagrangian and Duality

Recall a standard optimization problem:

$$\min_x \ f(x)$$

$$s.t. \ \ c_i(x) = 0, \ i \in E, \quad c_i(x) \geq 0, \ i \in I$$

Let's call this the **primal problem** and denote p^* as the the primal solution to this problem. The Lagrange Function:

$$L(x, \lambda) = f(x) - \sum_{i \in E \cup I} \lambda_i c_i(x)$$

Lower Bound Property of Lagrangian

For any feasible point of the primal problem $\bar{x}$ and $\bar{\lambda}_i$, $i \in E \cup I$:

$$L(\bar{x}, \bar{\lambda}) = f(\bar{x}) - \sum_{i \in E} \lambda_i c_i(x) - \sum_{i \in I} \lambda_i c_i(\bar{x})$$

From feasibility (constraints):

$$c_i(\bar{x}) = 0, \ i \in E$$

$$c_i(\bar{x}) \geq, \ i \in I$$

and by definition:

$$\lambda_i \geq 0$$

Hence,

$$L(\bar{x}, \bar{\lambda}) \leq f(\bar{x})$$

Definition (Lagrange Dual Function)

$$q(\lambda) = \inf_x L(x, \lambda) \quad ^a$$

If $q(\lambda) = -\infty$, then we call variable λ dual infeasible.

[a] $\Leftrightarrow \min_x L(x, \lambda)$

Lower Bound Property of the Dual Function

$$q(\lambda) = \inf_x L(x, \lambda) \leq L(x, \lambda) \leq f(x^*) = p^*$$

$$q(\lambda) \leq p^*$$

Concavity of Dual Function

Lagrangian Function is a affine in $\lambda \rightarrow$ convex.
Through operations that retain convexity, we know that $\sup_x -L(x, \lambda) = -q(\lambda)$ is convex.

$$q(\lambda) \text{ is concave}$$

It is always concave regardless of the concavity property of the primal function.

Here we have very nice properties from the dual function. Since q is concave, $-q$ is always convex regardless the convexity of the primal function. We can always get an answer as long as q^* is not negative infinity.

Dual problem:

$$\min q(\lambda)$$

$$s.t. \quad \lambda_i \geq 0, \quad i \in I$$

Theorem (Weak Duality)

It always holds that $q^* \leq p^*$
Duality Gap is always non-negative

But this is not pretty helpful for us to solve the primal function.
Instead, we want the $q^* = p^*$ so that we can solve the dual problem and get the solution for the primal automatically.
The equality requires **Strong Duality**

Definition (Slater's Condition)

If:
(1) $c_i, \forall i \in E$ is linear,
(2) $-c_i, \forall i \in I$ is convex,
(3) there exists $\bar{x} \in X$ such that $c_i(\bar{x}) = 0,\ \forall i \in E,\ c_i(\bar{x}) > 0,\ \forall i \in I$
Then Slater's condition holds.

Slater condition is satisfied for all feasible LP and convex QP problems

Theorem (Strong Duality(Sufficient Condition))

If:
(1) the primal optimization problem is convex
(2) Slater's condition holds
Then the primal and dual optimal values are equal, $p^* = q^*$.

For Strong Duality to hold: we need $f(x)$ to be convex, $c_i = 0$ to be linear, $c_i \geq 0$ to be concave.

Theorem

If the problem is convex and the strong duality holds, then x^* is the primal optimal and λ^* is the dual optimal iff KKT conditions are satisfied. (No critical cone condition any more).
Note the KKT condition become very powerful when we deal with convex problem, we do not need to do all those critical cone analysis, etc.

8.3 Perturbation Analysis of Constrained Problems

Sometimes we are uncertain about the constraints, so we may want to perturb them.
Consider the perturbed primal problem:

$$\min_x f(x)$$

$$s.t.\ \ c_i(x) = \delta_i,\ i \in E,\ \ c_i(x) \geq \delta_i,\ i \in I$$

Parameter δ_i capture the inaccuracy of satisfying the constraints (Perturbation in constraints).
The dual problem becomes:

$$\min_x q(\lambda)\ ^{15} + \sum_{i \in E \cup I} \delta_i \lambda_i$$

$$s.t.\ \ \lambda_i \geq \delta_i,\ \ i \in I$$

[15] Dual of the unperturbed problem

Let x^* and λ^* satisfies the KKT condition $\rightarrow$ primal and dual optimal in unperturbed system. From the weak and strong duality of the original problem:

$$p^*(\delta) \geq q(\lambda^*) - \sum_{i \in E \cup I} \delta_i \lambda_i^* \Rightarrow$$

$$p^*(\delta) \geq p^* - \sum_{i \in E \cup I} \delta_i \lambda_i^* \quad ^{16}$$

For small perturbation values we have: (it comes from the above relationship and continuity of the primal value)

$$\frac{\partial p^*(\delta)}{\partial \delta} = -\lambda^*$$

If $\lambda^* = 0$ then perturbation will not have any impact

8.4 Sub-gradient Methods for Dual Optimization

Let's consider the dual problem:

$$\max_{\lambda} q(\lambda)$$
$$s.t. \quad \lambda_i \geq 0, \quad i \in I$$

This is the same as $\min_\lambda -q(\lambda)$

Definition (Subgradient and Subdifferential)

Given a convex function f we say a vector g is a subgradient of f at point x if:

$$f(z) \geq f(x) + g^T(z - x), \quad \forall z \in R^n$$

The set of all subgradients of f at x is called the subdifferential of f at x and is denoted by $\partial f(x)$.

Note the form is very similar to gradient (BUT NOT gradient)

$$\frac{f(z) - f(x)}{z - x} = g^T$$

[16] $q^* \leq p^*$

Note:
(1) g is not unique
(2) if f is differential (and convex), we get the definition of differentiable convex function $\Rightarrow$ g is unique (i.e. $g = \nabla f(x)$, and $\partial f(x) = \{\nabla f(x)\}$), which means that f gradient at x is a subgradient.
(3) A subgradient can exist even when f is not differentiable at x.

(4) g is a subgradient of f, iff the hyperplane that passes $[x, f(x)]$ with normal $[-g, 1]$ supports the epigraph of f [a]
(5) $\partial f(x)$ is non-empty, convex and compact.

[a] A function is convex iff the region above its graph is a convex set, and this region is called the function's epigraph

Definition (Subgradient Optimality Condition)

A vector x minimises a convex function f over a convex set X iff there exists a subgradient $g \in \partial f(x)$ such that:

$$g^T(z - x) \geq 0, \ \forall x \in X$$

Theorem (Subgradient Calculation in Dual Problem)

Let $q(\lambda)$ be the dual function associated with a standard constrained problem.
Let $x_\lambda \in argmin_x[f(x) + \lambda^T c(x)]$ for some λ.
Then $c(x_\lambda) \in \partial[-q(\lambda)]$, i.e.

$$q(v) \leq q(\lambda) - (v - \lambda)^T c(x_\lambda), \quad \forall v$$

If at x, it is not differentiable, then subgradient is a direction of move along so that you can minimise your cost function.

So the update rule for λ can now become:

$$\lambda_{k+1} = \lambda_k - \alpha_k c_i(x_{k+1}), \quad \forall i \in E$$

$$\lambda_{k+1} = P(\lambda_k - \alpha_k c_i(x_{k+1})) \ [17], \quad \forall i \in I$$

Subgradient Boundedness Assumption

Assume that, for some scalar L, $\sup\{||c(x_k)|| \ | k \geq 0\} \leq L$

How do we choose step size α_k?:

1. Constant

[17] P projects the elements of its argument onto the positive orthant

2. Selected from a prior known diminishing sequence
3. Adaptive

Theorem (Convergence within a Neighbourhood for Constant Step-length)

Let Subgradient Boundedness Assumption hold,
Let $\alpha_k = \alpha$ and q^* be bounded where p^* is the dual optimal value.
Then under subgradient boundedness assumption, for $k \to \infty$

$$q(\lambda_k) \geq q^* - \frac{\alpha L}{2}$$

Theorem (Convergence for Diminishing Step-Length)

Let Subgradient Boundedness Assumption hold, and α_k satisfy

$$\lim_{k\to\infty} \alpha_k = 0, \sum_{k=1}^{\infty} \alpha_k = \infty$$

then $q(\lambda_k) = q^*$.
Moreover, if

$$\sum_{k=0}^{\infty} < \infty$$

then λ_k converges to λ^*

8.5 Termintion Conditions Based on the Duality Gap

1. This will only work if we have **strong duality condition**
2. Let λ to be dual feasible (all λ_k generated by the projected subgradient method are dual feasible): $p^* \geq q(\lambda)$, $-p^* \leq -q(\lambda)$
3. If x_k is primal feasible:

$$f(x_k) - \ p^* \leq f(x_k) - q(\lambda_k)$$

This means that how far away I am from the primal points is bounded above by duality gap (value of primal function - value of dual function)
4. Here we propose a method / a possible stopping criteria that can ensure the duality gap to be below a tolerance τ. 5. Method: The following condition guarantees that $\frac{f(x_k - p^*)}{|p^*} \leq \varepsilon$

$$\begin{cases} \frac{f(x_k)-q(\lambda_k)}{q(\lambda} \leq \varepsilon, & \text{if } q(\lambda) > 0 \\ \frac{f(x_k)-q(\lambda_k)}{-q(\lambda} \leq \varepsilon, & \text{if } f(x_k) > 0 \end{cases}$$

Therefore, even though we don't know p^*, we can still guarantee the gap is within some distance (so that we get convergence and terminates the algorithm)

9 Lecture 9: Gradient Projection Algorithm

10 Subject Outline

1. Gradient Projection Algorithm
2. Constant step length, varying step length, diminishing step length
3. Complexity issues
4. Gradient projection with exploration
5. Projection
6. Solving Quadratic Problems: active set method and ADMM
7. Approximating the constraint set.

10.1 Gradient Projection Algorithm

The problem:

$$\min_x \ f(x), \ s.t., \quad x \in X$$

Assumption: f is continuously differentiable, X is closed and convex.
The following iterative method:

$$x_{k+1} = P_X(x_k - \alpha \nabla f(x_k)))$$

Definition (Projection Arc)

The projection arc is the set of all possible next iterates parameterised by α

We want to show that unless $x_k(\alpha) = x$ (which is a condition for optimality of x_k), the vector $x_k(\alpha) - x_k$ is a feasible descent direction

Theorem (Projection Theorem)

(1) Let X be a nonempty closed convex subset of R^n
(2) There exists a unique vector that minimises $||z - x||$ over $x \in X$ called the projection of z on X.
(3) Furthermore, x^* is the projection of z on X iff:

$$(z - x^*)^T(x - x^*) \leq 0, \quad \forall x \in X$$

Proof:
The problem that describes the hypothesis of this theorem:

$$\min_x \ ||z - x||$$
$$s.t. \ \ x \in X$$

which is equivalent to the following problem:

$$\min_x \ \frac{1}{2}||z-x||^2$$

$$s.t. \quad x \in X$$

This is a convex optimization problem. They have the same minimiser (use the first order optimality condition for convex problems to prove)
Take Grad of the norm $||z-x||$ yields us:

$$||z-x|| = \sqrt{\sum (z_i - x_i)^2}$$

$$\nabla||z-x|| = \frac{1}{2}(\sqrt[-\frac{1}{2}]{\sum (z_i-x_i)^2}) \sum 2(z_i-x_i)(-1)$$

$$\nabla||z-x|| = \frac{-z+x}{||z-x||}$$

If z belongs to x, then $z = x^*$ and the equality holds, if z does not belong to x, then

$$(x^*-z)^T(x-x^*) \geq 0 \text{ or } (z-x^*)^T(x-x^*) \leq 0$$

Theorem (Descent Properties of Gradient Projection)

(i) If $x_k(\alpha) - x_k$ is a feasible descent direction [a] and particularly

$$\nabla f(x_k)^T(x_k(\alpha)-x_k) \leq -\frac{1}{\alpha}||x_k(\alpha)-x_k||^2, \quad \forall \alpha > 0$$

(ii) if $x_k(\alpha) = x_k$ for some $\alpha > 0$ then x_k satisfies the necessary condition for minimising $f(x)$ over X (Convex), i.e.

$$\nabla f(x_k)^T(x-x_k) \geq 0, \quad \forall x \in X$$

either (i) we move along the descent direction or (ii) we have the first order necessary and sufficient optimality condition

[a] i.e. the p_k and the descent direction condition is $\nabla f(x)^T p_k < 0$ because $p_k = -B_k^{-1}\nabla f_k$

An important assumption for constant step-size convergence: Lipschitz continuity of the gradient. i.e. choose α_k so that:

$$||\nabla f(x) - \nabla f(y)|| \leq L||x-y||, \quad \forall x, y \in X$$

And this will result in an important inequality:

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2}||y - x||^2, \quad \forall x, y \in X$$

Theorem (Constant Step Length)

Assume:
(1) gradient is Lipschitz continuous
(2) $\alpha_k = \alpha$, $\alpha \in (0, \frac{2}{L})$ (constant value α)
Then every limit point $\bar{x}$ of the generated sequence $\{x_k\}$ satisfies the necessary optimality condition.

$$\nabla f(\bar{x})^T (x - \bar{x}) \geq 0, \quad \forall x \in X$$

The cost function will reduce, or the cost function is monotonically decreasing, the next value of the cost function is less. (All we need is $\frac{1}{\alpha} - \frac{1}{2} > 0$)

Theorem (Convergence for Convex Cost Function)

Let $\alpha_k \downarrow \bar{\alpha}$ [a] is selected via any step length rule and for all k

$$f(x_{k+1}) \leq f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \frac{1}{2\alpha_k}||x_{k+1} - x_k||^2$$

Then $\{x_k\}$ converges to optimal x^* and

$$f(x_k) - f^* \leq \frac{\min_{x^* \in X^*} ||x_0 - x^*||^2}{2k\bar{\alpha}}, \quad k \geq 0$$ [b]

This is sub-linear convergence
A linear convergence:

$$\lim_{k \to \infty} \frac{x_{k+1} - x^*}{x_k - x^*} \leq c < 1$$

Here it is sub-linear

[a] Choose some large α_k at the beginning and then converge to $\bar{\alpha}$, note that α_k should $\in (0, \frac{2}{L})$
[b] constant value

Theorem (Convergence for Strongly Convex Cost Function)

Let:
(1) $\alpha \in (0, \frac{2}{L})$
(2) f is strongly convex with modulus σ [a]
Then,

$$||x_{k+1} - x^*|| \leq \max(|1 - \alpha L|, |1 - \alpha\sigma|)||x_k - x^*||$$

The bound is minimised if $\alpha = \frac{2}{\sigma + L}$
L/σ is the condition number [b] of the problem ($L \geq \sigma$)
This gives us the convergence rate, it can be shown that it is less than 1

[a] $\sigma \leq ||\nabla^2 f|| \leq L$
[b] we call this the condition number because it bounds the smallest eigenvalue of the hessian and the largest eigenvalue of hessian, if this ratio increases, α is smaller, and we have worse convergence behavior

For $\max(|1 - \alpha L|, |1 - \alpha\sigma|)$, we have max (convex, convex) = convex

10.2 Gradient Projection with Exploration: Heavy Ball Method

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1})$$

Heavy ball takes advantage of memory (x_{k-1}) to improve the performance. Adding more memory is not necessarily useful.
Now we have another method (better than constant α method if we can find convergence, but no guarantee on global convergence, only local convergence)
The iteration becomes ($x_{-1} = x_0, \beta_k \in (0, 1)$)

$$y_k = x_k + \beta_k(x_k - x_{k-1}), \text{ exploration step}$$

$$x_{k+1} = P_X(y_k - \alpha \nabla f(x_k)), \text{ gradient projection step}$$

where

$$\beta_k = \frac{\theta_k(1 - \theta_{k-1})}{\theta_{k-1}}$$

$\theta_0 = \theta_1 \in (0, 1]$, $\beta_k \in (0, 1)$)

10.3 Gradient Projection: What Projection?

Recall $P_X(x) = \xi$ where

$$\xi \in argmin_z ||x - z||, \ s.t. \ \ z \in X$$

Really, the projection is finding the minimizer of the above optimization problem.
So, do we need to solve another optimization problem for each iteration of an optimization problem? (Yes, and that is why the implementation of the Gradient Projection method becomes very limited in practice, it only works with a family of constraints). We introduce some constraints:

Box Constraints

A simple **Box constraints**:

$$X = \{x | l \leq x \leq u\}$$

Hence the projection is:

$$xi_i \begin{cases} l_i, & x_i < l_i \\ x_i, \ _i \leq x_i \leq u_i \\ u_i, & u_i \leq x_i \end{cases}$$

Linear Subspace (linear constraints)

$$P_X(x) = \xi \text{ where}$$

$$\xi \in argmin_z ||x - z||, \quad s.t. \ \ z \in X = \{x | Ax = b\}$$

$$\xi = (I - A^T(AA^T)^{-1}A)x + A^T(AA^T)^{-1}b$$

Constraints Set Defined by Inequalities

$P_X(x) = \xi$ where

$$\xi \in argmin_z ||x - z||, \quad s.t. \ \ z \in X = \{x | c_i(x) \geq, \ i \in I\}$$

c_i is concave and I is the inequality constraints index set So effectively, this is solving a quadratic optimization problem with inequality constraints, it is as hard as the original problem (at each step)

Consider the case where the constraints are linear (i.e. QP with linear constraints)

$$\min_x \frac{1}{2} x^T Q x + q^T x$$

$$s.t. \ \ a_i^t x \geq b_i, \quad i \in I$$

Two approaches to solving QPs will be considered:
(1). Primal Active Set method
(2). Alternating Direction Method of Multipliers (ADMM)

10.3.1 Primal Active-Set

Idea: some of the inequality constraints (and all equality constraints), i.e. active sets, are imposed as equalities.
This subset of the above equality constraints is referred to as the working set, W_k
It is required that the constraints in the working set W_k be linearly independent (LICQ).

10.3.2 Alternating Direction Method of Multipliers (ADMM)

Consider the following problem:

$$\min_{x,z} f_1(x) + f_2(x)$$

$$s.t. \;\; Ax = z$$

Augmented Lagrangian:

$$L_A(x, z, \lambda; \mu) = f_1(x) + f_2(z) - \lambda^T(Ax - z) + \frac{\mu}{2}||Ax - z||^2 \;\; ^{18}$$

The iterations become:

$$x_{k+1} \in argmin_x L_A(x, z_k, \lambda_k; \mu)$$

$$z_{k+1} \in argmin_z L_A(x_{k+1}, z, \lambda_k; \mu)$$

$$\lambda_{k+1} = \lambda_k + \mu(Ax_{k+1} - z_{k+1})$$

A very similar update rule that we have in the dual optimization problems.
The method converges for convex f_1 and f_2 for ANY $\mu > 0$, so it is powerful because you don't need to worry about the step size at all, you don't need the f to be strongly convex.
The algorithm converges R-linearly to the solution for $Q > 0$: projection, $Q = I, q = 0$ so the algorithm becomes easier when you implement it.
We can come up with an **optimum step size**:

$$\mu^* = (\sqrt{\lambda_{\max}(AQ^{-1}A^T)\lambda_{\min}(AQ^{-1}A^T)})^{-1}$$

> The above two methods only make sense if the complexity of solving the original problem is much larger bigger than solving the inner quadratic problem.

10.4 Projection: Approximating constraint sets X

We know that projection onto boxes is easy. So why not approximate constraints X with a box?
Consider a ball centered at 0 and has a radius R, the projection of a point z onto the ball is:

$$\begin{cases} \frac{z}{||z||}R, & \text{if point is outside the ball, } ||z|| > R \\ z, & \text{if the point is inside or on the ball, } ||z|| \leq R \end{cases}$$

A valid question to ask is, if I have constraints, can I approximate it with a norm constraint?
Idea is I want a norm ball that fits inside the constraint set that has the maximum gradient. In general, a box is a ∞-norm ball centred at x_C and with radius R:

$$B(x_c, R) = \{x | ||x - x_c||_\infty \leq R\}$$

[18] if the constraints satisfied, then the last term is zero

Let $X = \{x | a_i^T x \leq b_i, \quad i \in I\}$, and the goal is to find the largest ball (square box) in X (we want to find the centre and the corresponding length of the edge of the box that is the maximum that we can have within a constraint set).
To form a problem of finding the largest ball (x_c is called the Chebyshev centre):

$$\max_{x_c, R} R$$

$$s.t. \;\; c_i(x_c, R) \leq 0, \quad i \in I, \quad R \geq 0$$

$$c_i(x_c, R) = \sup_{||u|| \leq 1} a_i^T (x + Ru) - b_i = a_i^T x + R(\sup_{||u|| \leq 1} a_i^T u) - b_i$$

$$= a_i^T x + R||a_i||_* - b_i$$

$$(||a_i||_{\inf *} = ||a_i||_1)$$

We want $\sup_{||u|| \leq 1} a_i^T u$, the term is equivalent to:

$$\sum_{i=1}^{n} sign(a_i) a_i = |a_i|$$

the sum of absolute value of a vector is norm 1. So the constraint $c_i(x_c, R) \leq 0$ is a very easy constraint to write.

11 Lecture 10: Quadratic Penalty Method and Regularization

11.1 Lecture Outline

1. Quadratic penalty method (useless but introductory)
2. Nonsmooth exact penalty methods (useful)
3. Method of multipliers or augmented Lagrangian method (a family of algorithms)

11.2 Penalty and Augmented Lagrangian Methods

One of the ways of solving constrained optimization problems, at least approximately, is by adding a penalty function to the objective function that depends - in some logical way - on the value of the constraints.
The idea is to minimize a sequence of unconstrained minimization problems where the infeasibility of the constraints is minimised together with the objective function.
There are two main types of penalization methods: *exterior penalty functions*, which impose a penalty for violation of constraints, and **interior penalty functions**, which impose a penalty for approaching the boundary of an inequality constraint.
Idea of Penalty Function methods: take away the constraints and add them to the function in a way that when we don't satisfy the constraints.

11.3 Quadratic Penalty Method

$$Q(x;\mu) = f(x) + \frac{\mu}{2}\sum_{i\in E} c_i^2(x)$$

where $\mu > 0$ is the penalty parameter. how big μ is depends on whether we care about the violations of the constraints.
So let u_k increase as $k \to \infty$. At each step the approximate minimiser x_k of $Q(x;\mu_k)$ is found.
Some problems associate with this method:
1. Termination condition $||\nabla_x Q(x;\mu_k)|| \leq \tau_k$ (tolerance level) may not be satisfied because the iterates may move away from the feasible region when the penalty parameter is not large enough.
2. As μ_k becomes large the Hessian $\nabla^2 Q(x;\mu_k)$ becomes ill-conditioned near the minimizer. As a result, many algorithms that requires Hessian or Hessian approximation can have poorly performance. e.g. Quasi-Newton algorithms. Newton method is not directly react to the problem but there are other problems. (A way to resolve this is to decrease the grow rate of μ_k)

Theorem

suppose $Q(x;\mu_k)$ has a (finite) minimiser for each value μ_k and each x_k is its exact global minimiser, and that $\{\mu_k\}$ increases to ∞. Then every limit point x^* of the sequence $\{x_k\}$ is a global solution of the original problem.

Therefore, if you have $f(x)$ convex, $c_i(x)$ linear or convex $\Rightarrow c_i(x)^2$ convex, then this is a way to get away from the constraints.

12 Lecture 11: Barrier Transformation and Interior Point Method

12.1 Lecture Outline

1. Indicator Functions and Barrier transformation
2. Optimality Conditions for the Barrier Transformed Problem
3. Primal Interior Point Method (IPM)
4. A primal-dual reformulation of the optimality conditions
5. IPM for linear programmes
6. IPM for general convex problems

12.2 Indicator Functions and Barrier Transformation

Exterior penalty methods generate infeasible points and are therefore not suitable when feasibility has to be strictly maintained. This might be the case if the objective function is undefined or ill-defined outside the feasible region.
The interior method is analogous to the external penalty method: it creates the sequence of unconstrained modified differentiable functions whose unconstrained minima converge to the optimum solution of the constrained problem in the limit.
A standard optimization problem:

$$\min_x f(x)$$

$$s.t. \;\; c_i(x) \geq 0, \;\; i \in I, \;\; a_i^T x - b_i = 0, \;\; i \in E \;\; [19]$$

Assumption:
(1) f is convex, c_i are concave and all are twice differentiable
(2) $I \cap E = \emptyset$ and $I \cup E = \{1, ..., m\}$
(3) Optimal value exists and can be attained (KKT)
(4) Feasible set has an interior, that is, the Slaters' condition are satisfied $\Rightarrow$ The Strong duality holds.

We can reformulate the problem via indicator functions:

$$\min_x f(x) + \sum_{i \in I} \| \, (c_i(x))$$

$$s.t. \;\; a_i^T - b_i = 0, \;\; i \in E$$

where:

$$\| \, (c_i(x)) = \begin{cases} 0, & x \geq 0 \\ \infty, & otherwise \end{cases}$$

[19] Linear equaity constraints

This indicates an extreme penalty and the effect is a perturbed problem where an infinite cost is incurred as $c_i(x) \to 0$
A *canonical* choice of indicator function (smooth) is a logarithmic function.

12.3 Barrier Transformation

$$\min_x f(x) - \mu \sum_{i \in I} \log(c_i(x))$$

$$s.t. \;\; a_i^T - b_i = 0, \quad i \in E$$

Note that $c_i(x)$ is concave, hence $-c_i(x)$ is convex function on convex set $\{x | c_i(x) > 0, \quad i \in I\}$. Hence, the objective function $F(x; \mu)$ is convex.
The approximation gets better as $\mu \to 0$. Therefore, the idea is we let μ decrease overtime in the algorithm.

12.3.1 First Order KKT condition for transformed problem

Lagrangian Multiplier Function:

$$L(x, \lambda) = f(x) - \mu \sum_{i \in I} \log(c_i(x)) - \sum_{i \in E} \lambda_i (a_i^T x - b_i)$$

$$\nabla f(x) - \sum_{i \in I} \nabla c_i(x) \frac{\mu}{c_i(x)} - \sum_{i \in E} a_i \lambda_i = 0$$

When x is close to the minimizer $x(\mu)$ and μ is small the optimal Lagrange multiplier $\lambda_i^*, \quad i \in i$, can be estimated as,

$$\lambda_i^* = \frac{\mu}{c_i(x)}, \quad i \in I$$

problem: $x(\mu)$ becomes prohibitively difficult to find as $\mu \to 0$ because of the nonlinearity of the barrier transformed problem. (Numerically unstable as $\mu \to 0$.

12.4 Primal-Dual Reformulation

Let $\lambda_i = \frac{\mu}{c_i(x)}, \quad i \in I$
If $c_i(x) > 0$ then $\lambda_i - \frac{\mu}{c_i(x)} = 0$ iff $c_i(x)\lambda_i - \mu = 0$
We formulate the following (primal-dual nonlinear equations):

$$\nabla f(x) - \sum_{i \in I} \nabla c_i(x) \lambda_i - \sum_{i \in E} a_i \lambda_i = 0$$

$$a_i^T x - b_i = 0, \quad i \in e$$

$$c_i(x)\lambda_i - \mu = 0, \quad i \in I$$

Note that $c_i(x)\lambda_i - \mu = 0, \quad i \in I$ is a perturbed complementary slackness (as $\mu \to 0$, it becomes the original complementary slackness)

A method based on approximately solving these equations is called a **primal-dual interior point method**

12.5 Interior Point Methods

We consider the special case of linear programming, for simplicity

Primal Linear Programming

$$\min_x \ c^T x$$

$$s.t. Ax = b, \quad x \geq 0$$

Dual Linear Programming

$$\max_y \ b^T y \quad ^a$$

$$s.t. A^T y + s = c, \quad s \geq 0$$

[a] y really is the λ

13 Basics

13.1 Some Aspects of Linear Algebra Relevant to Optimization

[20]

13.1.1 Definition 1

A symmetric matrix A is said to be a positive-definite matrix if for any vector y, $||y|| > 0$

$$y^T Ay > 0$$

13.1.2 Definition 2

A symmetric matrix A is said to be a positive-semidefinite matrix if it is not a positive-definite matrix and for any vector y, $||y|| > 0$

$$y^T Ay \geq 0$$

13.1.3 Definition 3

Let A be an $n \times n$ symmetric matrix, then there exists n orthonormal vectors, $v_1, ..., v_2$ and n scalars $\lambda_1, ..., \lambda_n$ such that

$$Av_i = \lambda_i v_1, \quad i = 1, .., n$$

Vector v_i is an eigenvector of A and λ_i is its associated eigenvalue.

13.1.4 Definition 4

A set of vectors $a_1, a_2, ..., a_n$ is said to be linearly independent
if

$$\sum_{j=1}^{n} \beta_j a_j = 0$$

then (implies),

$$\beta_j = 0, \quad j = 1, 2, 3, 4, ...$$

13.1.5 Definition 5

The rank of a matrix A is equal to the maximum number of linearly independent rows.

13.1.6 Definition 6

The space spanned by a set of vectors is the space generated by all linear combinations of those vectors.

[20] Numerical Methods for Unconstrained Optimization, edited by W.Murray (1972)

13.1.7 Definition 7

The range of a matrix, denoted say by $R(A)$, is the space spanned by the columns of A. If $y \in R(A)$ then there exists a vector x such that

$$y = Ax$$

13.1.8 Definition 8

The null space of a matrix A, denoted say by $N(A)$, is the space spanned by the vectors orthogonal to the columns of A. If $y \in R(A)$ and $w \in N(A)$ then

$$y^T w = 0$$

13.1.9 Definition 9

The **pseudo-inverse** of an $m \times n$ matrix a is defined as the $n \times m$ matrix X which satisfies the four equations

$$AXA = A$$

$$XAX = X$$

$$(AX)^T = AX$$

$$(XA)^T = XA$$

13.1.10 Definition 10

the condition number of a non-singular matrix A is defined to be k where

$$k = ||A||\ ||A^{-1}||$$

13.1.11 Definition 11

The spectral norm of a symmetric non-singular matrix is defined to be

$$||A|| = |\lambda_{\max}|$$

where $\lambda_{\max}$ is the eigenvalue of maximum modulus of A.

13.2 Results

13.2.1 Results 1

Consider the set of equations

$$Ax = b$$

where A is an $m \times n$ matrix and b is an $n \times 1$ vector
when a solution exists, a particular solution is given by

$$x = Yb$$

where Y is any $n \times m$ matrix satisfying the first equation in definition (8). The matrix Y is said to be a generalized inverse of A.

13.2.2 Results 2

If the equations $Ax = b$ have a solution then it follows from the result (1) that

$$x = Xb$$

where X is the **pseudo-inverse** of A, is a particular solution. It is the solution whose Euclidean length is a minimum. Moreover, if $Ax = b$ does not have a solution then x given by $x = Xb$ is the solution of minimum Euclidean length to the problem

$$\min_x \{(Ax - b)^T (Ax - b)\}$$

13.2.3 Results 3

A symmetric matrix of rank r has r non-zero eigenvalues

13.2.4 Results 4

A positive-definite matrix has positive eigenvalues

13.2.5 Results 5

A positive-semidefinite matrix has non-negative eigenvalues with at least one zero eigenvalue

13.2.6 Results 6

If A is an $n \times n$ symmetric matrix whose eigenvalues are $\lambda_1, ..., \lambda_n$ with corresponding orthonormal eigenvectors $v_1, ..., v_n$ then

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^T$$

In addition, if A is non-singular then

$$A^{-1} = \sum_{i=1}^{n} \lambda_i^{-1} v_i v_i^T$$

13.2.7 Results 7

It follows from definition (10) and (11) that the condition number of a symmetric non-singular matrix under the spectral norm (spectral condition number) is given by:

$$k = |\lambda_{\max}/\lambda_{\min}|$$

where $\lambda_{\max}$ is the eigenvalue of the largest modulus of A.
and $\lambda_{\min}$ is the eigenvalue of the smallest modulus of A.

13.2.8 Results 8

If A is an $n \times n$ non-singular matrix and x and y are two $n \times 1$ column vectors such that $A + xy^T$ is non-singular then

$$(A + xy^T)^{-1} = A^{-1} - \frac{A^{-1}xy^TA^{-1}}{1 + y^TA^{-1}x}$$

This is often referred to as Householder's modification rule.

13.2.9 Matrix Norm

$$||AB|| \leq ||A||\ ||B||$$

13.3 Real Analysis

13.4 Taylor's Expansion

If in interval $[x,\ x + h]$, function f is
(1) continuous
(2) n times differentiable,
then there is exists some point in this interval, denoted by $x + th$ for some $t \in [0, 1]$, such that:

$$f(x + h) = f(x) + hf^{'}(x) + \frac{1}{2}h^2f^{''}(x) + ... + \frac{1}{(n-1)!}h^nf^{n-1}(x) + ...$$

$$f(x + p) = f(x) + \nabla f(x + tp)^Tp$$

$$f(x + p) \approx f(x) + p^T\nabla f(x) + \frac{1}{2}p^T\nabla^2 f(x)p$$

$$f(x_k + \varepsilon p_k) = f(x_k) + \varepsilon p_k^T\nabla f_k + o(\varepsilon^2)$$

$$\nabla f(x + p) = \nabla f(x) + \int_0^1 \nabla^2 f(x + tp)pdt$$

13.4.1 Triangle Inequality

$$||x + y|| \leq ||x|| + ||y||$$

13.5 Directional Gradient and Gradient Vector

Gradient point is the direction of maximum increase.
The gradient vector is orthogonal to level curves and level surface (sets in higher dimensions)

Definition (Directional Gradient)

$$D_u f(x,y) = f_x(x,y)\cos(\theta) + f_y(x,y)\sin(\theta)$$

If $\theta = 0$, $\Rightarrow f_x(x,y)$
If $\theta = \frac{\pi}{2}$, $\Rightarrow f_y(x,y)$
Partial derivative with respect to x gives the slope in the x direction.
Partial derivative with respect to y gives the slope in the y direction.

Definition (Gradient)

$$\nabla f(x,y) = f_x(x,y)i + f_y(x,y)j \quad \text{(vector)}$$

Let unit vector $u =< \cos(\theta), \sin(\theta) >$,

$$D_u f(x,y) = \nabla f(x,y)\ u$$

$$D_u f(x,y) = ||\nabla f(x,y)||\cos(\phi)$$

where ϕ is the angle between unit vector u and gradient vector $\nabla f(x,y)$
$D_u f(x,y)$ is maximized when $\cos(\phi) = 1$ or $\phi = 0$

$$D_u f(x,y) = ||\nabla f(x,y)||$$

And indicates that the unit vector u (direction that we move along) should be the same as the gradient vector if we want to max the value

Theorem

If f is differentiable at the point (x_0, y_0) and if $\nabla f(x_0, y_0) \neq 0$.
Then $\nabla f(x_0, y_0)$ is normal to the level curve through the point (x_0, y_0)
In 3-Dimension, the gradient is normal to the level surface through the point (x_0, y_0, z_0)

Consider the following example:

$$f(x,y,z) = x^2 + y^2 - 4z$$

$$\nabla f(x,y,z) = 2xi + 2yj - 4k$$

At point $(x, y, z) = (2, -1, 1)$

$$\nabla f(2, -1, 1) = 4i - 2j - 4k$$

$$f(2, -1, 1) = 1$$

The level surface is:

$$x^2 + y^2 - 4z = 1$$

$$z = \frac{1}{4}(x^2 + y^2 - 1)$$

Tangent plane:, say, we have the following

$$z = x^2 + y^2$$

and we want to know the tangent plane at $(1, 1, 2)$

$$F(x, y, z) = x^2 + y^2 - z$$

$$\nabla F(x, y, z) =< 2x, 2y, -1 >$$

$$\nabla F(1, 1, 2) =< 2, 2, -1 >$$

The Tangent plane:

$$2(x - 1) + 2(y - 1) - 1(z - 2) = 0$$

In general, if we have $z = f(x, y)$

$$F(x, y, z) = f(x, y) - z$$

the tangent plane at the point (x_0, y_0, z_0)

$$F_x(x - x_0) + F_y(y - y_0) + F_z(z - z_0) = 0$$

$$\Rightarrow f_x(x - x_0) + f_y(y - y_0) - (z - z_0) = 0$$

$$z - z_0 = f_x(x - x_0) + f_y(y - y_0)$$

$$\Rightarrow \Delta z \approx f_x(x_0, y_0)\Delta x + f_y(x_0, , y_0)\Delta y$$

Note: the tangent plane is a good approximation of that surface near the point of tangency.