Optimization Methods (CS1.404)
Spring 2025

Naresh Manwani

Machine Learning Lab, IIIT-H

January 23th, 2025

Affine Set

A set $C \subseteq \mathbb{R}^d$ is said to be an affine set if for any two distinct points, the line passing through these points also lied in the set C. Thus, if $\mathbf{x}_1, \mathbf{x}_2 \in C$, then $\theta\mathbf{x}_1 + (1-\theta)\mathbf{x}_2 \in C$, $\forall\theta \in \mathbb{R}$.

- C is an affine set if and only if it contains every affine combination of its points.
- For example, solution of a linear equation is an affine set.

Convex Set

Definition

A set $\mathcal{X} \subseteq \mathbb{R}^d$ is called convex, if for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$, $\lambda \mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2 \in \mathcal{X}$, $\forall \lambda \in [0, 1]$.

Convex Set

Definition

A set $\mathcal{X} \subseteq \mathbb{R}^d$ is called convex, if for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$, $\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \in \mathcal{X}, \ \forall \lambda \in [0,1]$.

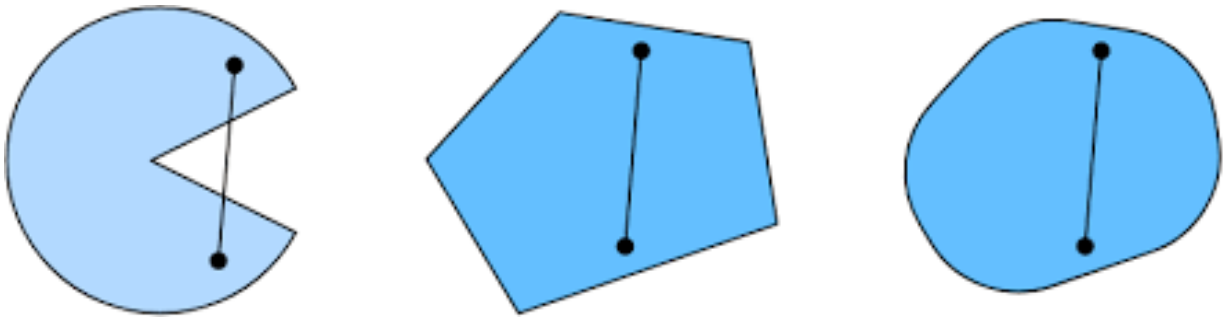
- Note that $\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2, \ \forall \lambda \in [0,1]$ represents the line segment joining $\mathbf{x}_1, \mathbf{x}_2$. For $\mathcal{X}$ to be a convex set, this line segment has to lie inside the set $\mathcal{X}$.

Convex Set

Definition

A set $\mathcal{X} \subseteq \mathbb{R}^d$ is called convex, if for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$, $\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \in \mathcal{X}, \ \forall \lambda \in [0,1]$.

- Note that $\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2, \ \forall \lambda \in [0,1]$ represents the line segment joining $\mathbf{x}_1, \mathbf{x}_2$. For $\mathcal{X}$ to be a convex set, this line segment has to lie inside the set $\mathcal{X}$.



Convex Combination

A convex combination is a linear combination of points (which can be vectors, scalars, or more generally points) where all coefficients are non-negative and sum to 1.

Convex Combination

A convex combination is a linear combination of points (which can be vectors, scalars, or more generally points) where all coefficients are non-negative and sum to 1.

Definition: Convex Combination

Given a finite number of points $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n$ in a real vector space, a convex combination of these points is a point of the form $\lambda_0\mathbf{x}_0 + \lambda_1\mathbf{x}_1 + \ldots + \lambda_n\mathbf{x}_n$ where $\lambda_i \geq 0,\ i = 0, 1, \ldots, n$ and $\sum_{i=0}^{n} \lambda_i = 1$.

Convex Combination

A convex combination is a linear combination of points (which can be vectors, scalars, or more generally points) where all coefficients are non-negative and sum to 1.

Definition: Convex Combination

Given a finite number of points $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n$ in a real vector space, a convex combination of these points is a point of the form $\lambda_0\mathbf{x}_0 + \lambda_1\mathbf{x}_1 + \ldots + \lambda_n\mathbf{x}_n$ where $\lambda_i \geq 0,\ i = 0, 1, \ldots, n$ and $\sum_{i=0}^{n} \lambda_i = 1$.

As a particular example, every convex combination of two points lies on the line segment between the points.

Intersection of Convex Sets

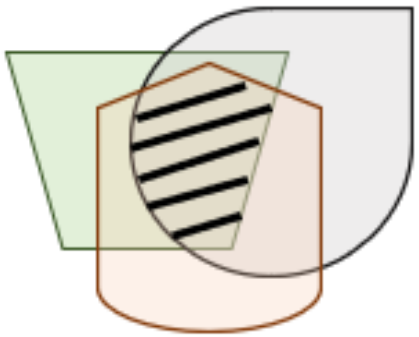
Result

Let $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k \subseteq \mathbb{R}^d$ be convex sets. Then $\cap_{i=1}^{k} \mathcal{X}_i$ is also a convex set.

Intersection of Convex Sets

Result

Let $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k \subseteq \mathbb{R}^d$ be convex sets. Then $\cap_{i=1}^{k} \mathcal{X}_i$ is also a convex set.



Union of Convex Sets

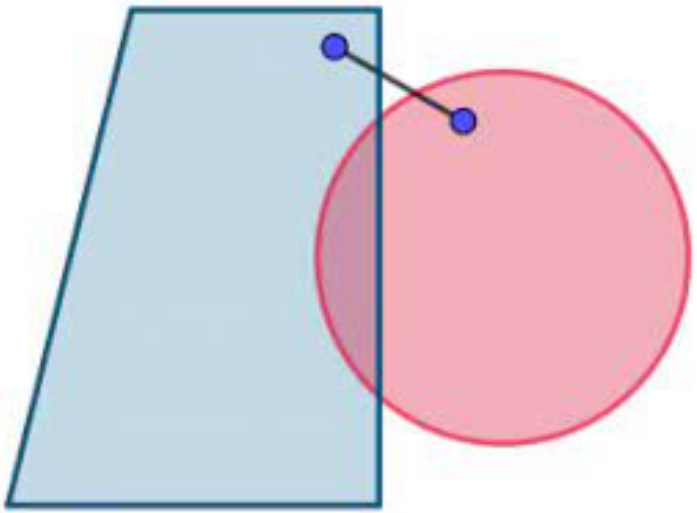
Result

Let $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k \subset \mathbb{R}^d$ be convex sets. Then $\cup_{i=1}^{k} \mathcal{X}_i$ may not be a convex set.

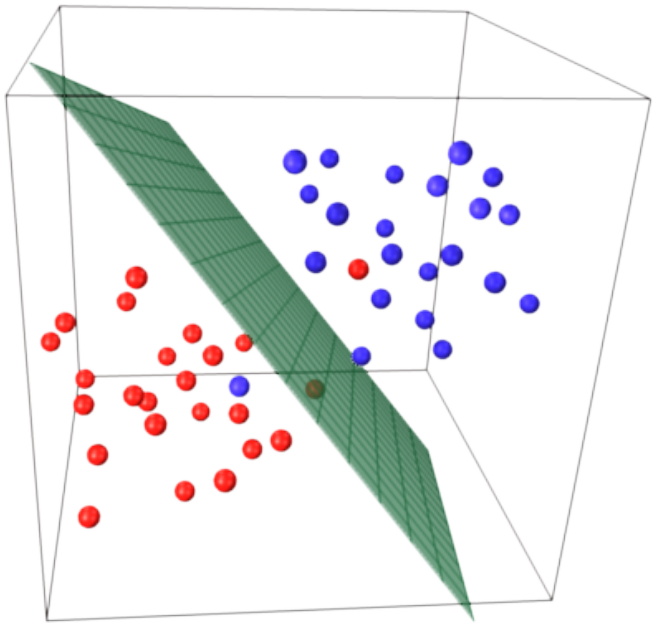
Union of Convex Sets

Result

Let $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_k \subset \mathbb{R}^d$ be convex sets. Then $\cup_{i=1}^{k} \mathcal{X}_i$ may not be a convex set.



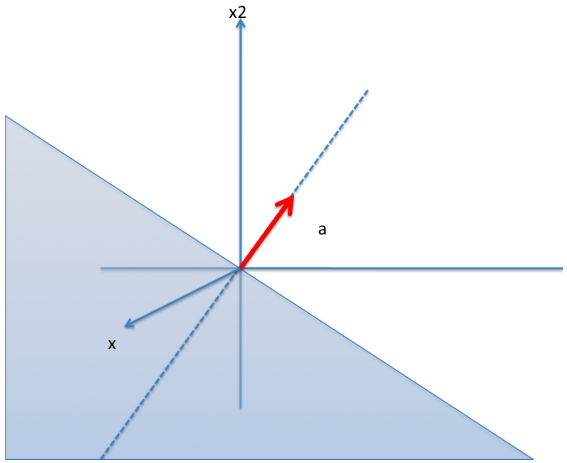
Hyperplane



A hyperplane in $\mathbb{R}^d$ is a set of the form $\{\mathbf{x} \mid \mathbf{w}^T\mathbf{x} = b\}$ where $\mathbf{w} \in \mathbb{R}^d$ is normal to the hyperplane and $b \in \mathbb{R}$ is offset parameter.

Hyperplane is convex set also.

Halfspaces



- A half-space is either of the two parts into which a hyperplane divides.
- A half-space may be specified by a linear inequality, derived from the linear equation that specifies the defining hyperplane.
- A strict linear inequality specifies an open half-space:
  $w_1x_1 + w_2x_2 + \ldots + w_dx_d > b$
- A non-strict inequality specifies a closed half-space:
  $w_1x_1 + w_2x_2 + \ldots + w_dx_d \geq b$
- Here, one assumes that not all of the real numbers a1, a2, ..., an are zero.

Closed half-spaces $\{\mathbf{x} \in \mathbb{R}^d \mid w_1x_1 + w_2x_2 + \ldots + w_dx_d \geq b\}$ and $\{\mathbf{x} \in \mathbb{R}^d \mid w_1x_1 + w_2x_2 + \ldots + w_dx_d \leq b\}$ are convex sets.

Weierstrass Theorem

Theorem

Let $\mathcal{X} \subset \mathbb{R}^n$ be a nonempty compact (closed and bounded) set and $f : \mathcal{X} \to \mathbb{R}$ be a continuous function on $\mathcal{X}$. Then, f attains a minimum and a maximum on $\mathcal{X}$.

- Weierstrass Theorem is not a necessary condition.

Closest Point Theorem

Theorem

let $S \subset \mathbb{R}^n$ be a nonempty, closed convex set and $\mathbf{y} \notin S$. Then there exists a unique point $\mathbf{x}_0 \in S$ with minimum distance from $\mathbf{y}$. Further $\mathbf{x}_0$ is the minimum distance point if and only if $(\mathbf{y} - \mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0) \leq 0, \forall \mathbf{x} \in S$.

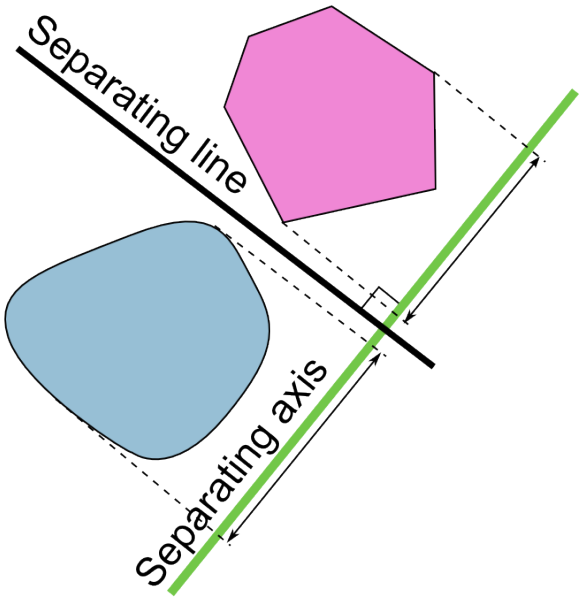
Optimization Methods (CS1.404)
Spring 2025

Naresh Manwani

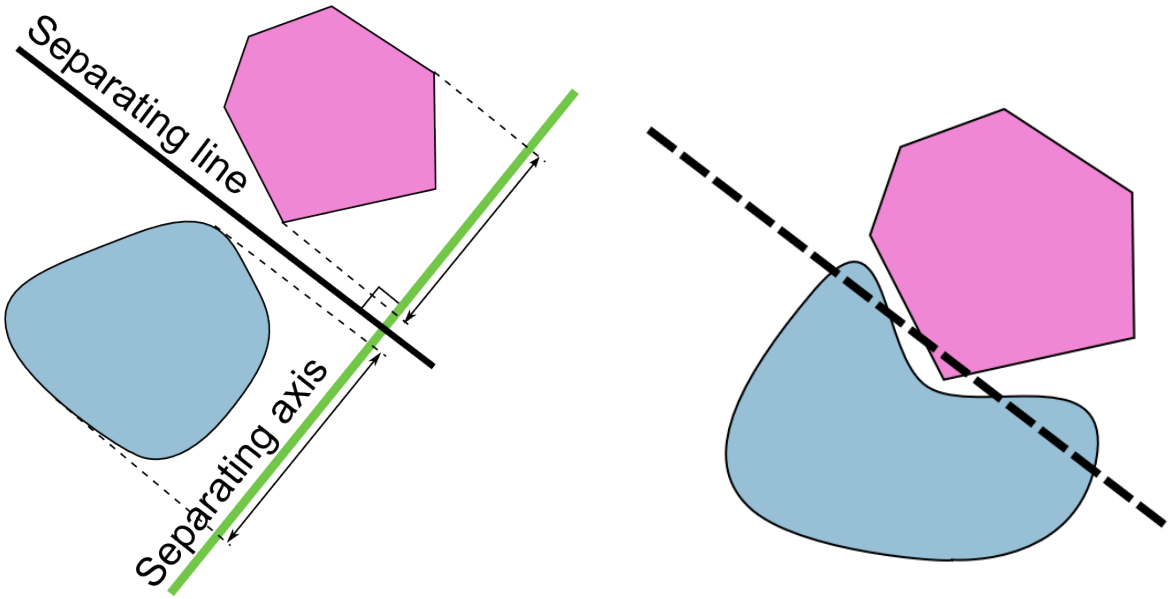
Machine Learning Lab, IIIT-H

February 3rd, 2025

Separating Hyperplane



Separating Hyperplane



Separating Hyperplane Theorem

Theorem

Let $\mathcal{X}_1$ and $\mathcal{X}_2$ be two disjoint, non-empty convex subsets of $\mathbb{R}^d$, then there exists a non-zero vector $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that $\mathbf{w}^T\mathbf{x} \geq b$, $\forall \mathbf{x} \in \mathcal{X}_1$ and $\mathbf{w}^T\mathbf{x} \leq b$, $\forall \mathbf{x} \in \mathcal{X}_2$. Thus, $\mathbf{w}^T\mathbf{x} = b$ separates $\mathcal{X}_1$ and $\mathcal{X}_2$.

- If $\mathbf{w}^T\mathbf{x} > b$, $\forall \mathbf{x} \in \mathcal{X}_1$ and $\mathbf{w}^T\mathbf{x} < b$, $\forall \mathbf{x} \in \mathcal{X}_2$, the the hyperplane $\mathbf{w}^T\mathbf{x} = b$ is said to **strictly separate** $\mathcal{X}_1$ and $\mathcal{X}_2$.
- The hyperplane is said to **strongly separate** $\mathcal{X}_1$ and $\mathcal{X}_2$ if $\mathbf{w}^T\mathbf{x} \geq b + \epsilon$, $\forall \mathbf{x} \in \mathcal{X}_1$ and $\mathbf{w}^T\mathbf{x} \leq b - \epsilon$, $\forall \mathbf{x} \in \mathcal{X}_2$ for some $\epsilon > 0$. The strong separation between disjoint convex sets $\mathcal{X}_1$ and $\mathcal{X}_2$ happens if both are closed, and at least one of them is also bounded.

Strict Separation between a point and a closed convex set

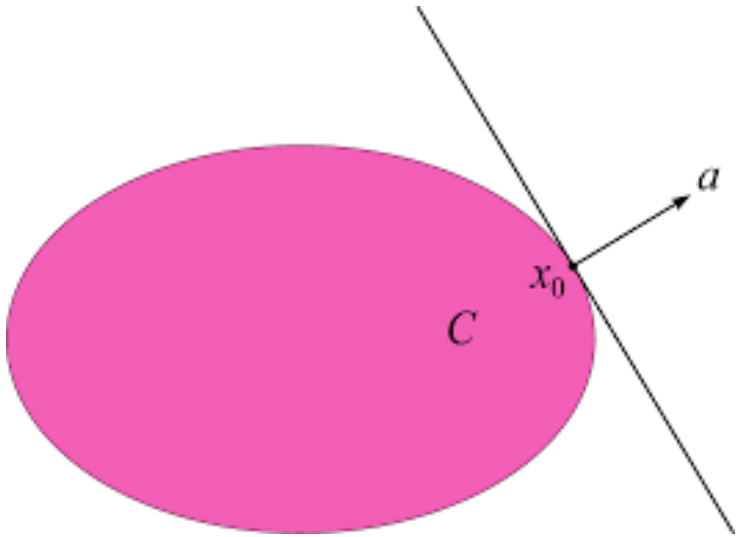
Theorem

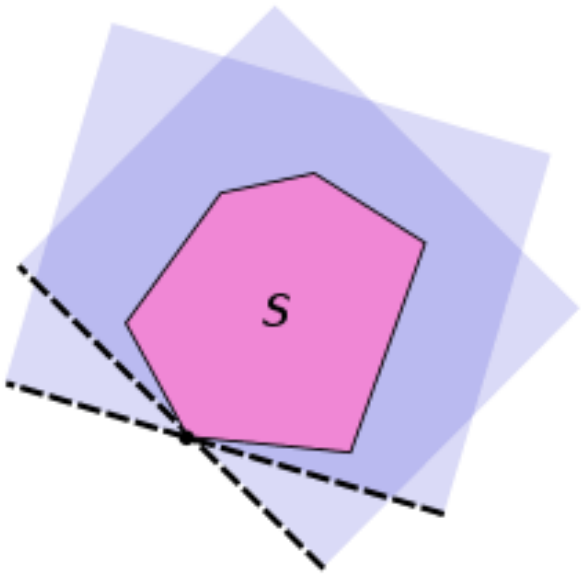
Let $\mathcal{X} \subset \mathbb{R}^d$ be a closed convex set and $\mathbf{y} \notin \mathcal{X}$. Then, there exists a hyperplane that strictly separates $\mathcal{X}$ and $\mathbf{y}$.

Supporting Hyperplane Theorem

Theorem

If C is convex, then a supporting hyperplane exists at every boundary point of C. Let $\mathbf{x}_0$ be a boundary point of set C, then there exists $\mathbf{a} \neq \mathbf{0}$ such that $\{\mathbf{x} : \mathbf{a}^T\mathbf{x} = \mathbf{a}^T\mathbf{x}_0\}$ is a supporting hyperplane to C and $\mathbf{a}^T\mathbf{x} \leq \mathbf{a}^T\mathbf{x}_0$, $\forall \mathbf{x} \in C$.





Convex Hull

Definition

Let $\mathcal{X} \subset \mathbb{R}^n$ be a set. The **convex hull** of $\mathcal{X}$ is the intersection of all convex sets containing it.

Convex Hull

Definition

Let $\mathcal{X} \subset \mathbb{R}^n$ be a set. The **convex hull** of $\mathcal{X}$ is the intersection of all convex sets containing it.

Lemma

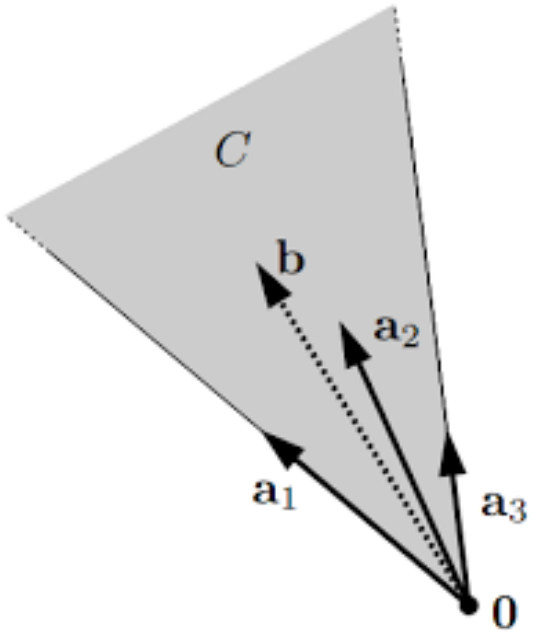
The convex hull of the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ is the set of all convex combinations of these vectors.

Convex Cone

Definition

Given vectors $\mathbf{a}_1, \ldots, \mathbf{a}_n$, the convex cone generated by these vectors is the set of all non-negative linear combinations of $\mathbf{a}_i$'s. That is,

$$\text{Convex-Cone}(\mathbf{a}_1, \ldots, \mathbf{a}_n) = \{t_1\mathbf{a}_1 + \ldots + t_n\mathbf{a}_n \mid t_1, \ldots, t_n \geq 0\}$$

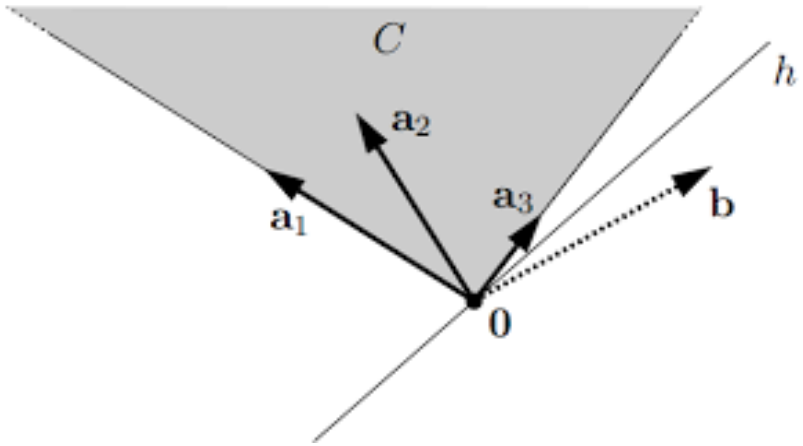


Farkas Lemma

Lemma

Let $A \in \mathbb{R}^{m \times n}$ and let $C = \{A\mathbf{x} \mid \mathbf{x} \geq \mathbf{0}\}$. Note that C is a closed convex cone. Then, exactly, one of the two systems has a solution:

1. $A\mathbf{x} = \mathbf{b}$ and $\mathbf{x} \geq \mathbf{0}$
2. $A^T\mathbf{y} \geq \mathbf{0}$ and $\mathbf{b}^T\mathbf{y} < 0$



Convex Functions
Optimization Methods (CS1.404), Spring 2024

Naresh Manwani

Machine Learning Lab, IIIT-H

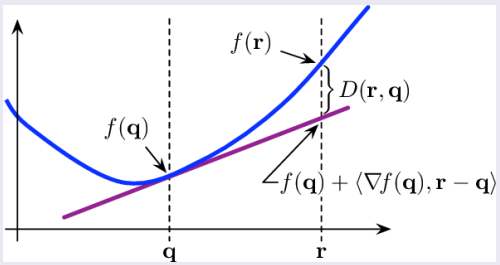
February 5th, 2024

First Order Characterization of Convex Functions

The Gradient Inequality

Let $f : S \to \mathbb{R}$ be a continuously differentiable function defined on a convex set $S \subseteq \mathbb{R}^n$ Then, f is convex over S if and only if

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}),\ \forall \mathbf{x}, \mathbf{y} \in S$$



The Gradient Inequality for Strictly Convex Function

Let $f : S \to \mathbb{R}$ be a continuously differentiable function defined on a convex set $S \subseteq \mathbb{R}^n$ Then, f is strictly convex over S if and only if

$$f(\mathbf{y}) > f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}),\ \forall \mathbf{x}, \mathbf{y} \in S\ \ (\mathbf{x} \neq \mathbf{y})$$

Sufficiency of Stationarity Under Convexity

Proposition

Let f be a continuously differentiable function which is convex over a convex set $S \subseteq \mathbb{R}^n$ Suppose that $\nabla f(\mathbf{x}^*) = \mathbf{0}$ for some $\mathbf{x}^* \in S$, then $\mathbf{x}^*$ is global minimizer of f over S.

Monotonicity of the Gradient of Convex Functions

Theorem

Suppose that f is a continuously differentiable function over a convex set $S \subseteq \mathbb{R}^n$. Then, f is convex over S if and only if

$$(\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})^T(\mathbf{x} - \mathbf{y}) \geq 0, \ \forall \mathbf{x}, \mathbf{y} \in S$$

Second Order Characterization of Convex Function

Theorem

Let f be a twice continuously differentiable function over an open convex set $S \subseteq \mathbb{R}^n$. Then, f is convex over S if and only if $\nabla^2 f(\mathbf{x})$ is positive semi-definite for any $\mathbf{x} \in S$.

Sufficient Second Order Characterization for Strict Convexity

Let f be a twice continuously differentiable function over a convex set $S \subseteq \mathbb{R}^n$, and suppose that $\nabla^2 f(\mathbf{x})$ is strictly positive definite for all $\mathbf{x} \in S$. Then, f is strictly convex over S.

Examples of Convex Functions:

1. log-sum-exponential function: $f(\mathbf{x}) = \ln(e^{x_1} + e^{x_2} + \ldots + e^{x_n})$
2. quadratic over linear: $f(x_1, x_2) = \frac{x_1^2}{x_2}$
3. $f(x) = x \log x$ where f is defined over $S = \{x \in \mathbb{R} \mid x > 0\}$
4. $f(\mathbf{x}) = \|A\mathbf{x} - \mathbf{b}\|^2$

Jensen's Inequality

Theorem

Let $f : S \to \mathbb{R}$ be a convex function defined on the convex set $S \subseteq \mathbb{R}^n$. Then for any $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k \in S$ and $\boldsymbol{\lambda} \in \Delta_k$, the following inequality holds:

$$f(\sum_{j=1}^{k} \lambda_k \mathbf{x}_k) \leq \sum_{j=1}^{k} \lambda_k f(\mathbf{x}_k)$$

where Δ_k is k-dimensional probability simplex.

Optimization Methods (CS1.404)
Spring 2025

Naresh Manwani

Machine Learning Lab, IIIT-H

February 13th, 2025

Descent Direction Methods

- We consider the unconstrained minimization problem as follows:

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

  where we assume that f is continuously differentiable over $\mathbb{R}^n$.
- In many cases, it might be very difficult to solve the equation $\nabla f(\mathbf{x}) = \mathbf{0}$ to find the stationary points.
- Even if it is possible to find the solutions of $\nabla f(\mathbf{x}) = \mathbf{0}$, if there are infinitely many solutions, finding the one corresponding to a local minima might be as difficult problem as original optimization problem.
- Due to these reasons, instead of finding the stationary points analytically, we consider adopting an iterative algorithm to find them.
- Iterative algorithms to find the stationary points are of the following form:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + t_k \mathbf{d}_k, \quad k = 0, 1, 2, \ldots,$$

  where $\mathbf{d}_k$ is the so-called direction t_k is the stepsize.

Descent Direction

Definition

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function over $\mathbb{R}^n$. A vector $\mathbf{d} \in \mathbb{R}^n$ $(\mathbf{d} \neq \mathbf{0})$ is said a **descent direction** of f at $\mathbf{x}$ if the directional derivative of f at $\mathbf{x}$ along the direction $\mathbf{d}$ is negative, i.e.,

$$\nabla f(\mathbf{x})^T \mathbf{d} < 0$$

Remark: Taking small enough steps along descent directions lead to a decrease of the function f.

Descent Property of Descent Directions

Lemma

Let f be a continuously differentiable function over an open set S of $\mathbb{R}^n$ and let $\mathbf{x} \in S$. Suppose that $\mathbf{d}$ is a descent direction of f at $\mathbf{x}$. Then there exist $\epsilon > 0$ such that

$$f(\mathbf{x} + \alpha \mathbf{d}) < f(\mathbf{x})$$

for any $\alpha \in (0, \epsilon]$.

Schematic Descent Directions Method

Schematic Descent Directions Method

- **Initialization:** Pick $\mathbf{x}_0 \in \mathbb{R}^n$ arbitrarily
- **General Step:** For any $k = 0, 1, 2, \ldots$, set
  1. Pick a descent direction $\mathbf{d}_k$.
  2. Find a step size t_k satisfying $f(\mathbf{x}_k + t_k \mathbf{d}_k) < f(\mathbf{x}_k)$.
  3. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + t_k \mathbf{d}_k$.
  4. STOP if the stopping condition is satisfied and Output $\mathbf{x}_{k+1}$. Else go to Step (1).

Challanges:

1. How to choose the initial point $\mathbf{x}_0$?
2. How to choose the descent direction $\mathbf{d}_k$?
3. How to choose the stepsize t_k?
4. What should be the stopping condition?
5. Does the algorithm converge? If yes, then how fast does it converge? Does the convergence depend on $\mathbf{x}_0$?

Stopping Condition

1. Stopping condition for a minimization problem is $\nabla f(\mathbf{x}_k) = \mathbf{0}$ and $\nabla^2 f(\mathbf{x}_k)$ is positive semi-definite.
2. A practical stopping condition is $\|\nabla f(\mathbf{x}_k)\| \leq \epsilon$.
3. Other stopping conditions

$$\|\nabla f(\mathbf{x}_k)\| < \epsilon(1 + |f(\mathbf{x}_k)|)$$

$$\frac{f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})}{|f(\mathbf{x}_k)|} \leq \epsilon$$

Finding Step Size t_k

- Step size t_k is chosen in such a way that $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$.
- The method of finding step size is called line search, since it a minimization of one dimensional function $g(t) = f(\mathbf{x}_k + t\mathbf{d}_k)$.
- Four popular choices for step size selection are as follows:
  - **Constant Step size:** $t_k = \eta$, $\forall k$. It is very simple approach, but it is unclear how to choose η. A large value of η might cause the algorithm to be nondecreasing and small η can cause very slow convergence.
  - **Diminishing Step Size:** $\alpha_k \to 0$, $\sum_{k=1}^{\infty} \alpha_k = \infty$. For example, $\alpha_k = \frac{1}{k}$.
    - Descent not guaranteed at each step; only later when becomes small.
    - $\sum_{k=1}^{\infty} \alpha_k = \infty$ imposed to guarantee progress does not become too slow.
    - Good theoretical guarantees, but unless the right sequence is chosen, can also be a slow method.

Finding Step Size t_k

- **Exact Line Search:** Here, t_k is the minimizer of f along the ray $\mathbf{x}_k + t\mathbf{d}_k$.

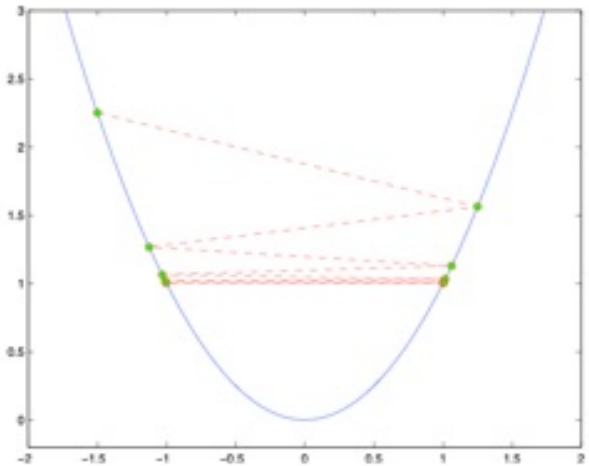
$$t_k = \arg\min_{t \geq 0} \; f(\mathbf{x}_k + t\mathbf{d}_k)$$

  It is an attractive approach, but it is not always possible to find the exact minimizer of $g(t) = f(\mathbf{x}_k + t\mathbf{d}_k)$.
- **Inexact Line Search:** This method iteratively finds t_k which minimizes f along the ray $\mathbf{x}_k + t\mathbf{d}_k$. It finds good enough step size which ensures sufficient decrease.

Example 1: How line search methods fail!

Large Step Sizes

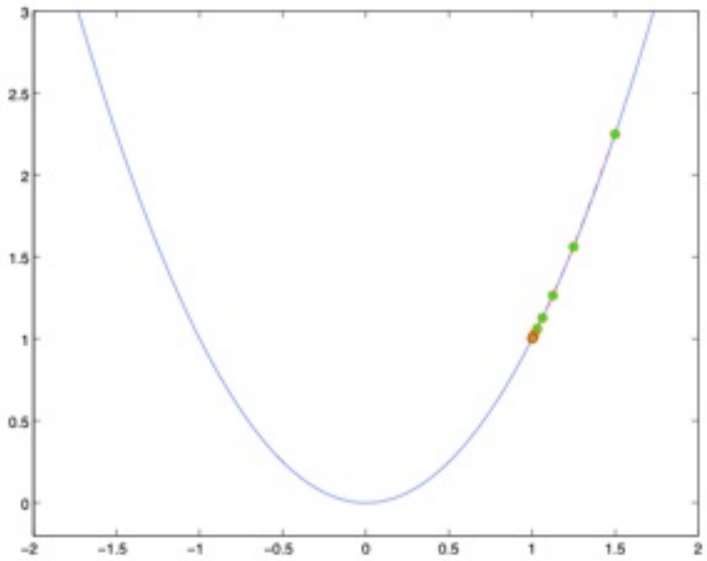
- The objective function $f(x) = x^2$. Global minimizer is $x^* = 0$ and optimal value of $f(x^*) = 0$.
- Iterates $x_{k+1} = x_k + \alpha_k d_k$ generated by the descent directions $d_k = (-1)^k$ and steps $\alpha_k = 2 + 3/2^k$ from $x_1 = 2$.
- $\{x\} = \{2, -3/2, 5/4, -9/8, \ldots\}$. As $k \to \infty$, x_k will oscillate between $+1$ and -1. Thus, the sequence x_k, $k = 1, 2, 3, \ldots$ does not converge.
- $\{f\} = \{4, 9/4, 25/16, 81/64, \ldots\}$. Thus, the function value decreases in each iteration. As $k \to \infty$, $f(x_k)$ will remain close to 1.
- **Key reason is a small decrease in function values relative to the step length.**



Example 2: How line search methods fail !

Small Step Sizes

- The objective function $f(x) = x^2$. Global minimizer is $x^* = 0$ and optimal value of $f(x^*) = 0$.
- Iterates $x_{k+1} = x_k + \alpha_k d_k$ generated by the descent directions $d_k = -1$, $\forall k$ and steps $\alpha_k = 1/2^k$ from $x_1 = 2$.
- $\{x\} = \{2, 3/2, 5/4, 9/8, \ldots\}$. As $k \to \infty$, x_k will converge to $+1$. But, $\lim_{k\to\infty} x_k \neq x^*$.
- $\{f\} = \{4, 9/4, 25/16, 81/64, \ldots\}$. Thus, the function value decreases in each iteration. As $k \to \infty$, $f(x_k)$ will remain close to 1.
- **Key reason is step sizes are too small compared to the initial rate of decrease of f.**



Sufficient Decrease Condition

Lemma

Let f be a continuously differentiable function over $\mathbb{R}^n$ and let $\mathbf{x} \in \mathbb{R}^n$. Suppose that $\mathbf{d} \in \mathbb{R}^n$ ($\mathbf{d} \neq \mathbf{0}$) is a descent direction of $\mathbf{d}$ at $\mathbf{x}$ and let $\alpha \in (0, 1)$. Then there exist $\epsilon > 0$ such that the inequality

$$f(\mathbf{x}) - f(\mathbf{x} + t\mathbf{d}) \geq -\alpha t \nabla f(\mathbf{x})^T \mathbf{d}$$

holds for all $t \in [0, \epsilon]$.

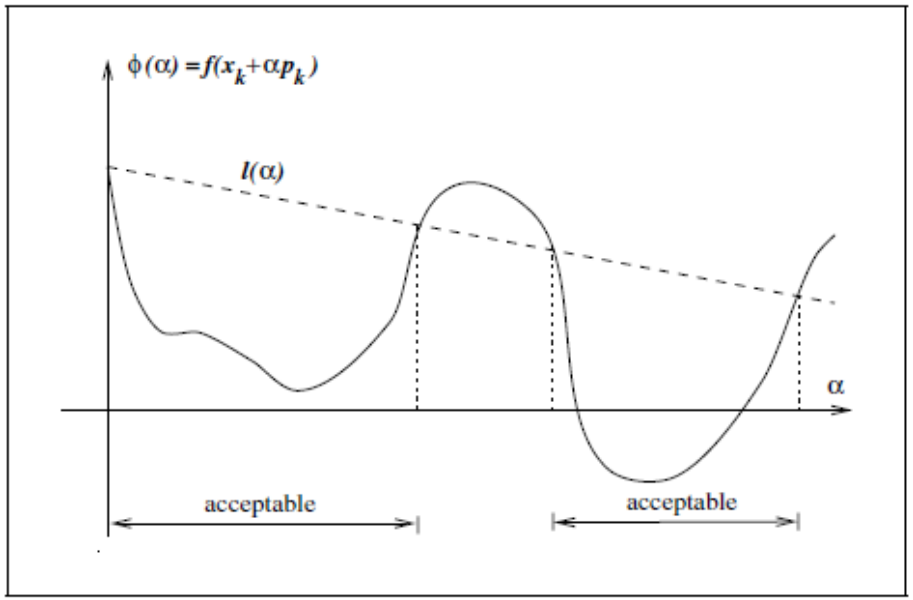
Armijo Line Search Method

- Armijo inexact line search condition stipulates that α_k should, first of all, give sufficient decrease in the objective function f, as measured by the following inequality:

$$f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$

  for some constant $c_1 \in (0, 1)$.
- Thus, the reduction in f should be proportional to both the step length α_k and the directional derivative $\nabla f(\mathbf{x}_k)\mathbf{d}_k$.

Geometric Interpretation of Armijo Condition



- Consider $\phi(\alpha) = f(\mathbf{x}_k + \alpha \mathbf{d}_k)$ and $l(\alpha) = f(\mathbf{x}_k) + c_1 \alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$.
- The function $l(\alpha)$ has negative slope $c_1 \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$, but because $c_1 \in (0, 1)$, it lies above the graph of ϕ for small positive values of α.
- The sufficient decrease condition states that α is acceptable only if $\phi(\alpha) \leq l(\alpha)$. In practice, c_1 is chosen to be quite small, say $c_1 = 10^{-4}$.

Backtracking Line Search With Armijo

Backtracking

1. **Initialize:** $\alpha^{(0)} \in (0,1)$, $\tau \in (0,1)$, $l = 0$
2. Until $f(\mathbf{x}_k + \alpha^{(l)}\mathbf{d}_k) > f(\mathbf{x}_k) + c_1\alpha^{(l)}\nabla f(\mathbf{x}_k)^T\mathbf{d}_k$
   1. Set $\alpha^{(l+1)} = \tau\alpha^{(l)}$
   2. $l = l + 1$
3. $\alpha_k = \alpha^{(l)}$

In practice, the following choices are used

- $\tau \in (0.1, 0.5]$
- $c_1 \in [10^{-5}, 10^{-1}]$

Issue with Armijo's condition:

- It does not ensure that the step size is sufficiently large because Armijo's condition can be satisfied even with a very small step size.
- Backtracking partially addresses this by starting from large step sizes and checking Armijo's condition.
- But can we add some other condition to Armijo?

Armijo-Goldstein Line Search

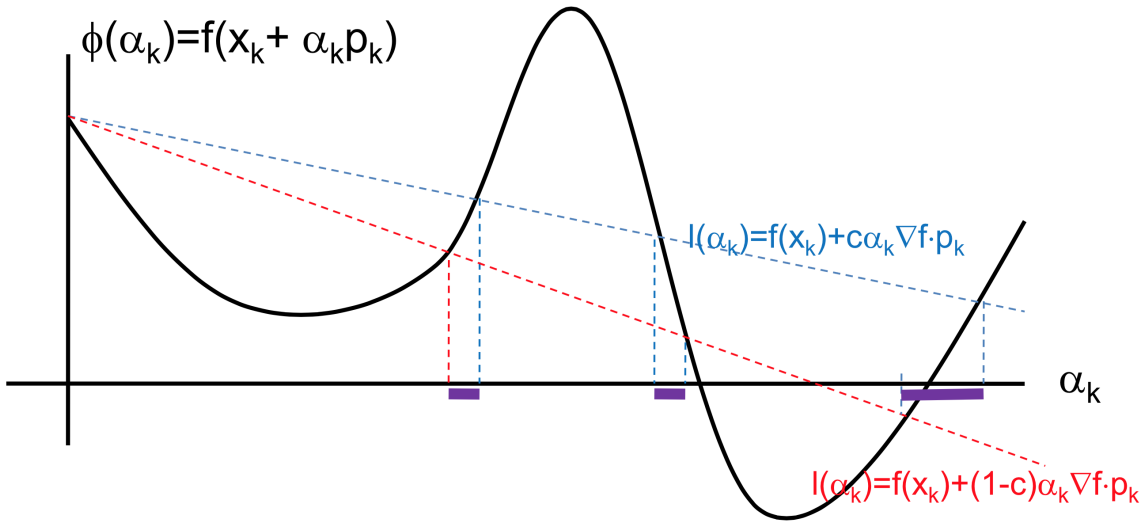
- Armijo-Goldstein inexact line search condition requires that α_k should be sufficiently large and it should give sufficient decrease in the objective function f as well.
- The condition is as follows.

  $$f(\mathbf{x}_k) + (1 - c_1)\alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k \leq f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$

  for some constant $c_1 \in (0, 1/2)$.
- The first inequality is introduced to control the step length from below.
- **Issue:** First inequality may exclude all minimizers of ϕ (see in figure). One can see that the Goldstein condition misses the first local minima.

Geometrical Interpretation of Goldstein Conditions

$$(1-c)\alpha_k \nabla f(x_k)^T p_k + f(x_k) \leq f(x_k + \alpha_k p_k) \leq c\alpha_k \nabla f(x_k)^T p_k + f(x_k)$$
$$(0 < c < 1/2)$$



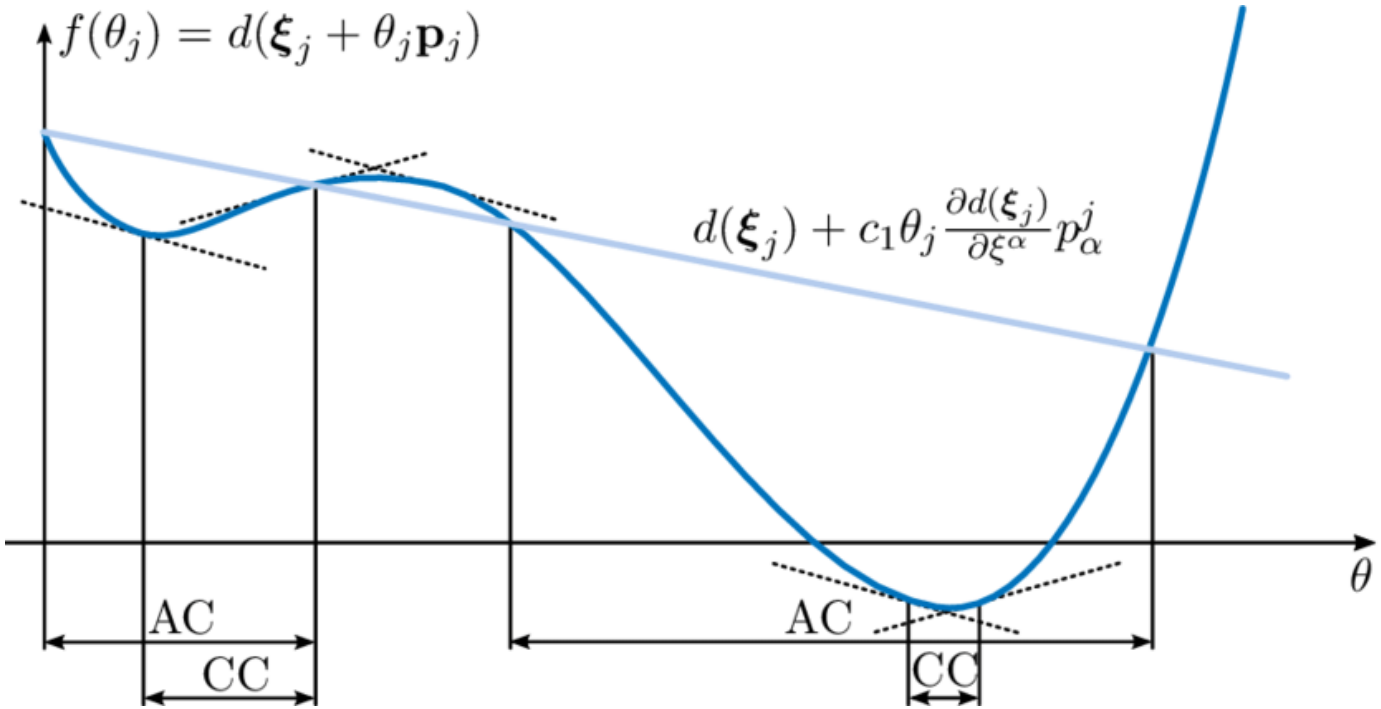
Armijo-Wolfe Condition

- Armijo-Wolfe condition is also used to rule out unacceptably short steps (called the curvature condition) and ensure sufficient decrease.
- The conditions are

$$f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$
$$\nabla f(\mathbf{x}_k + \alpha_k \mathbf{d}_k)^T \mathbf{d}_k \geq c_2 \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$

  for some constants $0 < c_2 < c_1 < 1$.
- LHS in the curvature condition is simply the derivative $\phi'(\alpha_k)$. So, the curvature condition ensures that the slope of ϕ at α_k is greater than c_2 times the initial slope $\phi'(0)$.
- If the slope $\phi'(\alpha)$ is strongly negative, we have an indication that we can reduce f significantly by moving further along the chosen direction. if $\phi'(\alpha_k)$ is only slightly negative or even positive, then we cannot expect more decrease in f in this direction, so it makes sense to terminate the line search.
- Thus, Wolf condition ensures sufficient rate of decrease of function value in the given direction.
- **Issue:** A step length may satisfy the Armijo-Wolfe conditions without being particularly close to a minimizer of ϕ.

Armijo-Wolfe Condition



Optimization Methods (CS1.404), Spring 2025

Naresh Manwani

Machine Learning Lab, IIIT-H

March 3rd, 2025

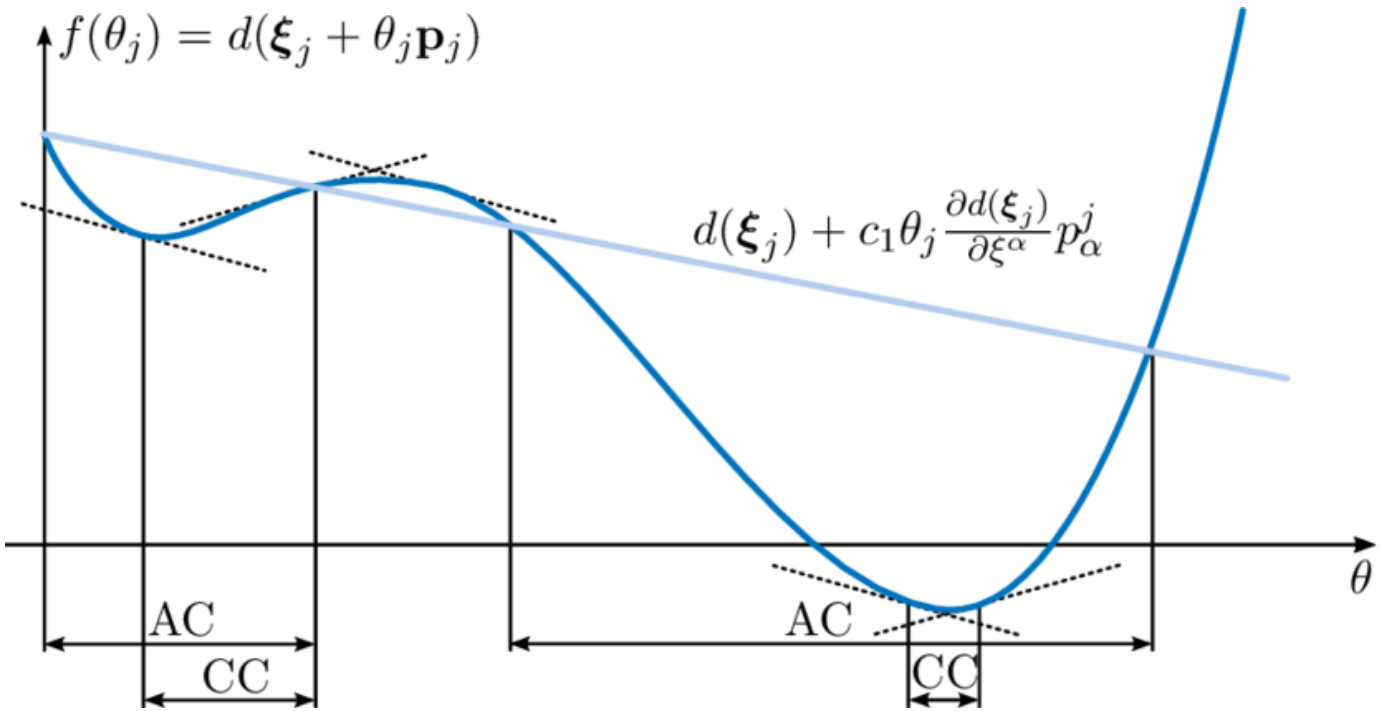
Armijo-Wolfe Condition

- Armijo-Wolfe condition is also used to rule out unacceptably short steps (called the curvature condition) and ensure sufficient decrease.
- The conditions are

$$f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) \leq f(\mathbf{x}_k) + c_1 \alpha_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$
$$\nabla f(\mathbf{x}_k + \alpha_k \mathbf{d}_k)^T \mathbf{d}_k \geq c_2 \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$

  for some constants $0 < c_2 < c_1 < 1$.
- LHS in the curvature condition is simply the derivative $\phi'(\alpha_k)$. So, the curvature condition ensures that the slope of ϕ at α_k is greater than c_2 times the initial slope $\phi'(0)$.
- If the slope $\phi'(\alpha)$ is strongly negative, we have an indication that we can reduce f significantly by moving further along the chosen direction. if $\phi'(\alpha_k)$ is only slightly negative or even positive, then we cannot expect more decrease in f in this direction, so it makes sense to terminate the line search.
- Thus, Wolf condition ensures sufficient rate of decrease of function value in the given direction.
- **Issue:** A step length may satisfy the Armijo-Wolfe conditions without being particularly close to a minimizer of ϕ.

Armijo-Wolfe Condition



Exact Line Search

Here, α_k is the minimizer of f along the ray $\mathbf{x}_k + \alpha \mathbf{d}_k$.

$$\alpha_k = \arg\min_{\alpha \geq 0} f(\mathbf{x}_k + \alpha \mathbf{d}_k).$$

Example: Exact line search for quadratic function

- Let $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} + \mathbf{b}^T\mathbf{x} + c$, where A is an $n \times n$ symmetric positive definite matrix, $\mathbf{b} \in \mathbb{R}^n$ and $c \in \mathbb{R}$.
- Let $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{d} \in \mathbb{R}^n$ be a descent direction of f at $\mathbf{x}$.

Then

$$\arg\min_{t \geq 0} \; f(\mathbf{x} + t\mathbf{d}) = -\frac{\nabla f(\mathbf{x})^T \mathbf{d}}{\mathbf{d}^T A \mathbf{d}}$$

Steepest Gradient Descent

- In the gradient method, the descent direction is chosen as the negative of the gradient at the current point: $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$. For such $\mathbf{d}_k$, we see that $\nabla f(\mathbf{x}_k)^T \mathbf{d}_k = -\|\mathbf{d}_k\|^2 < 0$.
- This is also called the steepest gradient descent direction.

Lemma: Optimality of the Steepest Gradient Descent Diirection

Let f be a continuously differentiable function, and let $\mathbf{x} \in \mathbb{R}^n$ be a non-stationary point (i.e., $\nabla f(\mathbf{x}) \neq \mathbf{0}$). Then, the optimal solution of

$$\min_{\mathbf{d} \in \mathbb{R}^n} \nabla f(\mathbf{x})^T \mathbf{d}$$
$$\text{s.t. } \|\mathbf{d}\| = 1$$

is $\mathbf{d} = -\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}$.

Steepest Gradient Descent Algorithm

- **Input:** $\epsilon > 0$ - tolerance parameter
- **Initialization:** Pick $\mathbf{x}_0 \in \mathbb{R}^n$ arbitrarily.
- **General Step:** For any $k = 0, 1, 2, \ldots$ execute the following steps
  1. Fix $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$
  2. Pick stepsize t_k by a line search on the function
  $$g(t) = f(\mathbf{x}_k + t\mathbf{d}_k)$$
  3. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + t_k\mathbf{d}_k$
  4. If $\|\nabla f(\mathbf{x}_k)\| \leq \epsilon$, then stop and $\mathbf{x}_{k+1}$ is the output.

Example 1: Gradient Descent with Exact Line Search on Quadratic Function

- Consider function $f(x, y) = x^2 + 2y^2$, whose optimal solution is $(0, 0)$ with optimal value 0.
- Let $(x_0, y_0) = (2, 1)$, $\epsilon = 10^{-5}$.
- The Gradient descent approach stops in 13 iterations and finds a solution that is pretty close to the optimal value. $(x^*, y^*) = (0.1254 * 10^{-5}, -0627 * 10^{-5})$.

```
iter_number =   1 norm_grad = 1.885618 fun_val = 0.666667
iter_number =   2 norm_grad = 0.628539 fun_val = 0.074074
iter_number =   3 norm_grad = 0.209513 fun_val = 0.008230
iter_number =   4 norm_grad = 0.069838 fun_val = 0.000914
iter_number =   5 norm_grad = 0.023279 fun_val = 0.000102
iter_number =   6 norm_grad = 0.007760 fun_val = 0.000011
iter_number =   7 norm_grad = 0.002587 fun_val = 0.000001
iter_number =   8 norm_grad = 0.000862 fun_val = 0.000000
iter_number =   9 norm_grad = 0.000287 fun_val = 0.000000
iter_number =  10 norm_grad = 0.000096 fun_val = 0.000000
iter_number =  11 norm_grad = 0.000032 fun_val = 0.000000
iter_number =  12 norm_grad = 0.000011 fun_val = 0.000000
iter_number =  13 norm_grad = 0.000004 fun_val = 0.000000
```

Example 1: Gradient Descent with Constant Step Size on Quadratic Function

- Consider function $f(x, y) = x^2 + 2y^2$, whose optimal solution is $(0, 0)$ with optimal value 0.
- Let $(x_0, y_0) = (2, 1)$, $\epsilon = 10^{-5}$, $t_k = 0.1$.
- The Gradient descent approach stops in 58 iterations.
- The stepsize was too small, which caused slow convergence.

```
iter_number =   1 norm_grad = 4.000000 fun_val = 3.280000
iter_number =   2 norm_grad = 2.937210 fun_val = 1.897600
iter_number =   3 norm_grad = 2.222791 fun_val = 1.141888
      :                :                :
iter_number =  56 norm_grad = 0.000015 fun_val = 0.000000
iter_number =  57 norm_grad = 0.000012 fun_val = 0.000000
iter_number =  58 norm_grad = 0.000010 fun_val = 0.000000
```

Example 1: Gradient Descent with Backtracking Line Search on Quadratic Function

- Consider function $f(x, y) = x^2 + 2y^2$, whose optimal solution is $(0, 0)$ with optimal value 0.
- Let $(x_0, y_0) = (2, 1)$, $\epsilon = 10^{-5}$, $\tau = 0.5$, $s = 2$, $c_1 = 0.25$.
- The Gradient descent approach stops in 2 iterations and outputs the exact optimal solution.
- **For this example, inexact line search performs better than exact line search.**

```
iter_number =    1 norm_grad = 2.000000 fun_val = 1.000000
iter_number =    2 norm_grad = 0.000000 fun_val = 0.000000
```

Example 2: Gradient Descent with Backtracking Line Search on Quadratic Function

- Consider function $f(x, y) = x^2 + \frac{1}{100}y^2$, whose optimal solution is $(0, 0)$ with optimal value 0.
- Let $(x_0, y_0) = (\frac{1}{100}, 1)$, $\epsilon = 10^{-5}$, $\tau = 0.5$, $s = 2$, $c_1 = 0.25$.
- The Gradient descent approach stops in 201 iterations.

```
iter_number =   1 norm_grad = 0.028003 fun_val = 0.009704
iter_number =   2 norm_grad = 0.027730 fun_val = 0.009324
iter_number =   3 norm_grad = 0.027465 fun_val = 0.008958
      :                   :                  :
iter_number = 201 norm_grad = 0.000010 fun_val = 0.000000
```

Convergence of Steepest Gradient Descent

- For different quadratic functions, we observe that the convergence time varies for gradient descent.
- Can we find a measure that can predict how many iterations are needed for the convergence of the Gradient method?
- This measure would quantify, in some sense, the hardness of the problem.
- One such measure that can partially answer the above question is **condition number**.

Convergence of Steepest Gradient Descent with Exact Line Search for Quadratic Function

- Let $f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x}$, where A is a symmetric positive definite matrix.
- For Steepest descent, $\mathbf{d}_k = -\nabla f(\mathbf{x}_k) = -2A\mathbf{x}_k$.
- Exact line search will result in $t_k = \arg\min_{t\geq 0} \; f(\mathbf{x}_k + t\mathbf{d}_k) = \frac{\mathbf{d}_k^T \mathbf{d}_k}{2\mathbf{d}_k^T A\mathbf{d}_k}$. Using this, we get

$$
\begin{aligned}
f(\mathbf{x}_k + t_k\mathbf{d}_k) &= f(\mathbf{x}_k) + t_k^2 \mathbf{d}_k^T A\mathbf{d}_k + 2t_k \mathbf{d}_k^T A\mathbf{x}_k \\
&= \mathbf{x}_k^T A\mathbf{x}_k + \frac{(\mathbf{d}_k^T\mathbf{d}_k)^2}{4\mathbf{d}_k^T A\mathbf{d}_k} + \frac{\mathbf{d}_k^T\mathbf{d}_k}{2\mathbf{d}_k^T A\mathbf{d}_k}\mathbf{d}_k^T(-\mathbf{d}_k) \\
&= \mathbf{x}_k^T A\mathbf{x}_k - \frac{1}{4}\frac{(\mathbf{d}_k^T\mathbf{d}_k)^2}{\mathbf{d}_k^T A\mathbf{d}_k} = \mathbf{x}_k^T A\mathbf{x}_k \left(1 - \frac{1}{4}\frac{(\mathbf{d}_k^T\mathbf{d}_k)^2}{(\mathbf{d}_k^T A\mathbf{d}_k)(\mathbf{x}_k^T A\mathbf{x}_k)}\right) \\
&= \mathbf{x}_k^T A\mathbf{x}_k \left(1 - \frac{1}{4}\frac{(\mathbf{d}_k^T\mathbf{d}_k)^2}{(\mathbf{d}_k^T A\mathbf{d}_k)(\mathbf{x}_k^T AA^{-1}A\mathbf{x}_k)}\right) \\
&= \left(1 - \frac{(\mathbf{d}_k^T\mathbf{d}_k)^2}{(\mathbf{d}_k^T A\mathbf{d}_k)(\mathbf{d}_k^T A^{-1}\mathbf{d}_k)}\right) f(\mathbf{x}_k)
\end{aligned}
$$

Convergence of Steepest Gradient Descent with Exact Line Search for Quadratic Function

Kantorovich Inequality

Let A be a positive definite $n \times n$ matrix. Then for any $\mathbf{x} \in \mathbb{R}^n$ ($\mathbf{x} \neq \mathbf{0}$), the inequality

$$\frac{(\mathbf{x}^T\mathbf{x})^2}{(\mathbf{x}^TA\mathbf{x})(\mathbf{x}^TA^{-1}\mathbf{x})} \geq \frac{4\lambda_{max}(A)\lambda_{min}(A)}{(\lambda_{max}(A)+\lambda_{min}(A))^2}$$

holds.

Lemma

Let $\{\mathbf{x}_k\}_{k\geq 0}$ be the sequence generated by the gradient descent method with exact line search for finding the minimizer of $f(\mathbf{x}) = \mathbf{x}^TA\mathbf{x}$. Then, for any $k = 0, 1, 2, \ldots$

$$f(\mathbf{x}_{k+1}) \leq \left(\frac{M-m}{M+m}\right)^2 f(\mathbf{x}_k)$$

where $M = \lambda_{max}(A)$ and $m = \lambda_{min}(A)$.

Condition Number and Convergence

Condition Number

Let A be an $n \times n$ positive definite matrix. Then the **condition number** of A is defined as

$$\chi(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$$

- For quadratic functions with large condition numbers, the gradient method might require a large number of iterations to converge.
- Matrices with large condition numbers are called **ill conditioned**.
- Matrices with small condition numbers are called **well conditioned**.
- In the case of non-quadratic functions, the rate of convergence of $\mathbf{x}_k$ to a given stationary point $\mathbf{x}^*$ depends on the condition number of $\nabla^2 f(\mathbf{x}^*)$.

Example: Rosenbrock Function

- The Rosenbrock function is the following function

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- The optimal solution is $(1, 1)$ with the optimal value 0.
- The Rosenbrock function is extremely ill-conditioned at the optimal solution.

$$\nabla f(\mathbf{x}) = \begin{pmatrix} -400x_1(x_2 - x_1^2) - 2(1 - x_1) \\ 200(x_2 - x_1^2) \end{pmatrix}$$

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} -400x_2 + 1200x_1^2 + 2 & -400x_1 \\ -400x_1 & 200 \end{pmatrix}$$

- $(1, 1)$ is unique stationary point.
- $\nabla^2 f(1, 1) = \begin{pmatrix} 802 & -400 \\ -400 & 200 \end{pmatrix}$
- Condition number of $\nabla^2 f(1, 1)$ is 2.508×10^3

Example: Steepest Descent with Backtracking on Rosenbrock Function

- Starting point $\mathbf{x}_0 = [2, 5]^T$. The run required 6890 iterations. So, ill-conditioning of $\nabla^2 f(1, 1)$ has significant impact.

```
iter_number =    1 norm_grad = 118.254478 fun_val = 3.221022
iter_number =    2 norm_grad = 0.723051 fun_val = 1.496586
      :                :                :
iter_number = 6889 norm_grad = 0.000019 fun_val = 0.000000
iter_number = 6890 norm_grad = 0.000009 fun_val = 0.000000
```

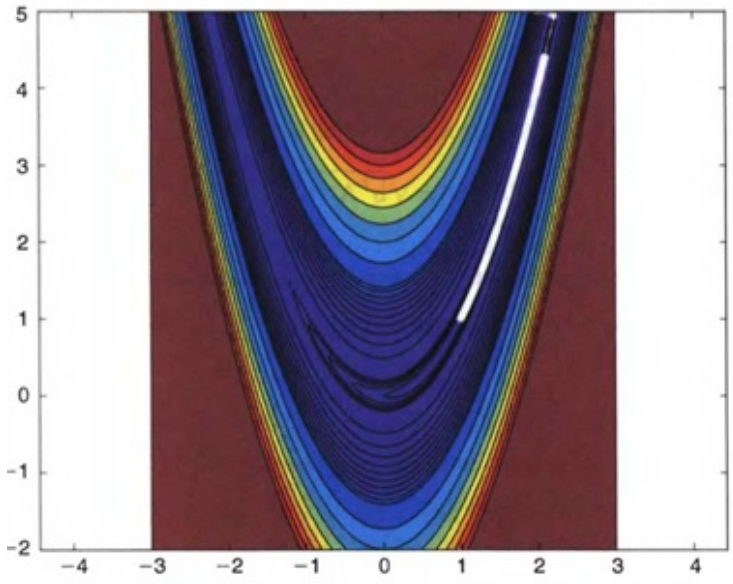


Figure: Banana-shaped contour lines of the Rosenbrock function surrounding the unique stationary point $(1, 1)$. Along with it thousands of iterations of steepest descent.

L-Smooth Functions

An L-smooth function is continuously differentiable and that its gradient ∇f is Lipschitz continuous over $\mathbb{R}^n$, meaning that there exists $L > 0$ for which

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \quad \text{for any } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n.$$

The class of functions with Lipschitz gradient with constant L are denoted by $\mathcal{C}_L^1$.

Examples:

- **Linear Functions:** Given $\mathbf{a} \in \mathbb{R}^n$, the function $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$ is in $\mathcal{C}_0^1$.
- **Quadratic Functions:** Let A be an $n \times n$ symmetric matrix, $\mathbf{b} \in \mathbb{R}^n$ and $c \in \mathbb{R}$. Then,

$$\|f(\mathbf{x}) - f(\mathbf{y})\| = 2\|A(\mathbf{x} - \mathbf{y})\| \leq 2\|A\|.\|\mathbf{x} - \mathbf{y}\|$$

Thus, the Lipschitz constant of ∇f is $2\|A\|$.

Interpretation of L-Smoothness

- The gradient of a function measures how the function changes when we move in a particular direction from a point.
- If the gradient were to change arbitrarily quickly, the old gradient does not give us much information at all, even if we take a small step.
- In contrast, smoothness assures us that the gradient cannot change too quickly. Therefore, we have an assurance that the gradient information is informative within a region around where it is taken. The implication is that we can decrease the function's value by moving in the direction opposite of the gradient.

Properties of L-Smooth Function

Theorem 4.20 (Chapter 4: Introduction to Nonlinear Optimization by Amir Beck)

Let f be a twice continuously differentiable function over $\mathbb{R}^n$. Then the following two claims are equivalent.

- $f \in \mathcal{C}_L^1(\mathbb{R}^n)$
- $\|\nabla^2 f(\mathbf{x})\| \leq L$ for any $\mathbf{x} \in \mathbb{R}^n$.

See the proof in the book.

Example: Let $f : \mathbb{R} \to \mathbb{R}$ be given by $f(x) = \sqrt{1+x^2}$. Then,

$$0 \leq f''(x) = \frac{1}{(1+x^2)^{3/2}} \leq 1$$

for any $x \in \mathbb{R}$. Thus, $f \in \mathcal{C}_1^1$.

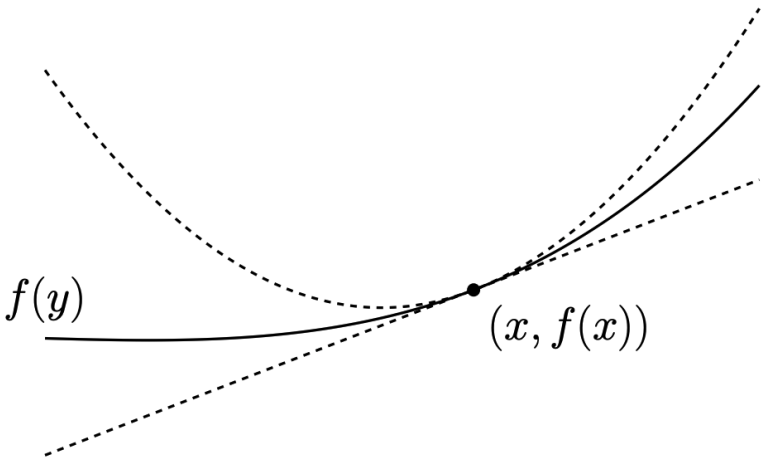
Descent Property of L-Smooth Functions

Lemma 4.22 (Chapter 4: Introduction to Nonlinear Optimization by Amir Beck)

Let $f \in C_L^1(\mathbb{R}^n)$ for some $L > 0$. Then, for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, it holds that

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|^2$$

See the proof in the book.



Comments:

1. This result shows that an L-smooth function can be bounded above by a quadratic function over the entire space.
2. This result is very useful in the convergence proofs of gradient-based methods.

Descent Property of Steepest Descent for L-Smooth Functions

Lemma (Sufficient Decrease of the Gradient Method)

Suppose that $f \in \mathcal{C}^1_L(\mathbb{R}^n)$. Let $\{\mathbf{x}_k\}_{k\geq 0}$ be the sequence generated by the gradient method for solving $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$ with one of the following stepsize strategies:

- constant stepsize $\bar{t} \in (0, \frac{2}{L})$
- exact line search
- backtracking procedure with parameters $s \in \mathbb{R}_{++}$, $\alpha \in (0,1)$, $\beta \in (0,1)$.

Then for any $\mathbf{x} \in \mathbb{R}^n$ and $t > 0$

$$f(\mathbf{x}) - f(\mathbf{x} - t\nabla f(\mathbf{x})) \geq M\|\nabla f(\mathbf{x})\|^2$$

where

$$M = \begin{cases} \bar{t}\left(1 - \frac{\bar{t}L}{2}\right), & \text{constant stepsize} \\ \frac{1}{2L}, & \text{exact line search} \\ \alpha \min\left\{s, \frac{2(1-\alpha)\beta}{L}\right\}, & \text{backtracking} \end{cases}$$

- Above result shows that at each iteration the decrease in the function value is at least a constant times the squared norm of the gradient.

Convergence of the Steepest Descent for L-Smooth Functions

Lemma (Sufficient Decrease of the Gradient Method)

Suppose that $f \in \mathbb{C}_L^{1,1}(\mathbb{R}^n)$. Let $\{\mathbf{x}_k\}_{k\geq 0}$ be the sequence generated by the gradient method for solving $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$ with one of the following stepsize strategies:

- constant stepsize $\bar{t} \in (0, \frac{2}{L})$
- exact line search
- backtracking procedure with parameters $s \in \mathbb{R}_{++}$, $\alpha \in (0,1)$, $\beta \in (0,1)$.

Assume that f is bounded below over $\mathbb{R}^n$, that is, there exists $m \in \mathbb{R}$ such that $f(\mathbf{x}) > m$ for all $\mathbf{x} \in \mathbb{R}^n$. Then we have the following:

1. The sequence $\{f(\mathbf{x}_k)\}_{k\geq 0}$ is non-increasing. In addition, for any $k \geq 0$, $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$ unless $\nabla f(\mathbf{x}_k) = \mathbf{0}$.
2. $\nabla f(\mathbf{x}_k) \to \mathbf{0}$ as $k \to \infty$.

Optimization Methods (CS1.404), Spring 2024
Lecture 12

Naresh Manwani

Machine Learning Lab, IIIT-H

February 19th, 2024

Backtracking Line Search With Armijo

Backtracking

1. **Initialize:** $\alpha^{(0)} \in (0,1)$, $\tau \in (0,1)$, $l = 0$
2. Until $f(\mathbf{x}_k + \alpha^{(l)}\mathbf{d}_k) > f(\mathbf{x}_k) + c_1\alpha^{(l)}\nabla f(\mathbf{x}_k)^T\mathbf{d}_k$
   1. Set $\alpha^{(l+1)} = \tau\alpha^{(l)}$
   2. $l = l + 1$
3. $\alpha_k = \alpha^{(l)}$

In practice the following choices are used

- $\tau \in (0.1, 0.5]$
- $c_1 \in [10^{-5}, 10^{-1}]$

Convergence of Steepest Gradient Descent with Exact Line Search for Quadratic Function

- Let $f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x}$, where A is a symmetric positive definite matrix.
- For Steepest descent, $\mathbf{d}_k = -\nabla f(\mathbf{x}_k) = -2A\mathbf{x}_k$.
- Exact line search will result in $t_k = \arg\min_{t\geq 0} \; f(\mathbf{x}_k + t\mathbf{d}_k) = \frac{\mathbf{d}_k^T\mathbf{d}_k}{2\mathbf{d}_k^T A\mathbf{d}_k}$. Using this, we get

$$
\begin{aligned}
f(\mathbf{x}_k + t_k\mathbf{d}_k) &= f(\mathbf{x}_k) + t_k^2\mathbf{d}_k^T A\mathbf{d}_k + 2t_k\mathbf{d}_k^T A\mathbf{x}_k \\
&= \mathbf{x}_k^T A\mathbf{x}_k + \frac{(\mathbf{d}_k^T\mathbf{d}_k)^2}{4\mathbf{d}_k^T A\mathbf{d}_k} + \frac{\mathbf{d}_k^T\mathbf{d}_k}{2\mathbf{d}_k^T A\mathbf{d}_k}\mathbf{d}_k^T(-\mathbf{d}_k) \\
&= \mathbf{x}_k^T A\mathbf{x}_k - \frac{1}{4}\frac{(\mathbf{d}_k^T\mathbf{d}_k)^2}{\mathbf{d}_k^T A\mathbf{d}_k} = \mathbf{x}_k^T A\mathbf{x}_k\left(1 - \frac{1}{4}\frac{(\mathbf{d}_k^T\mathbf{d}_k)^2}{(\mathbf{d}_k^T A\mathbf{d}_k)(\mathbf{x}_k^T A\mathbf{x}_k)}\right) \\
&= \mathbf{x}_k^T A\mathbf{x}_k\left(1 - \frac{1}{4}\frac{(\mathbf{d}_k^T\mathbf{d}_k)^2}{(\mathbf{d}_k^T A\mathbf{d}_k)(\mathbf{x}_k^T AA^{-1}A\mathbf{x}_k)}\right) \\
&= \left(1 - \frac{(\mathbf{d}_k^T\mathbf{d}_k)^2}{(\mathbf{d}_k^T A\mathbf{d}_k)(\mathbf{d}_k^T A^{-1}\mathbf{d}_k)}\right) f(\mathbf{x}_k)
\end{aligned}
$$

Convergence of Steepest Gradient Descent with Exact Line Search for Quadratic Function

Kantorovich Inequality

Let A be a positive definite $n \times n$ matrix. Then for any $\mathbf{x} \in \mathbb{R}^n$ $(\mathbf{x} \neq \mathbf{0})$, the inequality

$$\frac{(\mathbf{x}^T\mathbf{x})^2}{(\mathbf{x}^TA\mathbf{x})(\mathbf{x}^TA^{-1}\mathbf{x})} \geq \frac{4\lambda_{max}(A)\lambda_{min}(A)}{(\lambda_{max}(A)+\lambda_{min}(A))^2}$$

holds.

Lemma

Let $\{\mathbf{x}_k\}_{k\geq 0}$ be the sequence generated by the gradient descent method with exact line search for finding the minimizer of $f(\mathbf{x}) = \mathbf{x}^TA\mathbf{x}$. Then, for any $k = 0, 1, 2, \ldots$

$$f(\mathbf{x}_{k+1}) \leq \left(\frac{M-m}{M+m}\right)^2 f(\mathbf{x}_k)$$

where $M = \lambda_{max}(A)$ and $m = \lambda_{min}(A)$.

Condition Number and Convergence

Condition Number

Let A be an $n \times n$ positive definite matrix. Then the **condition number** of A is defined as

$$\chi(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$$

- For quadratic functions with large condition number, gradient method might require large number of iterations to converge.
- Matrices with large condition number are called **ill conditioned**.
- Matrices with small condition number are called **well conditioned**.
- In case of non-quadratic functions, the rate of convergence of $\mathbf{x}_k$ to a given stationary point $\mathbf{x}^*$ depend on the condition number of $\nabla^2 f(\mathbf{x}^*)$.

Example: Rosenbrock Function

- The Rosenbrock function is the following function

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

- The optimal solution is $(1, 1)$ with the optimal value 0.
- The Rosenbrock function is extremely ill conditioned at the optimal solution.

$$\nabla f(\mathbf{x}) = \begin{pmatrix} -400x_1(x_2 - x_1^2) - 2(1 - x_1) \\ 200(x_2 - x_1^2) \end{pmatrix}$$

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} -400x_2 + 1200x_1^2 + 2 & -400x_1 \\ -400x_1 & 200 \end{pmatrix}$$

- $(1, 1)$ is unique stationary point.
- $\nabla^2 f(1, 1) = \begin{pmatrix} 802 & -400 \\ -400 & 200 \end{pmatrix}$
- Condition number of $\nabla^2 f(1, 1)$ is 2.508×10^3

Example: Steepest Descent with Backtracking on Rosenbrock Function

- Starting point $\mathbf{x}_0 = [2, 5]^T$. The run required 6890 iterations. So, ill conditioning of $\nabla^2 f(1, 1)$ has significant impact.

```
iter_number =    1 norm_grad = 118.254478 fun_val = 3.221022
iter_number =    2 norm_grad = 0.723051 fun_val = 1.496586
      :                :                :
iter_number = 6889 norm_grad = 0.000019 fun_val = 0.000000
iter_number = 6890 norm_grad = 0.000009 fun_val = 0.000000
```

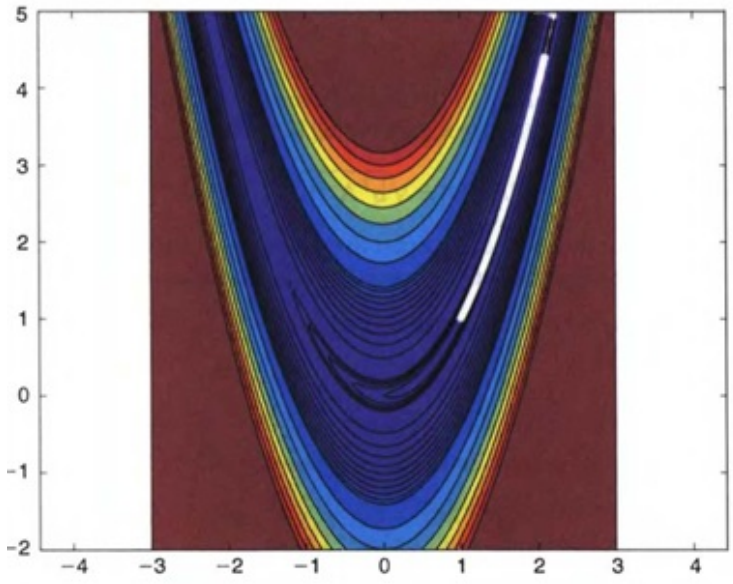


Figure: Banana shaped contour lines of the Rosenbrock function surrounding the unique stationary point $(1, 1)$. Along with it thousands of iterations of steepest descent.

Convergence Analysis of Gradient Descent

L-Smooth Functions

An L-smooth function is continuously differentiable and that its gradient ∇f is Lipschitz continuous over $\mathbb{R}^n$, meaning that there exists $L > 0$ for which

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \quad \text{for any } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n.$$

The class of functions with Lipschitz gradient with constant L are denoted by $\mathbb{C}_L^{1,1}$.

Examples:

- **Linear Functions:** Given $\mathbf{a} \in \mathbb{R}^n$, the function $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x}$ is in $\mathbb{C}_0^{1,1}$.
- **Quadratic Functions:** Let A be an $n \times n$ symmetric matrix, $\mathbf{b} \in \mathbb{R}^n$ and $c \in \mathbb{R}$. Then,

$$\|f(\mathbf{x}) - f(\mathbf{y})\| = 2\|A(\mathbf{x} - \mathbf{y})\| \leq 2\|A\|.\|\mathbf{x} - \mathbf{y}\|$$

Thus, the Lipschitz constant of ∇f is $2\|A\|$.

Interpretation of L-Smoothness

- The gradient of a functions measures how the function changes when we move in a particular direction from a point.
- If the gradient were to change arbitrarily quickly, the old gradient does not give us much information at all even if we take a small step.
- In contrast, smoothness assures us that the gradient cannot change too quickly. Therefore, we have an assurance that the gradient information is informative within a region around where it is taken. The implication is that we can decrease the function's value by moving the direction opposite of the gradient.

Properties of L-Smooth Function

Theorem 4.20 (Chapter 4: Introduction to Nonlinear Optimization by Amir Beck)

Let f be a twice continuously differentiable function over $\mathbb{R}^n$. Then the following two claims are equivalent.

- $f \in \mathbb{C}_L^{1,1}(\mathbb{R}^n)$
- $\|\nabla^2 f(\mathbf{x})\| \leq L$ for any $\mathbf{x} \in \mathbb{R}^n$.

See the proof in the book.

Example: Let $f : \mathbb{R} \to \mathbb{R}$ be given by $f(x) = \sqrt{1+x^2}$. Then,

$$0 \leq f''(x) = \frac{1}{(1+x^2)^{3/2}} \leq 1$$

for any $x \in \mathbb{R}$. Thus, $f \in \mathbb{C}_1^{1,1}$.

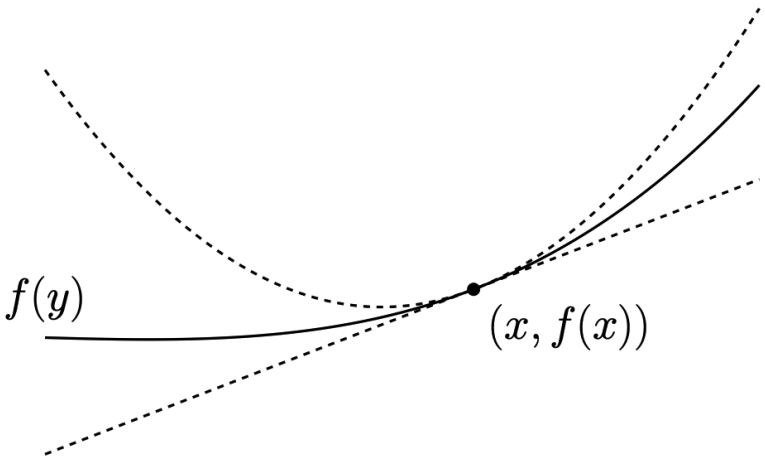
Descent Property of L-Smooth Functions

Lemma 4.22 (Chapter 4: Introduction to Nonlinear Optimization by Amir Beck)

Let $D \subseteq \mathbb{R}^n$ and $f \in \mathbb{C}_L^{1,1}(D)$ for some $L > 0$. Then, for any $\mathbf{x}, \mathbf{y} \in D$ satisfying $[\mathbf{x}, \mathbf{y}] \subseteq D$, it holds that

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|^2$$

See the proof in the book.



Comments:

1. This result shows that an L-smooth function can be bounded above by a quadratic function over the entire space.
2. This result is very useful in the convergence proofs of gradient based methods.

Descent Property of Steepest Descent for L-Smooth Functions

Lemma (Sufficient Decrease of the Gradient Method)

Suppose that $f \in \mathbb{C}_L^{1,1}(\mathbb{R}^n)$. Let $\{\mathbf{x}_k\}_{k\geq 0}$ be the sequence generated by the gradient method for solving $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$ with one of the following stepsize strategies:

- constant stepsize $\bar{t} \in (0, \frac{2}{L})$
- exact line search
- backtracking procedure with parameters $s \in \mathbb{R}_{++}$, $\alpha \in (0,1)$, $\beta \in (0,1)$.

Then for any $\mathbf{x} \in \mathbb{R}^n$ and $t > 0$

$$f(\mathbf{x}) - f(\mathbf{x} - t\nabla f(\mathbf{x})) \geq M\|\nabla f(\mathbf{x})\|^2$$

where

$$M = \begin{cases} \bar{t}\left(1 - \frac{\bar{t}L}{2}\right), & \text{constant stepsize} \\ \frac{1}{2L}, & \text{exact line search} \\ \alpha \min\left\{s, \frac{2(1-\alpha)\beta}{L}\right\}, & \text{backtracking} \end{cases}$$

- Above result shows that at each iteration the decrease in the function value is at least a constant times the squared norm of the gradient.

Convergence of the Steepest Descent for L-Smooth Functions

Lemma (Sufficient Decrease of the Gradient Method)

Suppose that $f \in \mathbb{C}_L^{1,1}(\mathbb{R}^n)$. Let $\{\mathbf{x}_k\}_{k\geq 0}$ be the sequence generated by the gradient method for solving $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$ with one of the following stepsize strategies:

- constant stepsize $\bar{t} \in (0, \frac{2}{L})$
- exact line search
- backtracking procedure with parameters $s \in \mathbb{R}_{++}$, $\alpha \in (0,1)$, $\beta \in (0,1)$.

Assume that f is bounded below over $\mathbb{R}^n$, that is, there exists $m \in \mathbb{R}$ such that $f(\mathbf{x}) > m$ for all $\mathbf{x} \in \mathbb{R}^n$. Then we have the following:

1. The sequence $\{f(\mathbf{x}_k)\}_{k\geq 0}$ is non-increasing. In addition, for any $k \geq 0$, $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$ unless $\nabla f(\mathbf{x}_k) = \mathbf{0}$.
2. $\nabla f(\mathbf{x}_k) \to \mathbf{0}$ as $k \to \infty$.

Optimization Methods (CS1.404), Spring 2025

Naresh Manwani

Machine Learning Lab, IIIT-H

March 5th, 2025

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

Diagonal Scaling to Improve Condition Number

- Consider the problem $\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} - \mathbf{c}^T\mathbf{x}$, where $\mathbf{H}$ is a symmetric positive definite matrix.
- Condition number of Hessian matrix controls the convergence rate of steepest descent.
- Faster convergence if Hessian is closer to a scalar multiple of the identity matrix.
- Can we transform the problem into another space in which the condition number of the Hessian becomes Identity?
- Let $\mathbf{H} = \mathbf{L}\mathbf{L}^T$ be the Cholesky decomposition of H.
- Define $\mathbf{y} = \mathbf{L}^T\mathbf{x}$.
- Consider the transformed function $h(\mathbf{y}) = f(\mathbf{L}^{-T}\mathbf{y})$.

Diagonal Scaling to Improve Condition Number

$$h(\mathbf{y}) = f(\mathbf{L}^{-T}\mathbf{y}) = \frac{1}{2}\mathbf{y}^T\mathbf{L}^{-1}\mathbf{H}\mathbf{L}^{-T}\mathbf{y} - \mathbf{c}^T(\mathbf{L}^{-T}\mathbf{y})$$
$$= \frac{1}{2}\mathbf{y}^T\mathbf{L}^{-1}\mathbf{L}\mathbf{L}^T\mathbf{L}^{-T}\mathbf{y} - \mathbf{c}^T(\mathbf{L}^{-T}\mathbf{y})$$
$$= \frac{1}{2}\mathbf{y}^T\mathbf{y} - \mathbf{c}^T(\mathbf{L}^{-T}\mathbf{y})$$

- The hessian of $h(\mathbf{y})$ is identity matrix.
- Let us apply steepest descent on $\mathbf{y}$ space.

$$\mathbf{y}^{k+1} = \mathbf{y}^k - \nabla h(\mathbf{y}^k) = \mathbf{y}^k - \mathbf{L}^{-1}\nabla f(\mathbf{L}^{-T}\mathbf{y}^k)$$

- Applying transformation $\mathbf{L}^{-T}$ on both sides

$$\mathbf{L}^{-T}\mathbf{y}^{k+1} = \mathbf{L}^{-T}\mathbf{y}^k - \mathbf{L}^{-T}\mathbf{L}^{-1}\nabla f(\mathbf{L}^{-T}\mathbf{y}^k)$$
$$\Rightarrow \quad \mathbf{x}^{k+1} = \mathbf{x}^k - \mathbf{H}^{-T}\nabla f(\mathbf{x}^k) = \mathbf{x}^k - \mathbf{H}^{-1}\nabla f(\mathbf{x}^k)$$

- This method is called **Newton Method**.

Newton Method

- Consider $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$, where $f \in \mathcal{C}^2(\mathbb{R}^n)$.
- Newton's method used second-order information to determine the descent direction.
- At each iteration, it uses a second-order Taylor series approximation of f at $\mathbf{x}_k$ and finds the minimum of it to get $\mathbf{x}_{k+1}$.

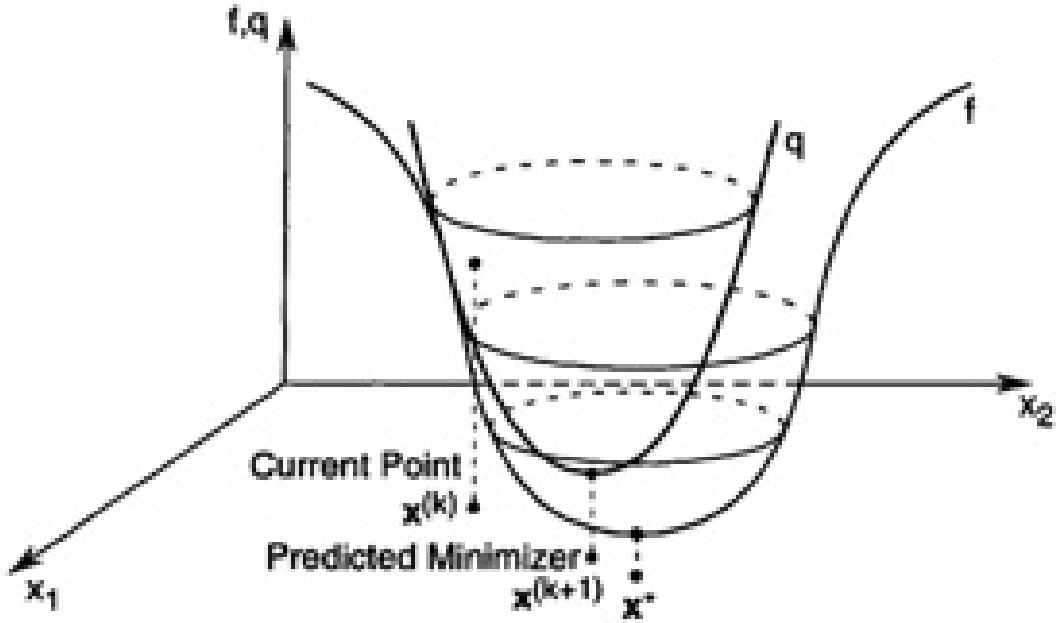
$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \left\{ f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T(\mathbf{x}-\mathbf{x}_k) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_k)^T\nabla^2 f(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k) \right\}$$

- The above formula is well defined only if we further assume that $\nabla^2 f(\mathbf{x}_k)$ is positive definite. Under this assumption, the unique minimizer is

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\nabla^2 f(\mathbf{x}_k))^{-1}\nabla f(\mathbf{x}_k)$$

- **Newton Direction:** $\mathbf{d}_N = -(\nabla^2 f(\mathbf{x}_k))^{-1}\nabla f(\mathbf{x}_k)$.

Geometry of Newton Method



Pure Newton Method

- **Input:** $\epsilon > 0$ -tolerance parameter
- **Initialization:** Pick $\mathbf{x}_0 \in \mathbb{R}^n$ arbitrarily
- **General Step :** For any $k = 0, 1, 2, \ldots$ execute the following steps:
  1. Compute the Newton's direction, which is the solution to the linear system: $\nabla^2 f(\mathbf{x}_k)\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$.
  2. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k$
  3. If $\|\nabla f(\mathbf{x}_{k+1})\| \leq \epsilon$, then STOP and output $\mathbf{x}_{k+1}$.

Convergence of Newton Method for Quadratic Functions

- Newton method requires that $\nabla^2 f(\mathbf{x})$ is positive definite for every $\mathbf{x}$ (strict convexity).
- Consider quadratic function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} - \mathbf{c}^T\mathbf{x}$ such that matrix $\mathbf{H}$ is real symmetric and positive definite matrix.
- We know that the unique global minimizer of f is $\mathbf{x}^* = \mathbf{H}^{-1}\mathbf{c}$.
- We see that $\nabla f(\mathbf{x}) = \mathbf{H}\mathbf{x} - \mathbf{c}$ and $\nabla^2 f(\mathbf{x}) = \mathbf{H}$.
- Applying Newton method on this function for $\mathbf{x}_0$ as initial point, we see that

$$\mathbf{x}_1 = \mathbf{x}_0 - \nabla^2 f(\mathbf{x}_0)^{-1}\nabla f(\mathbf{x}_0) = \mathbf{x}_0 - \mathbf{H}^{-1}(\mathbf{H}\mathbf{x}_0 - \mathbf{c}) = \mathbf{H}^{-1}\mathbf{c}$$

- Thus, using Newton's method, we reach to the global minima of a quadratic and strictly convex function in one step.

Convergence of Newton Method for General Functions

- Newton method requires that $\nabla^2 f(\mathbf{x})$ is positive definite for every $\mathbf{x}$ (strict convexity).
- Which implies a unique optimal solution $\mathbf{x}^*$ exists.
- However, this is not enough to guarantee convergence.
- Consider the following example.

Example

- Consider the function $f(x) = \sqrt{1+x^2}$. The minimizer of f is $x = 0$.
- $f'(x) = \frac{x}{\sqrt{1+x^2}}$, $f''(x) = \frac{1}{(1+x^2)^{3/2}}$.
- Therefore, the Pure Newton method update equations are

$$x_{k+1} = x_k - (1+x_k^2)^{3/2} \frac{x_k}{\sqrt{1+x_k^2}} = x_k - x_k(1+x_k^2) = -x_k^3$$

- Newton method converges to $x^* = 0$ when $|x_0| < 1$. For $|x_0| > 1$, it diverges.

Quadratic Local Convergence of Newton's Method

Theorem

Let f be a twice continuously differentiable function defined over $\mathbb{R}^n$. Assume that

- There exists $m > 0$ for which $\nabla^2 f(\mathbf{x}) \succeq m\mathbf{I}$ for any $\mathbf{x} \in \mathbb{R}^n$,
- There exists $L > 0$ for which $\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$ for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

Let $\{\mathbf{x}_k\}_{k \geq 0}$ be the sequence generated by Newton's Method, and let $\mathbf{x}^*$ be the unique minimizer of f over $\mathbb{R}^n$. Then for any $k = 0, 1, 2, \ldots$ the inequality

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| \leq \frac{L}{2m}\|\mathbf{x}_k - \mathbf{x}^*\|^2$$

holds. In addition, if $\|\mathbf{x}^* - \mathbf{x}_0\| \leq \frac{m}{L}$, then

$$\|\mathbf{x}^* - \mathbf{x}_k\| \leq \frac{2m}{L}\left(\frac{1}{2}\right)^{2^k}, \ k = 0, 1, 2, \ldots$$

- Thus, near the optimal solution, the error $e_k = \|\mathbf{x}^* - \mathbf{x}_k\|$ satisfies the inequality $e_{k+1} \leq Me_k^2$ for some positive $M > 0$.

Example 2: $\nabla f(\mathbf{x}) \succeq m\mathbf{I}$ not satisfied

- Consider the problem $\min_{x_1,x_2} \sqrt{1+x_1^2} + \sqrt{1+x_2^2}$. The optimal solution is $(0,0)$.
- Hessian of the function is $\nabla^2 f(\mathbf{x}) = \begin{pmatrix} \frac{1}{(1+x_1^2)^{3/2}} & 0 \\ 0 & \frac{1}{(1+x_2^2)^{3/2}} \end{pmatrix} \succeq \mathbf{0}$.
- Even though the Hessian is positive definite, there does not exist an $m > 0$ for which $\nabla^2 f(\mathbf{x}) \succeq m\mathbf{I}$. As $x_1, x_2 \to \infty$, $\nabla^2 f(\mathbf{x})$ becomes a zero matrix.
- Basic assumption for convergence is not satisfied.
- This is reflected in implementation also.
- Newton's method with initial vector $\mathbf{x}_0 = (1,1)$ and tolerance parameter $\epsilon = 10^{-8}$ we obtain convergence after 37 iterations.
- Newton's method with initial vector $\mathbf{x}_0 = (10,10)$ diverges.

```
iter=  1 f(x)=2.8284271247
iter=  2 f(x)=2.8284271247
   :                :
iter= 30 f(x)=2.8105247315
iter= 31 f(x)=2.7757389625
iter= 32 f(x)=2.6791717153
iter= 33 f(x)=2.4507092918
iter= 34 f(x)=2.1223796622
iter= 35 f(x)=2.0020052756
iter= 36 f(x)=2.0000000081
iter= 37 f(x)=2.0000000000
```

(a) Starting point (1,1). Not much progress in 30 iterations. Converges in 37 iterations.

```
iter=  1 f(x)=2000.0009999997
iter=  2 f(x)=1999999999.9999990000
iter=  3 f(x)=19999999999999973000000000.0000000
iter=  4 f(x)=199999999999999230000000000000000000....
iter=  5 f(x)=       Inf
```

(b) Starting point (10,10). Newton's method diverges.

Issues with the Newton Method

- Requires computing inverse of hessian in each iteration. It can be computationally intensive if the number of variables is large.
- No guarantee that $\mathbf{d}_N = -\nabla^2 f(\mathbf{x})^{-1} \nabla f(\mathbf{x})$ is descent direction as the algorithm does not check if the hessian is positive definite.
- Problem happens when hessian is singular in some iteration.
- No guarantee that the function value decreases in each iteration (as no line search is used).
- Convergence is sensitive to the initial point.

Newton method with Backtracking Line Search

Damped Newton Method

- **Input:** $\alpha, \beta \in (0,1)$ - parameters for the backtracking procedure. $\epsilon > 0$ -tolerance parameter
- **Initialization:** Pick $\mathbf{x}_0 \in \mathbb{R}^n$ arbitrarily
- **General Step :** For any $k = 0, 1, 2, \ldots$ execute the following steps:
  1. Compute the Newton's direction, which is the solution to the linear system: $\nabla^2 f(\mathbf{x}_k)\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$.
  2. Set $t_k = 1$. While,
  $$f(\mathbf{x}_k) - f(\mathbf{x}_k + t_k\mathbf{d}_k) < -\alpha t_k \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$$
  Set $t_k = \beta t_k$.
  3. $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k$
  4. If $\|\nabla f(\mathbf{x}_{k+1})\| \leq \epsilon$, then STOP and output $\mathbf{x}_{k+1}$.

One can also use other step-size selection methods.

Newton Method with Backtracking on Example 2

- Consider the problem $\min_{x_1,x_2} \sqrt{1+x_1^2} + \sqrt{1+x_2^2}$. The optimal solution is $(0,0)$.
- Take initial point (10,10).
- Using backtracking line search with $\alpha = \beta = 0.5$ and $\epsilon = 10^{-8}$ Newton method converges in 17 iterations.

Levenberg Marquardt Algorithm

- If the hessian matrix $\nabla^2 f(\mathbf{x}_k)^{-1}$ is not positive definite, the Newton direction $\mathbf{d}_N = -\nabla^2 f(\mathbf{x})^{-1} \nabla f(\mathbf{x})$ may not remain a descent direction.
- This issue can be resolved by updating the Newton update in the following way.

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (\nabla^2 f(\mathbf{x}_k) + \mu_k \mathbf{I})^{-1} \nabla f(\mathbf{x}_k)$$

  where $\mu_k \geq 0$.
- The idea is as follows.
  - Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of $\nabla^2 f(\mathbf{x}_k)$ and $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be the corresponding eigenvectors.
  - If $\nabla^2 f(\mathbf{x}_k)$ is not positive definite, then some of the eigenvalues of it are negative.
  - Matrix $\nabla^2 f(\mathbf{x}_k) + \mu_k \mathbf{I}$ has eigenvalues $\lambda_1 + \mu_k, \ldots, \lambda_n + \mu_k$ with $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be the corresponding eigenvectors.
  - if μ_k is chosen sufficiently large, all eigenvalues of $\nabla^2 f(\mathbf{x}_k) + \mu_k \mathbf{I}$ can become positive.
  - In that case $-(\nabla^2 f(\mathbf{x}_k) + \mu_k \mathbf{I})^{-1} \nabla f(\mathbf{x}_k)$ becomes a descent direction.

Choosing μ_k

Choosing μ_k

1. Start with some μ_k (a small value)
2. Do the Cholesky factorization of $\nabla^2 f(\mathbf{x}_k) + \mu_k \mathbf{I}$.
3. If Unsuccessful, increase the value of μ_k and go to step 2,

- If μ_k is very large, then this method becomes same as steepest descent.
- If μ_k is very small, then this method becomes same as Newton method.

Levenberg Marquardt Algorithm

- **Input:** Tolerance parameter $\epsilon > 0$, lower bound on minimum eigenvalue $\delta > 0$
- **Initialization:** Pick $\mathbf{x}_0 \in \mathbb{R}^n$ arbitrarily. Set $k = 0$.
- While $(\|\nabla f(\mathbf{x}_k)\| > \epsilon)$
  1. Find the smallest $\mu_k \geq 0$ such that the smallest eigenvalue of $\nabla^2 f(\mathbf{x}_k) + \mu_k \mathbf{I}$ is greater than δ.
  2. Set $\mathbf{d}_k = -(\nabla^2 f(\mathbf{x}_k) + \mu_k \mathbf{I})^{-1} \nabla f(\mathbf{x}_k)$
  3. Find $\alpha_k > 0$ using backtracking
  4. Update $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
  5. $k = k + 1$
- **Output:** $\mathbf{x}^* = \mathbf{x}_k$ as stationary point of $f(\mathbf{x})$.

Need for Cholesky Factorization

- In Levenberg Marquardt algorithm, it is required to validate the positive definiteness of the matrix $\nabla^2 f(\mathbf{x}_k) + \mu_k \mathbf{I}$.
- Another issue is to solve the equation $\nabla^2 f(\mathbf{x}_k)\mathbf{d} = -\nabla f(\mathbf{x}_k)$ in general for Newton method.
- These two issues are resolved using Cholesky factorization.

Solving $\mathbf{Ax} = \mathbf{b}$ using Cholesky Factorization

- Let $\mathbf{A}$ be $n \times n$ positive definite matrix. Cholesky factorization of $\mathbf{A}$ has the form $\mathbf{A} = \mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is a lower triangular $n \times n$ matrix whose diagonal is positive
- Given the Cholesky factorization, equation $\mathbf{Ax} = \mathbf{b}$ can be solved in following two steps.
  - Find the solution $\mathbf{u}$ of $\mathbf{Lu} = \mathbf{b}$
  - Find the solution $\mathbf{x}$ of $\mathbf{L}^T\mathbf{x} = \mathbf{u}$.

Optimization Methods (CS1.404), Spring 2025

Naresh Manwani

Machine Learning Lab, IIIT-H

March 13th, 2025

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

Need for Cholesky Factorization

- In Levenberg Marquardt algorithm, it is required to validate the positive definiteness of the matrix $\nabla^2 f(\mathbf{x}_k) + \mu_k \mathbf{I}$.
- Another issue is to solve the equation $\nabla^2 f(\mathbf{x}_k)\mathbf{d} = -\nabla f(\mathbf{x}_k)$ in general for Newton method.
- These two issues are resolved using Cholesky factorization.

Solving $\mathbf{Ax} = \mathbf{b}$ using Cholesky Factorization

- Let $\mathbf{A}$ be $n \times n$ positive definite matrix. Cholesky factorization of $\mathbf{A}$ has the form $\mathbf{A} = \mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is a lower triangular $n \times n$ matrix whose diagonal is positive
- Given the Cholesky factorization, equation $\mathbf{Ax} = \mathbf{b}$ can be solved in following two steps.
  - Find the solution $\mathbf{u}$ of $\mathbf{Lu} = \mathbf{b}$
  - Find the solution $\mathbf{x}$ of $\mathbf{L}^T\mathbf{x} = \mathbf{u}$.

Cholesky Factorization Algorithm

- The computation of Cholesky factorization is done using a simple recursive approach.
- Consider the following block matrix partitioning of the matrices A and L.

$$\mathbf{A} = \begin{pmatrix} A_{11} & \mathbf{A}_{21} \\ \mathbf{A}_{12} & \mathbf{A}_{22} \end{pmatrix} \qquad \mathbf{L} = \begin{pmatrix} L_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{pmatrix}$$

where $A_{11} \in \mathbb{R}, \mathbf{A}_{21} \in \mathbb{R}^{(n-1)\times 1}, \mathbf{A}_{22} \in \mathbb{R}^{(n-1)\times(n-1)}, L_{11} \in \mathbb{R}$, $\mathbf{L}_{21} \in \mathbb{R}^{(n-1)\times 1}, \mathbf{L}_{22} \in \mathbb{R}^{(n-1)\times(n-1)}$.

- Since $\mathbf{A} = \mathbf{L}\mathbf{L}^T$, we have

$$\begin{pmatrix} A_{11} & \mathbf{A}_{21} \\ \mathbf{A}_{12} & \mathbf{A}_{22} \end{pmatrix} = \begin{pmatrix} L_{11}^2 & L_{11}\mathbf{L}_{21}^T \\ L_{11}\mathbf{L}_{21} & \mathbf{L}_{21}\mathbf{L}_{21}^T + \mathbf{L}_{22}\mathbf{L}_{22}^T \end{pmatrix}$$

Cholesky Factorization Algorithm: Continue

- Therefore, in particular $L_{11} = \sqrt{A_{11}}$, $\mathbf{L}_{21} = \frac{1}{\sqrt{A_{11}}}\mathbf{A}_{12}^T$.
- $\mathbf{L}_{22}\mathbf{L}_{22}^T = \mathbf{A}_{22} - \mathbf{L}_{21}\mathbf{L}_{21}^T = \mathbf{A}_{22} - \frac{1}{A_{11}}\mathbf{A}_{12}^T\mathbf{A}_{12}$.
- We are left with the task of Cholesky factorization of $(n-1) \times (n-1)$ matrix $\mathbf{A}_{22} - \frac{1}{A_{11}}\mathbf{A}_{12}^T\mathbf{A}_{12}$.
- We keep following the above procedure and we can get the complete Cholesky factorization.
- The algorithm for Cholesky factorization will find a solution only if all the diagonal elements l_{ii} that are computed during the process are positive, so that computing their square root is possible.
- The positiveness of these elements is equivalent to the property that the matrix to be factored is positive definite.
- Therefore, the Cholesky factorization process can be viewed as a criteria for positive definiteness.

Coordinate Descent Method

- Consider the problem $\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$, where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function.
- Coordinate descent method works as follows.
  1. For every coordinate variable x_i, $i \in \{1, \ldots, n\}$, minimize $f(\mathbf{x})$ with respect to x_i, keeping other variables x_j, $j \neq i$ constant.
  2. Repeat the above process in step 1 until some stopping condition is satisfied.

Coordinate Descent Method

Algorithm

- **Input:** $\epsilon > 0$ (tolerance parameter)
- **Initialize:** $\mathbf{x}_1$, $k = 1$
- **Step 1:** Set $\mathbf{d}_k = \mathbf{e}_k$, where $\mathbf{e}_k$ is k^{th} basis vector of standard basis of $\mathbb{R}^n$
- **Step 2:** Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$, where $\alpha_k = \arg\min_{\alpha \in \mathbb{R}} f(\mathbf{x}_k + \alpha \mathbf{d}_k)$.
- **Step 3:**
  - If $(\|\nabla f(\mathbf{x}_k)\| \leq \epsilon)$
    - then output $\mathbf{x}^* = \mathbf{x}_k$
  - Else if $(k = n)$
    - Set $\mathbf{x}_1 = \mathbf{x}_{k+1}$ and repeat from Step 2.
  - Else,
    - Set $k = k + 1$ and repeat from Step 2.

Coordinate Descent Method on Quadratic Functions

- **For convex quadratic functions of n variables, above algorithm converges in n-steps.**
  - Example 1: $\min_{\mathbf{x}\in\mathbb{R}^2}\ 4x_1^2 + x_2^2$ (spherical contours). Take $\mathbf{x}_0 = (-1, -1)$. Coordinate descent method finds minimizer in two steps.
  - Example 2: $\min_{\mathbf{x}\in\mathbb{R}^2}\ 4x_1^2 + x_2^2 - 2x_1x_2$ (elliptical contours) . Take $\mathbf{x}_0 = (-1, -1)$. Coordinate descent method does not converge in two steps.
- In other words, when the objective function is separable in terms of variables (hessian is diagonal), then coordinate descent method will find $\mathbf{x}^*$ in n-steps if there are n-variables.
- When objective function is not separable in variables, then Hessian is not diagonal. Coordinate descent method will not find minimizer in n-steps.
- Can we choose $\mathbf{d}_1, \ldots, \mathbf{d}_n$ in such a way that it converges in n-steps?

Conjugate Directions

Definition

Let Q be a real symmetric $n \times n$ matrix. The directions $\mathbf{d}_0, \ldots, \mathbf{d}_{n-1}$ are Q-conjugate if, for all $i \neq j$, we have $\mathbf{d}_i^T Q \mathbf{d}_j = 0$.

Lemma

Let Q be a symmetric positive definite $n \times n$ matrix. Let directions $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_k \in \mathbb{R}^n$ $(k \leq n-1)$ are Q-conjugate, then they are linearly independent.

Conjugate Directions: Example 1

- Let $f(x_1, x_2) = 4x_1^2 + x_2^2 - 2x_1x_2$
- Hessian $H = \begin{pmatrix} 8 & -2 \\ -2 & 2 \end{pmatrix}$
- Let $\mathbf{d}_0 = (1, 0)^T$
- Then the conjugate direction $\mathbf{d}_1 = (a, b)^T$ would satisfy $\mathbf{d}_0^T H \mathbf{d}_1 = 0$.
- This results in relation $8a - 2b = 0$. Thus, we can take $\mathbf{d}_1 = (1, 4)^T$.

Conjugate Directions: Example 2

- Let $Q = \begin{pmatrix} 3 & 0 & 1 \\ 0 & 4 & 2 \\ 1 & 2 & 3 \end{pmatrix}$
- Let $\mathbf{d}_0 = (1, 0, 0)^T$.
- Now, we want to find $\mathbf{d}_1 = (a, b, c)^T$ which is Q-conjugate to $\mathbf{d}_0$. We require $\mathbf{d}_0^T H \mathbf{d}_1 = 0$. Which results in relation $3a + c = 0$. So, we can choose $\mathbf{d}_1 = (1, 0, -3)^T$.
- Now, we want to find $\mathbf{d}_2 = (e, f, g)^T$ which is Q-conjugate to $\mathbf{d}_0$ and $\mathbf{d}_1$. So, we get the conditions $3e + g = 0$ and $-6f - 8g = 0$. We can choose $\mathbf{d}_2 = (1, 4, -3)^T$.

Choosing Conjugate Directions

- A systematic procedure for finding Q-conjugate directions can be developed using the idea of Gram-Schmidt algorithm of transforming a given basis of $\mathbb{R}^n$ into an orthogonal basis of $\mathbb{R}^n$.
- For a symmetric matrix matrix H, orthogonal eigenvectors of H itself are H-conjugate.
  - Let $\mathbf{v}_1$ and $\mathbf{v}_2$ are mutually orthonormal eigenvectors of H corresponding to eigenvalues λ_1 and λ_2.
  - Then $\mathbf{v}_1^T H \mathbf{v}_2 = \lambda_2 \mathbf{v}_1^T \mathbf{v}_2 = 0$.

Conjugate Direction Method

- Consider minimization problem $\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\mathbf{x}^T H\mathbf{x} + \mathbf{c}^T\mathbf{x}$, where H is symmetric positive definite matrix.
- Let $\mathbf{x}_0$ be the initial parameters.
- Let $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{n-1} \in \mathbb{R}^n$ be H-conjugate directions.
- As we know that these conjugate directions are linearly independent. We can write any $\mathbf{x} - \mathbf{x}_0 \in \mathbb{R}^n$ as a linear combination of these conjugate directions. Thus,

$$\mathbf{x} - \mathbf{x}_0 = \sum_{i=0}^{n-1} \alpha_i \mathbf{d}_i$$

Conjugate Direction Method - Continue

- Given $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{n-1} \in \mathbb{R}^n$ and $\mathbf{x}_0 \in \mathbb{R}^n$, the above minimization problem becomes

$$\begin{aligned}
\phi(\boldsymbol{\alpha}) &= \frac{1}{2}\left(\mathbf{x}_0 + \sum_{i=0}^{n-1}\alpha_i\mathbf{d}_i\right)^T H\left(\mathbf{x}_0 + \sum_{i=0}^{n-1}\alpha_i\mathbf{d}_i\right) + \mathbf{c}^T\left(\mathbf{x}_0 + \sum_{i=0}^{n-1}\alpha_i\mathbf{d}_i\right) \\
&= \frac{1}{2}\mathbf{x}_0^T H\mathbf{x}_0 + \frac{1}{2}\left(\sum_{i=0}^{n-1}\alpha_i\mathbf{d}_i\right)^T H\left(\sum_{i=0}^{n-1}\alpha_i\mathbf{d}_i\right) + \left(\sum_{i=0}^{n-1}\alpha_i\mathbf{d}_i\right) H\mathbf{x}_0 \\
&\quad + \mathbf{c}^T\left(\mathbf{x}_0 + \sum_{i=0}^{n-1}\alpha_i\mathbf{d}_i\right) \\
&= \frac{1}{2}\mathbf{x}_0^T H\mathbf{x}_0 + \frac{1}{2}\sum_{i=0}^{n-1}\alpha_i^2\mathbf{d}_i^T H\mathbf{d}_i + \sum_{i=0}^{n-1}\alpha_i\mathbf{d}_i^T H\mathbf{x}_0 + \mathbf{c}^T\left(\mathbf{x}_0 + \sum_{i=0}^{n-1}\alpha_i\mathbf{d}_i\right) \\
&= \sum_{i=0}^{n-1}\left(\frac{1}{2}(\mathbf{x}_0 + \alpha_i\mathbf{d}_i)^T H(\mathbf{x}_0 + \alpha_i\mathbf{d}_i) + \mathbf{c}^T(\mathbf{x}_0 + \alpha_i\mathbf{d}_i)\right) \\
&\quad - (n-1)\left(\frac{1}{2}\mathbf{x}_0^T H\mathbf{x}_0 + \mathbf{c}^T\mathbf{x}_0\right)
\end{aligned}$$

Conjugate Direction Method - Continue

- Ignoring the constant term, we define a new function

$$\psi(\boldsymbol{\alpha}) = \sum_{i=0}^{n-1} \left(\frac{1}{2}(\mathbf{x}_0 + \alpha_i \mathbf{d}_i)^T H(\mathbf{x}_0 + \alpha_i \mathbf{d}_i) + \mathbf{c}^T(\mathbf{x}_0 + \alpha_i \mathbf{d}_i) \right)$$

- ψ is separable in terms of $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$, which are our new optimization variables.
- Minimizing ψ with respect to α_i, we get

$$\alpha_i^* = -\frac{\mathbf{d}_i^T(H\mathbf{x}_0 + \mathbf{c})}{\mathbf{d}_i^T H \mathbf{d}_i}$$

- $\mathbf{x}^* = \mathbf{x}_0 + \sum_{i=0}^{n-1} \alpha_i^* \mathbf{d}_i$

Basic Conjugate Direction Algorithm

Given starting point $\mathbf{x}_0$ and H conjugate directions $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{n-1} \in \mathbb{R}^n$, the Conjugate Direction Algorithm works as follows:

- For $(k = 0, 1, \ldots, n-1)$
  - $\nabla f(\mathbf{x}_k) = H\mathbf{x}_k + \mathbf{c}$
  - $\alpha_k = -\frac{\nabla f(\mathbf{x}_k)^T \mathbf{d}_k}{\mathbf{d}_k^T H \mathbf{d}_k}$
  - $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$

Theorem

Consider minimization problem $\min_{\mathbf{x} \in \mathbb{R}^n} \ \frac{1}{2}\mathbf{x}^T H \mathbf{x} + \mathbf{c}^T \mathbf{x}$, where H is symmetric positive definite matrix. For any starting point $\mathbf{x}_0$ and H conjugate directions $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{n-1} \in \mathbb{R}^n$, the **Basic Conjugate Direction Algorithm** converges to the unique $\mathbf{x}^*$ (that solves $H\mathbf{x}^* + \mathbf{c} = \mathbf{0}$) in n-steps; that is $\mathbf{x}_n = \mathbf{x}^*$.

Optimization Methods (CS1.404), Spring 2024
Lecture 14

Naresh Manwani

Machine Learning Lab, IIIT-H

March 4th, 2024

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Basic Conjugate Direction Algorithm

Given starting point $\mathbf{x}_0$ and H conjugate directions $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{n-1} \in \mathbb{R}^n$, the Conjugate Direction Algorithm works as follows:

- For $(k = 0, 1, \ldots, n-1)$
  - $\nabla f(\mathbf{x}_k) = H\mathbf{x}_k + \mathbf{c}$
  - $\alpha_k = -\frac{\nabla f(\mathbf{x}_k)^T \mathbf{d}_k}{\mathbf{d}_k^T H \mathbf{d}_k}$
  - $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$

Theorem

Consider minimization problem $\min_{\mathbf{x} \in \mathbb{R}^n} \ \frac{1}{2}\mathbf{x}^T H \mathbf{x} + \mathbf{c}^T \mathbf{x}$, where H is symmetric positive definite matrix. For any starting point $\mathbf{x}_0$ and H conjugate directions $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{n-1} \in \mathbb{R}^n$, the **Basic Conjugate Direction Algorithm** converges to the unique $\mathbf{x}^*$ (that solves $H\mathbf{x}^* + \mathbf{c} = \mathbf{0}$) in n-steps; that is $\mathbf{x}_n = \mathbf{x}^*$.

Basic Conjugate Algorithm: Example

- $f(x_1, x_2) = 4x_1^2 + x_2^2 - 2x_1x_2$
- **Step 1:** $\mathbf{d}_0 = (1, 0)^T$, $\mathbf{x}_0 = (-1, -1)^T$
  - Find $x_1 = x_0 + \alpha_0 \mathbf{d}_0$ where $\alpha_0 = \arg\min_{\alpha>0} \ f(\mathbf{x}_0 + \alpha \mathbf{d}_0)$
  - Let $\phi(\alpha) = f(\mathbf{x}_0 + \alpha \mathbf{d}_0) = f(-1 + \alpha, -1) = 4(\alpha - 1)^2 + 1 + 2(\alpha - 1)$
  - $\phi'(\alpha) = 0 \Rightarrow 8(\alpha - 1) + 2 = 0 \Rightarrow \alpha_0 = \frac{3}{4}$
  - $\mathbf{x}_1 = \mathbf{x}_0 + \alpha_0 \mathbf{d}_0 = (-1, -1)^T + \frac{3}{4}(1, 0)^T = (-\frac{1}{4}, -1)^T$.
- **Step 2:** Choosing $\mathbf{d}_1 = (1, 4)^T$, as it becomes H-conjugate for $\mathbf{d}_0$.
  - Find $x_2 = x_1 + \alpha_1 \mathbf{d}_1$ where $\alpha_1 = \arg\min_{\alpha>0} \ f(\mathbf{x}_1 + \alpha \mathbf{d}_1)$
  - Let $\phi(\alpha) = f(\mathbf{x}_1 + \alpha \mathbf{d}_1) = f(-\frac{1}{4} + \alpha, -1 + 4\alpha) =$ $4(\alpha - \frac{1}{4})^2 + (4\alpha - 1)^2 - 2(\alpha - \frac{1}{4})(4\alpha - 1) = \frac{3}{4}(4\alpha - 1)^2$
  - $\phi'(\alpha) = 0 \Rightarrow \alpha_1 = \frac{1}{4}$
  - $\mathbf{x}_2 = \mathbf{x}_1 + \alpha_1 \mathbf{d}_1 = (-\frac{1}{4}, -1)^T + \frac{1}{4}(1, 4)^T = (0, 0)^T$.
- Because f is quadratic function in two variables, $\mathbf{x}_2 = \mathbf{x}^*$.

Expanding Subspace Theorem

Theorem

Let $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$. In the Conjugate Direction algorithm,

- $\mathbf{g}_{k+1}^T \mathbf{d}_i = 0$ for all k, $0 \leq k \leq n-1$, and $0 \leq i \leq k$.
- $\mathbf{x}_{k+1} = \arg\min\ f(\mathbf{x})$ s.t. $\mathbf{x} \in \mathbf{x}_0 + \mathcal{B}_k$.

- Let $\mathcal{B}_k$ be the subspace spanned by $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_k$
- By this lemma, $\mathbf{g}_{k+1}$ is orthogonal to any vector from the subspace spanned $\mathcal{B}_k$.

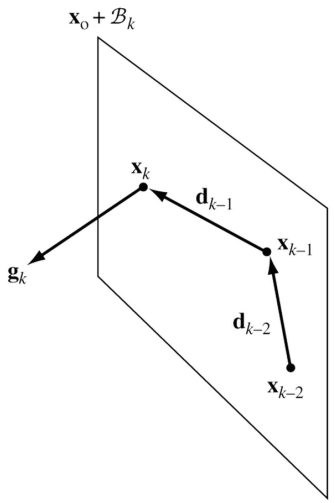


Figure: Illustration of Lemma

The Conjugate Gradient Algorithm

- Conjugate direction method uses pre-specified directions.
- Conjugate gradient algorithm does not use pre-specified directions.
- At each stage, the conjugate gradient algorithm, the direction is calculated as a linear combination of the previous direction and the current gradient in such a way that all the directions are mutually H-conjugate.

The Conjugate Gradient Algorithm

- Consider the quadratic function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T H\mathbf{x} + \mathbf{c}^T\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^n$ and H is a symmetric positive definite $n \times n$ matrix.
- At $\mathbf{x}_0$, we choose $\mathbf{d}_0$ as the steepest descent direction. I.e., $\mathbf{d}_0 = -\nabla f(\mathbf{x}_0) = -\mathbf{g}_0$.
- Thus, $\mathbf{x}_1 = \mathbf{x}_0 + \alpha_0\mathbf{d}_0$, where $\alpha_0 = \arg\min_{\alpha \geq 0} \; f(\mathbf{x}_0 + \alpha\mathbf{d}_0) = -\frac{\mathbf{g}_0^T\mathbf{d}_0}{\mathbf{d}_0 H\mathbf{d}_0}$.
- Next, we search direction $\mathbf{d}_1$ that is H conjugate of $\mathbf{d}_0$
- We choose $\mathbf{d}_1$ as linear combination of $\mathbf{d}_0$ and $\mathbf{g}_1$.
- In general, at the step $(k+1)$, we choose $\mathbf{d}_{k+1}$ as linear combination of $\mathbf{d}_k$ and $\mathbf{g}_{k+1}$.
- Specifically, we choose $\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k\mathbf{d}_k, \;\; k = 0, 1, 2, \ldots$.
- The coefficients $\beta_k, \;\; k = 0, 1, 2, \ldots$, are chosen such that $\mathbf{d}_{k+1}$ is H-conjugate to $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_k$.
- This is accomplished by choosing $\beta_k = \frac{\mathbf{g}_{k+1}^T H\mathbf{d}_k}{\mathbf{d}_k^T H\mathbf{d}_k}$.

The Conjugate Gradient Algorithm

For quadratic function, $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T H\mathbf{x} + \mathbf{c}^T\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^n$ and H is a symmetric positive definite $n \times n$ matrix.

The Conjugate Gradient Algorithm

- **Initialize:** $\mathbf{x}_0$, $\epsilon > 0$, $\mathbf{d}_0 = -\mathbf{g}_0$, $k = 0$
- While $(\|\mathbf{g}_k\| > \epsilon)$
  - Choose $\alpha_k = \arg\min_{\alpha \geq 0} \ f(\mathbf{x}_k + \alpha \mathbf{d}_k) = -\frac{\mathbf{g}_k^T \mathbf{d}_k}{\mathbf{d}_k H \mathbf{d}_k}$
  - $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
  - $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1}) = H\mathbf{x}_{k+1} + \mathbf{c}$
  - $\beta_k = \frac{\mathbf{g}_{k+1}^T H \mathbf{d}_k}{\mathbf{d}_k^T H \mathbf{d}_k}$
  - $\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k$
  - $k = k + 1$
- **Output:** $\mathbf{x}^* = \mathbf{x}_k$, global minimum of $f(\mathbf{x})$.

Conjugate Gradient Algorithm: Conjugate Property of Directions Generated

Proposition

In the conjugate gradient algorithm, the directions $\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{n-1}$ are H-conjugate.

Optimization Methods (CS1.404), Spring 2024
Lecture 16

Naresh Manwani

Machine Learning Lab, IIIT-H

March 11th, 2024

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Conjugate Gradient Algorithm for Non-Quadratic Problems

- To minimize a non-quadratic function, we first find a quadratic approximation at $\mathbf{x}_k$ using Taylor series and minimize it using conjugate descent to find $\mathbf{x}_{k+1}$.
- We replace H by Hessian at that iteration.
- The conjugate descent algorithm requires computation of Hessian at each iteration which makes it computationally expensive.
- An efficient implementation of conjugate descent eliminates the evaluation of Hessian at each step.
- Note that in conjugate descent algorithm, Hessian appears in the expression of α_k and β_k.
- Because $\alpha_k = \arg\min_{\alpha \geq 0} \ f(\mathbf{x}_k + \alpha \mathbf{d}_k)$, closed form formula for α_k can be replaced by numerical line search method.
- To eliminate Hessian from the formula of β_k, there are three possible ways.

Hestenes-Stiefel Formula

- Recall that $\beta_k = \frac{\mathbf{g}_{k+1}^T H \mathbf{d}_k}{\mathbf{d}_k^T H \mathbf{d}_k}$.
- Here, we replace $H\mathbf{d}_k$ by the term $\frac{\mathbf{g}_{k+1} - \mathbf{g}_k}{\alpha_k}$.
  $\left(\frac{\mathbf{g}_{k+1} - \mathbf{g}_k}{\alpha_k} = \frac{H\mathbf{x}_{k+1} + \mathbf{c} - H\mathbf{x}_k - \mathbf{c}}{\alpha_k} = \frac{H(\mathbf{x}_{k+1} - \mathbf{x}_k)}{\alpha_k} = H\mathbf{d}_k\right)$
- Using this in the β_k formula, we get $\beta_k = \frac{\mathbf{g}_{k+1}^T (\mathbf{g}_{k+1} - \mathbf{g}_k)}{\mathbf{d}_k^T (\mathbf{g}_{k+1} - \mathbf{g}_k)}$.
- **For quadratic functions, $\beta_k = \frac{\mathbf{g}_{k+1}^T (\mathbf{g}_{k+1} - \mathbf{g}_k)}{\mathbf{d}_k^T (\mathbf{g}_{k+1} - \mathbf{g}_k)}$ is same as $\beta_k = \frac{\mathbf{g}_{k+1}^T H \mathbf{d}_k}{\mathbf{d}_k^T H \mathbf{d}_k}$.**

Hestenes-Stiefel Approach

1: **Initialize:** The starting point $\mathbf{x}_0$ and the tolerance parameter $\epsilon > 0$, Set $k = 0$
2: Assign $\mathbf{d}_0 = -\mathbf{g}_0$
3: **while** $\|\mathbf{g}_k\| > \epsilon$ **do**
4: $\quad \alpha_k = \arg\min_{\alpha>0} \ f(\mathbf{x}_k + \alpha \mathbf{d}_k)$
5: $\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
6: $\quad$ Compute $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1})$
7: $\quad$ **if** $(k < n-1)$ **then**
8: $\quad\quad \beta_k = \frac{\mathbf{g}_{k+1}^T(\mathbf{g}_{k+1}-\mathbf{g}_k)}{\mathbf{d}_k^T(\mathbf{g}_{k+1}-\mathbf{g}_k)}$
9: $\quad\quad \mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k$
10: $\quad\quad k = k+1$
11: $\quad$ **else**
12: $\quad\quad \mathbf{x}_0 = \mathbf{x}_{k+1}$
13: $\quad\quad \mathbf{d}_0 = -\mathbf{g}_{k+1}$
14: $\quad\quad k = 0$
15: $\quad$ **end if**
16: **end while**
17: **Output:** $\mathbf{x}^* = \mathbf{x}_k$, a stationary point of f.

Polak-Ribiere Formula

- Starting from Hestenes-Stiefel formula, we multiply out the denominator to get $\beta_k = \frac{\mathbf{g}_{k+1}^T(\mathbf{g}_{k+1}-\mathbf{g}_k)}{\mathbf{d}_k^T\mathbf{g}_{k+1}-\mathbf{d}_k^T\mathbf{g}_k}$.
- But, we know that $\mathbf{d}_k^T\mathbf{g}_{k+1} = 0$.
- Also, since $\mathbf{d}_k = -\mathbf{g}_k + \beta_{k-1}\mathbf{d}_{k=1}$, we get

$$\mathbf{g}_k^T\mathbf{d}_k = -\mathbf{g}_k^T\mathbf{g}_k + \beta_{k-1}\mathbf{g}_k^T\mathbf{d}_{k-1} = -\mathbf{g}_k^T\mathbf{g}_k$$

- Thus, we get $\beta_k = \frac{\mathbf{g}_{k+1}^T(\mathbf{g}_{k+1}-\mathbf{g}_k)}{\mathbf{g}_k^T\mathbf{g}_k}$.
- This expression for β_k is called Polak-Ribiere Formula.
- **For quadratic functions, $\beta_k = \frac{\mathbf{g}_{k+1}^T(\mathbf{g}_{k+1}-\mathbf{g}_k)}{\mathbf{g}_k^T\mathbf{g}_k}$ is same as $\beta_k = \frac{\mathbf{g}_{k+1}^T H\mathbf{d}_k}{\mathbf{d}_k^T H\mathbf{d}_k}$.**

Polak-Ribiere Approach

1: **Initialize:** The starting point $\mathbf{x}_0$ and the tolerance parameter $\epsilon > 0$, Set $k = 0$
2: Assign $\mathbf{d}_0 = -\mathbf{g}_0$
3: **while** $\|\mathbf{g}_k\| > \epsilon$ **do**
4: $\quad \alpha_k = \arg\min_{\alpha>0} \ f(\mathbf{x}_k + \alpha \mathbf{d}_k)$
5: $\quad \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
6: $\quad$ Compute $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1})$
7: $\quad$ **if** $(k < n-1)$ **then**
8: $\quad\quad \beta_k = \frac{\mathbf{g}_{k+1}^T(\mathbf{g}_{k+1} - \mathbf{g}_k)}{\mathbf{g}_k^T \mathbf{g}_k}$
9: $\quad\quad \mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k$
10: $\quad\quad k = k + 1$
11: $\quad$ **else**
12: $\quad\quad \mathbf{x}_0 = \mathbf{x}_{k+1}$
13: $\quad\quad \mathbf{d}_0 = -\mathbf{g}_{k+1}$
14: $\quad\quad k = 0$
15: $\quad$ **end if**
16: **end while**
17: **Output:** $\mathbf{x}^* = \mathbf{x}_k$, a stationary point of f.

Fletcher Reeves Formula

- Starting with the Polak-Ribiere Formula, we get $\beta_k = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1} - \mathbf{g}_{k+1}^T \mathbf{g}_k}{\mathbf{g}_k^T \mathbf{g}_k}$.
- We know that $\mathbf{d}_k = -\mathbf{g}_k + \beta_k \mathbf{d}_{k-1}$. Thus, $\mathbf{g}_{k+1}^T \mathbf{d}_k = -\mathbf{g}_{k+1}^T \mathbf{g}_k + \beta_k \mathbf{g}_{k+1}^T \mathbf{d}_{k-1}$.
- But, we know that $\mathbf{g}_{k+1}^T \mathbf{d}_k = \mathbf{g}_{k+1}^T \mathbf{d}_{k-1} = 0$.
- Thus, $\mathbf{g}_{k+1}^T \mathbf{g}_k = 0$.
- This leads to $\beta_k = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k}$.
- This is called Fletcher Reeves formula.
- **For quadratic functions, $\beta_k = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k}$ is same as $\beta_k = \frac{\mathbf{g}_{k+1}^T H \mathbf{d}_k}{\mathbf{d}_k^T H \mathbf{d}_k}$.**

Fletcher Reeves Approach

1: **Initialize:** The starting point $\mathbf{x}_0$ and the tolerance parameter $\epsilon > 0$, Set $k = 0$
2: Assign $\mathbf{d}_0 = -\mathbf{g}_0$
3: **while** $\|\mathbf{g}_k\| > \epsilon$ **do**
4: $\alpha_k = \arg\min_{\alpha>0} f(\mathbf{x}_k + \alpha \mathbf{d}_k)$
5: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
6: Compute $\mathbf{g}_{k+1} = \nabla f(\mathbf{x}_{k+1})$
7: **if** $(k < n-1)$ **then**
8: $\beta_k = \frac{\mathbf{g}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{g}_k^T \mathbf{g}_k}$
9: $\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k$
10: $k = k + 1$
11: **else**
12: $\mathbf{x}_0 = \mathbf{x}_{k+1}$
13: $\mathbf{d}_0 = -\mathbf{g}_{k+1}$
14: $k = 0$
15: **end if**
16: **end while**
17: **Output:** $\mathbf{x}^* = \mathbf{x}_k$, a stationary point of f.

Summary: Conjugate Gradient Methods

- Conjugate direction methods can be regarded as being between the method of steepest descent (first-order method that uses gradient) and Newton's method (second-order method that uses Hessian as well).
  - Steepest descent is slow.
  - Newton method is fast, but we need to calculate the inverse of the Hessian matrix.
  - **Conjugate gradient uses gradient only and faster than steepest descent.**
- Conjugate gradient method attempts to accelerate gradient descent by building in momentum.
  - Recall $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
  - Using $\mathbf{d}_k = -\mathbf{g}_k + \beta_{k-1}\mathbf{d}_{k-1}$, we get

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{d}_k \\ &= \mathbf{x}_k - \alpha_k \mathbf{g}_k + \alpha_k \beta_{k-1}\mathbf{d}_{k-1}\end{aligned}$$

  - Using $\mathbf{d}_{k-1} = \frac{\mathbf{x}_k - \mathbf{x}_{k-1}}{\alpha_{k-1}}$, we get

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k + \frac{\alpha_k \beta_{k-1}}{\alpha_{k-1}}(\mathbf{x}_k - \mathbf{x}_{k-1})$$

momentum term

Quasi Newton Methods

- **Newton Method:** Given a function $f \in \mathbb{C}^2(\mathbb{R}^n)$, Newton method finds the descent direction by solving $H_k \mathbf{d}_k = -\mathbf{g}_k$, where $H_k = \nabla^2 f(\mathbf{x}_k)$ and $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$.
- **Quasi Newton Method:** Given a function $f \in \mathbb{C}^1(\mathbb{R}^n)$, quasi-Newton method finds descent direction as $\mathbf{d}_k = -B_k \mathbf{g}_k$, where B_k is a positive definite matrix.
  - B_k^{-1} is either H_k or its approximation.
  - $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k^{QN} = \mathbf{x}_k - \alpha_k B_k \mathbf{g}_k$
  - Given $\mathbf{x}_k$, $\mathbf{x}_{k+1}$, $\mathbf{g}_k$, $\mathbf{g}_{k+1}$ and B_k, how to get symmetric positive definite B_{k+1}?
  - Are there any conditions that B_{k+1} needs to satisfy?

Quasi-Newton Method

- We find quadratic approximation of f at $\mathbf{x}_{k+1}$ using B_{K+1} as follows.
  $f_{k+1}(\mathbf{x}) = f(\mathbf{x}_{k+1}) + \mathbf{g}_{k+1}^T(\mathbf{x} - \mathbf{x}_{k+1}) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_{k+1})^T B_{k+1}^{-1}(\mathbf{x} - \mathbf{x}_{k+1})$
- We require that $\nabla f_{k+1}(\mathbf{x}_k) = \mathbf{g}_k$ and $\nabla f_{k+1}(\mathbf{x}_{k+1}) = \mathbf{g}_{k+1}$.
- Therefore, using the first condition, we require
  $\nabla f_{k+1}(\mathbf{x}_k) = \mathbf{g}_k = \mathbf{g}_{k+1} + B_{k+1}^{-1}(\mathbf{x}_k - \mathbf{x}_{k-1})$.
- Letting $\gamma_k = \mathbf{g}_{k+1} - \mathbf{g}_k$ and $\boldsymbol{\delta}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, we get $B_{k+1}\gamma_k = \boldsymbol{\delta}_k$.
- This condition is also called **Quasi-Newton condition**.
- B_{k+1} should be positive definite. Thus, $\gamma_k^T B_{k+1}\gamma_k = \gamma_k^T \boldsymbol{\delta}_k > 0$.
  - From Wolfe line search condition

$$\mathbf{g}_{k+1}^T \mathbf{d}_k \geq c_2 \mathbf{g}_k^T \mathbf{d}_k, \quad \text{where } c_2 \in (0,1)$$

$$\Rightarrow (\mathbf{g}_{k+1} - \mathbf{g}_k)^T \mathbf{d}_k \geq (c_2 - 1)\mathbf{g}_k^T \mathbf{d}_k$$

We know that $c_2 - 1 < 0$ and $\mathbf{g}_k^T \mathbf{d}_k = -\mathbf{g}_k^T B_k \mathbf{g}_k < 0$ as B_k is positive definite matrix. Thus, we get

$$(\mathbf{g}_{k+1} - \mathbf{g}_k)^T \mathbf{d}_k \geq (c_2 - 1)\mathbf{g}_k^T \mathbf{d}_k > 0 \Rightarrow \gamma_k^T \mathbf{d}_k > 0$$

$$\Rightarrow \gamma_k^T \boldsymbol{\delta}_k > 0, \quad \text{using } \mathbf{d}_k = \frac{1}{\alpha_k}(\mathbf{x}_{k+1} - \mathbf{x}_k) = \frac{1}{\alpha_k}\boldsymbol{\delta}_k$$

- When Wolfe condition is satisfied in a line search, $\exists B_{k+1}$ which satisfies Quasi-Newton condition.

Symmetric Rank One Correction

- Here, we want to update B_k to B_{k+1} by adding a rank one matrix $a_k \mathbf{z}_k \mathbf{z}_k^T$, where $a_k \in \mathbb{R}(a \neq 0)$ and $\mathbf{z}_k \in \mathbb{R}^n(\mathbf{z} \neq \mathbf{0})$. Thus,

$$B_{k+1} = B_k + a_k \mathbf{z}_k \mathbf{z}_k^T$$

- Now, we choose a_k and $\mathbf{z}_k$ such that B_{k+1} satisfies Quasi-Newton condition. Thus, we want

$$\begin{aligned} & B_{k+1}\gamma_k = \boldsymbol{\delta}_k \\ \Rightarrow & (B_k + a_k \mathbf{z}_k \mathbf{z}_k^T)\gamma_k = \boldsymbol{\delta}_k \\ \Rightarrow & a_k \mathbf{z}_k \mathbf{z}_k^T \gamma_k = \boldsymbol{\delta}_k - B_k \gamma_k \end{aligned}$$

- Let $\mathbf{z}_k = \boldsymbol{\delta}_k - B_k \gamma_k$. Therefore, $a_k \mathbf{z}_k^T \gamma_k = 1$.
- That gives $\alpha_k = \frac{1}{(\boldsymbol{\delta}_k - B_k \gamma_k)^T \gamma_k}$.

Thus, using $\mathbf{x}_k$, $\mathbf{x}_{k+1}$, $\mathbf{g}_{k+1}$ and $\mathbf{g}_k$, we get

$$B_{k+1}^{SR1} = B_k + \frac{(\boldsymbol{\delta}_k - B_k \gamma_k)(\boldsymbol{\delta}_k - B_k \gamma_k)^T)}{(\boldsymbol{\delta}_k - B_k \gamma_k)^T \gamma_k}$$

Quasi-Newton Method (Rank One Correction)

1: **Initialize:** The starting point $\mathbf{x}_0$, Symmetric positive definite matrix B_0 and the tolerance parameter $\epsilon > 0$, Set $k = 0$
2: **while** $\|\mathbf{g}_k\| > \epsilon$ **do**
3: $\mathbf{d}_k = -B_k \mathbf{g}_k$
4: Find α_k along $\mathbf{d}_k$ such that

- $f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) < f(\mathbf{x}_k)$
- α_k satisfies Armijo-Wolfe condition

5: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
6: Find B_{k+1} as

$$B_{k+1} = B_k + \frac{(\boldsymbol{\delta}_k - B_k \boldsymbol{\gamma}_k)(\boldsymbol{\delta}_k - B_k \boldsymbol{\gamma}_k)^T)}{(\boldsymbol{\delta}_k - B_k \boldsymbol{\gamma}_k)^T \boldsymbol{\gamma}_k}$$

7: $k = k + 1$
8: **end while**
9: **Output:** $\mathbf{x}^* = \mathbf{x}_k$, a stationary point of f.

Example: Rank One Correction

- Consider the problem min $f(x, y) = 4x^2 + y^2 - 2xy$
- For this problem, $\mathbf{x}^* = [0 \;\; 0]^T$, $H = \begin{bmatrix} 8 & -2 \\ -2 & 2 \end{bmatrix}$, $H^{-1} = \begin{bmatrix} 0.1667 & 0.1667 \\ 0.1667 & 0.6667 \end{bmatrix}$
- We run **rank one correction** approach with $\mathbf{x}_0 = [-2 \;\; -2]^T$ and B_0 as identity matrix.
- We see that the algorithm converges in 3 steps. Below are the updates in each step.

k	x_k	y_k	B_k	$\|\mathbf{g}_k\|$
0	-2	-2	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	12.0
1	0	-2	$\begin{bmatrix} 0.1833 & 0.2333 \\ 0.2333 & 0.9333 \end{bmatrix}$	5.65
2	0.1538	0.1536	$\begin{bmatrix} 0.1667 & 0.1667 \\ 0.1667 & 0.6667 \end{bmatrix}$	0.92
3	0	0	H^{-1}	0

Quasi-Newton Algorithm Applied on Quadratic Functions

- Consider the problem $\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^T H\mathbf{x} + \mathbf{c}^T\mathbf{x}$, where H is a symmetric positive definite matrix.
- To solve this problem using rank one correction method, at every iteration k
  - B_{k+1} is symmetric positive definite.
  - B_{k+1} is obtained from $\mathbf{x}_k$, $\mathbf{x}_{k+1}$, $\mathbf{g}_{k+1}$ and $\mathbf{g}_k$.
  - B_{k+1} satisfies Quasi-Newton condition, $B_{k+1}\gamma_k = \delta_k$
- Note that $\mathbf{g}_{k+1} - \mathbf{g}_k = H\mathbf{x}_{k+1} + \mathbf{c} - H\mathbf{x}_k - \mathbf{c} = H(\mathbf{x}_{k+1} - \mathbf{x}_k)$. Which means, $\gamma_k = H\delta_k$.

Lemma: Hereditary Property

SR1 correction approach applied to quadratic function with positive definite Hessian H, we have

$$B_{k+1}\gamma_i = \delta_i,\ 0 \leq i \leq k.$$

When f is quadratic, the hereditary property is satisfied by SR1 regardless of how the line search is performed.

Convergence of SR1 Applied on Quadratic Functions

Theorem1: For Quadratic Functions

Consider SR1 quasi-Newton algorithm applied to a quadratic function with positive definite Hessian H. Then, for any starting point $\mathbf{x}_0$ and any symmetric starting matrix B_0, the sequence of iterates $\mathbf{x}_k$ generated by SR1 converges to the minimizer in n-steps, provided $(\boldsymbol{\delta}_k - B_k\boldsymbol{\gamma}_k)^T\boldsymbol{\gamma}_k \neq 0, \forall k$. Moreover, if n-steps are performed and $\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{n-1}$ are linearly independent, then $B_n = H^{-1}$.

Optimization Methods (CS1.404), Spring 2024
Lecture 17

Naresh Manwani

Machine Learning Lab, IIIT-H

March 14th, 2024

Rank Two Correction Quasi newton Method

- Given that B_k is symmetric and positive definite matrix, let

$$B_{k+1} = B_k + \alpha \mathbf{u}\mathbf{u}^T + \beta \mathbf{v}\mathbf{v}^T$$

- B_{k+1} is required to satisfy Quasi-Newton condition. Thus,

$$\alpha \mathbf{u}^T \gamma_k \mathbf{u} + \beta \mathbf{v}^T \gamma_k \mathbf{v} = \boldsymbol{\delta}_k - B_k \gamma_k$$

- Letting $\alpha \mathbf{u}^T \gamma_k = \beta \mathbf{v}^T \gamma_k = 1$, we get $\mathbf{u} + \mathbf{v} = \boldsymbol{\delta}_k - B_k \gamma_k$. Taking $\mathbf{u} = \boldsymbol{\delta}_k$ and $\mathbf{v} = -B_k \gamma_k$, we get

$$\alpha^{-1} = \mathbf{u}^T \gamma_k = \boldsymbol{\delta}_k^T \gamma_k$$
$$\beta^{-1} = \mathbf{v}^T \gamma_k = -\gamma_k^T B_k \gamma_k$$

- Therefore, we get the following update for B_{k+1}:

$$B_{k+1} = B_k + \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T}{\boldsymbol{\delta}_k^T \gamma_k} - \frac{B_k \gamma_k \gamma_k^T B_k}{\gamma_k^T B_k \gamma_k}$$

- This update is called DFP named after Davidson, Fletcher and Powell.

Rank Two Correction Quasi newton Method

Theorem

Given that B_k is symmetric and positive definite, B_{k+1} generated by DFP is symmetric and positive definite.

DFP Quasi-Newton Algorithm

1: **Initialize:** The starting point $\mathbf{x}_0$, Symmetric positive definite matrix B_0 and the tolerance parameter $\epsilon > 0$, Set $k = 0$
2: **while** $\|\mathbf{g}_k\| > \epsilon$ **do**
3: $\mathbf{d}_k = -B_k \mathbf{g}_k$
4: Find α_k along $\mathbf{d}_k$ such that

- $f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) < f(\mathbf{x}_k)$
- α_k satisfies Armijo-Wolfe condition

5: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
6: Find B_{k+1} as

$$B_{k+1} = B_k + \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T}{\boldsymbol{\delta}_k^T \boldsymbol{\gamma}_k} - \frac{B_k \boldsymbol{\gamma}_k \boldsymbol{\gamma}_k^T B_k}{\boldsymbol{\gamma}_k^T B_k \boldsymbol{\gamma}_k}$$

7: $k = k + 1$
8: **end while**
9: **Output:** $\mathbf{x}^* = \mathbf{x}_k$, a stationary point of f.

BFGS Algorithm

- Quasi Newton condition requires $B_{k+1}\gamma_k = \delta_k$ to hold for all k.
- Assume that we want to approximate the Hessian H_{k+1} rather than its inverse. Let $G_{k+1} = B_{k+1}^{-1}$ which approximates Hessian H_{k+1}.
- Then, Quasi-Newton condition would result into $G_{k+1}\delta_k = \gamma_k$.
- Rank two update of G_{k+1} will have the form

$$G_{k+1} = G_k + \alpha \mathbf{u}\mathbf{u}^T + \beta \mathbf{v}\mathbf{v}^T$$

- G_{k+1} is required to satisfy Quasi-Newton condition. Thus,

$$\alpha \mathbf{u}^T \delta_k \mathbf{u} + \beta \mathbf{v}^T \delta_k \mathbf{v} = \gamma_k - G_k \delta_k$$

- Letting $\alpha \mathbf{u}^T \delta_k = \beta \mathbf{v}^T \delta_k = 1$, we get $\mathbf{u} + \mathbf{v} = \gamma_k - G_k \delta_k$. Taking $\mathbf{u} = \gamma_k$ and $\mathbf{v} = -G_k \delta_k$, we get

$$\alpha^{-1} = \mathbf{u}^T \delta_k = \gamma_k^T \delta_k$$
$$\beta^{-1} = \mathbf{v}^T \delta_k = -\delta_k^T G_k \delta_k$$

- Therefore, we get the following update for B_{k+1}:

$$G_{k+1} = G_k + \frac{\gamma_k \gamma_k^T}{\gamma_k^T \delta_k} - \frac{G_k \delta_k \delta_k^T G_k}{\delta_k^T G_k \delta_k}$$

BFGS Algorithm

- Next step is to find B_{k+1} as G_{k+1}^{-1}.
- We use Sherman-Morrison Formula to find G_{k+1}^{-1}.
  $(A + \mathbf{uv}^T)^{-1} = A^{-1} - \frac{A^{-1}\mathbf{uv}^T A^{-1}}{1+\mathbf{v}^T A^{-1}\mathbf{u}}$
- Applying this formula twice to G_{k+1}, we get

$$B_{k+1}^{BFGS} = B_k^{BFGS} + \left(1 + \frac{\gamma_k^T B_k^{BFGS} \gamma_k}{\boldsymbol{\delta}_k^T \gamma_k}\right) \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T}{\boldsymbol{\delta}_k^T \gamma_k} - \left(\frac{\boldsymbol{\delta}_k \gamma_k^T B_k^{BFGS} + B_k^{BFGS} \gamma_k \boldsymbol{\delta}_k^T}{\boldsymbol{\delta}_k^T \gamma_k}\right)$$

Broyden Family

- $B_{k+1}(\phi) = \phi B_{k+1}^{BFGS} + (1-\phi) B_{k+1}^{DFP}$, where $\phi \in [0,1]$

Broyden Familty Quasi-Newton Algorithm

1: **Initialize:** The starting point $\mathbf{x}_0$, Symmetric positive definite matrix B_0 and the tolerance parameter $\epsilon > 0$, Set $k = 0$
2: **while** $\|\mathbf{g}_k\| > \epsilon$ **do**
3: $\mathbf{d}_k = -B_k\mathbf{g}_k$
4: Find α_k along $\mathbf{d}_k$ such that

- $f(\mathbf{x}_k + \alpha_k\mathbf{d}_k) < f(\mathbf{x}_k)$
- α_k satisfies Armijo-Wolfe condition

5: $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{d}_k$
6: Find B_{k+1} as

$$B_{k+1}(\phi) = \phi B_{k+1}^{BFGS} + (1-\phi)B_{k+1}^{DFP}$$

where $\phi \in [0, 1]$.
7: $k = k + 1$
8: **end while**
9: **Output:** $\mathbf{x}^* = \mathbf{x}_k$, a stationary point of f.

Optimization Methods (CS1.404), Spring 2025

Naresh Manwani

Machine Learning Lab, IIIT-H

April 7, 2025

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

Constrained Optimization Problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ s.t. \quad & h_j(\mathbf{x}) \leq 0, \ j = 1, \ldots, l \\ & e_i(\mathbf{x}) = 0, \ i = 1, \ldots, m \end{aligned}$$

where

- $h_j : \mathbb{R}^n \to \mathbb{R}, \ j = 1, \ldots, l$
- $e_i : \mathbb{R}^n \to \mathbb{R}, \ i = 1, \ldots, m$
- Assume that all h_j and e_i are sufficiently smooth functions.
- **Feasible set:** Any point that satisfies constraints is called feasible point. Set of all feasible points is called feasible set and is described as $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid h_j(\mathbf{x}) \leq 0, \ e_i(\mathbf{x}) = 0, \ j = 1 \ldots l, \ i = 1 \ldots m\}$.

Local and Global Minima

Definition: Global Minima

A point $\mathbf{x}^* \in \mathcal{X}$ is said to be global minimum point of f over $\mathcal{X}$ if $f(\mathbf{x}) \geq f(\mathbf{x}^*)$, $\forall \mathbf{x} \in \mathcal{X}$. If $f(\mathbf{x}) > f(\mathbf{x}^*)$, $\forall \mathbf{x} \in \mathcal{X}$, $\mathbf{x} \neq \mathbf{x}^*$, then $\mathbf{x}^*$ is called strict global minima.

Definition: Local Minima

A point $\mathbf{x}^* \in \mathcal{X}$ is said to be local minimum point of f over $\mathcal{X}$ if there exists $\epsilon > 0$ such that $f(\mathbf{x}) \geq f(\mathbf{x}^*)$, $\forall \mathbf{x} \in \mathcal{X} \cap B(\mathbf{x}^*, \epsilon)$. If $f(\mathbf{x}) > f(\mathbf{x}^*)$, $\forall \mathbf{x} \in \mathcal{X} \cap B(\mathbf{x}^*, \epsilon)$, $\mathbf{x} \neq \mathbf{x}^*$, then $\mathbf{x}^*$ is called strict local minima.

Constrained Convex Optimization Problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ s.t. \quad & h_j(\mathbf{x}) \leq 0, \ j = 1, \ldots, l \\ & e_i(\mathbf{x}) = 0, \ i = 1, \ldots, m \end{aligned}$$

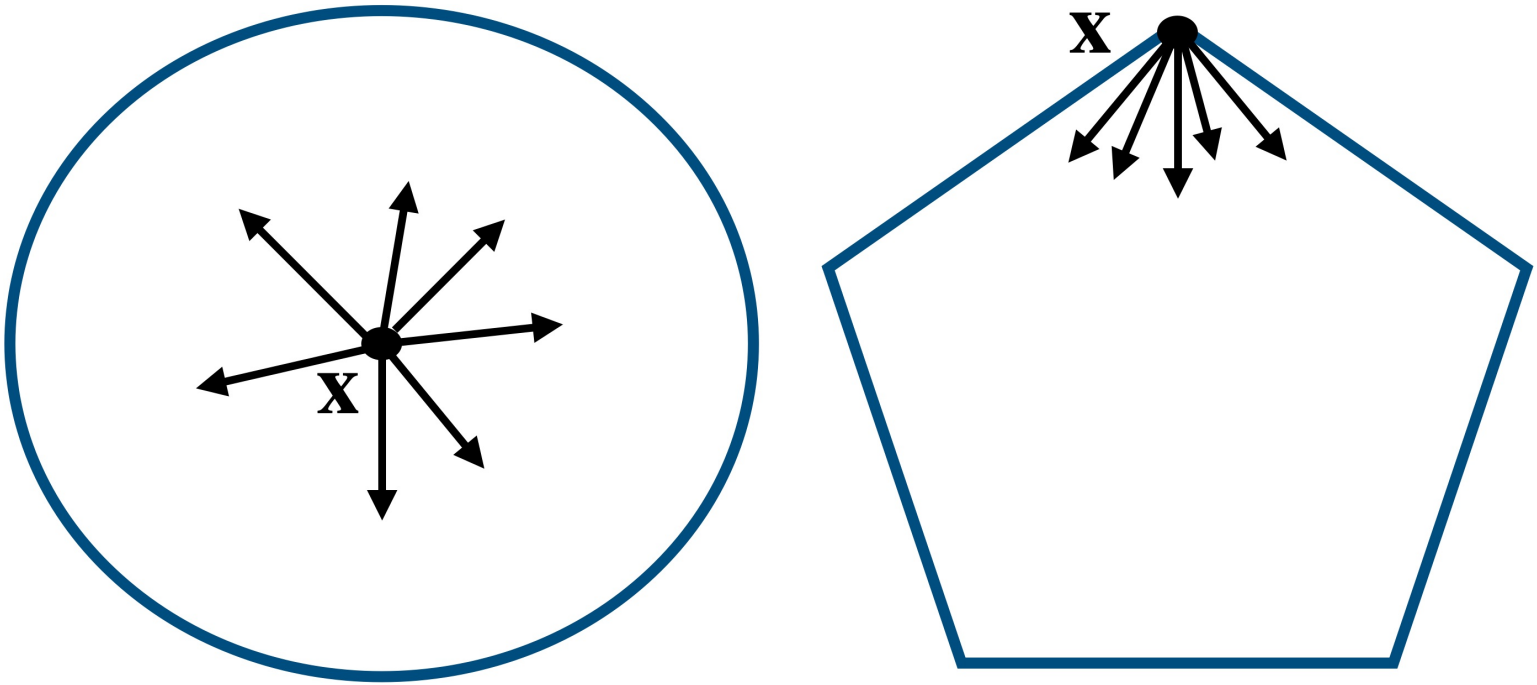
where

- $f(\mathbf{x})$ is convex.
- $h_j : \mathbb{R}^n \to \mathbb{R}, \ j = 1, \ldots, l$ are convex functions.
- $e_i : \mathbb{R}^n \to \mathbb{R}, \ i = 1, \ldots, m$ are affine functions.
- Any local minima is a global minima.

Set of Feasible Directions at $\mathbf{x} \in \mathcal{X}$

Definition

A vector $\mathbf{d} \in \mathbb{R}^n$, $\mathbf{d} \neq \mathbf{0}$ is said to be a feasible direction at $\mathbf{x} \in \mathcal{X}$ if there exist $\delta_1 > 0$ such that $\mathbf{x} + \alpha\mathbf{d} \in \mathcal{X}$, $\forall \alpha \in (0, \delta_1)$.



Let $\mathcal{F}(\mathbf{x})$ represent the set of feasible directions at $\mathbf{x} \in \mathcal{X}$. Thus,

$$\mathcal{F}(\mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n \mid \exists \delta_1 > 0 \text{ s.t. } \mathbf{x} + \alpha\mathbf{d} \in \mathcal{X}\ \forall \alpha \in (0, \delta_1)\}$$

Set of Descent Directions at $\mathbf{x} \in \mathcal{X}$

Definition

A vector $\mathbf{d} \in \mathbb{R}^n$, $\mathbf{d} \neq \mathbf{0}$ is said to be a descent direction at $\mathbf{x} \in \mathcal{X}$ if there exists $\delta_2 > 0$ such that $f(\mathbf{x} + \alpha\mathbf{d}) < f(\mathbf{x})$, $\forall \alpha \in (0, \delta_2)$.

Let $\mathcal{D}(\mathbf{x})$ represent the set of descent directions at $\mathbf{x} \in \mathcal{X}$.. Thus,

$$\mathcal{D}(\mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n \mid \exists \delta_2 > 0 \text{ s.t. } f(\mathbf{x} + \alpha\mathbf{d}) < f(\mathbf{x}) \; \forall \alpha \in (0, \delta_2)\}.$$

Characterization of Local Minima for Constrained Optimization Problem

Theorem

Let $\mathcal{X}$ be a nonempty set in $\mathbb{R}^n$ and $\mathbf{x}^* \in \mathcal{X}$ be a local minimum of f over $\mathcal{X}$. Then $\mathcal{F}(\mathbf{x}^*) \cap \mathcal{D}(\mathbf{x}^*) = \phi$.

- $\mathbf{x}^* \in \mathcal{X}$ is a local minima $\Rightarrow \mathcal{F}(\mathbf{x}^*) \cap \mathcal{D}(\mathbf{x}^*) = \phi$.
- Consider any $\mathbf{x} \in \mathcal{X}$ and assume $f \in \mathcal{C}^1$. Then $\nabla f(\mathbf{x})^T \mathbf{d} < 0$ implies there exists $\delta > 0$ such that $f(\mathbf{x} + \alpha \mathbf{d}) < f(\mathbf{x})$, $\forall \alpha \in (0, \delta)$. Such $\mathbf{d}$ is a descent direction ($\mathbf{d} \in \mathcal{D}(\mathbf{x})$).
- Let $\tilde{\mathcal{D}}(\mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n \mid \nabla f(\mathbf{x})^T \mathbf{d} < 0\}$, then $\tilde{\mathcal{D}}(\mathbf{x}) \subseteq \mathcal{D}(\mathbf{x})$.

Active Constraints

- Consider the problem min $f(\mathbf{x})$ such that $h_j(\mathbf{x}) \leq 0,\ j = 1 \ldots l$, $\mathbf{x} \in \mathbb{R}^n$.
- Assume $h_j \in \mathcal{C}^1,\ j = 1 \ldots l$.
- Let $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid h_j(\mathbf{x}) \leq 0,\ j = 1 \ldots l\}$.
- An inequality constraint $h_j(\mathbf{x}) \leq 0$ is said to be active at $\mathbf{x}^*$ if $h_j(\mathbf{x}^*) = 0$. It is inactive if $h_j(\mathbf{x}^*) < 0$.
- Set of active constraints $\mathcal{A}(\mathbf{x}) = \{j \in \{1, \ldots, l\} \mid h_j(\mathbf{x}) = 0\}$.

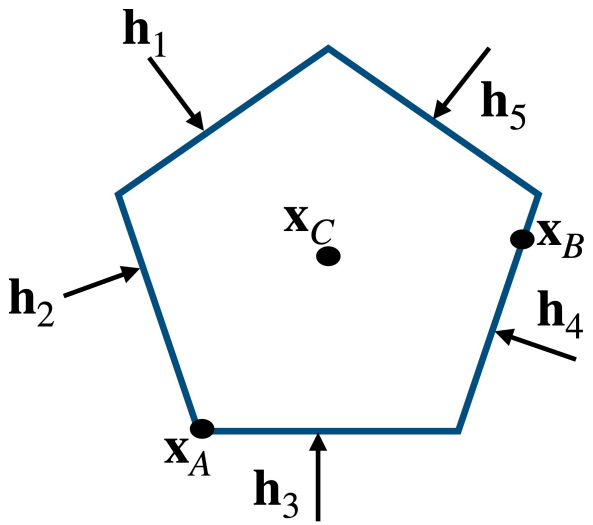


Figure: At $\mathbf{x}_A$, h_2 and h_3 are active. At $\mathbf{x}_B$, h_4 is active. At $\mathbf{x}_C$, no constraint is active. Thus, $\mathcal{A}(\mathbf{x}_A) = \{2, 3\}$, $\mathcal{A}(\mathbf{x}_B) = \{4\}$ and $\mathcal{A}(\mathbf{x}_C) = \phi$.

Further Characterization of Set of Feasible Directions

- Consider the set $\tilde{\mathcal{F}}(\mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n \mid \nabla h_j(\mathbf{x})^T \mathbf{d} < 0,\ j \in \mathcal{A}(\mathbf{x})\}$.
- $\tilde{\mathcal{F}}(\mathbf{x})$ is the set of those directions at $\mathbf{x}$ for which the directional derivative of active constraints is negative.

Lemma

For any $\mathbf{x} \in \mathcal{X}$, we have

$$\tilde{\mathcal{F}}(\mathbf{x}) \subseteq \mathcal{F}(\mathbf{x}).$$

Necessary Condition for Local Minima

- Consider the problem min $f(\mathbf{x})$ such that $h_j(\mathbf{x}) \leq 0,\ j = 1 \ldots l$, $\mathbf{x} \in \mathbb{R}^n$.
- Let $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid h_j(\mathbf{x}) \leq 0,\ j = 1 \ldots l\}$.
- $\tilde{\mathcal{D}}(\mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n \mid \nabla f(\mathbf{x})^T \mathbf{d} < 0\} \subseteq D(\mathbf{x})$.
- $\tilde{\mathcal{F}}(\mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n \mid \nabla h_j(\mathbf{x})^T \mathbf{d} < 0,\ j \in \mathcal{A}(\mathbf{x})\} \subseteq \mathcal{F}(\mathbf{x})$.
- $\mathbf{x}^* \in \mathcal{X}$ is a local minima implies $\Rightarrow \mathcal{F}(\mathbf{x}^*) \cap \mathcal{D}(\mathbf{x}^*) = \phi$. Which implies $\tilde{\mathcal{F}}(\mathbf{x}^*) \cap \tilde{\mathcal{D}}(\mathbf{x}^*) = \phi$.
- $\tilde{\mathcal{F}}(\mathbf{x}^*) \cap \tilde{\mathcal{D}}(\mathbf{x}^*) = \phi$ is only necessary condition for $\mathbf{x}^*$ to be local minimum. It is not a sufficient condition. This means if this condition is satisfied for $\mathbf{x}^*$, it does not mean that $\mathbf{x}^*$ is a local minimum.

Examples: Necessary Condition for Local Minima

- **Example 1:** Consider optimization problem min $x_1^2 + x_2^2$ such that $(x_1 + x_2 - 1)^3 \leq 0$ and $x_1, x_2 \geq 0$.
  - Here, $h_1(x_1, x_2) = (x_1 + x_2 - 1)^3$, $h_2(x_1, x_2) = -x_1$, $h_3(x_1, x_2) = -x_2$
  - Consider a point $\mathbf{x}_A = (a, b) \in \mathbb{R}^2$ such that $a + b = 1$ and $a > 0, b > 0$.
  - We can see that $\nabla h_1(\mathbf{x}_A) = \mathbf{0}$. Thus, $\tilde{\mathcal{F}}(\mathbf{x}_A) = \{\mathbf{d} \mid \nabla h_1(\mathbf{x}_A)^T \mathbf{d} < 0\} = \phi$.
  - Here, $\tilde{\mathcal{F}}(\mathbf{x}_A) \cap \tilde{\mathcal{D}}(\mathbf{x}_A) = \phi$, but $\mathbf{x}_A$ is not a local minima.
- **Example 2:** Consider optimization problem min $x_1^2 + x_2^2$ such that $(x_1 + x_2 - 1) \leq 0$ and $x_1, x_2 \geq 0$.
  - Here, $h_1(x_1, x_2) = (x_1 + x_2 - 1)$, $h_2(x_1, x_2) = -x_1$, $h_3(x_1, x_2) = -x_2$
  - Consider a point $\mathbf{x}_A = (a, b) \in \mathbb{R}^2$ such that $a + b = 1$ and $a > 0, b > 0$.
  - We can see that $\nabla h_1(\mathbf{x}_A) = [1\ 1]^T$. Thus, $\tilde{\mathcal{F}}(\mathbf{x}_A) = \{\mathbf{d} \mid \nabla h_1(\mathbf{x}_A)^T \mathbf{d} < 0\} = \{\mathbf{d} \mid d_1 + d_2 < 0\} \neq \phi$.
  - Here, $\mathbf{x}_A$ is not a local minima and $\tilde{\mathcal{F}}(\mathbf{x}_A) \cup \tilde{\mathcal{D}}(\mathbf{x}_A) \neq \phi$.

Examples: Necessary Condition for Local Minima

Example 3: Consider optimization problem min $x_1^2 + x_2^2$ such that $(x_1 + x_2 - 1) = 0$.

- Here, $h_1(x_1, x_2) = (x_1 + x_2 - 1)$, $h_2(x_1, x_2) = -x_1 - x_2 + 1$
- Consider a point $\mathbf{x}_A = (a, b) \in \mathbb{R}^2$ such that $a + b = 1$.
- Thus, $\tilde{\mathcal{F}}(\mathbf{x}_A) = \{\mathbf{d} \mid \nabla h_1(\mathbf{x}_A)^T \mathbf{d} < 0,\ \nabla h_2(\mathbf{x}_A)^T \mathbf{d} < 0\} = \phi$.
- This does not guarantee that $\mathbf{x}_A$ is a local minima.
- **The above Necessary Condition for Local Minima is only applicable when there are inequality constraints.**
- **It is not applicable when there are equality constraints.**

Necessary Condition for Local Minima

- Consider the problem min $f(\mathbf{x})$ such that $h_j(\mathbf{x}) \leq 0,\ j = 1 \ldots l$, $\mathbf{x} \in \mathbb{R}^n$.
- Let $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid h_j(\mathbf{x}) \leq 0,\ j = 1 \ldots l\}$.
- $\tilde{\mathcal{D}}(\mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n \mid \nabla f(\mathbf{x})^T \mathbf{d} < 0\} \subseteq D(\mathbf{x})$.
- $\tilde{\mathcal{F}}(\mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n \mid \nabla h_j(\mathbf{x})^T \mathbf{d} < 0,\ j \in \mathcal{A}(\mathbf{x})\} \subseteq \mathcal{F}(\mathbf{x})$.
- $\mathbf{x}^* \in \mathcal{X}$ is a local minima, then $\tilde{\mathcal{F}}(\mathbf{x}^*) \cap \tilde{\mathcal{D}}(\mathbf{x}^*) = \phi$.
- Let $A = \begin{bmatrix} \nabla f(\mathbf{x}^*)^T \\ \nabla h_{j_1}(\mathbf{x}^*)^T \\ \vdots \\ \nabla h_{j_k}(\mathbf{x}^*)^T \end{bmatrix}$ assuming that there are k active constraints $h_{j_1}, \ldots, h_{j_k}$.
- If $\mathbf{x}^* \in \mathcal{X}$ is a local minima, then $\{\mathbf{d} \in \mathbb{R}^n \mid A\mathbf{d} < \mathbf{0}\} = \phi$.

Farkas Lemma

Lemma

Let $A \in \mathbb{R}^{m \times n}$ and $\mathbf{c} \in \mathbb{R}^n$. Then, exactly one of the two systems has a solution:

1. $A\mathbf{x} \leq \mathbf{0},\ \mathbf{c}^T\mathbf{x} > 0$ for some $\mathbf{x} \in \mathbb{R}^n$
2. $A^T\mathbf{y} = \mathbf{c},\ \mathbf{y} \geq \mathbf{0}$ for some $\mathbf{y} \in \mathbb{R}^m$

Corollary

1. $A\mathbf{x} < \mathbf{0}$ for some $\mathbf{x} \in \mathbb{R}^n$
2. $A^T\mathbf{y} = 0,\ \mathbf{y} \geq \mathbf{0}$ for some $\mathbf{y} \in \mathbb{R}^m$

Local Minima Characterization

- If $\mathbf{x}^* \in \mathcal{X}$ is a local minima, then $\{\mathbf{d} \in \mathbb{R}^n \mid A\mathbf{d} < \mathbf{0}\} = \phi$.
- Using the Corollary above, $\exists \lambda_0 \geq 0$ and $\lambda_j \geq 0$, $j \in \mathcal{A}(\mathbf{x}^*)$, not all λ's zero, such that

$$\lambda_0 \nabla f(\mathbf{x}^*) + \sum_{j \in \mathcal{A}(\mathbf{x}^*)} \lambda_j \nabla h_j(\mathbf{x}^*) = \mathbf{0}$$

Optimization Methods (CS1.404), Spring 2025

Naresh Manwani

Machine Learning Lab, IIIT-H

April 10, 2025

Necessary Condition for Local Minima

- Consider the problem min $f(\mathbf{x})$ such that $h_j(\mathbf{x}) \leq 0,\ j = 1 \ldots l$, $\mathbf{x} \in \mathbb{R}^n$.
- Let $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid h_j(\mathbf{x}) \leq 0,\ j = 1 \ldots l\}$.
- $\tilde{\mathcal{D}}(\mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n \mid \nabla f(\mathbf{x})^T \mathbf{d} < 0\} \subseteq D(\mathbf{x})$.
- $\tilde{\mathcal{F}}(\mathbf{x}) = \{\mathbf{d} \in \mathbb{R}^n \mid \nabla h_j(\mathbf{x})^T \mathbf{d} < 0,\ j \in \mathcal{A}(\mathbf{x})\} \subseteq \mathcal{F}(\mathbf{x})$.
- $\mathbf{x}^* \in \mathcal{X}$ is a local minima if $\tilde{\mathcal{F}}(\mathbf{x}) \cap \tilde{\mathcal{D}}(\mathbf{x}) = \phi$.
- Let $A = \begin{bmatrix} \nabla f(\mathbf{x}^*) \\ \nabla h_{j_1}(\mathbf{x}^*) \\ \vdots \\ \nabla h_{j_k}(\mathbf{x}^*) \end{bmatrix}$ assuming that there are k active constraints $h_{j_1}, \ldots, h_{j_k}$.
- Then $\mathbf{x}^* \in \mathcal{X}$ is a local minima if $\{\mathbf{d} \in \mathbb{R}^n \mid A\mathbf{d} < \mathbf{0}\} = \phi$.

Farkas Lemma

Lemma

Let $A \in \mathbb{R}^{m \times n}$ and $\mathbf{c} \in \mathbb{R}^n$. Then, exactly one of the two systems has a solution:

1. $A\mathbf{x} \leq \mathbf{0},\ \mathbf{c}^T\mathbf{x} > 0$ for some $\mathbf{x} \in \mathbb{R}^n$
2. $A^T\mathbf{y} = \mathbf{c},\ \mathbf{y} \geq \mathbf{0}$ for some $\mathbf{y} \in \mathbb{R}^m$

Gordon's Theorem

1. $A\mathbf{x} < \mathbf{0}$ for some $\mathbf{x} \in \mathbb{R}^n$
2. $A^T\mathbf{y} = 0,\ \mathbf{y} \geq \mathbf{0}$ for some $\mathbf{y} \in \mathbb{R}^m$

Fritz-John Condition for Local Minima

- $\mathbf{x}^* \in \mathcal{X}$ is a local minima if $\{\mathbf{d} \in \mathbb{R}^n \mid A\mathbf{d} < \mathbf{0}\} = \phi$.
- Using Gordon's Theorem, $\exists \lambda_0 \geq 0$ and $\lambda_j \geq 0$, $j \in \mathcal{A}(\mathbf{x}^*)$, not all λ's zero, such that

$$\lambda_0 \nabla f(\mathbf{x}^*) + \sum_{j \in \mathcal{A}(\mathbf{x}^*)} \lambda_j \nabla h_j(\mathbf{x}^*) = \mathbf{0}$$

- Define $\lambda_j = 0$, $\forall j \notin \mathcal{A}(\mathbf{x}^*)$. Then the above condition is same as

$$\lambda_0 \nabla f(\mathbf{x}^*) + \sum_{j=1}^{l} \lambda_j \nabla h_j(\mathbf{x}^*) = \mathbf{0}$$
$$\lambda_j h_j(\mathbf{x}^*) = 0; \quad j = 1, \ldots, l$$
$$\lambda_j \geq 0; \quad j = 0, 1, \ldots, l$$

Issues with Fritz-John Condition

- A major drawback of the Fritz-John conditions is in the fact that they allows λ_0 to be zero.
- The case $\lambda_0 = 0$ is not particularly informative since condition. In this case, Fritz-John condition becomes

$$\sum_{j \in \mathcal{A}(\mathbf{x}^*)} \lambda_j \nabla h_j(\mathbf{x}^*) = \mathbf{0}$$

- This means that the gradients of the active constraints are linearly dependent.
- This condition has nothing to do with the objective function.
- This implies that there can be many points which satisfy Fritz-John condition which are not local minima.

Next we will see how KKT condition can overcome this issue.

Regular Point

Definition

A point $\mathbf{x}^* \in \mathcal{X}$ is said to be regular point if the gradient vectors $\nabla h_j(\mathbf{x}^*)$, $j \in \mathcal{A}(\mathbf{x}^*)$, are linearly independent. Then, $\sum_{j \in \mathcal{A}(\mathbf{x}^*)} \lambda_j \nabla h_j(\mathbf{x}^*) = \mathbf{0}$ only if $\lambda_j = 0$, $\forall j \in \mathcal{A}(\mathbf{x}^*)$.

Lemma

Consider min $f(\mathbf{x})$ such that $h_j(\mathbf{x}) \leq 0$, $j = 1 \ldots l$, $\mathbf{x} \in \mathbb{R}^n$. If $\mathbf{x}^*$ is a regular point and a local minima, then $\lambda_0 \neq 0$.

Proof:

- If $\mathbf{x}^*$ is a regular point and local minima, then Frtiz-John optimality condition implies,

$$\lambda_0 \nabla f(\mathbf{x}^*) + \sum_{j \in \mathcal{A}(\mathbf{x}^*)} \lambda_j \nabla h_j(\mathbf{x}^*) = \mathbf{0}.$$

- Let $\lambda_0 = 0$. Then, $\sum_{j \in \mathcal{A}(\mathbf{x}^*)} \lambda_j \nabla h_j(\mathbf{x}^*) = \mathbf{0}$, making gradients of the active constraints linearly dependent.
- However, this contradicts that $\mathbf{x}^*$ is a regular point. Thus, $\lambda_0 \neq 0$.

This is the key idea in KKT conditions taking $\lambda_0 = 1$.

KKT Optimality Conditions: First Order

KKT Optimality Conditions of First Order

Consider the problem min $f(\mathbf{x})$ such that $h_j(\mathbf{x}) \leq 0$, $j = 1 \ldots l$, $\mathbf{x} \in \mathbb{R}^n$. Assume that $\mathbf{x}^* \in \mathcal{X}$ to be a regular point and $\mathbf{x}^*$ is a local minima. Then there exist λ_j $j = 1 \ldots l$, such that

$$\nabla f(\mathbf{x}^*) + \sum_{j=1}^{l} \lambda_j \nabla h_j(\mathbf{x}^*) = \mathbf{0}$$

$$\lambda_j h_j(\mathbf{x}^*) = 0; \quad j = 1, \ldots, l$$

$$\lambda_j \geq 0; \quad j = 1, \ldots, l$$

- These are first order KKT necessary conditions.
- KKT point: $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$, where $\boldsymbol{\lambda}^* = [\lambda_1^* \ \lambda_2^* \ \ldots \ \lambda_l^*]^T$.

Lagrangian Function

- Lagrangian function is represented as:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{j=1}^{l} \lambda_j h_j(\mathbf{x})$$

- KKT Conditions imply

$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = 0$

$\lambda_j^* \geq 0; \;\; j = 1 \ldots l$ ($\lambda' s$ are called Lagrange multipliers.)

$\lambda_j^* h_j(\mathbf{x}^*) = 0; \;\; j = 1 \ldots l$ (Complementary slackness conditions.)

$\lambda_j^* = 0; \;\; \forall j \in \mathcal{A}(\mathbf{x}^*)$

- Note that for active constraints, $\lambda_j^* h_j(\mathbf{x}^*) = 0$ because $h_j(\mathbf{x}^*) = 0$. Thus, λ_j^* can be zero or greater than zero.
- For non-active constraints, $h_j(\mathbf{x}^*) < 0$. Thus, $\lambda_j^* h_j(\mathbf{x}^*) = 0$ implies $\lambda_j^* = 0$.

Example 1

$$\begin{aligned} \min_{x_1,x_2} \quad & x_1^2 + x_2^2 \\ s.t. \quad & h_1 : \ x_1 + x_2 \geq 1 \\ & h_2 : \ x_2 \leq 1 \end{aligned}$$

Case 1:

- Let h_1 and h_2 both are active constraints. $\mathbf{z} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
- At this point both h_1 and h_2 are active. Thus, $\lambda_1^* \geq 0$ and $\lambda_2^* \geq 0$.
- $\mathcal{L} = x_1^2 + x_2^2 + \lambda_1^*(x_2 - 1) + \lambda_2^*(1 - x_1 - x_2)$
- $\nabla f(\mathbf{z}) = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$, $\nabla h_2(\mathbf{z}) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $\nabla h_1(\mathbf{z}) = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$.
- $\nabla \mathcal{L} = \mathbf{0}$ implies $\begin{pmatrix} 0 \\ 2 \end{pmatrix} + \lambda_1^* \begin{pmatrix} -1 \\ -1 \end{pmatrix} + \lambda_2^* \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.
- This results in $\lambda_1^* = 0$ and $\lambda_2^* = -2$. Thus, $\mathbf{z} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ is not a local minima.

Example 1: Continue

Case 2:

- Assume constraint h_1 is active and h_2 is inactive. Thus, $x_1 + x_2 = 1$ and $x_2 < 1$ at the solution.
- This implies $\lambda_2^* = 0$.
- KKT condition results $\nabla f(\mathbf{x}^*) + \lambda_1^* \nabla h_1(\mathbf{x}^*) = \mathbf{0}$.
- Which results in $\begin{pmatrix} 2x_1^* \\ 2x_2^* \end{pmatrix} + \lambda_1^* \begin{pmatrix} -1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.
- This means $2x_1^* - \lambda_1^* = 0$, $2x_2^* - \lambda_1^* = 0$ and $x_1^* + x_2^* = 1$.
- This results in $x_1^* = x_2^* = 0.5$ and $\lambda_1^* = 1$ and $\lambda_2^* = 0$.
- $x_1^* = x_2^* = 0.5$ is a KKT point at which $f(\mathbf{x}^*) = 0.5$.

Example 1: Continue

Case 3:

- Assume constraint h_1 is inactive and h_2 is active. Thus, $x_1 + x_2 > 1$ and $x_2 = 1$ at the solution.
- This implies $\lambda_1^* = 0$.
- KKT condition results $\nabla f(\mathbf{x}^*) + \lambda_2^* \nabla h_2(\mathbf{x}^*) = \mathbf{0}$.
- Which results in $\begin{pmatrix} 0 \\ 2 \end{pmatrix} + \lambda_2^* \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.
- This means $\lambda_2^* = -2$, which is not a feasible solution.

Thus, $x_1^* = x_2^* = 0.5$ **is the minima.**

Finding KKT Point

Consider the minimization problem.

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ s.t. \quad & h_j(\mathbf{x}) \leq \mathbf{0} \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned}$$

- At a local minimum, active set is unknown.
- Need to investigate all possible active sets for finding KKT points.

Is KKT Point Always Optimal?

KKT Point may not be optimal always. See the example below.

- Consider min $-x^2$ such that $x \leq 0$.
- $\mathcal{L}(x, \lambda) = -x^2 + \lambda x$
- $\frac{\partial \mathcal{L}}{\partial x} = 0 \Rightarrow -2x + \lambda = 0$
- At x^*, the constraint is active. Thus, $x^* = 0$.
- $\frac{\partial \mathcal{L}(x^*, \lambda^*)}{\partial x} = 0 \Rightarrow \lambda^* = 0$.
- (0,0) is a KKT point.
- However, $-x^2$ is unbounded in $x \leq 0$ and $x^* = 0$ is not a local minimum.

Constrained Convex Optimization

Consider the optimization problem:

$$\begin{aligned} CP: \quad & \min \ f(\mathbf{x}) \\ & s.t. \ h_j(\mathbf{x}) \leq 0, \ j = 1 \ldots l \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned}$$

- Assumption: f, $h_j; j = 1 \ldots l$ are differentiable convex functions.
- $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid h_j(\mathbf{x}) \leq 0; \ j = 1 \ldots l\}$

Result

If $\mathbf{x}^*$ is a regular point, then for $\mathbf{x}^*$ to be a global minimum of CP, first order KKT conditions are necessary and sufficient.

Necessity of the KKT Conditions Under Regularity Condition for Convex Optimization Problem

Theorem

Let $\mathbf{x}^*$ be a regular point and is an optimal solution of the problem

$$CP: \quad \min_{\mathbf{x}\in\mathbb{R}^n} \; f(\mathbf{x})$$
$$s.t. \;\; h_j(\mathbf{x}) \leq 0, \; j = 1 \ldots l$$

where $f(\mathbf{x})$ and $h_1(\mathbf{x}), \ldots, h_l(\mathbf{x})$ are continuously differentiable convex functions over $\mathbb{R}^n$. Then, there exists multipliers $\lambda_1, \ldots, \lambda_l \geq 0$, such that

$$\nabla f(\mathbf{x}^*) + \sum_{j=1}^{l} \lambda_j \nabla h_j(\mathbf{x}^*) = \mathbf{0}$$
$$\lambda_j h_j(\mathbf{x}^*) = 0; \;\; j = 1 \ldots l.$$

Sufficiency of the KKT conditions Under Regularity Condition for Convex Optimization Problems

- KKT conditions are necessary optimality conditions under the regularity condition.
- When the problem is convex, the KKT conditions are always sufficient and no further condition is required.

Theorem

Consider the convex optimization problem:

$$CP: \quad \min_{\mathbf{x}\in\mathbb{R}^n} \; f(\mathbf{x})$$
$$s.t. \;\; h_j(\mathbf{x}) \leq 0, \; j = 1 \ldots l$$

where $f(\mathbf{x})$, $h_1(\mathbf{x}), \ldots, h_l(\mathbf{x})$ are continuously differentiable convex functions over $\mathbb{R}^n$. Let there exist multipliers $\lambda_1, \ldots, \lambda_l \geq 0$ such that

$$\nabla f(\mathbf{x}^*) + \sum_{j=1}^{l} \lambda_j \nabla h_j(\mathbf{x}^*) = \mathbf{0}$$
$$\lambda_j h_j(\mathbf{x}^*) = 0; \;\; j = 1 \ldots l.$$

Then, $\mathbf{x}^*$ is an optimal solution.

Slater's Condition

Slater's Condition

Let $h_j(\mathbf{x}^*)$, $j = 1 \ldots l$ are convex functions. Slater's condition is satisfied for these inequalities if there exists a point $\hat{\mathbf{x}}$ such that

$$h_j(\hat{\mathbf{x}}) < 0;\ j = 1 \ldots l.$$

Thus, Slater's condition requires that there exists a point that strictly satisfies the constraints. In other words, the interior of the feasible set is non-empty.

- Slater's condition does not require, like in the regularity condition, an apriori knowledge on the point that is a candidate to be an optimal solution.
- Checking the validity of Slater's condition is much easier task than checking regularity.

Necessity of the KKT Conditions Under Slater's Condition for Convex Optimization Problem

Theorem

Let $\mathbf{x}^*$ be an optimal solution of the problem

$$CP: \quad \min_{\mathbf{x}\in\mathbb{R}^n} \; f(\mathbf{x})$$
$$s.t. \;\; h_j(\mathbf{x}) \leq 0, \; j = 1 \ldots l$$

where $f(\mathbf{x})$ and $h_1(\mathbf{x}), \ldots, h_l(\mathbf{x})$ are continuously differentiable convex functions over $\mathbb{R}^n$. In addition, suppose there exists a point $\hat{\mathbf{x}}$ such that

$$h_j(\hat{\mathbf{x}}) < 0; \; j = 1 \ldots l.$$

Then, there exists multipliers $\lambda_1, \ldots, \lambda_l \geq 0$, such that

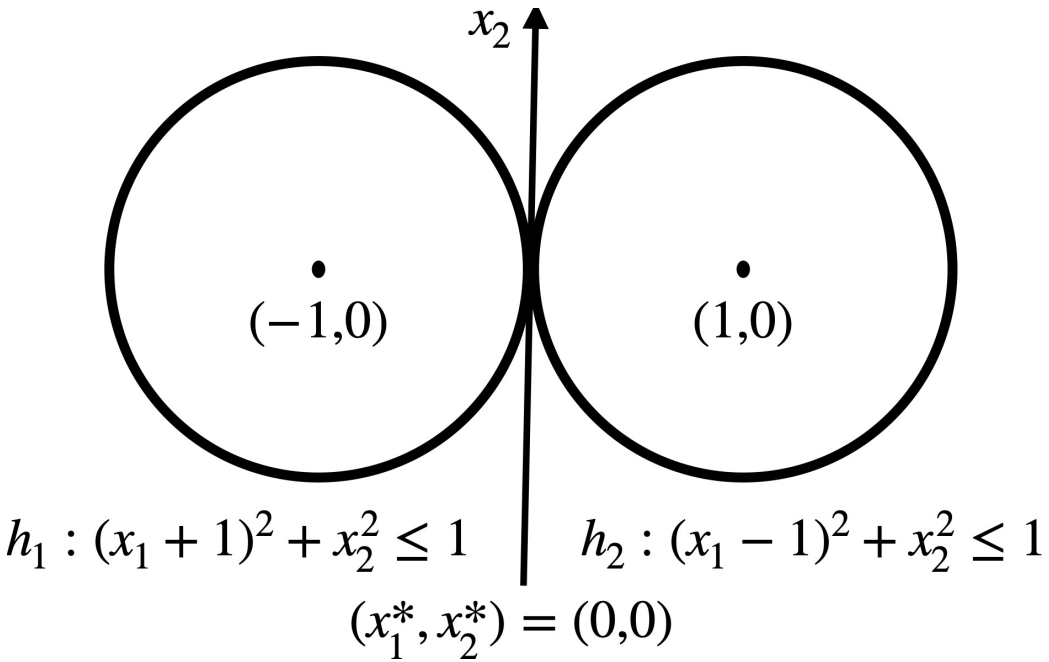
$$\nabla f(\mathbf{x}^*) + \sum_{j=1}^{l} \lambda_j \nabla h_j(\mathbf{x}^*) = \mathbf{0}$$
$$\lambda_j h_j(\mathbf{x}^*) = 0; \;\; j = 1 \ldots l.$$

Not all Problems Satisfy Slater's Condition

Consider the optimization problem as follows.

$$\begin{aligned} \min \quad & x_1 + x_2 \\ & (x_1+1)^2 + x_2^2 \leq 1 \\ & (x_1-1)^2 + x_2^2 \leq 1 \end{aligned}$$

Here, Feasible set $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^2 \mid (x_1+1)^2 + x_2^2 \leq 1,\ (x_1-1)^2 + x_2^2 \leq 1\} = \{(0,0)\}$. At this point, both the constraints are satisfied with equality. Thus, it does not satisfy Slater's condition.



Optimization Methods (CS1.404), Spring 2025

Naresh Manwani

Machine Learning Lab, IIIT-H

April 23, 2025

General Nonlinear Constrained Optimization Problems

Consider the optimization problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ s.t. \quad & h_j(\mathbf{x}) \leq 0,\ j = 1 \ldots l \\ & e_i(\mathbf{x}) = 0;\ i = 1 \ldots m \end{aligned}$$

where $f(\mathbf{x})$, $h_1(\mathbf{x}), \ldots, h_l(\mathbf{x})$, $e_1(\mathbf{x}), \ldots, e_m(\mathbf{x})$ are smooth functions over $\mathbb{R}^n$.

- Let $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid h_j(\mathbf{x}) \leq 0,\ j = 1 \ldots l;\ e_i(\mathbf{x}) = 0;\ i = 1 \ldots m\}$ be the feasible set.
- Let $\mathbf{x}^* \in \mathcal{X}$ and $\mathcal{A}(\mathbf{x}^*)$ denote set of active inequality constraints at $\mathbf{x}^*$. Then, $\mathcal{A}(\mathbf{x}^*) = \{j \in \{1, \ldots, l\} \mid h_j(\mathbf{x}^*) = 0\}$.

Regular Point for General Constraint Problems

Definition

A point $\mathbf{x}^* \in \mathcal{X}$ is said to be a regular point, if the gradient vectors $\nabla h_j(\mathbf{x}^*),\ j \in \mathcal{A}(\mathbf{x}^*)$ and $\nabla e_i(\mathbf{x}^*),\ i \in \{1, \ldots, m\}$ are linearly independent, where $\mathcal{A}(\mathbf{x}^*) = \{j \in \{1, \ldots, l\} \mid h_j(\mathbf{x}^*) = 0\}$. Which means,

$$\sum_{j \in I(\mathbf{x}^*)} \lambda_j \nabla h_j(\mathbf{x}^*) + \sum_{i=1}^{m} \mu_i \nabla e_i(\mathbf{x}^*) = \mathbf{0}$$

only if $\lambda_j = 0,\ j \in \mathcal{A}(\mathbf{x}^*)$ and $\mu_i = 0,\ i = 1 \ldots m$.

KKT Necessary Conditions of First Order

Theorem

Consider the optimization problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ s.t. \quad & h_j(\mathbf{x}) \leq 0,\ j = 1 \dots l \\ & e_i(\mathbf{x}) = 0;\ i = 1 \dots m \end{aligned}$$

where $f(\mathbf{x})$, $h_1(\mathbf{x}), \dots, h_l(\mathbf{x})$, $e_1(\mathbf{x}), \dots, e_m(\mathbf{x})$ are smooth functions over $\mathbb{R}^n$. If $\mathbf{x}^* \in \mathcal{X}$ is a local minimum and a regular point, then there exist unique vectors $\boldsymbol{\lambda}^* = [\lambda_1^* \ \dots \ \lambda_l^*]^\top \in \mathbb{R}^l_+$ and $\boldsymbol{\mu}^* = [\mu_1^* \ \dots \ \mu_m^*]^\top \in \mathbb{R}^m$, such that

$$\nabla f(\mathbf{x}^*) + \sum_{j=1}^{l} \lambda_j^* \nabla h_j(\mathbf{x}^*) + \sum_{i=1}^{m} \mu_i^* \nabla e_i(\mathbf{x}^*) = \mathbf{0}$$

$$\lambda_j^* h_j(\mathbf{x}^*) = 0,\ j = 1 \dots l$$

$$h_j(\mathbf{x}^*) \leq 0,\ j = 1 \dots l$$

$$e_i(\mathbf{x}^*) = 0,\ i = 1 \dots m$$

KKT Point: A point $(\mathbf{x}^* \in \mathcal{X}, \boldsymbol{\lambda} \in \mathbb{R}^l_+, \boldsymbol{\mu} \in \mathbb{R}^m)$ satisfying above conditions is called KKT point.

Necessity and Sufficiency of KKT Conditions for Convex Optimization Problem Under Slater's Condition

Theorem

Consider the convex optimization problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ s.t. \quad & h_j(\mathbf{x}) \leq 0,\ j = 1 \dots l \\ & e_i(\mathbf{x}) = 0;\ i = 1 \dots m \end{aligned}$$

where

- $f(\mathbf{x}),\ h_1(\mathbf{x}), \dots, h_l(\mathbf{x})$ are smooth convex functions over $\mathbb{R}^n$.
- $e_i(\mathbf{x}) = \mathbf{a}_i^\top \mathbf{x} - b_i,\ i = 1 \dots m$

Let $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid h_j(\mathbf{x}) \leq 0,\ j = 1 \dots l;\ e_i(\mathbf{x}) = 0;\ i = 1 \dots m\}$ be the feasible set and it satisfies Slater's Condition. Then first order KKT conditions are necessary and sufficient for a global minima of convex optimization problem above.

Constraint Classification

Strongly Active Constraint

- An inequality constraint is strongly active if it belongs to $\mathcal{A}(\mathbf{x}^*)$ and it has strictly positive Lagrange multiplier ($\lambda_j > 0$).
- An equality constraint is strongly active if its Lagrange multiplier is strictly non-zero ($\mu_i \neq 0$).

Weakly Active Constraint

- An inequality constraint weakly active at if it belongs to $\mathcal{A}(\mathbf{x}^*)$ and it has a zero-valued Lagrange multiplier ($\lambda_j = 0$).
- An equality constraint is weakly active at if it has a zero-valued Lagrange multiplier ($\mu_i = 0$).

Inactive Constraint

An inequality constraint is inactive at if it does not belongs to $\mathcal{A}(\mathbf{x}^*)$. Thus, it has a zero-valued Lagrange multiplier ($\lambda_j = 0$).

Weakly Active and Inactive Constraints do not participate !!

Example: Constraint Classification

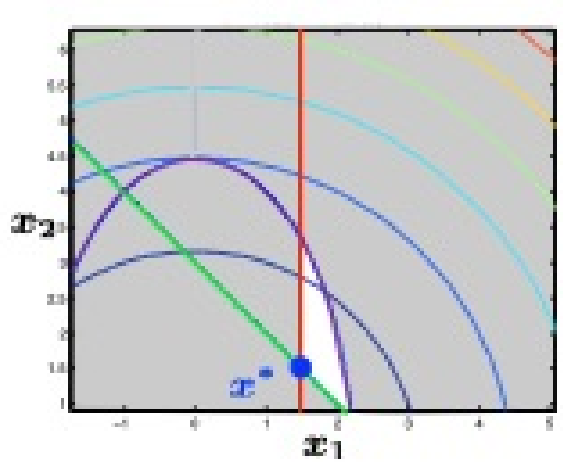
Consider Optimization Problem as follows.

$$\min_{(x_1,x_2)\in\mathbb{R}^2} \quad x_1^2 + x_2^2$$

$$s.t. \quad h_1(x_1, x_2) = x_1 + x_2 - 3 \geq 0 \qquad \text{(strongly active)}$$

$$h_2(x_1, x_2) = x_1 - 1.5 \geq 0 \qquad \text{(weakly active)}$$

$$h_3(x_1, x_2) = -x_1^2 - 4x_2^2 + 5 \geq 0 \qquad \text{(inactive)}$$



The solution is unchanged even if constraints h_2 and h_3 are removed.

Second Order Necessary Conditions

Theorem

Let $\mathbf{x}^*$ be a local minimum of the optimization problem described below.

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ s.t. \quad & h_j(\mathbf{x}) \leq 0,\ j = 1 \dots l \\ & e_i(\mathbf{x}) = 0;\ i = 1 \dots m \end{aligned}$$

where $f(\mathbf{x}) \in \mathbb{C}^2(\mathbb{R}^n)$, $h_j \in \mathbb{C}^2(\mathbb{R}^n)$, $j = 1 \dots l$ and $e_i \in \mathbb{C}^2(\mathbb{R}^n)$, $i = 1 \dots m$. Suppose that $\mathbf{x}^*$ is regular, which means $\nabla h_j(\mathbf{x}^*)$, $j \in \mathcal{A}(\mathbf{x}^*)$ and $\nabla e_i(\mathbf{x}^*)$, $i \in \{1, \dots, m\}$ are linearly independent, where $\mathcal{A}(\mathbf{x}^*) = \{j \in \{1, \dots, l\} \mid h_j(\mathbf{x}^*) = 0\}$.

1. Then there exist $\boldsymbol{\lambda}^* = [\lambda_1^* \ \dots \ \lambda_l^*]^\top \in \mathbb{R}^l_+$ and $\boldsymbol{\mu}^* = [\mu_1^* \ \dots \ \mu_m^*]^\top \in \mathbb{R}^m$, such that

$$\nabla f(\mathbf{x}^*) + \sum_{j=1}^{l} \lambda_j^* \nabla h_j(\mathbf{x}^*) + \sum_{i=1}^{m} \mu_i^* \nabla e_i(\mathbf{x}^*) = \mathbf{0}$$

$$\lambda_j^* h_j(\mathbf{x}^*) = 0,\ j = 1 \dots l$$

2. and $\mathbf{y}^\top [\nabla^2 f(\mathbf{x}^*) + \sum_{j=1}^{l} \lambda_j^* \nabla h_j^2(\mathbf{x}^*) + \sum_{i=1}^{m} \mu_i^* \nabla e_i^2(\mathbf{x}^*)]\mathbf{y} \geq 0$ for all $\mathbf{y} \in \hat{T}(\mathbf{x}^*)$ where

$$\hat{T}(\mathbf{x}^*) = \{\mathbf{y} \in \mathbb{R}^n \mid \nabla h_j(\mathbf{x}^*)^\top \mathbf{y} = 0,\ j \in \mathcal{A}(\mathbf{x}^*);\ \nabla e_i(\mathbf{x}^*)^\top \mathbf{y} = 0,\ i = 1 \dots m\}$$

Second Order Sufficiency Conditions

Theorem

Consider the optimization problem described below.

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ s.t. \quad & h_j(\mathbf{x}) \leq 0,\ j = 1 \dots l \\ & e_i(\mathbf{x}) = 0;\ i = 1 \dots m \end{aligned}$$

where $f(\mathbf{x}) \in \mathbb{C}^2(\mathbb{R}^n)$, $h_j \in \mathbb{C}^2(\mathbb{R}^n)$, $j = 1 \dots l$ and $e_i \in \mathbb{C}^2(\mathbb{R}^n)$, $i = 1 \dots m$. Let $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum_{j\in\mathcal{A}(\mathbf{x})} \lambda_j h_j(\mathbf{x}) + \sum_{i=1}^{m} \mu_i e_i(\mathbf{x})$. Suppose there exist a feasible point $\mathbf{x}^*$, $\boldsymbol{\lambda}^* = [\lambda_1^* \ \dots \ \lambda_l^*]^\top \in \mathbb{R}^l_+$ and $\boldsymbol{\mu} = [\mu_1^* \ \dots \ \mu_m^*]^\top \in \mathbb{R}^m$, such that

1. $\lambda_j^* h_j(\mathbf{x}^*) = 0,\ j = 1 \dots l$ and $\nabla\mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \mathbf{0}$
2. Also, for all $\mathbf{y} \in \tilde{T}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$, $\mathbf{y} \neq \mathbf{0}$, we have $\mathbf{y}^\top \nabla^2\mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)\mathbf{y} > 0$. where

$$\begin{aligned} \tilde{T}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \{\mathbf{y} \in \mathbb{R}^n \mid & \nabla h_j(\mathbf{x}^*)^\top \mathbf{y} = 0,\ \forall j \in \hat{A}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*); \\ & \nabla e_i(\mathbf{x}^*)^\top \mathbf{y} = 0,\ i = 1 \dots m\}. \end{aligned}$$

for $\hat{A}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \{j \in \{1, \dots, l\} \mid h_j(\mathbf{x}^*) = 0, \lambda_j^* > 0\}$.

Then $\mathbf{x}^*$ is a local minimizer.

Example 1

- Consider the problem

$$\begin{aligned} \min \ & (x_1 - 1)^2 + x_2 - 2 \\ & h(x_1, x_2) = x_2 - x_1 - 1 = 0 \\ & g(x_1, x_2) = x_1 + x_2 - 2 \leq 0 \end{aligned}$$

- For all $(x_1, x_2) \in \mathbb{R}^2$, we have $\nabla h(x_1, x_2) = [-1 \ \ 1]^\top$ and $\nabla g(x_1, x_2) = [1 \ \ 1]^\top$.
- Thus, $\nabla h(x_1, x_2)$ and $\nabla g(x_1, x_2)$ are linearly independent and hence all feasible points are regular.
- $\nabla f(x_1, x_2) = [2(x_1 - 1) \ \ 1]^\top$.
- KKT conditions are as follows.

$$\nabla f(x_1, x_2) + \lambda \nabla g(x_1, x_2) + \mu \nabla h(x_1, x_2) = [2x_1 - 2 - \mu + \lambda, 1 + \mu + \lambda]^\top = [0, 0]^\top$$

$$\lambda(x_1 + x_2 - 2) = 0$$

$$\lambda \geq 0$$

$$x_2 - x_1 - 1 = 0$$

$$x_1 + x_2 - 2 \leq 0$$

- To find points that satisfy above conditions, we analyse two cases: (a) $\lambda > 0$, (b) $\lambda = 0$.

Example 1 - Case 1 ($\lambda > 0$)

- $\lambda > 0$ implies that $x_1 + x_2 - 2 = 0$. Thus, we are faced with a system of four linear equations.

$$
\begin{aligned}
2x_1 - 2 - \mu + \lambda &= 0 \\
1 + \mu + \lambda &= 0 \\
x_2 - x_1 - 1 &= 0 \\
x_1 + x_2 - 2 &= 0
\end{aligned}
$$

- Solving the above system of equations, we obtain $x_1 = \frac{1}{2},\ x_2 = \frac{3}{2},\ \lambda = 0,\ \mu = -1$.
- However, this is not a legitimate solution to KKT condition, because we obtain $\lambda = 0$, which contradicts the assumption that $\lambda > 0$.

Example 1 - Case 2 ($\lambda = 0$)

- Assuming $\lambda = 0$, we are faced with a system of three linear equations.

$$2x_1 - 2 - \mu = 0$$
$$1 + \mu = 0$$
$$x_2 - x_1 - 1 = 0$$

  And the solution must satisfy $x_1 + x_2 - 2 \leq 0$.

- Solving the above system of equations, we obtain $x_1 = \frac{1}{2},\ x_2 = \frac{3}{2},\ \mu = -1$.
- Note that $(x_1^*, x_2^*) = [\frac{1}{2}, \frac{3}{2}]^\top$ satisfy the constraint $x_1 + x_2 - 2 \leq 0$.
- $(x_1^*, x_2^*) = [\frac{1}{2}, \frac{3}{2}]^\top$ is a candidate for being a minimizer.
- We now verify that the point $(x_1^*, x_2^*) = [\frac{1}{2}, \frac{3}{2}]^\top$, $\lambda^* = 0$ and $\mu^* = -1$ satisfy the second order sufficient conditions.
- For this, we form the matrix

$$\nabla^2 \mathcal{L}(x_1^*, x_2^*, \lambda^*, \mu^*) = \nabla^2 f(x_1^*, x_2^*) + \mu^* \nabla^2 h(x_1^*, x_2^*) + \lambda^* \nabla g(x_1^*, x_2^*)$$
$$= \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$$

- We then find the subspace $\tilde{T}(x_1^*, x_2^*, \lambda^*, \mu^*) = \{\mathbf{y} \mid \nabla h(x_1^*, x_2^*)^\top \mathbf{y} = 0\}$.
- Note that $\lambda^* = 0$, the active constraint $x_1 + x_2 = 2$ does not enter into the computation of $\tilde{T}(x_1^*, x_2^*, \lambda^*, \mu^*)$.

Example 1 - Case 2 ($\lambda = 0$)

- We have $\tilde{T}(x_1^*, x_2^*, \lambda^*, \mu^*) = \{\mathbf{y} \mid [-1, 1]\mathbf{y} = 0\} = \{[a, a]^\top \mid a \in \mathbb{R}\}$.
- We then check for positive semi-definiteness of $\nabla^2 \mathcal{L}(x_1^*, x_2^*, \lambda^*, \mu^*)$ on $\tilde{T}(x_1^*, x_2^*, \lambda^*, \mu^*)$.
- We have $\mathbf{y}^\top \nabla^2 \mathcal{L}(x_1^*, x_2^*, \lambda^*, \mu^*)\mathbf{y} = [a, a] \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ a \end{bmatrix} = 2a^2$.
- Thus, $\nabla^2 \mathcal{L}(x_1^*, x_2^*, \lambda^*, \mu^*)$ is positive definite on $\tilde{T}(x_1^*, x_2^*, \lambda^*, \mu^*)$.
- By second order sufficient conditions, we conclude that $(x_1^*, x_2^*) = [\frac{1}{2}, \frac{3}{2}]^\top$ is a strict local minimizer.

Test Positive Definiteness in a Subspace

- In the second-order sufficiency conditions requires that $\mathbf{d}^\top \nabla^2 \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)\mathbf{d} > 0$ for all $\mathbf{d} \in \tilde{T}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$, $\mathbf{d} \neq \mathbf{0}$, where

  $$\tilde{T}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \{\mathbf{d} \in \mathbb{R}^n \mid \nabla h_j(\mathbf{x}^*)^\top \mathbf{d} = 0,\ j \in \hat{A};\ \nabla e_i(\mathbf{x}^*)^\top \mathbf{d} = 0,\ i = 1 \ldots m\}.$$

  for $\hat{A} = \{j \in \{1, \ldots, l\} \mid h_j(\mathbf{x}^*) = 0, \lambda_j^* > 0\}$.

- Let $Q = \nabla^2 \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ and $B = \begin{bmatrix} \nabla e_1(\mathbf{x}^*)^\top \\ \vdots \\ \nabla e_m(\mathbf{x}^*)^\top \\ \nabla h_{j_1}(\mathbf{x}^*)^\top \\ \vdots \\ \nabla h_{j_{|\hat{A}|}}(\mathbf{x}^*)^\top \end{bmatrix}$.

- Then, second-order sufficiency conditions requires that $\mathbf{d}^\top Q\mathbf{d} > 0$, $\forall \mathbf{d} \neq \mathbf{0}$ such that $B\mathbf{d} = \mathbf{0}$. (In this case, the subspace is the null space of matrix B.) This test itself might be a nonconvex optimization problem.

Test Positive Definiteness in a Subspace

- Consider any vector $\mathbf{u} \in \mathbb{R}^n$ can be decomposed into two orthogonal components: (a) one which lies in the null space of matrix B, (b) one which lies in the space spanned by the rows of B.
  - If we project $\mathbf{u}$ in the row space of B, we can get the component of $\mathbf{u}$ which lies in the row space of B. The corresponding projection matrix is $P = B^\top (BB^\top)^{-1} B$.
  - Thus, the component of $\mathbf{u}$ in the null space of B is $\mathbf{u} - B^\top (BB^\top)^{-1} B\mathbf{u} = [I - B^\top (BB^\top)^{-1} B]\mathbf{u}$.
- Thus, $\mathbf{d}$ is in the null space of matrix B if and only if $\mathbf{d} = (I - B^\top (BB^\top)^{-1} B)\mathbf{u} = P_B \mathbf{u}$ for some $\mathbf{u} \in \mathbb{R}^n$.
- Thus, the test becomes whether or not

$$\mathbf{u}^\top P_B Q P_B \mathbf{u} > 0, \ \forall \mathbf{u} \in \mathbb{R}^n.$$

- That is, we just need to test positive definiteness of matrix $P_B Q P_B$ as usual.

Dual Problem

- Consider the optimization problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ s.t. \quad & h_j(\mathbf{x}) \leq 0,\ j = 1 \dots l \\ & e_i(\mathbf{x}) = 0;\ i = 1 \dots m \end{aligned}$$

  where $f(\mathbf{x})$, h_j, $j = 1 \dots l$ and e_i, $i = 1 \dots m$ are sufficiently smooth functions over $\mathbb{R}^n$.
- This problem is referred as primal problem. Let p^* be the optimal value of the above problem.
- The Lagrangian of the problem is

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum_{j=1}^{l} \lambda_j h_j(\mathbf{x}) + \sum_{i=1}^{m} \mu_i e_i(\mathbf{x})$$

  where $\boldsymbol{\lambda} = [\lambda_1 \ \dots \ \lambda_l]^\top \in \mathbb{R}^l_+$ are nonnegative Lagrange multipliers associated with the inequality constraints and $\boldsymbol{\mu} = [\mu_1 \ \dots \ \mu_m]^\top \in \mathbb{R}^m$ are the Lagrange multipliers associated with the equality constraints.

Dual Problem

- The dual objective function $g : \mathbb{R}^l_+ \times \mathbb{R}^m \rightarrow \mathbb{R} \cup \{\infty\}$ is defined to be

$$g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \min_{\mathbf{x}} \ \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$$

- Note that above minimization problem can be unbounded, i.e., there may be values $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ for which $g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = -\infty$.
- We define the domain of dual function as

$$dom(g) = \{(\boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{R}^l_+ \times \mathbb{R}^m \mid g(\boldsymbol{\lambda}, \boldsymbol{\mu}) > -\infty\}$$

- The **Dual Problem** is defined as

$$\begin{aligned} g^* = \max \ & g(\boldsymbol{\lambda}, \boldsymbol{\mu}) \\ & s.t. \ (\boldsymbol{\lambda}, \boldsymbol{\mu}) \in dom(g) \end{aligned}$$

Theorem: Convexity of the Dual Problem

Domain of dual function g is convex and g is a concave function over the $dom(g)$.

Example 1: Linear Programming

- Consider the linear programming problem

$$\min_{\mathbf{x}} \mathbf{c}^\top \mathbf{x}$$
$$s.t.\ A\mathbf{x} \leq \mathbf{b}$$

where $\mathbf{c} \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}$, and $\mathbf{b} \in \mathbb{R}^m$. We assume that the problem is feasible (which means, constraint set is nonempty).

- The Lagrangian function is

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{c}^\top \mathbf{x} + \boldsymbol{\lambda}^\top (A\mathbf{x} - \mathbf{b})$$

where $\boldsymbol{\lambda} = [\lambda_1 \ \ldots \ \lambda_m]^\top \in \mathbb{R}^m_+$ are nonnegative Lagrange multipliers associated with the inequality constraints

- The dual objective function is

$$\begin{aligned} g(\boldsymbol{\lambda}) &= \min_{\mathbf{x}} \ \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \min_{\mathbf{x}} \ \mathbf{c}^\top \mathbf{x} + \boldsymbol{\lambda}^\top (A\mathbf{x} - \mathbf{b}) \\ &= \min_{\mathbf{x}} \ (\mathbf{c} + A^\top \boldsymbol{\lambda})^\top \mathbf{x} - \mathbf{b}^\top \boldsymbol{\lambda} \\ &= \begin{cases} -\mathbf{b}^\top \boldsymbol{\lambda}, & \mathbf{c} + A^\top \boldsymbol{\lambda} = \mathbf{0} \\ -\infty, & \text{else} \end{cases} \end{aligned}$$

- The dual problem is

$$\max \ -\mathbf{b}^\top \boldsymbol{\lambda}$$
$$s.t.\ \mathbf{c} + A^\top \boldsymbol{\lambda} = \mathbf{0}$$
$$\boldsymbol{\lambda} \geq \mathbf{0}$$

Example 2: Strictly Convex Quadratic Programming

- Consider the linear programming problem

$$\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^\top Q\mathbf{x} + \mathbf{c}^\top \mathbf{x}$$
$$s.t.\ A\mathbf{x} \leq \mathbf{b}$$

  where $Q \in \mathbb{R}^{n\times n}$ is positive definite, $\mathbf{c} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$.

- The Lagrangian function is

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{x}^\top Q\mathbf{x} + \mathbf{c}^\top \mathbf{x} + \boldsymbol{\lambda}^\top (A\mathbf{x} - \mathbf{b}) = \frac{1}{2}\mathbf{x}^\top Q\mathbf{x} + (A^\top \boldsymbol{\lambda} + \mathbf{c})^\top \mathbf{x} - \mathbf{b}^\top \boldsymbol{\lambda}$$

  where $\boldsymbol{\lambda} = [\lambda_1 \ \ldots \ \lambda_m]^\top \in \mathbb{R}^m_+$ are nonnegative Lagrange multipliers.

- To find the dual function, we minimize the Lagrangian with respect to $\mathbf{x}$. The minimizer is attained at the stationary point which is the solution to

$$\nabla_{\mathbf{x}}\, \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = Q\mathbf{x}^* + A^\top \boldsymbol{\lambda} + \mathbf{c} = \mathbf{0} \Rightarrow \mathbf{x}^* = -Q^{-1}(A^\top \boldsymbol{\lambda} + \mathbf{c})$$

- Using $g(\boldsymbol{\lambda}) = \min_{\mathbf{x}}\ \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda})$, we obtain

$$g(\boldsymbol{\lambda}) = \frac{1}{2}(A^\top \boldsymbol{\lambda} + \mathbf{c})^\top Q^{-1} Q Q^{-1}(A^\top \boldsymbol{\lambda} + \mathbf{c}) - (A^\top \boldsymbol{\lambda} + \mathbf{c})^\top Q^{-1}(A^\top \boldsymbol{\lambda} + \mathbf{c}) - \mathbf{b}^\top \boldsymbol{\lambda}$$
$$= -\frac{1}{2}\boldsymbol{\lambda}^\top A Q^{-1} A^\top \boldsymbol{\lambda} - (AQ^{-1}\mathbf{c} + \mathbf{b})^T \boldsymbol{\lambda} - \mathbf{c}^\top Q^{-1}\mathbf{c}$$

- The dual problem is

$$\max\ -\frac{1}{2}\boldsymbol{\lambda}^\top A Q^{-1} A^\top \boldsymbol{\lambda} - (AQ^{-1}\mathbf{c} + \mathbf{b})^T \boldsymbol{\lambda} - \mathbf{c}^\top Q^{-1}\mathbf{c}$$
$$s.t.\ \boldsymbol{\lambda} \geq \mathbf{0}$$

Weak Duality Theorem

Theorem

Consider the primal problem

$$\begin{aligned} p^* = \min_{\mathbf{x} \in \mathbb{R}^n} \quad & f(\mathbf{x}) \\ s.t. \quad & h_j(\mathbf{x}) \leq 0, \; j = 1 \ldots l \\ & e_i(\mathbf{x}) = 0; \; i = 1 \ldots m \end{aligned}$$

and dual problem

$$\begin{aligned} d^* = \max \; & g(\boldsymbol{\lambda}, \boldsymbol{\mu}) \\ s.t. \; & (\boldsymbol{\lambda}, \boldsymbol{\mu}) \in dom(g) \end{aligned}$$

where $g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \min_{\mathbf{x}} \; \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$. Then,

$$d^* \leq p^*$$

Example

- Consider the problem

$$\min x_1^2 - 3x_2^2$$
$$s.t.\ x_1 = x_2^3$$

- Substituting $x_1 = x_2^3$ into the objective function, the resulting unconstrained optimization problem is $\min_{x_2}\ x_2^6 - 3x_2^2$.
- The stationary points are $x_2 = 0, \pm 1$. Thus, the candidates for optimal solution are $(0,0), (1,1), (-1,-1)$.
- It is can be easily verified that the optimal solutions are $(1,1)$ and (-1,-1) with optimal value $p^* = -2$.
- Let us consider the dual problem. The Lagrangian is

$$\mathcal{L}(x_1, x_2, \mu) = x_1^2 - 3x_2^2 + \mu(x_1 - x_2^3) = x_1^2 + \mu x_1 - 3x_2^2 - \mu x_2^3$$

- Obviously, for any value of $\mu \in \mathbb{R}$, $\min_{x_1,x_2}\ \mathcal{L}(x_1, x_2, \mu) = -\infty$.
- Hence, the dual optimal value is $d^* = -\infty$, which is an extremely poor lower bound on the primal optimal value $p^* = -2$.

Optimization Methods (CS1.404), Spring 2025

Naresh Manwani

Machine Learning Lab, IIIT-H

April 21, 2025

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY

HYDERABAD

Equality Constraint Problems

The optimization problem with equality constraints is given below.

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ s.t. \quad & e_i(\mathbf{x}) = 0;\ i = 1 \dots m \\ & h_j(\mathbf{x}) \leq 0;\ j = 1 \dots l \end{aligned}$$

where $f(\mathbf{x})$, $e_1(\mathbf{x}), \dots, e_m(\mathbf{x}), h_1(\mathbf{x}), \dots, h_l(\mathbf{x})$ are smooth functions over $\mathbb{R}^n$.

Let $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid e_i(\mathbf{x}) = 0,\ i = 1 \dots m;\ h_j(\mathbf{x}) \leq 0,\ j = 1 \dots l\}$ be the set of feasible points.

Regular Point for Equality Constraint Problems

Definition

A point $\mathbf{x}^* \mathcal{X}$ is said to be a regular point of the constraints if the gradient vectors $\nabla e_1(\mathbf{x}^*), \ldots, \nabla e_m(\mathbf{x}^*)$ and $\nabla h_j(\mathbf{x}^*)$, $\forall j \in \mathcal{A}(\mathbf{x}^*)$ are linearly independent. Let $D\mathbf{e}(\mathbf{x}^*)$ be the Jacobian matrix of $\mathbf{e} = [e_1, \ldots, e_m]^T$ at $\mathbf{x}^*$, given by

$$D\mathbf{e}(\mathbf{x}^*) = \begin{bmatrix} De_1(\mathbf{x}^*) \\ \vdots \\ e_m(\mathbf{x}^*) \end{bmatrix} = \begin{bmatrix} \nabla e_1(\mathbf{x}^*)^T \\ \vdots \\ \nabla e_m(\mathbf{x}^*)^T \end{bmatrix}$$

Then, $\mathbf{x}^*$ is regular if and only if

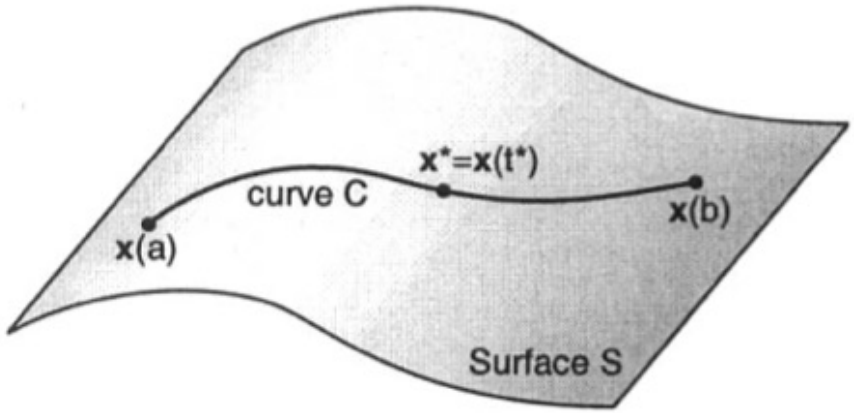
$$\text{rank} \left(\begin{bmatrix} D\mathbf{e}(\mathbf{x}^*) \\ \nabla h_{j_1}(\mathbf{x}^*)^T \\ \vdots \\ \nabla h_{j_k}(\mathbf{x}^*)^T \end{bmatrix} \right) = m + |\mathcal{A}(\mathbf{x}^*)|.$$

where $j_1, j_2, \ldots, j_k \in \mathcal{A}(\mathbf{x}^*)$ be the set of inequality constraints active at $\mathbf{x}^*$.

Curve on the Surface

Definition

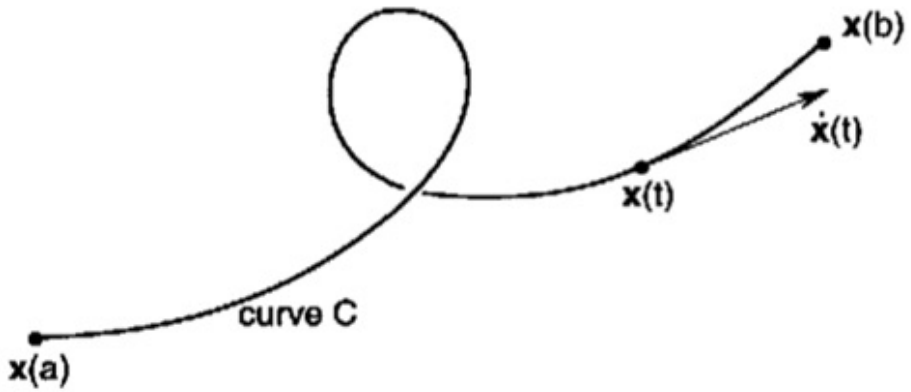
A curve C on a surface S is a set of points $\{\mathbf{x}(t) \in S \mid t \in (a, b)\}$, continuously parameterized by $t \in (a, b)$, that is, $\mathbf{x} : (a, b) \to S$ is a continuous function.



- All the points on the curve satisfy the equation describing the surface.
- The curve passes through the point $\mathbf{x}^*$ if there exist $t^* \in (a, b)$ such that $\mathbf{x}(t^*) = \mathbf{x}^*$.

Curve on the Surface

- The curve $C = \{\mathbf{x}(t) \in S \mid t \in (a, b)\}$ is differentiable if $\mathbf{x}'(t) = \frac{\partial \mathbf{x}(t)}{\partial t} = \begin{bmatrix} x_1'(t) \\ \vdots \\ x_n'(t) \end{bmatrix}$ exists for all $t \in (a, b)$.
- The curve $C = \{\mathbf{x}(t) \in S \mid t \in (a, b)\}$ is twice-differentiable if $\mathbf{x}''(t) = \frac{\partial^2 \mathbf{x}(t)}{\partial t^2} = \begin{bmatrix} x_1''(t) \\ \vdots \\ x_n''(t) \end{bmatrix}$ exists for all $t \in (a, b)$.
- The vector $\mathbf{x}'(t)$ is the direction of the tangent to the curve at $\mathbf{x}(t)$.

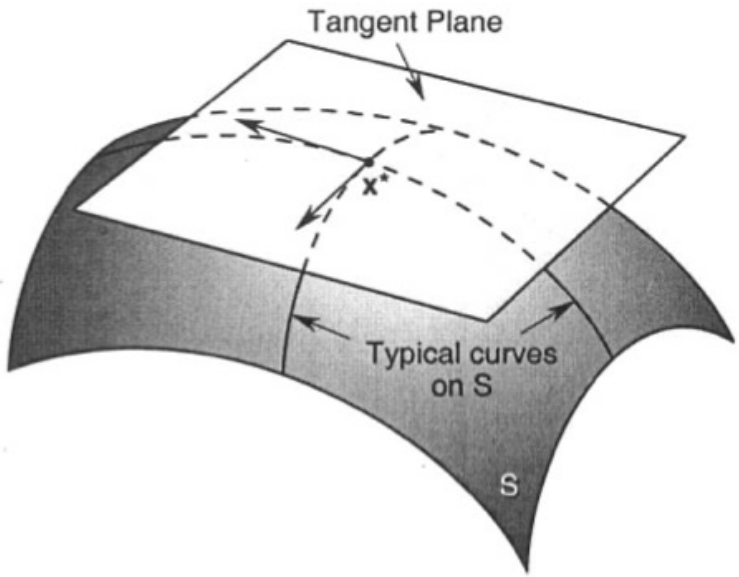


Tangent Space

Definition

Tangent space at a point $\mathbf{x}^*$ on the surface
$S = \{\mathbf{x} \in \mathbb{R}^n \mid e_1(\mathbf{x}^*) = 0, \ldots, e_m(\mathbf{x}^*) = 0\}$ is the set

$$\begin{aligned} T(\mathbf{x}^*) &= \{\mathbf{d} \mid D\mathbf{e}(\mathbf{x}^*)\mathbf{d} = \mathbf{0}\} \\ &= \{\mathbf{d} \mid \nabla e_1(\mathbf{x}^*)^T\mathbf{d} = 0, \ldots, \nabla e_m(\mathbf{x}^*)^T\mathbf{d} = 0\} \end{aligned}$$



- Tangent space at $\mathbf{x}^*$ is the null-space of $D\mathbf{e}(\mathbf{x}^*)$, which is a subspace of $\mathbb{R}^n$.
- Assuming $\mathbf{x}^*$ is a regular point, dimension of the tangent space $T(\mathbf{x}^*)$ is $n - m$.
- Tangent space passes through the origin.

Necessary Condition for Local Optimality

Theorem

Consider the optimization problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ s.t. \quad & e_i(\mathbf{x}) = 0;\ i = 1 \ldots m \\ & h_j(\mathbf{x}) \leq 0;\ j = 1 \ldots l \end{aligned}$$

Let $\tilde{\mathcal{D}}(\mathbf{x}), \tilde{\mathcal{H}}(\mathbf{x})$ and $\tilde{\mathcal{E}}(\mathbf{x})$ be defined as follows.

$$\begin{aligned} \tilde{\mathcal{D}}(\mathbf{x}) &= \{\mathbf{d} \in \mathbb{R}^n \mid \nabla f(\mathbf{x})^T \mathbf{d} < 0\} \\ \tilde{\mathcal{H}}(\mathbf{x}) &= \{\mathbf{d} \in \mathbb{R}^n \mid \nabla h_j(\mathbf{x})^T \mathbf{d} < 0,\ j \in \mathcal{A}(\mathbf{x})\} \\ \tilde{\mathcal{E}}(\mathbf{x}) &= \{\mathbf{d} \in \mathbb{R}^n \mid \nabla e_i(\mathbf{x})^T \mathbf{d} = 0,\ i = 1 \ldots l\} \end{aligned}$$

Let $\mathbf{x}^*$ be a local minimum of this optimization problem and $\nabla e_1(\mathbf{x}^*), \ldots, \nabla e_m(\mathbf{x}^*)$ are linearly independent, then $\tilde{\mathcal{D}}(\mathbf{x}^*) \cap \tilde{\mathcal{H}}(\mathbf{x}^*) \cap \tilde{\mathcal{E}}(\mathbf{x}^*) = \phi$.

Another Theorem of Alternatives

Theorem

Let $A_1 \in \mathbb{R}^{m \times d}$ and $A_2 \in \mathbb{R}^{n \times d}$. Then one of the two systems has a solution.

1. There exists $\mathbf{d} \in \mathbb{R}^d$ such that $A_1\mathbf{d} < \mathbf{0}$ and $A_2\mathbf{d} = \mathbf{0}$.
2. There exists vectors $\mathbf{p}_1 \in \mathbb{R}^m$ $(\mathbf{p}_1 \geq \mathbf{0})$ and $\mathbf{p}_2 \in \mathbb{R}^n$ such that $A_1^T\mathbf{p}_1 + A_2^T\mathbf{p}_2 = \mathbf{0}$.

Fritz John Necessary Conditions

Theorem

Consider the optimization problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ s.t. \quad & e_i(\mathbf{x}) = 0;\ i = 1 \ldots m \\ & h_j(\mathbf{x}) \leq 0;\ j = 1 \ldots l \end{aligned}$$

Let $\mathbf{x}^*$ be a local minimum of this optimization problem. Let $\mathcal{A}(\mathbf{x}^*) = \{j \in \{1, \ldots, l\} \mid h_j(\mathbf{x}^*) = 0\}$ be the set of active inequality constraints at $\mathbf{x}^*$. Then, there exist $\lambda_0 \geq 0$ and $\lambda_j \geq 0$, $\forall j \in \mathcal{A}(\mathbf{x}^*)$ and $\mu_1, \ldots, \mu_m \in \mathbb{R}$ such that

$$\lambda_0 \nabla f(\mathbf{x}^*) + \sum_{j\in\mathcal{A}(\mathbf{x}^*)} \nabla h_j(\mathbf{x}^*) + \sum_{i=1}^{m} \nabla e_i(\mathbf{x}^*) = \mathbf{0}$$

KKT Necessary Conditions

Theorem

Consider the optimization problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ s.t. \quad & e_i(\mathbf{x}) = 0;\ i = 1 \dots m \\ & h_j(\mathbf{x}) \leq 0;\ j = 1 \dots l \end{aligned}$$

Let $\mathbf{x}^*$ be a local minimum of this optimization problem and a regular point. Let $\mathcal{A}(\mathbf{x}^*) = \{j \in \{1, \dots, l\} \mid h_j(\mathbf{x}^*) = 0\}$ be the set of active inequality constraints at $\mathbf{x}^*$. Then, there exist $\lambda_j \geq 0$, $\forall j \in \mathcal{A}(\mathbf{x}^*)$ and $\mu_1, \dots, \mu_m \in \mathbb{R}$ such that

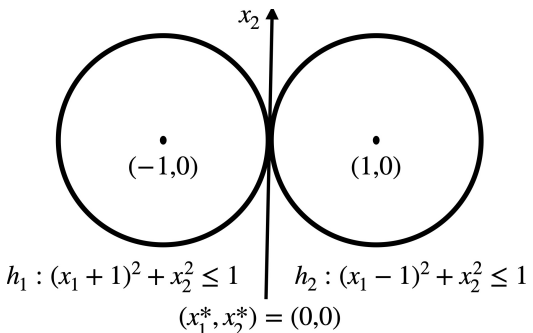
$$\nabla f(\mathbf{x}^*) + \sum_{j\in\mathcal{A}(\mathbf{x}^*)} \lambda_j \nabla h_j(\mathbf{x}^*) + \sum_{i=1}^{m} \mu_i \nabla e_i(\mathbf{x}^*) = \mathbf{0}$$

Example 1

- Consider the optimization problem as follows.

$$\begin{aligned} \min \quad & x_1 - 3x_2 \\ & e_1(\mathbf{x}) = (x_1 - 1)^2 + x_2^2 = 1 \\ & e_2(\mathbf{x}) = (x_1 + 1)^2 + x_2^2 = 1 \end{aligned}$$

- Here, Feasible set
  $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^2 \mid (x_1 + 1)^2 + x_2^2 = 1,\ (x_1 - 1)^2 + x_2^2 = 1\} = \{(0, 0)\}$.
- $\mathcal{L}(\mathbf{x}, \mu_1, \mu_2) = x_1 - 3x_2 + \mu_1[(x_1 - 1)^2 + x_2^2 - 1] + \mu_2[(x_1 + 1)^2 + x_2^2 - 1]$.
- $\nabla f(\mathbf{x}) = \begin{bmatrix} 1 \\ -3 \end{bmatrix}$, $\nabla e_1(\mathbf{x}) = \begin{bmatrix} 2(x_1 - 1) \\ 2x_2 \end{bmatrix}$, $\nabla e_2(\mathbf{x}) = \begin{bmatrix} 2(x_1 + 1) \\ 2x_2 \end{bmatrix}$.
- $\nabla f(0, 0) = \begin{bmatrix} 1 \\ -3 \end{bmatrix}$, $\nabla e_1(0, 0) = \begin{bmatrix} -2 \\ 0 \end{bmatrix}$, $\nabla e_2(\mathbf{x}) = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$.
- If $\mathbf{x}^* = (0, 0)$ is a local minima, then $\nabla\mathcal{L}(\mathbf{x}^*, \mu_1, \mu_2) = \mathbf{0}$. Which implies, $1 - 2\mu_1 + 2\mu_2 = 0$ and $-3 = 0$, which is impossible.
- Thus, $\mathbf{x}^* = (0, 0)$ is not a local minima.
- Note that $\mathbf{x}^* = (0, 0)$ is not a regular point.



Example 2

- Consider the following problem: $\max_{\mathbf{x}\in\mathbb{R}^n} \frac{\mathbf{x}^T Q\mathbf{x}}{\mathbf{x}^T\mathbf{x}}$ where Q is a symmetric positive semi-definite matrix.
- Note that is $\mathbf{x}$ is a solution to the problem, then $t\mathbf{x}$ is also a solution for any $t \neq 0$. $\left(\frac{(t\mathbf{x})^T Q(t\mathbf{x})}{(t\mathbf{x})^T(t\mathbf{x})} = \frac{\mathbf{x}^T Q\mathbf{x}}{\mathbf{x}^T\mathbf{x}}\right)$.
- To avoid the multiplicity of the solutions, we add the constraint $\mathbf{x}^T\mathbf{x} = 1$.
- Thus,

$$\max_{\mathbf{x}\in\mathbb{R}^n} \frac{\mathbf{x}^T Q\mathbf{x}}{\mathbf{x}^T\mathbf{x}} = \max_{\mathbf{x}\in\mathbb{R}^n} \mathbf{x}^T Q\mathbf{x}$$
$$s.t.\ \mathbf{x}^T\mathbf{x} = 1$$

- So, $f(\mathbf{x}) = \mathbf{x}^T Q\mathbf{x}$ and $e(\mathbf{x}) = 1 - \mathbf{x}^T\mathbf{x}$.
- Any feasible point for this problem is regular.
- Lagrange conditions yield $2\mathbf{x}^T Q - 2\mu\mathbf{x}^T = 0$ and $1 - \mathbf{x}^T\mathbf{x} = 0$.
- The first condition gives $Q\mathbf{x} = \mu\mathbf{x}$. Therefore, if it exists, the solution is an eigenvector of Q.
- Let $\mathbf{x}^*$ and μ^* be the optimal solution. Because $(\mathbf{x}^*)^T\mathbf{x}^* = 1$ and $Q\mathbf{x}^* = \mu^*\mathbf{x}^*$.. This gives

$$\mu^* = (\mathbf{x}^*)^T Q\mathbf{x}^*$$

- Hence μ^* is the maximum of the objective function, and therefore, the maximum eigenvalue of Q.

Sufficiency of KKT Conditions for Convex Optimization Problem Under Slater's Condition

Theorem

Consider the convex optimization problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ s.t. \quad & h_j(\mathbf{x}) \leq 0,\ j = 1 \ldots l \\ & e_i(\mathbf{x}) = 0;\ i = 1 \ldots m \end{aligned}$$

where

- $f(\mathbf{x})$, $h_1(\mathbf{x}), \ldots, h_l(\mathbf{x})$ are smooth convex functions over $\mathbb{R}^n$.
- $e_i(\mathbf{x}) = \mathbf{a}_i^\top \mathbf{x} - b_i,\ i = 1 \ldots m$

Let $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n \mid h_j(\mathbf{x}) \leq 0,\ j = 1 \ldots l;\ e_i(\mathbf{x}) = 0;\ i = 1 \ldots m\}$ be the feasible set and it satisfies Slater's Condition. The first-order KKT conditions are necessary and sufficient for global minima of the convex optimisation problem above.