MAT3007 Optimization
Lecture 19 Algorithms

Yuang Chen

School of Data Scienc
The Chinese University of Hong Kong, Shenzhen

July 17, 2025

Outline

Outline

Gradient Descent Algorithm

Start with any point $\boldsymbol{x}^0$. Set $k = 0$ and stopping criterion $\epsilon > 0$

1. Check $||\nabla f(\boldsymbol{x}^k)||$. If $||\nabla f(\boldsymbol{x}^k)|| \leq \epsilon$, stop and output $\boldsymbol{x}^k$. Otherwise, continue to Step 2
2. Let $\boldsymbol{d}^k = -\nabla f(\boldsymbol{x}^k)$
3. Use either exact line search or backtracking line search to find α_k
4. Let $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k$, let $k = k + 1$. Go back to step 1.

$$x^{k+1} \leftarrow x^k - \alpha_k \nabla f(x^k)$$

Step Size

- Constant step size: $\alpha_k = \bar{\alpha}$ for all k.
- Exact line search: choose α_k such that

$$\alpha_k = \arg\min_{\alpha} f(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k)$$

- Backtracking / Armijo line search:
  1. Let $\sigma, \gamma \in (0, 1)$ be given. Start with $\alpha = 1$.
  2. If $f(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k) \leq f(\boldsymbol{x}^k) + \gamma\alpha \cdot \nabla f(\boldsymbol{x}^k)^\top \boldsymbol{d}^k$, choose $\alpha_k = \alpha$. Otherwise, set $\alpha = \sigma\alpha$ and repeat this step.

Gradient Descent Convergence Rates

Consider unconstrained problem $\min_{x \in \mathbb{R}^n} f(x)$.

- **Sublinear convergence**: If f is convex and ∇f is Lipschitz continuous, GD with constant step size has sublinear convergence (in function value) to the global minimizer.
- **Linear convergence**: If f is strongly convex and ∇f is Lipschitz continuous, GD with constant step size / exact / Armijo line search will have linear convergence to the unique global minimizer.
- **Global convergence**: No matter where GD starts, it will always converge to a local minimizer (when $f(\cdot)$ is convex, it converges to a global minimizer).

References and further reading

- S. Boyd and L. Vandenberghe (2004), *Convex Optimization*, Chapter 9
- T. Hastie, R. Tibshirani and J. Friedman (2009), *The Elements of Statistical Learning*, Chapters 10 and 16
- Y. Nesterov (1998), *Introductory Lectures on Convex Optimization: A Basic Course*, Chapter 2
- L. Vandenberghe, Lecture notes for EE 236C, UCLA

Outline

Intuition 1

- First order Taylor expansion for linearly approximating $f(x)$:

$$f(x+\Delta x) = f(x) + \nabla f(x)\Delta x + o(|\Delta x|).$$

- Take gradient on both sides:

$$\nabla f(x+\Delta x) = \nabla f(x) + \nabla^2 f(x)\Delta x + o(|\Delta x|).$$

- If we want $x^* = x + \Delta x$, then

$$\nabla f(x^*) = \nabla f(x+\Delta x) = \nabla f(x) + \nabla^2 f(x)\Delta x = 0$$

$$\Rightarrow \quad x^\star - x = \Delta x = -[\nabla^2 f(x)]^{-1}\nabla f(x)$$

- Use this direction in each iteration

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1}\nabla f(x^k)$$

Intuition 2

- Let x^k be the current iterate and consider a second-order Taylor approximation of $f(x)$ at x^k, i.e.

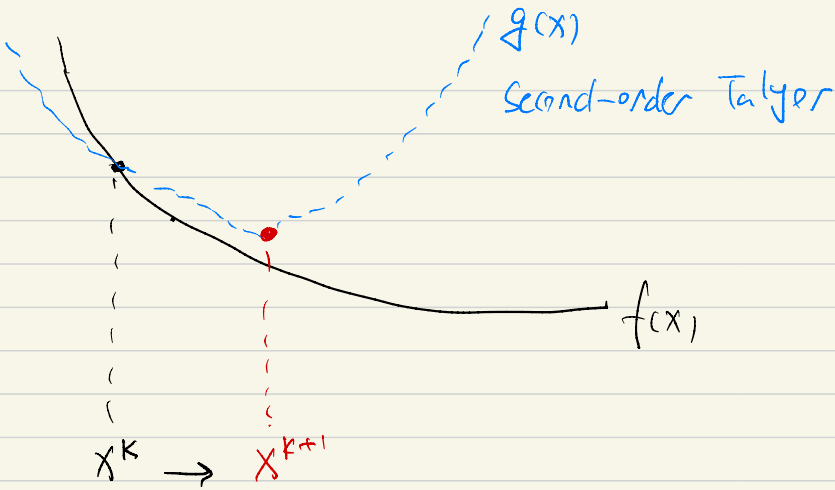
$$g(x) = f(x^k) + \nabla f(x^k)^\top (x - x^k) + \frac{1}{2}(x - x^k)^\top \nabla^2 f(x^k)(x - x^k).$$

- Choose the next iterate as the solution that minimizes the approximate function $g(x)$.
- Setting $\nabla g(x) = 0$, we get the linear system

$$\nabla f(x^k) + \nabla^2 f(x^k)(x - x^k) = 0.$$

- If the Hessian is non-singular, a solution to the above system is well-defined, and we set

$$x^{k+1} = x^k - [\nabla^2 f(x_k)]^{-1} \nabla f(x^k).$$


g(x)
Second-order Taylor
f(x)
x^k
x^{k+1}

Newton's Method in High Dimensional Case

- Newton's step:

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

- Direction: $\boldsymbol{d}^k = -(\nabla^2 f(\boldsymbol{x}^k))^{-1} \nabla f(\boldsymbol{x}^k)$.
- Then Newton's step direction is a descent direction when f is convex.
- Newton's method may not converge unless the starting point is close.
- One way to help convergence is to use a step size parameter α_k in

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k$$

  where $\boldsymbol{d}^k = -(\nabla^2 f(\boldsymbol{x}^k))^{-1} \nabla f(\boldsymbol{x}^k)$ is the Newton's step.
- One can use backtracking line search to determine α_k.
- In all above derivations, we assume that $\nabla^2 f(\boldsymbol{x}^k)$ is invertible.

$$d := -(\nabla^2 f(x))^{-1} \nabla f(x)$$

Prove d is a descent direction if f is convex.

Descent direction: $\{d : \nabla f(x)^T d < 0\}$

$$\nabla f(x)^T d = -\underbrace{\nabla f(x)^T}_{a} \underbrace{(\nabla^2 f(x))^{-1}}_{b} \underbrace{\nabla f(x)}_{a} \quad \not< 0 \; ?$$

$$= -ba^2$$

If f is convex, then $\nabla^2 f(x) \succeq 0 \Rightarrow [\nabla^2 f(x)]^{-1} \succeq 0$

Complete Procedure of Newton's Method

Start from a point $\boldsymbol{x}^0$, set tolerance $\epsilon > 0$ and $k = 0$.

1. Compute the Newton's step:

$$\boldsymbol{d}^k = -(\nabla^2 f(\boldsymbol{x}^k))^{-1} \nabla f(\boldsymbol{x}^k)$$

   If $||\nabla f(\boldsymbol{x}^k)|| < \epsilon$, then output $\boldsymbol{x}^k$, otherwise continue;
2. Choose step size α_k by backtracking line search
3. $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k$, $k = k + 1$, go back to step 1

$$x^{k+1} \leftarrow x^k - (\nabla^2 f(x^k))^{-1} \nabla f(x^k)$$

Linear Convergence vs Quadratic Convergence

Remember the gradient descent method can have linear convergence rate (strongly convex case):

$$|x^{k+1} - x^*| \leq \eta |x^k - x^*|.$$

Now, Newton's method has quadratic convergence rate:

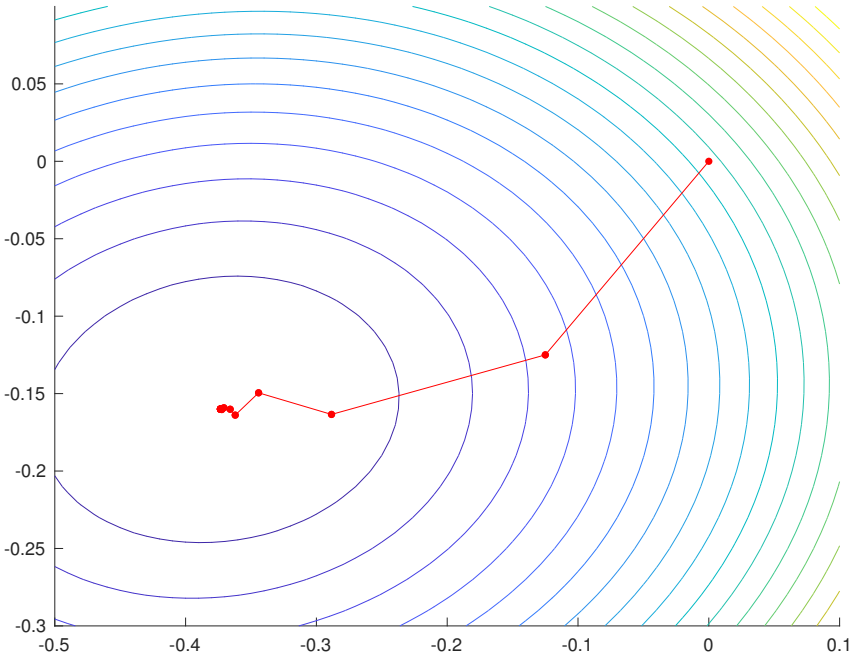
$$|x^{k+1} - x^*| \leq C |x^k - x^*|^2.$$

Convergence of Newton's Method

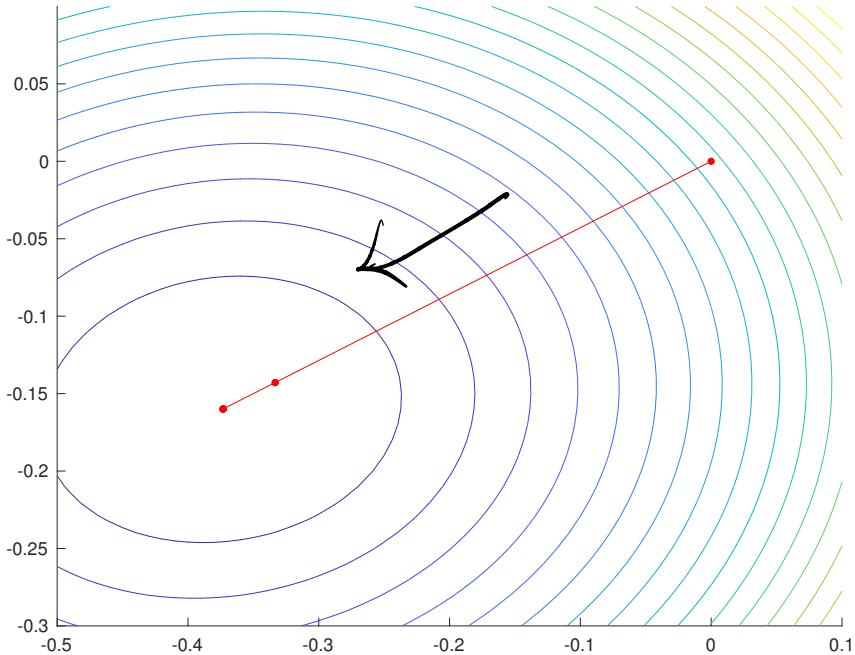
- Quadratic convergence in the neighborhood of a strict local minimum (under some conditions). Quadratic convergence is much faster than linear convergence in gradient descent.

Example: Minimize

$$f(x) = \exp(x_1 + x_2) + x_1^2 + 3x_2^2 - x_1 x_2$$



(a) Gradient Descent



(b) Newton's Method

Newton's Method is Fast in Iteration Count, But ...

It requires computing the second-order derivative (Hessian) in each iteration.

- It might be computationally expensive to do so, especially if the second-order derivative doesn't have a closed-form (which is the case for many useful applications)
- It also requires a lot of space to store the Hessian matrix (If it is an n-dimensional problem, then we need n^2 space, comparing to n space required for gradient descent method).
- It requires much more matrix computation in each iteration
- If the Hessian is singular (or close to singular) at some iteration, we cannot proceed.
- Newton's method may diverge, even for convex functions.

if it starts from some points far away from x^*.

Quasi-Newton Methods

- Goal: Avoids computation of the Hessian and its inverse.
- Update iterate using

$$x^{k+1} = x^k - \alpha_k H_k \nabla f(x^k),$$

where α_k is the step size and H_k is an approximation to $[\nabla^2 f(x^k)]^{-1}$.

- Since H_{k+1}^{-1} approximates the Hessian,

$$H_{k+1}(\nabla f(x^{k+1}) - \nabla f(x^k)) = x^{k+1} - x^k.$$

- A widely used formula is the Broyden–Fletcher–Goldfarb–Shanno (BFGS) formula:

$$H_{k+1} = H_k - \frac{d_k g_k^\top H_k + H_k g_k d_k^\top}{d_k^\top g_k} + \left(1 + \frac{g_k^\top H_k g_k}{d_k^\top g_k}\right) \frac{d_k d_k^\top}{d_k^\top g_k},$$

where $g_k = \nabla f(x^{k+1}) - \nabla f(x^k)$ and $d_k = x^{k+1} - x^k$.

Quiz

$\max. f(x) \rightarrow -\min. -f(x)$

What are the update equations in each iteration for gradient descent and Newton's method for a maximization problem?

GD: $x^{k+1} \leftarrow x^k + \alpha \nabla f(x^k)$

Newton: $x^{k+1} \leftarrow x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k)$

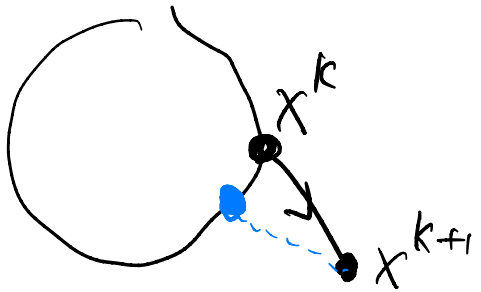
keeps same

Outline

Constrained Optimization

We consider the following constrained optimization problem:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{x} \in \Omega. \end{aligned}$$



where $\Omega \subset \mathbb{R}^n$ is a convex and closed set.

In the methods for unconstrained problems, the main idea is:

- At each $\boldsymbol{x}^k$, compute a descent direction $\boldsymbol{d}^k$.
- Then find an appropriate step size α_k and update $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k$.
- Both the gradient and Newton's method are based on this basic idea.
- $\boldsymbol{x}^{k+1}$ can become infeasible. One solution is to project the point back to Ω.

Euclidean Projection / Orthogonal Projection

Definition: Euclidean / Orthogonal Projection

Let $\Omega \subset \mathbb{R}^n$ be a nonempty, closed, convex set. The (Euclidean / orthogonal) projection of $\boldsymbol{x}$ onto Ω is defined as the unique minimizer $\boldsymbol{y}^*$ of the constrained optimization problem:

$$\min_{\boldsymbol{y}} \quad \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2 \quad \text{s.t.} \quad \boldsymbol{y} \in \Omega$$

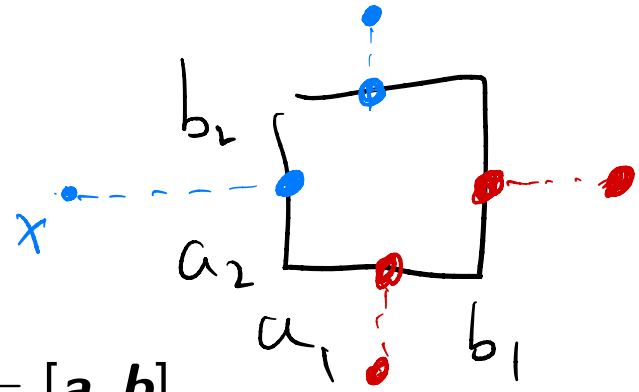
and we write $\boldsymbol{y}^* = \mathcal{P}_\Omega(\boldsymbol{x})$.

Observations:

- Interpretation: Projection is to find the closest point $\boldsymbol{y}^*$ in Ω to $\boldsymbol{x}$.
- Hence, the projection $\boldsymbol{y}^* = \mathcal{P}_\Omega(\boldsymbol{x})$ is the point in Ω that has the minimum distance to $\boldsymbol{x}$.

$P_\Omega(x) = x$ if $x \in \Omega$

Example I: Box Constraints



Suppose that Ω is given by box constraints:

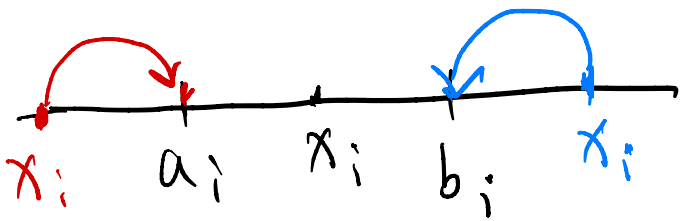
$$\Omega = \{\boldsymbol{x} \in \mathbb{R}^n : x_i \in [a_i, b_i],\ \forall i\} = [\boldsymbol{a}, \boldsymbol{b}],$$

where $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$, $\boldsymbol{a} \leq \boldsymbol{b}$, are given.

Euclidean Projection:

- The projection onto Ω can be computed as follows:

$$[\mathcal{P}_\Omega(\boldsymbol{x})]_i = \mathcal{P}_{[a_i,b_i]}(x_i) = \max\{\min\{x_i, b_i\}, a_i\}, \quad \forall i.$$



$$P_{[a_i,b_i]}(x_i) = \begin{cases} a_i & \text{if } x_i < a_i \\ x_i & \text{if } x_i \in [a_i, b_i] \\ b_i & \text{if } x_i > b_i \end{cases}$$

Example II: Linear Constraints

We now consider the simple case where Ω consists of linear equality constraints:

$$\Omega = \{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\},$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$ are given.

Euclidean Projection:

- Suppose that $\boldsymbol{A}$ has full row rank ($m \leq n$), then it holds that:

$$\mathcal{P}_\Omega(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}).$$

- This yields a special case when $\Omega = \{\boldsymbol{x} : \boldsymbol{a}^\top\boldsymbol{x} = \boldsymbol{b}\}$, i.e., projection onto a hyperplane. We have

$$\mathcal{P}_\Omega(\boldsymbol{x}) = \boldsymbol{x} - \frac{\boldsymbol{a}^\top\boldsymbol{x} - \boldsymbol{b}}{\|\boldsymbol{a}\|^2}\boldsymbol{a}.$$

x is given:

$$\left.\begin{array}{ll} \min\limits_{y} & \frac{1}{2}\|y-x\|^2 \\ \text{s.t.} & Ay=b \end{array}\right\} \text{ Convex opt}$$

Lagrangian: $L(y,\mu)=\frac{1}{2}\|y-x\|^2+\mu^T(Ay-b)$

KKT:

main. $\nabla_y L(y,\mu)=y-x+A^T\mu=0$

~~Complementary slackness: $\mu^T(Ay-b)=0$~~

Primal feasible: $Ay=b$

~~Dual feasible any μ~~

$A(y-x+A^T\mu)=A(0)$

$Ay-Ax+AA^T\mu=0$

$b-Ax+AA^T\mu=0$

$\mu=(AA^T)^{-1}(Ax-b)$

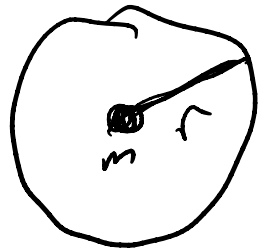
$y=x-A^T\mu=x-A^T(AA^T)^{-1}(Ax-b)$

$=P_{\Omega}(x)$

Example III: Ball Constraints

Suppose that Ω is a Euclidean ball with radius $r > 0$ and center $\boldsymbol{m} \in \mathbb{R}^n$, i.e.:

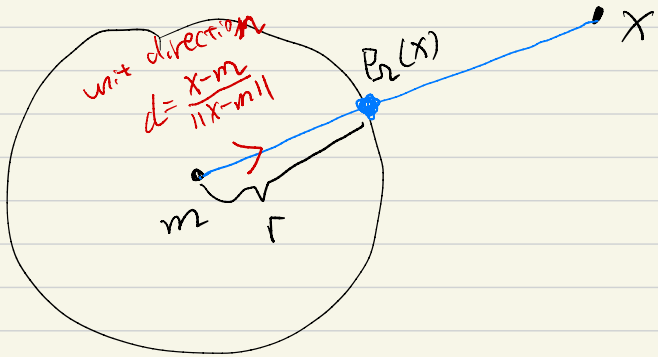
$$\Omega = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x} - \boldsymbol{m}\| \leq r\}.$$



Euclidean Projection:

- The projection onto Ω can be computed as follows:

$$y^* = \mathcal{P}_\Omega(\boldsymbol{x}) = \begin{cases} \boldsymbol{x} & \text{if } \|\boldsymbol{x} - \boldsymbol{m}\| \leq r, \checkmark \\ \boldsymbol{m} + \dfrac{r}{\|\boldsymbol{x} - \boldsymbol{m}\|}(\boldsymbol{x} - \boldsymbol{m}) & \text{if } \|\boldsymbol{x} - \boldsymbol{m}\| > r. \end{cases}$$



$$P_\Omega(x) = m + r\frac{x-m}{\|x-m\|} \quad \text{if } x \notin \Omega$$

$$\min_y \|y-x\|^2 \quad \text{s.t. } \|y-m\| \le r$$

Solve KKT

Projected Gradient Method: Basic Idea

In the gradient descent method, we perform a gradient step of the form:

$$\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k - \lambda_k \nabla f(\boldsymbol{x}^k),$$

where $\lambda_k > 0$ is a step size.

Motivation & Strategy:

- Setting $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \lambda_k \nabla f(\boldsymbol{x}^k)$ might likely generate infeasible iterates: $\boldsymbol{x}^{k+1} \notin \Omega$.
- Idea: We project the step $\boldsymbol{x}^k - \lambda_k \nabla f(\boldsymbol{x}^k)$ back onto Ω:

$$\boldsymbol{x}^{k+1} = \mathcal{P}_\Omega(\boldsymbol{x}^k - \lambda_k \nabla f(\boldsymbol{x}^k)).$$

Questions:

- How to choose $\lambda_k > 0$? If λ_k is determined by line search, every adjustment of λ_k requires re-evaluating the projection $\mathcal{P}_\Omega(\boldsymbol{x}^k - \lambda_k \nabla f(\boldsymbol{x}^k))$. This can be expensive.
- Can we guarantee descent?

Important Observation

Observation:

$$x^{k+1} \leftarrow x^k + \alpha d^k$$

$$\boldsymbol{x}^{k+1} = \mathcal{P}_\Omega(\boldsymbol{x}^k - \lambda_k \nabla f(\boldsymbol{x}^k)) = \boldsymbol{x}^k + \left[\mathcal{P}_\Omega(\boldsymbol{x}^k - \lambda_k \nabla f(\boldsymbol{x}^k)) - \boldsymbol{x}^k\right].$$

$\boldsymbol{d}^k$

- This is close to our usual update form: $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k$.
- Setting $\boldsymbol{d}^k = \mathcal{P}_\Omega(\boldsymbol{x}^k - \lambda_k \nabla f(\boldsymbol{x}^k)) - \boldsymbol{x}^k$, we can consider:

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k = (1 - \alpha_k)\boldsymbol{x}^k + \alpha_k \mathcal{P}_\Omega(\boldsymbol{x}^k - \lambda_k \nabla f(\boldsymbol{x}^k)) \in \Omega$$

$\in \Omega$ $\in \Omega$

If $\alpha_k \in [0, 1]$, then the convexity of Ω implies that $\boldsymbol{x}^{k+1}$ will be feasible if $\boldsymbol{x}^k \in \Omega$.

by Ω is a convex set.

Why does / can this work? Is $\boldsymbol{d}^k$ a descent direction?

The Projected Gradient Method

Projected Gradient Method

1. Initialization: Choose an initial point $\boldsymbol{x}^0 \in \Omega$.
2. For $k = 0, 1, \ldots$:
   1. Select $\lambda_k > 0$ and compute $\nabla f(\boldsymbol{x}^k)$ and the search direction
   $$\boldsymbol{d}^k = \mathcal{P}_\Omega(\boldsymbol{x}^k - \lambda_k \nabla f(\boldsymbol{x}^k)) - \boldsymbol{x}^k.$$
   2. Choose a step size ~~$\alpha_k > 0$~~ $\alpha_k \in [0, 1]$.
   3. Set $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha_k \boldsymbol{d}^k$.
   4. If $\|\boldsymbol{d}^{k+1}\| \leq \epsilon$, then STOP and $\boldsymbol{x}^{k+1}$ is the output.

- In terms of convergence theorem, under appropriate conditions on the step sizes α_k and λ_k, projected gradient method has similar convergence guarantees to those of GD.

Outline

Optimization Problems with Convex Constraints

Recall the constrained optimization problem:

$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x}) \quad \text{s.t.} \quad \boldsymbol{x} \in \Omega, \tag{1}$$

where $\Omega \subset \mathbb{R}^n$ is a convex and closed set.

- We can derive the following optimality condition:

Theorem: FONC for Problems with Convex Constraints

Let f be continuously differentiable on an open set that contains the convex and closed set $\Omega \subset \mathbb{R}^n$. Let $\boldsymbol{x}^* \in \Omega$ be a local minimizer of (1), then:

$$\nabla f(\boldsymbol{x}^*)^\top (\boldsymbol{x} - \boldsymbol{x}^*) \geq 0, \quad \forall \boldsymbol{x} \in \Omega. \tag{2}$$

- A point satisfying (2) is called a stationary point of problem (1).

Proof: Let x^* be a local optimal

$\exists\ \varepsilon > 0$ s.t.

$$f(x^*) \le f(x), \quad \forall x \in B_\varepsilon(x^*) \quad (*)$$

Let $x \in \Omega$ and $\lambda \in [0,1]$

Define $x(\lambda) = \lambda x + (1-\lambda) x^* \in \Omega$ because Ω is convex set

Rewrite $x(\lambda) = x^* + \lambda (x - x^*)$

$$\begin{aligned} & \|x(\lambda) - x^*\| \le \varepsilon \\ \Leftrightarrow\ & \|\lambda (x - x^*)\| \le \varepsilon \\ \Leftrightarrow\ & \lambda \|x - x^*\| \le \varepsilon \\ \Leftrightarrow\ & \lambda \le \frac{\varepsilon}{\|x - x^*\|} \end{aligned}$$

When $\lambda \in [0, \frac{\varepsilon}{\|x-x^*\|}]$, $x(\lambda) \in B_\varepsilon(x^*)$

By $(*)$, $f(x^*) \le f(x(\lambda))$

$$\Leftrightarrow \quad \frac{f(x(\lambda)) - f(x^*)}{\lambda} \ge 0 \quad \text{if } \lambda \neq 0$$

Take limit of $\lambda \to 0$, we get

$$\to \nabla f(x^*)(x - x^*) \ge 0, \quad \forall x \in \Omega$$

Projection Theorem

Projection Theorem

Let Ω be a nonempty, closed, and convex set. Then:

- A point $\boldsymbol{y}^*$ is the projection of $\boldsymbol{x}$ onto Ω, i.e., $\boldsymbol{y}^* = \mathcal{P}_\Omega(\boldsymbol{x})$, if and only if

$$(\boldsymbol{y}^* - \boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{y}^*) \geq 0, \quad \forall \boldsymbol{y} \in \Omega.$$

- The mapping $\mathcal{P}_\Omega : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous with constant $L = 1$, i.e.,

$$\|\mathcal{P}_\Omega(\boldsymbol{x}) - \mathcal{P}_\Omega(\boldsymbol{y})\| \leq \|\boldsymbol{x} - \boldsymbol{y}\|, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n.$$

- (FONC for Problem with Convex Constraints) For any $\lambda > 0$, the vector $\boldsymbol{x}^*$ is a stationary point of (1) if and only if

$$\boldsymbol{x}^* - \mathcal{P}_\Omega(\boldsymbol{x}^* - \lambda \nabla f(\boldsymbol{x}^*)) = 0.$$

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

s.t. $y \in \Omega$

Plug in FONC (2): $\nabla f(y^*)^T(y-y^*) \geq 0, \forall y \in \Omega$

$\Rightarrow (y^*-x)^T(y-y^*) \geq 0, \forall y \in \Omega$ (**)

2. From (**), $y^* = P_\Omega(x)$

$(P_\Omega(x)-x)^T(z-P_\Omega(x)) \geq 0, \forall z \in \Omega$

From (**), $y^* = P_\Omega(y)$

$(P_\Omega(y)-y)^T(z-P_\Omega(y)) \geq 0, \forall z \in \Omega$

Pick $z = P_\Omega(y)$

Pick $z = P_\Omega(x)$

$(P_\Omega(x)-x)^T(P_\Omega(y)-P_\Omega(x)) \geq 0$

$+ (P_\Omega(y)-y)^T(P_\Omega(x)-P_\Omega(y)) \geq 0$

$\Rightarrow (P_\Omega(x)-P_\Omega(y))^T(P_\Omega(y)-P_\Omega(x)+x-y) \geq 0$

$\Rightarrow -\|P_\Omega(x)-P_\Omega(y)\|^2 + (P_\Omega(x)-P_\Omega(y))^T(x-y) \geq 0$

$$\Rightarrow \quad \|P_\Omega(x)-P_\Omega(y)\|^2 \le (P_\Omega(x)-P_\Omega(y))^T(x-y)$$

$$\Rightarrow \|P_\Omega(x)-P_\Omega(y)\|^2 \le \|P_\Omega(x)-P_\Omega(y)\|\,\|x-y\|$$

$$\Rightarrow \|P_\Omega(x)-P_\Omega(y)\| \le \|x-y\|$$

3. $x^* = P_\Omega(x^* - \lambda \nabla f(x^*))$

From 1. $\Leftrightarrow (x^* - (x^* - \lambda \nabla f(x^*))^T (x - x^*) \ge 0, \quad \forall x \in \Omega$

proj　inside proj　any in Ω　proj

$$\Leftrightarrow \lambda \nabla f(x^*)^T (x - x^*) \ge 0 \quad \forall x \in \Omega$$

Since $\lambda > 0$

$$\Leftrightarrow \nabla f(x^*)(x - x^*) \ge 0 \quad \forall x \in \Omega$$

FONC

Descent Direction

Lemma: Descent Direction

Let $\boldsymbol{x} \in \Omega$ and $\lambda > 0$ be given. If $\boldsymbol{x}$ is not a stationary point of (1), then the direction

$$\boldsymbol{d} := \mathcal{P}_\Omega(\boldsymbol{x} - \lambda \nabla f(\boldsymbol{x})) - \boldsymbol{x}$$

is a descent direction and it holds that

$$\Rightarrow \quad \nabla f(\boldsymbol{x})^\top \boldsymbol{d} \leq -\frac{1}{\lambda}\|\boldsymbol{d}\|^2 < 0.$$

$\boldsymbol{d}$ is called the generalized gradient.

Proof:

$x \in \Omega$

$$\nabla f(x)^T d$$

$$= \nabla f(x)^T \left(P_\Omega(x - \lambda \nabla f(x)) - x \right)$$

$$= \nabla f(x)^T \left(P_\Omega(x - \lambda \nabla f(x)) - P_\Omega(x) \right)$$

$$= -\frac{1}{\lambda} (x - \lambda \nabla f(x) - x)^T \left(P_\Omega(x - \lambda \nabla f(x)) - P_\Omega(x) \right)$$

Cauchy Schwarz

$$\leq -\frac{1}{\lambda} \| x - \lambda \nabla f(x) - x \| \cdot \| P_\Omega(x - \lambda \nabla f(x)) - P_\Omega(x) \|$$

$$\leq -\frac{1}{\lambda} \| P_\Omega(x - \lambda \nabla f(x)) - P_\Omega(x) \| \cdot \| P_\Omega(x - \lambda \nabla f(x)) - P_\Omega(x) \|$$

$$= -\frac{1}{\lambda} \| P_\Omega(x - \lambda \nabla f(x)) - P_\Omega(x) \|^2$$

$$\leq -\frac{1}{\lambda} \| P_\Omega(x - \lambda \nabla f(x) - x \|^2$$

$$= -\frac{1}{\lambda} \| d \|^2 < 0$$