Convex Optimization

ScAi Lab Study Group

Zhiping (Patricia) Xiao

University of California, Los Angeles

Winter 2020

Outline

Textbook:

- Convex Optimization and Intro to Linear Algebra by Prof. Boyd and Prof. Vandenberghe

Course Materials:

- ECE236B, ECE236C offered by Prof. Vandenberghe
- CS260 Lecture 12 offered by Prof. Quanquan Gu

Notes:

- My previous ECE236B notes and ECE236C final report.
- My previous CS260 Cheat Sheet.

Related Papers:

- Accelerated methods for nonconvex optimization
- Lipschitz regularity of deep neural networks: analysis and efficient estimation

Introduction: Convex Optimization

Notations

- iff: if and only if
- $\mathbb{R}_+ = \{x \in \mathbb{R} \mid x \geq 0\}$
- $\mathbb{R}_{++} = \{x \in \mathbb{R} \mid x > 0\}$
- $\mathbf{int}\, K$: interior of set K, not its boundary.
- Generalized inequalities (textbook 2.4), based on a proper cone K (convex, closed, solid, pointed — if $x \in K$ and $-x \in K$ then $x = 0$):
  - $x \preceq_K y \iff y - x \in K$
  - $x \prec_K y \iff y - x \in \mathbf{int}\, K$
- Positive semidefinite matrix $X \in \mathbb{S}_+^n$, $\forall y \in \mathbb{R}^n$, $y^T X y \geq 0$ $\iff X \succeq 0$.

Convexity of Set

Set C is convex iff the line segment between any two points in C lies in C, i.e. $\forall x_1, x_2 \in C$ and $\forall \theta \in [0, 1]$, we have:

$$\theta x_1 + (1 - \theta) x_2 \in C$$

Both convex and nonconvex sets have convex hull, which is defined as:

$$\textbf{conv } C = \{\sum_{i=1}^{k} \theta_i x_i \mid x_i \in C, \theta_i \geq 0, i = 1, 2, \ldots, k, \sum_{i=1}^{k} \theta_i = 1\}$$

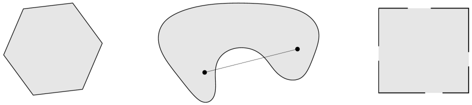


Figure: Left: convex, middle & right: nonconvex.

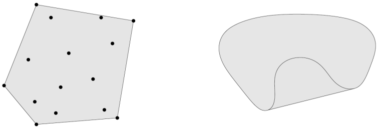


Figure: Left: convex hull of the points, right: convex hull of the kidney-shaped set above.

The most common operations that preserve convexity of convex sets include:

- *Intersection*
- *Image / inverse image* under affine function
- *Cartesian Product*, *Minkowski sum*, *Projection*
- *Perspective* function
- *Linear-fractional* functions

Convexity is preserved under ***intersection***:

- S_1, S_2 are convex sets then $S_1 \cap S_2$ is also convex set.
- If S_α is convex for $\forall \alpha \in \mathcal{A}$, then $\cap_{\alpha \in \mathcal{A}} S_\alpha$ is convex.

Proof: **Intersection** of a collection of convex sets is convex set. If the intersection is empty, or consists of only a single point, then proved by definition. Otherwise, for any two points A, B in the intersection, line AB must lie wholly within each set in the collection, hence must lie wholly within their intersection.

An ***affine function*** $f : \mathbb{R}^n \to \mathbb{R}^m$ is a sum of a linear function and a constant, i.e., if it has the form $f(x) = Ax + b$, where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, thus f represents a *hyperplane*.

Suppose that $S \subseteq \mathbb{R}^n$ is convex and then the *image* of S under f is convex:

$$f(S) = \{f(x) \mid x \in S\}$$

Also, if $f : \mathbb{R}^m \to \mathbb{R}^n$ is an affine function, the inverse image of S under f is *convex*:

$$f^{-1}(S) = \{x \mid f(x) \in S\}$$

Examples include *scaling* $\alpha S = \{f(x) \mid \alpha x,\ x \in S\}$ $(\alpha \in \mathbb{R})$ and *translation* $S + a = \{f(x) \mid x + a,\ x \in S\}$ $(a \in \mathbb{R}^n)$; they are both convex sets when S is convex.

Proof: the image of convex set S under affine function $f(x) = Ax + b$ is also convex.

If S is empty or contains only one point, then $f(S)$ is obviously convex. Otherwise, take $x_S, y_S \in f(S)$. $x_S = f(x) = Ax + b$, $x_S = f(y) = Ay + b$. Then $\forall \theta \in [0, 1]$, we have:

$$\begin{aligned} \theta x_S + (1-\theta) y_S &= A(\theta x + (1-\theta)y) + b \\ &= f(\theta x + (1-\theta)y) \end{aligned}$$

Since $x, y \in S$, and S is convex set, then $\theta x + (1-\theta)y \in S$, and thus $f(\theta x + (1-\theta)y) \in f(S)$.

The *Cartesian Product* of convex sets $S_1 \subseteq \mathbb{R}^n$, $S_2 \subseteq \mathbb{R}^m$ is obviously convex:

$$S_1 \times S_2 = \{(x_1, x_2) \mid x_1 \in S_1, x_2 \in S_2\}$$

The *Minkowski sum* of the two sets is defined as:

$$S_1 + S_2 = \{x_1 + x_2 \mid x_1 \in S_1, x_2 \in S_2\}$$

and it is also obviously convex.
The *projection* of a convex set onto some of its coordinates is also obviously convex. (consider the definition of convexity reflected on each coordinate)

$$T = \{x_1 \in \mathbb{R}^m \mid (x_1, x_2) \in S \text{ for some } x_2 \in \mathbb{R}^n\}$$

We define the *perspective* function $P : \mathbb{R}^{n+1} \to \mathbb{R}^n$, with domain $\textbf{dom}\, P = \mathbb{R}^n \times \mathbb{R}_{++}$, as $P(z,t) = z/t$.
The *perspective function* scales or normalizes vectors so the last component is one, and then drops the last component.

We can interpret the perspective function as the action of a pin-hole camera. (x_1, x_2, x_3) through a hold at $(0,0,0)$ on plane $x_3 = 0$ forms an image at $-(x_1/x_3, x_2/x_3, 1)$ at $x_3 = -1$. The last component could be dropped, since the image point is fixed.

Proof: That this operation preserves convexity is already proved by *affine function* + *projection* preserve convexity.

A *linear-fractional function* $f : \mathbb{R}^n \to \mathbb{R}^m$ is formed by composing the *perspective function* with an *affine function*. Consider the following affine function $g : \mathbb{R}^n \to \mathbb{R}^{m+1}$:

$$g(x) = \begin{bmatrix} A \\ c^T \end{bmatrix} x + \begin{bmatrix} b \\ d \end{bmatrix}$$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $d \in \mathbb{R}$.
Followed by a *perspective function* $P : \mathbb{R}^{m+1} \to \mathbb{R}^m$ we have:

$$f(x) = (Ax + b)/(c^T x + d), \qquad \mathbf{dom}\, f = x \mid c^T x + d > 0$$

And it naturally preserves convexity because both *affine function* and *perspective function* preserve convexity.

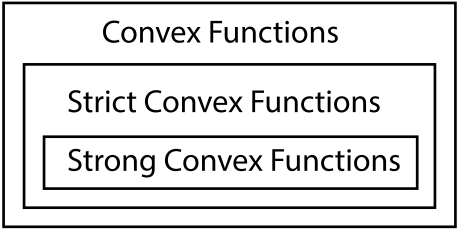


Figure: The three commonly-seen types of *convex functions* and their relations. In brief, *strong convex functions* $\Rightarrow$ *strict convex functions* $\Rightarrow$ *convex functions*.

$f : \mathbb{R}^n \to \mathbb{R}$ is *convex* iff it satisfies:

- $\mathbf{dom}\, f$ is a convex set.
- $\forall x, y \in \mathbf{dom}\, f$, $\theta \in [0, 1]$, we have the *Jensen's inequality*:

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y)$$

f is *strictly convex* iff when $x \neq y$ and $\theta \in (0, 1)$, strict inequality of the above inequation holds.

f is *concave* when $-f$ is *convex*, *strictly concave* when $-f$ *strictly convex*, and vice versa.

f is *strong convex* iff $\exists \alpha > 0$ such that $f(x) - \alpha \|x\|^2$ is convex. $\|\cdot\|$ is any norm.

Proof: *strong convex functions* $\Rightarrow$ *strict convex functions* $\Rightarrow$ *convex functions.*

That all *strict convex functions* are *convex functions*, and that *convex functions* are not necessarily *strict convex. Strong convexity* implies, $\forall x, y \in \textbf{dom}\, f, \theta \in [0,1]$, $x \neq y$, $\exists \alpha > 0$:

$$f(\theta x + (1-\theta)y) - \alpha\|\theta x + (1-\theta)y\|^2 \qquad (1.1)$$
$$\leq \theta f(x) + (1-\theta)f(y) - \theta\alpha\|x\|^2 - (1-\theta)\alpha\|y\|^2$$

Something we didn't prove yet but is true: $\|\cdot\|^2$ **is strictly convex**. We need it for this proof.

$$\|\theta x + (1-\theta)y\|^2 < \theta\|x\|^2 + (1-\theta)\|y\|^2$$

(proof continues)

$$\alpha\|\theta x + (1-\theta)y\|^2 < \theta\alpha\|x\|^2 + (1-\theta)\alpha\|y\|^2$$

$$t = -\alpha\|\theta x + (1-\theta)y\|^2 + \theta\alpha\|x\|^2 + (1-\theta)\alpha\|y\|^2 > 0$$

(1.1) is equivalent with:

$$f(\theta x + (1-\theta)y) + t \leq \theta f(x) + (1-\theta)f(y)$$

where $t > 0$, thus:

$$f(\theta x + (1-\theta)y) < \theta f(x) + (1-\theta)f(y)$$

Convexity of Function

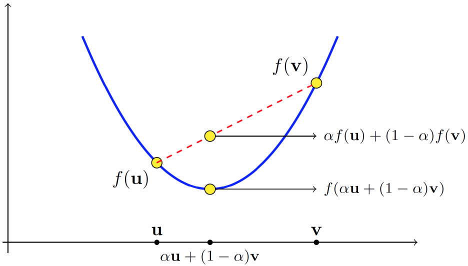


Figure: Convex function illustration from Prof. Gu's Slides. This figure shows a typical convex function f, and instead of our expression of x and y he used u & v instead.

Commonly-seen uni-variate convex functions include:

- Constant: C
- Exponential function: e^{ax}
- Power function: x^a ($a \in (-\infty, 0] \cup [1, \infty)$, otherwise it is *concave*)
- Powers of absolute value: $|x|^p$ ($p \geq 1$)
- Logarithm: $-\log(x)$ ($x \in \mathbb{R}_{++}$)
- $x \log(x)$ ($x \in \mathbb{R}_{++}$)
- All norm functions $\|x\|$
  - "The inequality follows from the triangle inequality, and the equality follows from homogeneity of a norm."

Affine functions are Convex & Concave

An *affine function* $f : \mathbb{R}^n \to \mathbb{R}^m$, $f(x) = Ax + b$, where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, is *convex* & *concave* (neither *strict convex* nor *strict concave*).

Conversely, all functions that are both *convex* and *concave* are *affine functions*.

Proof: $\forall \theta \in [0, 1], x, y \in \textbf{dom}\, f$, we have:

$$\begin{aligned} f(\theta x + (1-\theta)y) &= A(\theta x + (1-\theta)y) + b \\ &= \theta(Ax + b) + (1-\theta)(Ay + b) \\ &= \theta f(x) + (1-\theta) f(y) \end{aligned}$$

Subgradient[1]

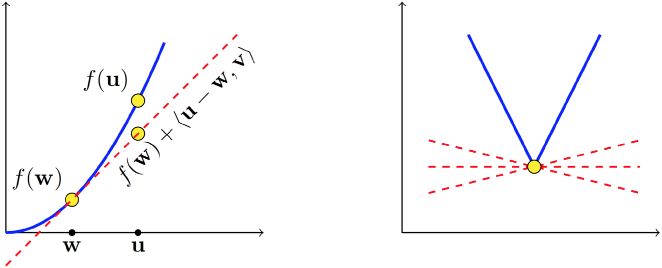


Figure: By using subgradient $g \in \partial f(x)$ instead of gradient $\nabla f(x)$, where $\forall u, w \in \mathbf{dom}\, f$, $f(u) \geq f(w) + g^T(u - w)$, we can handle the cases where the functions are not differentiable. $\partial f(x)$ is called *sub-differential*, the set of *sub-gradients* of f at x.

f is convex iff for every $x \in \mathbf{dom}\, f$, $\partial f(x) \neq \emptyset$.

[1] In Prof. Gu's Slides.

0^{th}-Order Characterization

f is convex iff it is convex when restricted to **any** line that intersects its domain.
In other words, f is convex iff $\forall x \in \mathbf{dom}\, f$ and $\forall v \in \mathbb{R}^n$, the function:

$$g(t) = f(x + tv)$$

is convex. $\mathbf{dom}\, g = \{t \mid x + tv \in \mathbf{dom}\, f\}$

This property allows us to check convexity of a function by restricting it to a line.

1^{st}-Order Characterization

Suppose f is differentiable (its gradient ∇f exists at each point in $\mathbf{dom}\, f$, which is open). Then f is convex iff:

- $\mathbf{dom}\, f$ is a convex set
- $\forall x, y \in \mathbf{dom}\, f$:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

It states that, for a convex function, the *first-order Taylor approximation* ($f(x) + \nabla f(x)^T (y - x)$ is the first-order Taylor approximation of f near x) is in fact a global underestimator of the function.

Could also be interpreted as "tangents lie below f".

Proof is on next page.

This proof comes from CVX textbook page *70*, 3.1.3.

Let $x, y \in \textbf{dom}\, f$, $t \in (0, 1]$, s.t. $x + t(y - x) \in \textbf{dom}\, f$, then, by convexity we have:

$$f(x + t(y - x)) = f((1 - t)x + ty) \leq (1 - t)f(x) + tf(y)$$

$$tf(y) \geq (t - 1)f(x) + f(x + t(y - x))$$

$$f(y) \geq f(x) + \frac{f(x + t(y - x)) - f(x)}{t}$$

take $\lim_{t \to 0}$ we have:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

1^{st}-Order Characterization w. Strong Convex

First we assume that α is the maximum value of the parameter before the norm.

Also note that all norms are equivalent [2], meaning that $\exists 0 < C_1 \leq C_2$ for $\forall a, b, x$:

$$C_1\|x\|_b \leq \|x\|_a \leq C_2\|x\|_b$$

and thus it is okay to treat $\|\cdot\|$ as ℓ_2 norm.
Consider the Taylor formula:

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

[2] https://math.mit.edu/~stevenj/18.335/norm-equivalence.pdf

2^{nd}-Order Characterization

We now assume that f is twice differentiable, that is, its Hessian or second derivative $\nabla^2 f$ exists at each point in $\mathbf{dom}\, f$, which is open. Then f is convex iff:

- $\mathbf{dom}\, f$ is convex
- f's Hessian is positive semidefinite, $\forall x \in \mathbf{dom}\, f$:

$$\nabla^2 f(x) \succeq 0$$

When $f : \mathbb{R}^n \to \mathbb{R}$, it is simply:

$$\nabla^2 f(x) \geq 0$$

(*) When f is **strongly convex** with constant m:

$$\nabla^2 f(x) \succeq mI \qquad \forall x \in \mathbf{dom}\, f$$

2^{nd}-Order Characterization: More on Bound

$$\nabla^2 f(x) \succeq 0$$

Then for strong convex, where $\nabla^2 \big(f(x) - \alpha \|x\|^2 \big) \succeq 0$, we have:

$$\nabla^2 f(x) \succeq \nabla_x^2 \alpha \|x\|^2$$

and we often take the bound of $\nabla_x^2 \alpha \|x\|^2$ as m. For instance, in the case of $\nabla_x^2 \alpha \|x\|_2^2$, $m = 2\alpha$.

Note that α and m are usually different constants. But it doesn't matter such much in practice.

The α-*sublevel set* of a function $f : \mathbb{R}^n \to \mathbb{R}$ is defined as:

$$C_\alpha = \{x \in \mathbf{dom}\, f \mid f(x) \leq \alpha\}$$

Sublevel sets of a convex function are convex, for any value of α.

Proof: $\forall x, y \in C_\alpha$, $f(x) \leq \alpha$, $\forall \theta \in [0, 1]$, $f(\theta x + (1-\theta)y) \leq \alpha$, and hence $\theta x + (1-\theta)y \in C_\alpha$.

The converse is **not** true: a function can have **all** its sublevel sets convex (a.k.a. *quasiconvex*), but **not** convex itself. e.g. $f(x) = -e^x$ is *concave* in $\mathbb{R}$ but all its sublevel sets are *convex*.

If f is concave, then its α-*superlevel set* is a convex set:

$$\{x \in \mathbf{dom}\, f \mid f(x) \geq \alpha\}$$

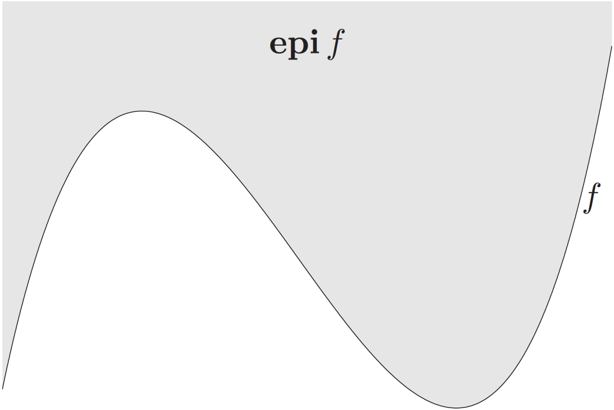


Figure: The illustration of graph and epigraph from textbook. Epigraph of f is the shaded part, graph of f is the dark line.

Graph and Epigraph

The graph of a function $f : \mathbb{R}^n \to \mathbb{R}$ is a subset of $\mathbb{R}^{n+1}$:

$$\{(x, f(x)) \mid x \in \mathbf{dom}\, f\}$$

The *epigraph* of it is also subset of $\mathbb{R}^{n+1}$, defined as:

$$\mathbf{epi}\, f = \{(x, t) \mid x \in \mathbf{dom}\, f,\ f(x) \leq t\}$$

The link between convex sets and convex functions is via the epigraph: A function is convex iff its epigraph is a convex set.

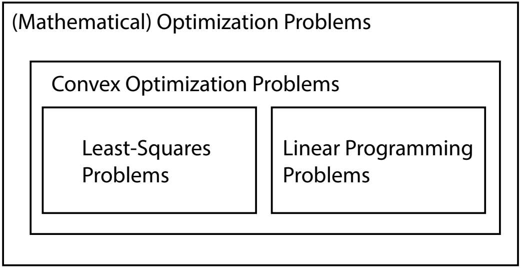


Figure: "... least-squares and linear programming problems have a fairly complete theory, arise in a variety of applications, and can be solved numerically very efficiently ... the same can be said for the larger class of convex optimization problems." — from textbook

Considering the following *mathematical optimization problem* (a.k.a *optimization problem*):

$$\begin{array}{rl} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le b_i, \; i = 1, 2, \ldots m \end{array}$$

- $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is the *optimization variable*
- $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the *objective function*
- $f_i : \mathbb{R}^n \to \mathbb{R}$ $(i = 1, 2, \ldots, m)$ are the *constraint functions*

A vector x^* is called *optimal*, or called a *solution* of the problem, iff: $\forall z$ satisfying every $f_i(z) \le b_i$ $(i = 1, 2, \ldots, m)$, we have $f_0(z) \ge f_0(x^*)$.

Least-Squares Problems (Linear ver.)

$$\text{minimize} \qquad f_0(x) = \|Ax - b\|_2^2 = \sum_{i=1}^{k} (a_i^T x - b_i)^2$$

It has no *constraints*. $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{k \times n}$, $k \geq n$. $a_i \in \mathbb{R}^n$ are the rows of the coefficient matrix A.

The solution can be reduced to solving a set of linear equations:

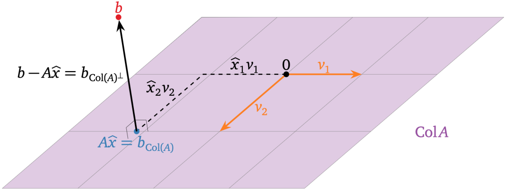
$$A^T A x = A^T b$$

We have analytical solution:

$$x = (A^T A)^{-1} A^T b$$

Can be solved in approximately $\mathcal{O}(n^2 k)$ time if A is dense, otherwise much faster.

Least-Square Problems’ Solution



$$-1.10\, v_1 + 1.00\, v_2 = b_{\mathrm{Col}(A)}$$

$$A\hat{x} = \begin{bmatrix} 0.00 & 1.10 \\ 1.00 & 0.00 \\ 0.00 & -0.20 \end{bmatrix} \begin{bmatrix} -1.10 \\ 1.00 \end{bmatrix} = \begin{bmatrix} 1.10 \\ -1.10 \\ -0.20 \end{bmatrix} = b_{\mathrm{Col}(A)}$$

Figure: Illustration of how we get the solution of a least-square problem. $k = 3$, $n = 1$. With $Col(A)$ be the set of all vectors of the form Ax (the column space, consistent), the closest vector of the form Ax to b is the orthogonal projection of b onto $Col(A)$. Figure from `https://textbooks.math.gatech.edu/ila/least-squares.html`.

$$\begin{array}{ll} \text{minimize} & f_0(x) = c^T x \\ \text{subject to} & f_i(x) = a_i^T x \leq b_i, \qquad i = 1, 2, \ldots m \end{array}$$

It is called *linear programming*, because the objective (parameterized by $c \in \mathbb{R}^n$) and all constraint functions (parameterized by $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$) are linear.

- No simple analytical solution.
- Cannot give exact number of arithmetic operations required.
- A lot of effective methods, include:
  - Dantzig's *simplex method* [3]
  - Interior-point methods (most recent)
    - Time complexity can be estimated to a given accuracy, usually around $\mathcal{O}(n^2 m)$ in practice (assuming $m \geq n$).
    - Could be extended to *convex optimization* problems.

[3] It's the thing you've be taught in junior high school.

Many optimization problems can be transformed to an equivalent linear program. For example, the *Chebyshev approximation problem*:

$$\text{minimize} \qquad \max_{i=1,2,\ldots k} |a_i^T x - b_i|$$

Many optimization problems can be transformed to an equivalent linear program. For example, the *Chebyshev approximation problem*:

$$\text{minimize} \qquad \max_{i=1,2,\dots k} |a_i^T x - b_i|$$

It can be solved by solving:

$$\begin{array}{rll} \text{minimize} & t & \\ \text{subject to} & a_i^T x - t \leq b_i, & i = 1, 2, \dots, k \\ & -a_i^T x - t \leq b_i, & i = 1, 2, \dots, k \end{array}$$

Here, $a_i, x \in \mathbb{R}^n$, $b_i, t \in \mathbb{R}$.

A convex optimization problem is one of the form

$$\begin{array}{rl} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i, \; i = 1, 2, \ldots m \end{array}$$

where the functions $f_0, f_1, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ are all convex functions. That is, they satisfy:

$$f_i(\theta x + (1-\theta)y) \leq \theta f_i(x) + (1-\theta) f_i(y)$$

$\forall x, y \in \mathbb{R}^n$, $\theta \in [0, 1]$.

The *least-squares problem* and *linear programming problem* are both special cases of the general *convex optimization problem*.

- No analytical formula for the solution.
- Interior-point methods work very well in practice, but no consensus has emerged yet as to what the best method or methods are, and it is still a very active research area.
- We cannot yet claim that solving general convex optimization problems is a mature technology.
- For some subclasses of *convex optimization* problems, e.g. *second-order cone* programming or *geometric programming*, interior-point methods are approaching mature technology.

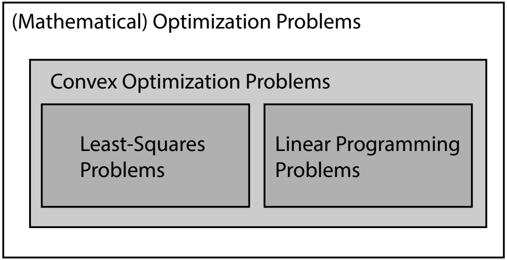


Figure: Illustration of what are included in the *nonlinear optimization problems* (grey parts are wiped out). The problems where (1) $\exists f_i$ not linear, (2) the problem is **not known** to be convex.

No effective methods for solving the general *nonlinear programming* problem, and the different approaches each of involves some compromise.

- Local optimization: "more art than technology"
- Global optimization: "the compromise is efficiency"

Convex optimization also helps with non-convex problems from:

- Initialization for local optimization:
  1. Find an approximate, but convex, formulation of the problem.
  2. Use the approximate convex problem's exact solution to handle the original non-convex problem.

Nonlinear Optimization Study II

- Introduce convex heuristics for solving nonconvex optimization problems, e.g:
  - Sparsity: when and why it is preferred.
  - The use of *randomized algorithms* to find the best parameters.
- Estimating the bounds, e.g. estimating the lower bound on the optimal value (the best-possible value):
  - Lagrangian relaxation:
    1. Solve the Lagrangian dual problem, which is convex
    2. It provides a lower bound on the optimal value
  - Relaxation:
    - Each nonconvex constraint is replaced with a looser, but convex, constraint.

Convex Optimization

The problem is often expressed as:

$$
\begin{array}{rll}
\text{minimize} & f_0(x) & \\
\text{subject to} & f_i(x) \le 0, & i = 1, 2, \ldots, m \\
& h_i(x) = 0, & i = 1, 2, \ldots, p
\end{array}
$$

The domain $\mathcal{D}$ is defined as:

$$
\mathcal{D} = \bigcap_{i=0}^{m} \mathbf{dom}\, f_i \cap \bigcap_{i=1}^{p} \mathbf{dom}\, h_i
$$

A point $x \in \mathcal{D}$ is *feasible* if it satisfies the constraints (f_i for $i = 1, \ldots, m$, and h_i for $i = 1, \ldots, p$).
The *optimal value* p^* is $\inf f_0(x)$ when x is feasible. An optimal point x^* satisfies $f_0(x^*) = p^*$.

The standard form optimization problem is convex optimization problem when satisfying three additional conditions:

1. The objective function f_0 must be convex;
2. The inequality constraint functions f_i $(i = 1, 2, \ldots, m)$ must be convex;
3. The equality constraint functions $h_i(x) = a_i^T x - b_i$ $(i = 1, 2, \ldots, p)$ must be affine.

An important property coming after: The *feasible set* $\mathcal{D}$ must be convex, as it is the *intersection* of the above-listed convex functions.

The *epigraph form* is in the form ($x \in \mathbb{R}^n$, $t \in \mathbb{R}$), obviously equivalent with standard form:

$$\begin{array}{rll} \text{minimize} & t & \\ \text{subject to} & f_0(x) - t \leq 0 & \\ & f_i(x) \leq 0, & i = 1, 2, \ldots, m \\ & h_i(x) = 0, & i = 1, 2, \ldots, p \end{array}$$

Note that the objective function of the epigraph form problem is a linear function of the variables x, t.
It can be interpreted geometrically as minimizing t over the epigraph of f_0, subject to the constraints on x.

Local and Global Optima

A fundamental property of convex optimization problems is that any locally optimal point is also (globally) optimal.

Proof: Assume x is local optima, then $x \in \mathcal{D}$, and for some $R > 0$,

$$f_0(x) = \inf\{f_0(z) \mid z \in \mathcal{D}, \|z - x\|_2 \leq R\}$$

Now assume it is not global optima, then $\exists y \in \mathcal{D}$, $f_0(y) < f_0(x)$. There must be $\|y - x\|_2 > R$. Consider point z given by:

$$z = (1 - \theta)x + \theta y \qquad \theta = \frac{R}{2\|y - x\|_2} < \frac{1}{2}$$

Therefore, $\|z - x\|_2 = \frac{R}{2} < R$. By convexity of feasible set $\mathcal{D}$, $z \in \mathcal{D}$, and f_0 is convex. Then it contradicts the assumption:

$$f_0(z) \leq (1 - \theta)f_0(x) + \theta f_0(y) < f_0(x)$$

When solving an optimization problem, we follow the following steps:

1. Reformulate the problem into the standard format / epigraph format / other known equivalent format (e.g. LP (Linear Program), QP (Quadratic Program), SOCP (Second-Order Cone Program), GP (Geometric Program), CP (Cone Program), SDP (Semidefinite Program)); [4]
2. We could form highly nontrivial bounds on convex optimization problems by duality. (Weak) duality works even for hard problems that are not necessarily convex (but the functions involved must be convex).
3. The problem could be solved by solving the KKT (Karush-Kuhn-Tucker) conditions.

[4] Textbook Chapter 4.

When f_0 is a constant, the problem becomes a *feasibility problem.*

When there's no $f_1, \ldots, f_m$ and no $h_1, \ldots h_p$, the problem is an *unconstrained minimization problem.*

- *Feasibility* $\rightarrow$ *unconstrained minimization*: make a new f_0' with value 0 (or other constants) when $x \in \mathcal{D}$, otherwise $f_0'(x) = +\infty$.
- *Unconstrained minimization* $\rightarrow$ *feasibility*: introduce $f_0(x) \leq p^* + \epsilon$ as the constraint and remove the objective.

Infeasible problem: $p^* = +\infty$; unbounded problem: $p^* = -\infty$.

LPs are normally in the form:

$$\begin{array}{rl} \text{minimize} & c^T x + d \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

With slack variable $s \in \mathbb{R}^m \succeq 0$ introduced and $x = x^+ - x^-$, $x^+, x^- \succeq 0$:

$$\begin{array}{rl} \text{minimize} & c^T x^+ - c^T x^- + d \\ \text{subject to} & Gx^+ - Gx^- + s = h \\ & Ax^+ - Ax^- = b \\ & s, x^+, x^- \succeq 0 \end{array}$$

Example: Converting a problem into LP I

Consider the Chebyshev center of a polyhedron $\mathcal{P}$, defined as:

$$\mathcal{P} = \{x \in \mathbb{R}^n \mid a_i^T x \leq b_i,\ i = 1, 2, \ldots m\}$$

We want to find the largest Euclidean ball that lies in $\mathcal{P}$, whose center is known as the Chebyshev center of the polyhedron.
The ball is represented as:

$$\mathcal{B} = \{x_c + u \mid \|u\|_2 \leq r\}$$

The variables: $x_c \in \mathbb{R}^n$, $r \in \mathbb{R}$, problem: maximize r subject to the constraint $\mathcal{B} \subseteq \mathcal{P}$.

Example: Converting a problem into LP II

We start from observing that $x = x_c + u$ from $\mathcal{B}$, and that $x \in \mathcal{P}$, thus:

$$a_i^T(x_c + u) = a_i^T x_c + a_i^T u \leq b_i$$

$\|u\|_2 \leq r$ infers that:

$$\sup\{a_i^T u \mid \|u\|_2 \leq r\} = r\|a_i\|_2$$

and that the condition we have is:

$$a_i^T x_c + r\|a_i\|_2 \leq b_i$$

a linear inequality in (x_c, r).

$$\begin{array}{ll} \text{minimize} & -r \\ \text{subject to} & a_i^T x_c + r\|a_i\|_2 \leq b_i, \qquad i = 1, 2, \ldots m \end{array}$$

Lagrangian Duality

Consider the standard form written as:

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0, & i = 1, 2, \ldots, m \\ & h_i(x) = 0, & i = 1, 2, \ldots, p \end{array}$$

Denote the optimal value as p^*. Its Lagrangian, $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$, with $\mathbf{dom}\, L = \mathcal{D} \times \mathbb{R}^m \times \mathbb{R}^p$:

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

It is basically a weighted sum of the objective and the constraints. The Lagrange dual function $g : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$:

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu)$$

Denote g's optimal point, or *dual optimal point*, as (λ^*, ν^*).

g is **always** *concave*, and could reach $-\infty$ at some λ, ν values.

There's an important **lower-bound property**: If $\lambda \succeq 0$, then $g(\lambda, \nu) \leq p^*$.

Proof: Since $x^* \in \mathcal{D}$, $f_i(x^*) \leq 0$, $h_i(x^*) = 0$, thus:

$$p^* = f_0(x^*) \geq L(x^*, \lambda^*, \nu^*) \geq \inf_{x \in \mathcal{D}} L(x, \lambda^*, \nu^*) = g(\lambda^*, \nu^*)$$

KKT: Proof of Complementary Slackness

Assume strong duality holds, then:

$$\begin{aligned} f_0(x^*) &= g(\lambda^*, \nu^*) \\ &= \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i^* f_i(x) + \sum_{i=1}^{p} \nu_i^* h_i(x) \right) \\ &\leq f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* f_i(x^*) + \sum_{i=1}^{p} \nu_i^* h_i(x^*) \\ &\leq f_0(x^*) \end{aligned}$$

because of $\lambda_i \geq 0$, $f_i(x^*) \leq 0$, $h_i(x^*) = 0$. Therefore,

$$\sum_{i=1}^{m} \lambda_i^* f_i(x^*)$$

Since each term is non-positive, $\lambda_i^* f_i(x^*) = 0$.

For a problem with differentiable f_i and h_i, we have four conditions that togetherly named KKT conditions:

- **Primal Constraints**:

$$\begin{cases} f_i(x) \leq 0 & i = 1, 2, \dots, m \\ h_i(x) = 0 & i = 1, 2, \dots, p \end{cases}$$

- **Dual Constraints**: $\lambda \succeq 0$
- **Complementary Slackness**: $\lambda_i f_i(x) = 0$ $(i = 1, 2, \dots, m)$
- **gradient of Lagrangian vanishes** (with respect to x):

$$\nabla_x L(x, \lambda, \nu) = \nabla_x f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla_x f_i(x) + \sum_{i=1}^{p} \nu_i \nabla_x h_i(x) = 0$$

KKT Properties

- If strong duality holds and (x, λ, ν) are optimal, then KKT condition must be satisfied.
- If the KKT condition is satisfied by (x, λ, ν), strong duality must hold and the variables are optimal.
- If Slater's Conditions (see textbook section 3.5.6, these conditions imply strong duality) is satisfied, and x is optimal $\iff \exists(\lambda, \nu)$ that satisfy KKT conditions.

Least Square into Standard Form

The original form of least-square problem

$$\text{minimize} \qquad \|Ax - b\|_2^2$$

could be transformed into:

$$\begin{array}{rl} \text{minimize} & x^T x \\ \text{subject to} & Ax = b \end{array}$$

With independent rows, we have that AA^T is nonsingular, and thus:

$$A^\dagger = A^T(AA^T)^{-1}$$

With independent columns, we have that A^TA is nonsingular, and thus:

$$A^\dagger = (A^TA)^{-1}A^T$$

Recall that previously, we said that for a least-square problem, $\|Ax - b\|_2^2$, sometimes it doesn't exist an A^{-1}, thus we use $A^TAx = A^Tb$ instead, the pseudo inverse is a formal definition of this operation.

Consider the following problem with $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$:

$$\begin{array}{rl} \text{minimize} & x^T x \\ \text{subject to} & Ax = b \end{array}$$

Example: Solving Least-squares I

Consider the following problem with $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$:

$$\begin{array}{rl} \text{minimize} & x^T x \\ \text{subject to} & Ax = b \end{array}$$

Solution: The Lagrangian of this problem is (no need λ):

$$L(x, \nu) = x^T x + \nu^T (Ax - b)$$

The KKT conditions are:

- **Primal Constraints**: $Ax = b$
- **Dual Constraints**: None
- **Complementary Slackness**: None
- **gradient of Lagrangian vanishes** (with respect to x):

$$\nabla_x L(x, \nu) = 2x + A^T \nu = 0$$

From **gradient of Lagrangian vanishes**, $x^* = -(1/2)A^T\nu^*$; from **Primal Constraints**, $Ax^* = b$. Therefore:

$$\begin{aligned} AA^T\nu^* &= -2b \\ \nu^* &= -2(AA^T)^{-1}b \\ x^* &= A^T(AA^T)^{-1}b = A^\dagger b \end{aligned}$$

It agrees with the previous version in terms of pseudo-inverse:

$$x^* = (A^TA)^{-1}A^Tb = A^\dagger b$$

- Descent Methods
  - To find the best step size we do *line search*, but the particular choice of line search does not matter such much, instead, the particular choice of search direction matters a lot.
  - SGD, AdaGrad, Adam, etc. Almost all popular optimizers today.
- Newton's Method
  - In theory faster convergence, in practice much larger space.
  - (*) Prof. Lin's course projects (Newton + CNN)
- Interior-point Methods
  - Applying Newton's method to a sequence of modified versions of the KKT conditions.

$$x^{(k+1)} = x^{(k)} + \eta \Delta x^{(k)} \qquad f(x^{(k+1)}) < f(x^{(k)})$$

where $\Delta x^{(k)}$ is called a step, and $|\eta| = -\eta$ the step size. From convexity, it implies:

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

Step size could be determined by line-search, optimized along the direction of $\nabla f(x) \Delta x$.

$$f(x + \eta \Delta x) \approx f(x) + \eta \nabla f(x)^T \Delta x$$

Exact line search:

$$\eta^* = \operatorname*{argmin}_{t>0} f(x + \eta \Delta x)$$

Backtracking line search:

- Parameters: $\alpha \in (0, 0.5), \beta \in (0, 1)$
- Start with $\eta = 1$, repeat:
  1. Stop when:

$$f(x + \eta \Delta x) < f(x) + \alpha \eta \Delta f(x)^T \Delta x$$

  2. If not stop, update $\eta := \beta \eta$.

Both strategies are used for selecting a proper step size. Not very important in practice.

In steepest descent methods, instead of optimizing towards the direction of $\nabla f(x)^T \Delta x$, it searches for the unit-vector v with the most negative $\nabla f(x)^T v$ — the directional derivative of f at x in the direction v. In other words:

$$x^{(k+1)} = x^{(k)} + \eta \Delta x_{nsd}^{(t)}$$

where x_{nsd} is defined as:

$$\Delta x_{nsd} = \operatorname*{argmin}_v \{\nabla f(x)^T v \mid \|v\| = 1\}$$

Use subgradient $g \in \partial f(x)$ instead of gradient $\nabla f(x)$, which means that,

$$f(y) \geq f(x) + g^T(y - x), \ \forall y$$

There could be multiple g for the same x. The advantage of using g is that it enables the function to handle non-derivative functions.

g for $x^{(t)}$ is denoted as $g^{(t)}$.

$$x^{(k+1)} = x^{(k)} + \eta g^{(t)}$$

[5]In Prof. Gu's Slides only, not included in textbook.

Giving starting point $x \in \mathbf{dom}\, f$ and tolerance $\epsilon > 0$, repeat the following steps:

1. Compute Newton step: $\Delta x_{nt} = -\frac{\nabla f(x)}{\nabla^2 f(x)}$
2. Compute Newton decrement:
   $\lambda(x)^2 = (\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x))$
3. Quit if $\frac{\lambda(x)^2}{2} \leq \epsilon$
4. Select step size η by backtracking line-search
5. $x = x + \eta \Delta x_{nt}$

- $x + \Delta x_{nt}$ minimized second-order approximation:

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

- $x + \Delta x_{nt}$ solves linearized optimality condition:

$$\nabla f(x+v) \approx \nabla \hat{f}(x+v) = \nabla f(x) + \nabla^2 f(x) v = 0$$

- Δx_{nt} is steepest descent direction at x in local Hessian norm:

$$\|u\|_{\nabla^2 f(x)} = \left(u^T \nabla^2 f(x) u\right)^{1/2}$$

- $\lambda(x)$ is an approximation of $f(x) - p^*$, with p^* estimated by $\inf_y \hat{f}(y)$:

$$f(x) - \inf_y \hat{f}(y) = \frac{1}{2}\lambda(x)^2$$

Define the logarithm barrier function:

$$\phi(x) = -\sum_{i=1}^{m} \log(-f_i(x)), \ \mathbf{dom}\,\phi(x) = \{x \mid f_i(x) < 0, \ i = 1, \dots m\}$$

it preserves the convexity and the twice continuously differentiable (if any) of f_i, and could turn the inequality constraints from explicit to implicit:

$$\begin{array}{rl} \text{minimize} & f_0(x) + \phi(x) \\ \text{subject to} & h_i(x) = 0, \ i = 1, 2, \dots, p \end{array}$$

$$
\begin{array}{ll}
\text{minimize} & f_0(x) + \phi(x) \\
\text{subject to} & Ax = b
\end{array}
$$

Interior point methods does not work well if some of the constraints are not strictly feasible:

- f_i is convex and twice continuously differentiable
- $A \in \mathbb{R}^{p \times n}$ and A's rank is p
- p^* is finite and attained
- The problem is strictly feasible (exists interior point), hence, strong duality holds and dual optimum is attained.

It is the algorithm coming directly from primal-dual methods. In brief, at iteration step t, we set $x^*(t)$ as the solution of:

$$\begin{array}{rl} \text{minimize} & t f_0(x) + \phi(x) \\ \text{subject to} & Ax = b \end{array}$$

t exists here as a balance of $\phi(x)$'s increasing value, forcing the algorithm to focus on f_0 more in the end.

We have central path defined as $\{x^*(t) \mid t > 0\}$, the path alone which we minimizes the Lagrangian, and:

$$\lim_{t \to \infty} f_0(x^*(t)) = p^*$$

Lipschitz constraint is a very common type of constraint applied to the functions, being L-Lipschitz meaning:

$$|f(x) - f(y)| \leq L\|x - y\|, \qquad \forall x, y \in \mathbf{dom}\, f$$

L is called the *coefficient*.

Lipschitzness is very important in analyzing convergence of optimization problems, in both convex cases and non-convex cases.

We need it to analyze from one step to the next, although sometimes it is omitted in the end.

The Lipschitz: Interpretation

The coefficient L of f can be interpreted as:

- A bound on the next-level derivative of f
  - Can taken to be zero if f is constant.
- More generally, L measures how well f can be approximated by a constant.
  - If $f = \nabla g$ then L measures how well g can be approximated by a linear model.
  - If $f = \nabla^2 h$ then L measures how well h can be approximated by a quadratic model.

Consider the convergence analysis of Newton, in unbounded optimization, where the objective f is:

- Twice continuously differentiable: $\nabla f(x)$ and $\nabla^2 f(x)$ exist;
- Strongly convex with constant m: $\nabla^2 f(x) \succeq mI \ (x \in \mathcal{D})$
  - It implies that $\exists M > 0, \forall x \in \mathcal{D}, \nabla^2 f(x) \preceq MI$. (Proof on next page)
- The Hessian of f is L-Lipschitz continuous on $\mathcal{D}$, $\forall x, y \in \mathcal{D}$:

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_2 \leq L\|x - y\|_2$$

This part's proof comes from textbook 9.1.2.

First, by using the 1^{st}-order characterization of convex function f, we have that, $\forall x, y \in \mathbf{dom}\, f$:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

with the previously-mentioned quadratic Taylor approximation:

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

In the case of strong convex with constant $m > 0$, we have $\nabla^2 f(x) \succeq mI$, thus

$$(y - x)^T \nabla^2 f(z)(y - x) \geq m\|y - x\|_2^2$$

Therefore we have:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2}\|y - x\|_2^2$$

This inequality implies that the sublevel sets contained in **dom** f are bounded, so **dom** f is bounded. It essentially means that the maximum eigenvalue of $\nabla^2 f(x)$, which is a continuous function of x on **dom** f, is bounded above on **dom** f, i.e., there exists a constant $M > 0$ such that:

$$\nabla^2 f(x) \preceq MI$$

Note that m are M are often *unknown in practice*.

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{M}{2}\|y - x\|_2^2$$

Still from textbook 9.1.2.

Previously we have had:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2} \|y - x\|_2^2$$

Now, considering a fixed x, it is obvious that the right-hand-side is a convex quadratic function of y. Where we find the $\tilde{y}$ that minimizes it is the one that achieves zero derivative:

$$\nabla_{\tilde{y}} \left(f(x) + \nabla f(x)^T (\tilde{y} - x) + \frac{m}{2} \|\tilde{y} - x\|_2^2 \right) = \nabla f(x) + m(\tilde{y} - x) = 0$$

$$\tilde{y} = x - \frac{1}{m} \nabla f(x)$$

And therefore,

$$\begin{aligned} f(y) &\geq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2}\|y - x\|_2^2 \\ &\geq f(x) + \nabla f(x)^T (\tilde{y} - x) + \frac{m}{2}\|\tilde{y} - x\|_2^2 \\ &= f(x) - \frac{1}{m}\nabla f(x)^T \nabla f(x) + \frac{1}{2m}\|\nabla f(x)\|_2^2 \\ &= f(x) - \frac{1}{2m}\|\nabla f(x)\|_2^2 \end{aligned}$$

Since it holds for $\forall y \in \mathcal{D}$, we can say that:

$$p^* \geq f(x) - \frac{1}{2m}\|\nabla f(x)\|_2^2$$

Strong-Convexity Other Implications III

It is often used as the upper-bound estimation of error:

$$\epsilon = f(x) - p^* \leq \frac{1}{2m} \|\nabla f(x)\|_2^2$$

Similarly, applying the same strategy to:

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{M}{2} \|y - x\|_2^2$$

we have an lower bound of the error ϵ:

$$p^* \leq f(x) - \frac{1}{2M} \|\nabla f(x)\|_2^2$$

$$\epsilon = f(x) - p^* \geq \frac{1}{2M} \|\nabla f(x)\|_2^2$$

Outline of the proof: $\exists \eta \in (0, \frac{m^2}{L}]$, $\gamma > 0$, such that

$$\begin{cases} f(x^{(k+1)}) - f(x^{(k)}) \le -\gamma & \|\nabla f(x)\|_2 \ge \eta \\ \frac{L}{2m^2}\|\nabla f(x^{(k+1)})\|_2 \le \left(\frac{L}{2m^2}\|\nabla f(x^{(k)})\|_2\right)^2 & \|\nabla f(x)\|_2 < \eta \end{cases}$$

There are two phases of the problem ($t^{(k)}$ is the step size here):

1. Damped Newton phase ($\|\nabla f(x)\|_2 \ge \eta$):
   - Most iterations require backtracking steps
   - Function value decreases by at least γ
   - If bounded ($p^* > -\infty$), this phase costs *iterations* no more than
   $$\frac{f(x^{(0)}) - p^*}{\gamma}$$
2. Quadratically convergent phase ($\|\nabla f(x)\|_2 < \eta$):
   - All iterations use step size $t^{(k)} = 1$

- $\|\nabla f(x)\|_2$ converges to zero quadratically:

$$\frac{L}{2m^2}\|\nabla f(x^{(k+1)})\|_2 \leq \left(\frac{L}{2m^2}\|\nabla f(x^{(k)})\|_2\right)^2 \leq \frac{1}{2}$$

We've set $\eta \leq \frac{L^2}{m}$, thus for $k+1$ and $\|\nabla f(x^{(k)})\|_2 < \eta$, we have:

$$\|\nabla f(x^{(k+1)})\|_2 \leq \frac{L}{2m^2}\|\nabla f(x^{(k)})\|_2^2 < \frac{\eta^2 L}{2m^2} \leq \frac{\eta}{2} < \eta$$

and it holds for $\forall l > k$. More generally:

$$\frac{L}{2m^2}\|\nabla f(x^{(l)})\|_2 \leq \left(\frac{L}{2m^2}\|\nabla f(x^{(k)})\|_2\right)^{2^{l-k}} \leq \left(\frac{1}{2}\right)^{l-k}$$

$$\|\nabla f(x^{(l)})\|_2^2 \leq \frac{4m^4}{L^2}\left(\frac{1}{2}\right)^{l-k+1}$$

From strong convexity we know:

$$f(x) - p^* \leq \frac{1}{2m} \|\nabla f(x)\|_2^2$$

$$f(x^{(l)}) - p^* \leq \frac{1}{2m} \|\nabla f(x^{(l)})\|_2^2 \leq \frac{2m^3}{L^2} \left(\frac{1}{2}\right)^{l-k+1} \leq \epsilon$$

It implies that it converges fast at this phase.

Define $\epsilon_0 = \frac{2m^3}{L}$, then we have that, We can bound the number of iterations in the quadratically convergent phase by:

$$\log_2 \log_2 \left(\frac{\epsilon_0}{\epsilon}\right)$$

Consider a function f which is ρ-Lipschitz, updated via sub-gradient descent.

$$\because f(x^{(t)}) - f(x^*) = (f(x^{(t+1)}) - f(x^{(t)})) - (f(x^{(t+1)}) - f(x^*)) \leq 0$$

$$\therefore \langle x^{(t+1)} - x^{(t)}, g^{(t+1)} \rangle - \langle x^{(t+1)} - x^*, g^{(t+1)} \rangle \leq 0$$

$$\therefore \langle x^{(t+1)} - x^*, x^{(t+1)} - x^{(t)} \rangle \leq 0$$

$$\iff \langle x^{(t+1)} - x^*, (x^{(t+1)} - x^*) - (x^{(t)} - x^*) \rangle \leq 0$$

$$\therefore \|(x^{(t)} - x^*) - (x^{(t+1)} - x^*)\|^2 \leq \|x^{(t+1)} - x^*\|^2 + \|x^{(t+1)} - x^*\|^2$$

$$(\because (x^{(t)} - x^{(k)})g^{(t)} \le f(x^{(t)}) - f(x^{(k)}))$$

$$\sum_{t=1}^{T} \left(f(x^{(t)}) - f(x^*)\right)$$

$$\le \sum_{t=1}^{T} (x^{(t)} - x^*)^T g^{(t)} \qquad (\because a^2 + b^2 \ge 2ab)$$

$$\le \sum_{t=1}^{T} \left(\frac{\|x^{(t)} - x^*\|^2}{2\eta} + \frac{\eta}{2}\|g^{(t)}\|^2\right) \approx \sum_{t=1}^{T} \left(\frac{\|x^{(t)} - x^{(t+1)}\|^2}{2\eta} + \frac{\eta}{2}\|g^{(t)}\|^2\right)$$

$$\le \sum_{t=1}^{T} \frac{\|x^{(t)} - x^*\|^2 - \|x^{(t+1)} - x^*\|^2}{2\eta} + \sum_{t=1}^{T} \frac{\eta}{2}\|g^{(t)}\|^2$$

$$= \frac{\|x^{(0)} - x^*\|^2 - \|x^{(T+1)} - x^*\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|g^{(t)}\|^2$$

Assume that f is ρ-Lipschitz, then we have $\|g^{(t)}\| \leq \rho$ $(\forall t)$.
Also, $x^{(0)} = 0$, $\lim_{t\to\infty} x^{(t)} \to x^*$.

$$f(x^{(t)}) - f(x^*) \leq \frac{\|x^{(0)} - x^*\|^2 - \|x^{(T+1)} - x^*\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|g^{(t)}\|^2$$

$$\frac{1}{T}\Big(f(x^{(t)}) - f(x^*)\Big) \leq \frac{\|x^*\|^2}{2\eta T} + \frac{\eta\rho^2}{2}$$

For every x^*, if $T \geq \frac{\|x^*\|^2\rho^2}{\epsilon^2}$ and $\eta = \sqrt{\frac{\|x^*\|^2}{\rho^2 T}}$, then the right hand side of the last inequation is at most ϵ.

Examples

Accelerated Methods for Non-Convex Optimization

- Design accelerated methods that doesn't rely on convexity of the optimization problem.
- It relies on that the problem has L_1-Lipschitz continuous gradient and L_2-Lipschitz continuous Hessian.
- Calculate a score α according to L_1, ϵ, and the gradient $\nabla f(x)$ to decide whether or not negative curvature descent should be conducted at each step.
- Apply accelerated gradient descent for *almost-convex function* made for the *almost-convex point* at each step to update that *point*.

Saving gradient and negative curvature computations: Finding local minima more efficiently

- Doesn't require the original problem to be convex.
- Develops an algorithm with fewer steps of computing the negative curvature descent [6].
- Divide the entire domain of the objective function into two regions (by comparing $\|\nabla f(x)\|_2$ with ϵ): large gradient region, small gradient region; and then perform gradient descent-based methods in the large gradient region, and only perform negative curvature descent in the small gradient region.

Official code in PyTorch: `https://github.com/yaodongyu/gose-nonconvex`.

[6] Useful for escaping the small-gradient regions.

Multi-Task Learning as Multi-Objective Optimization work on solving the problem of that multiple tasks might conflict.

- Use the multiple-gradient-descent algorithm (MGDA) optimizer;
- Define the Pareto optimality for MTL (in brief, no other solutions dominants the current solution);
- Use multi-objective KKT (Karush-Kuhn-Tucker) conditions and find a descent direction that decreases all objectives.
- Applicable to any problem that uses optimization **based on gradient descent**.

Implementation: `https://github.com/hav4ik/Hydra`