# Convex Optimization

Shan-Hung Wu
*shwu@cs.nthu.edu.tw*

Department of Computer Science,
National Tsing Hua University, Taiwan

NetDB-ML, Fall 2014

# Outline

1. Optimization Problems
   - Standard Forms and Terminology
   - Problem Classes
2. Convexity
   - Convex Sets
   - Convex Functions
3. Convex Optimization
   - Optimality
   - Disciplined Convex Programming and CVX
   - LP and QP
4. Algorithms
   - Unconstrained Problems
   - Constrained Problems
   - Large-Scale Problems**
5. Duality
   - Weak Duality
   - Strong Duality

# Outline


1. Optimization Problems
   - Standard Forms and Terminology
   - Problem Classes
2. Convexity
   - Convex Sets
   - Convex Functions
3. Convex Optimization
   - Optimality
   - Disciplined Convex Programming and CVX
   - LP and QP
4. Algorithms
   - Unconstrained Problems
   - Constrained Problems
   - Large-Scale Problems**
5. Duality
   - Weak Duality
   - Strong Duality

# Functional Form

- An ***optimization problem*** is to minimize an ***objective*** (or cost) function $f : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}$:

$$\begin{gathered}\min_{\boldsymbol{x}} f(\boldsymbol{x}) \\ \text{subject to } \boldsymbol{x} \in C\end{gathered}$$

  where $C \subseteq \mathbb{R}^n$ is called the ***feasible set*** containing ***feasible points*** (or variables)
- If $C = \mathbb{R}^n$, we say the optimization problem is unconstrained
- Maximizing $f$ equals to minimizing $-f$
- $C$ can be a set of function ***constrains***, i.e., $C = \{\boldsymbol{x} : g_i(\boldsymbol{x}) \leqslant 0, i = 1, \cdots, m\}$
  - Sometimes, we single out equality constrains $C = \{\boldsymbol{x} : g_i(\boldsymbol{x}) \leqslant 0, h_j(\boldsymbol{x}) = 0, i = 1, \cdots, m, j = 1, \cdots, p\}$
  - Each equality constrain can be written as two inequality constrains

# Epigraph form

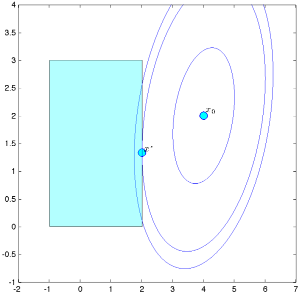
- We can always assume that the objective is a linear function of the variables, via the ***epigraph*** ($epi(f) := \{(\boldsymbol{x}, t) \in \mathbb{R}^{n+1} : \boldsymbol{x} \in \mathbb{R}^n, t \geqslant f(\boldsymbol{x})\}$) representation of the problem

$$\begin{gathered} \min_{\boldsymbol{x},t} t \\ \text{subject to } f(\boldsymbol{x}) - t \leqslant 0, \boldsymbol{x} \in C \end{gathered}$$

- The objective function is $\hat{f} : \mathbb{R}^{n+1} \to \mathbb{R}$, with values $\hat{f}(\boldsymbol{x}, t) = t$
- Consider the $t$-sublevel set of $\hat{f}$ (i.e.,$\{\boldsymbol{x} : t \geqslant \hat{f}(\boldsymbol{x})\}$), the problem amounts to finding the smallest $t$ for which the corresponding sub-level set intersects the set of points satisfying the constraints

# Geometric View

Functional form:
$\min_{\boldsymbol{x}} 0.9x_1^2 - 0.4x_1x_2 - 0.6x_2^2 - 6.4x_1 - 0.8x_2 : -1 \leqslant x_1 \leqslant 2, 0 \leqslant x_2 \leqslant 3$
Epigraph form:
$\min_{\boldsymbol{x},t} t : t \geqslant 0.9x_1^2 - 0.4x_1x_2 - 0.6x_2^2 - 6.4x_1 - 0.8x_2, -1 \leqslant x_1 \leqslant 2, 0 \leqslant x_2 \leqslant 3$



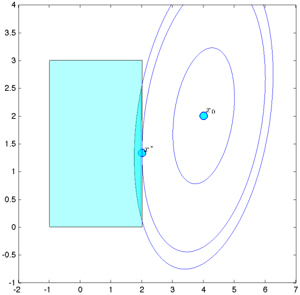

The level sets of the objective function are shown as blue lines, and the feasible set is the light-blue box. The problem amounts to find the smallest value of $t$ such that $t = f(\boldsymbol{x})$ for some feasible $\boldsymbol{x}$. The two dots are the unconstrained and constrained optimal values respectively

# Terminology (1)

- $p^* := \inf_{\boldsymbol{x}} f(\boldsymbol{x}) : \boldsymbol{x} \in C$ is called the ***optimal value***, which
  - may not exist if the problem is infeasible
  - may not be attained (e.g., in $\min_x e^{-x}$, $p^* = 0$ is attained only when $x \to \infty$)
- We allow $p^*$ to take on the values $\infty$ and $-\infty$ when the problem is either
  - infeasible (the feasible set is empty), or
  - unbounded below (there exists feasible points such that $f(\boldsymbol{x}) \to -\infty$), respectively
- A feasible point $\boldsymbol{x}^*$ is called the ***optimal point*** if $f(\boldsymbol{x}^*) = p^*$
- The ***optimal set*** $X^*$ is the set of all optimal points, i.e., $X^* := \{\boldsymbol{x} \in C : f(\boldsymbol{x}) = p^*\} = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}) : \boldsymbol{x} \in C$
- We say the problem is ***attained*** iff $C \neq \emptyset$ and $p^*$ is attained (or equivalently, $X^* \neq \emptyset$)

# Terminology (2)

- The *$\epsilon$-suboptimal set* $X^{\epsilon}$ is defined as $X^{\epsilon} := \{\boldsymbol{x} \in C : f(\boldsymbol{x}) \leqslant p^* + \epsilon\}$




An $\epsilon$-suboptimal set is marked in darker color. This corresponds to the set of feasible points that achieves an objective value less or equal than $p^* + \epsilon$

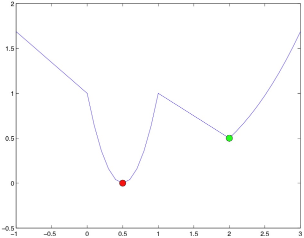
- In practice, we may be only interested in suboptimal solutions

# Local vs. Global Optimality

- A point $z$ is ***locally optimal*** if there is a value $\delta > 0$ such that $z$ is optimal for problem (with new objective $\widetilde{f}(\boldsymbol{x}, z) = f(\boldsymbol{x})$)

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) : z, \boldsymbol{x} \in C, \|\boldsymbol{x} - z\| \leqslant \delta$$

  - That is, a local minimizer minimizes $f$, but only for its nearby points in the feasible set




Minima of a nonlinear function. The value at a local minimizer is not necessarily the (global) optimal value of the problem, unless $f$ is a "convex" function (in the sense that $epi(f)$ is a "convex" set)

# Outline

1. Optimization Problems
   - Standard Forms and Terminology
   - Problem Classes
2. Convexity
   - Convex Sets
   - Convex Functions
3. Convex Optimization
   - Optimality
   - Disciplined Convex Programming and CVX
   - LP and QP
4. Algorithms
   - Unconstrained Problems
   - Constrained Problems
   - Large-Scale Problems**
5. Duality
   - Weak Duality
   - Strong Duality

# Linear Programming

- ***Linear Programming*** (***LP***) has the form:[1]

$$\min_{\boldsymbol{x}} \boldsymbol{c}^\top \boldsymbol{x}$$
$$\text{subject to } \boldsymbol{G}\boldsymbol{x} \leqslant \boldsymbol{h}, \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

  where $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{G} \in \mathbb{R}^{m\times n}$, $\boldsymbol{h} \in \mathbb{R}^m$,$\boldsymbol{A} \in \mathbb{R}^{p\times n}$, and $\boldsymbol{b} \in \mathbb{R}^p$
- The objective and the $m+p$ constrain functions are ***all affine*** (i.e., translated linear)
  - Note $\min_{\boldsymbol{x}} \boldsymbol{c}^\top \boldsymbol{x} + d$ for some fixed $d \in \mathbb{R}$ amounts to $\min_{\boldsymbol{x}} \boldsymbol{c}^\top \boldsymbol{x}$

[1] The term "programming" has nothing to do with computer programs. It is named so due to historical reasons.

# Quadratic Programming

- ***Quadratic Programming*** (***QP***) has the form:

$$\begin{gathered}\min_{\boldsymbol{x}} \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to } \boldsymbol{G}\boldsymbol{x} \leqslant \boldsymbol{h}, \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\end{gathered}$$

  where $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$, $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{G} \in \mathbb{R}^{m\times n}$, $\boldsymbol{h} \in \mathbb{R}^m$,$\boldsymbol{A} \in \mathbb{R}^{p\times n}$, and $\boldsymbol{b} \in \mathbb{R}^p$
- The objective is a quadratic function, and the $m+p$ constrain functions are affine

# Convex Optimization

- A ***convex optimization*** problem is of the form:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$
$$\text{subject to } \boldsymbol{x} \in C$$

  where $f$ is a ***convex function***, and $C$ is a ***convex set***
- In particular, with constrains
  $C = \{\boldsymbol{x} : g_i(\boldsymbol{x}) \leqslant 0, h_j(\boldsymbol{x}) = 0, i = 1, \cdots, m, j = 1, \cdots, p\}$
  - $g_i$ must be ***convex*** functions
  - $h_j$ must be ***affine*** functions (since $h_j$ can be expressed as two $g$'s, the only way to make both $g$'s convex is by letting $h_j$ affine)
- Includes LP, QP with positive semidefinite $\boldsymbol{Q}$, and more

# Combinatorial Optimization

- In combinatorial optimization, some (or all) the variables are Boolean or integers, reflecting discrete choices to be made
  - E.g., Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ be an incidence matrix of a directed graph where $A_{i,j}$ equals to 1 if the arc $j$ starts at node $i$; $-1$ if $j$ ends at $i$; 0 otherwise. The problem of finding the shortest path between nodes 1 and $m$ can be expressed as

$$\min_{\boldsymbol{x}} \mathbf{1}^\top \boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} = [1, 0, \cdots, 0, -1]^\top, \boldsymbol{x} \in \{0, 1\}^n$$

  - E.g., the traveling salesman problem
- Generally, extremely hard to solve
- However, they can often be approximately solved with linear or convex programming
  - E.g., the LP-***relaxed*** single-pair shortest path problem:

$$\min_{\boldsymbol{x}} \mathbf{1}^\top \boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} = [1, 0, \cdots, 0, -1]^\top, \mathbf{0} \leqslant \boldsymbol{x} \in \mathbb{R}^n \leqslant \mathbf{1}$$

# Hard vs. Easy Problems

- We say a problem is hard if cannot be solved in a reasonable amount of time and/or memory space
- Roughly speaking, ***convex problems are easy***; non-convex ones are hard
- Of course, not all convex problems are easy, but a (reasonably large) subset
  - E.g., LP and QP with positive semidefinite $\boldsymbol{Q}$
- Conversely, some non-convex problems are actually easy
  - E.g., the LP-relaxed single-pair shortest path problem has optimal points turn out to be Boolean, so these points are also optimal to the original problem

# Outline


1. Optimization Problems
   - Standard Forms and Terminology
   - Problem Classes
2. Convexity
   - Convex Sets
   - Convex Functions
3. Convex Optimization
   - Optimality
   - Disciplined Convex Programming and CVX
   - LP and QP
4. Algorithms
   - Unconstrained Problems
   - Constrained Problems
   - Large-Scale Problems**
5. Duality
   - Weak Duality
   - Strong Duality

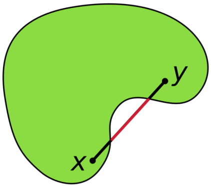
# Convex Sets

## Definition (Convex Set)

A set $C$ of points is ***convex*** iff for any $\boldsymbol{x}, \boldsymbol{y} \in C$ and $\theta \in [0,1]$, we have $(1-\theta)\boldsymbol{x} + \theta\boldsymbol{y} \in C$.

- The point $(1-\theta)\boldsymbol{x} + \theta\boldsymbol{y}$ is called the ***convex combination*** of points $\boldsymbol{x}$ and $\boldsymbol{y}$
- Non-convex set:
- Any convex set you know?

橢圓(內)、凸多邊形(內)、
拋物線(焦點所在的半平面)...

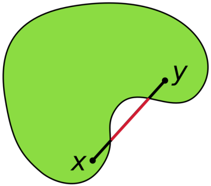


# Convex Sets

## Definition (Convex Set)

A set $C$ of points is ***convex*** iff for any $\boldsymbol{x}, \boldsymbol{y} \in C$ and $\theta \in [0,1]$, we have $(1-\theta)\boldsymbol{x} + \theta\boldsymbol{y} \in C$.

- The point $(1-\theta)\boldsymbol{x} + \theta\boldsymbol{y}$ is called the ***convex combination*** of points $\boldsymbol{x}$ and $\boldsymbol{y}$
- Non-convex set:
- Any convex set you know? $\mathbb{R}^n$, non-negative orthant $\mathbb{R}^n_+$, $\emptyset$, $\{\boldsymbol{x}\}$, line segments, etc.




- A set is said to be a ***convex cone*** if it is convex, and has the property that if $\boldsymbol{x} \in C$, then $\theta\boldsymbol{x} \in C$ for every $\theta \geqslant 0$
  - E.g., $\mathbb{R}^n$, $\mathbb{R}^n_+$, union of scalings of a convex set (must contains $\boldsymbol{0}$)

# More Examples

- Subspaces and affine subspaces such as lines, hyperplanes, and higher-dimensional "flat" sets
- Half-spaces, linear varieties (polyhedra, intersections of half-spaces)
- The ***convex hull***s of a set of points $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m\}$ is a convex set:
  將某集合中所有點包起來的 周長/表面積/...最小的形狀
  $$Co(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m) := \left\{ \sum_{i=1}^{m} \theta_i \boldsymbol{x}_i : \theta_i \geqslant 0, \forall i, \sum_{i=1}^{m} \theta_i = 1 \right\}$$
- Norm balls: $N = \{\boldsymbol{x} : \|\boldsymbol{x}\| \leqslant 1\}$, where $\|\cdot\|$ is some norm on $\mathbb{R}^n$
  - As for any $\boldsymbol{x}, \boldsymbol{y} \in N$,
    $\|(1-\theta)\boldsymbol{x} + \theta\boldsymbol{y}\| \leqslant \|(1-\theta)\boldsymbol{x}\| + \|\theta\boldsymbol{y}\| = (1-\theta)\|\boldsymbol{x}\| + \theta\|\boldsymbol{y}\| \leqslant 1$
- The set of all (symmetric) positive semidefinite matrices, denoted by $\mathbb{S}_+^n \subset \mathbb{R}^{n \times n}$, is a convex cone
  - For any $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{S}_+^n$ and $\boldsymbol{x} \in \mathbb{R}^n$,
    $\boldsymbol{x}^\top((1-\theta)\boldsymbol{A} + \theta\boldsymbol{B})\boldsymbol{x} = \boldsymbol{x}^\top(1-\theta)\boldsymbol{A}\boldsymbol{x} + \boldsymbol{x}^\top\theta\boldsymbol{B}\boldsymbol{x} \geqslant 0$

# Operations That Preserve Convexity

- Given a convex set $C_1, C_2 \subseteq \mathbb{R}^n$,
  - Scaling: $\beta C = \{\beta \boldsymbol{x} : \boldsymbol{x} \in C\}$ is convex for any $\beta \in \mathbb{R}$
  - Sum: $C_1 + C_2 = \{\boldsymbol{x}_1 + \boldsymbol{x}_2 : \boldsymbol{x}_1 \in C_1, \boldsymbol{x}_2 \in C_2\}$ is convex
  - Augmentation: $\{(\boldsymbol{x}_1, \boldsymbol{x}_2) : \boldsymbol{x}_1 \in C_1, \boldsymbol{x}_2 \in C_2\} \subseteq \mathbb{R}^{2n}$ is convex
  - ***Intersection:*** $C_1 \cap C_2$ is convex [Homework]
- Affine transformation: if a map $f : \mathbb{R}^n \to \mathbb{R}^m$ is affine, and $C$ is convex, then the set

$$f(C) := \{f(\boldsymbol{x}) : \boldsymbol{x} \in C\}$$

  is convex [Proof]
  - In particular, the projection of a convex set on a subspace is convex

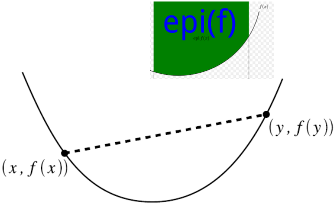
# Outline


1. Optimization Problems
   - Standard Forms and Terminology
   - Problem Classes
2. Convexity
   - Convex Sets
   - Convex Functions
3. Convex Optimization
   - Optimality
   - Disciplined Convex Programming and CVX
   - LP and QP
4. Algorithms
   - Unconstrained Problems
   - Constrained Problems
   - Large-Scale Problems**
5. Duality
   - Weak Duality
   - Strong Duality

# Convex Functions

## Definition (Convex Function)

A function $f : \mathcal{D} \subseteq \mathbb{R}^n \to \mathbb{R}$ is ***convex*** iff a) $\mathcal{D}$ is convex; and b) for any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}$ and $\theta \in [0, 1]$, we have $f((1-\theta)\boldsymbol{x} + \theta\boldsymbol{y}) \leqslant (1-\theta)f(\boldsymbol{x}) + \theta f \boldsymbol{y})$



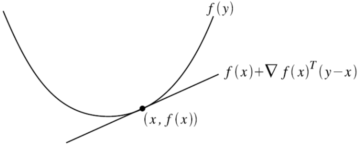

- Condition a) is necessary (what if $\mathcal{D}$ is union of two line segments?)
- Alternatively, $f$ is ***convex*** iff its epigraph $epi(f) := \{(\boldsymbol{x}, t) \in \mathbb{R}^{n+1} : \boldsymbol{x} \in \mathbb{R}^n, t \geqslant f(\boldsymbol{x})\}$ is convex

function f 是 convex function
iff epi (f) 是 convex set

- We say that a function $f$ is
  - ***strictly convex*** if $f((1-\theta)\boldsymbol{x} + \theta\boldsymbol{y}) < (1-\theta)f(\boldsymbol{x}) + \theta f \boldsymbol{y})$ for $\boldsymbol{x} \neq \boldsymbol{y}$
- ***concave*** if $-f$ is convex

# More Alternate Definitions

- First-order condition: if $f \in \mathcal{C}^1$ is differentiable (that is, $\mathcal{D}$ is open and the gradient exists everywhere on $\mathcal{D}$), then $f$ is convex iff for any $\boldsymbol{x}$ and $\boldsymbol{y}$, $f(\boldsymbol{y}) \geqslant f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$
  - I.e., the graph of $f$ is bounded below everywhere by anyone of its tangent planes




- Restriction to a line: $f$ is convex iff its restriction to ***any*** line is convex, i.e., for every $\boldsymbol{x}_0, \boldsymbol{v} \in \mathbb{R}^n$, the function $g(t) := f(\boldsymbol{x}_0 + t\boldsymbol{v})$ is convex when $\boldsymbol{x}_0 + t\boldsymbol{v} \in \mathcal{D}$
- Second-order condition: If $f$ is twice differentiable, then it is convex iff its Hessian $\nabla^2 f$ is positive semidefinite everywhere on $\mathcal{D}$; i.e., for any $\boldsymbol{x} \in \mathcal{D}$, $\nabla^2 f(\boldsymbol{x}) \succeq \boldsymbol{O}$

# Examples

- $f(x) = e^{ax}$ for $a \in \mathbb{R}$, $f(x) = |x|$, $f(x) = -\log x$ on $\mathbb{R}_{++}$ (strict positive real numbers), negative entropy $f(x) = x \log x$ on $\mathbb{R}_{++}$
- Affine functions $f(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$
- Quadratic functions $f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}\boldsymbol{x} + c$ with positive semidefinite $\boldsymbol{A}$
- Function $\lambda_{\max}(\boldsymbol{X})$ that maps an $n \times n$ symmetric matrix $\boldsymbol{X}$ to it maximum eigenvalue $\lambda_{\max}$
    - Since the condition $\lambda_{\max}(\boldsymbol{X}) \leqslant t$ is equivalent to the condition that $t\boldsymbol{I} - \boldsymbol{X} \in \mathbb{S}_+^n$, the epigraph is convex
- Norms
    - As $\|(1-\theta)\boldsymbol{x} + \theta\boldsymbol{y}\| \leqslant \|(1-\theta)\boldsymbol{x}\| + \|\theta\boldsymbol{y}\| = (1-\theta)\|\boldsymbol{x}\| + \theta\|\boldsymbol{y}\|$
- Log-sum-exp $f(\boldsymbol{x}) = \log \sum_i e^{x_i}$ (a smooth approximation to $f(\boldsymbol{x}) = \max\{x_i\}$)

# Convexity of Sublevel Sets

- Convex functions give rise to a particularly important type of convex set, the $t$-sublevel set:

**Theorem**

*Given a convex function $f : \mathcal{D} \rightarrow \mathbb{R}$ and $t \in \mathbb{R}$. The $t$-sublevel set (i.e., $\{\boldsymbol{x} \in \mathcal{D} : f(\boldsymbol{x}) \leqslant t\}$ is Convex.*

**Proof.**

[Homework] □

- Consider a inequality constrain $g \leqslant 0$ in a convex optimization problem, if $g$ is a convex function, then it defines a convex feasible set, the 0-sublevel set
  - When there are multiple inequality constrains, the final feasible set is the intersection of multiple convex sets, which is still convex

# Operations That Preserve Convexity (1)

- Composition with an affine function: if $\boldsymbol{A}$ in $\mathbb{R}^{m\times n}$, $\boldsymbol{b}$ in $\mathbb{R}^m$ and $f:\mathbb{R}^m\to\mathbb{R}$ is convex, then the function $g:\mathbb{R}^n\to\mathbb{R}$ with values $g(\boldsymbol{x})=f(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b})$ is convex
- ***Point-wise maximum***: the pointwise maximum of a family of convex functions is convex—if $\{f_i\}_{i\in\mathcal{A}}$ is a family of convex functions, then the function $f(\boldsymbol{x}):=\max_{i\in\mathcal{A}} f_i(\boldsymbol{x})$ is convex
  - E.g., $f(\boldsymbol{x})=\max\{x_i\}$, induced matrix norm $\|\boldsymbol{A}\|=\max_{\boldsymbol{x}:\|\boldsymbol{x}\|=1}\|\boldsymbol{A}\boldsymbol{x}\|$ is convex
  - Extension: $\sup_{y\in\mathcal{A}} f(\boldsymbol{x},\boldsymbol{y})$ is convex if for each $\boldsymbol{y}\in\mathcal{A}$, $f(\boldsymbol{x},\boldsymbol{y})$ is convex in $\boldsymbol{x}$
- Nonnegative weighted sum of convex functions is convex
  - E.g., entropy $f(\boldsymbol{x})=-\sum_{i=1}^n x_i\log x_i$ for a distribution $\boldsymbol{x}\in[0,1]^n$ and $\mathbf{1}^\top\boldsymbol{x}=1$ is concave
- Partial minimum: If $f$ is a convex function in $(\boldsymbol{y},\boldsymbol{z})$, then the function $g(\boldsymbol{y}):=\min_{\boldsymbol{z}} f(\boldsymbol{y},\boldsymbol{z})$ is convex
  - Note that joint convexity in $(\boldsymbol{y},\boldsymbol{z})$ is essential

# Operations That Preserve Convexity (2)

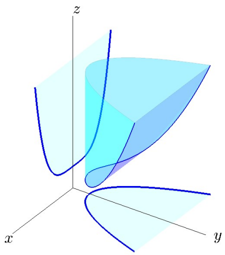
- ***Composition with monotone convex functions***: if $f(\boldsymbol{x}) = h(g_1(\boldsymbol{x}), \cdots, g_k(\boldsymbol{x}))$, with $g_i : \mathbb{R}^n \to \mathbb{R}$ convex, $h : \mathbb{R}^k \to \mathbb{R}$ convex and non-decreasing in each variable, then $f$ is convex
  - For simplicity, assume $k = 1$ and $h, g \in \mathcal{C}^2$. The above conditions ensure that $\nabla^2 g_1(\boldsymbol{x}) \in \mathbb{R}^{n \times n} \succeq \boldsymbol{O}$, $h''(\boldsymbol{y}) \in \mathbb{R}^n \geqslant 0$, and $h'(\boldsymbol{y}) \in \mathbb{R}^n \geqslant 0$
  - Then for any $\boldsymbol{x} \in \mathcal{D}$, (remember the chain and product rules?)

$$\begin{aligned}
\nabla^2 f(\boldsymbol{x}) &= (\nabla f)'(\boldsymbol{x})^\top = \left\{ [\nabla g_1(\boldsymbol{x}) h'(g_1(\boldsymbol{x}))]' \right\}^\top \\
&= \{\nabla g_1(\boldsymbol{x}) h''(g_1(\boldsymbol{x})) g_1'(\boldsymbol{x}) + (\nabla g_1)'(\boldsymbol{x}) h'(g_1(\boldsymbol{x}))\}^\top \\
&= h''(g_1(\boldsymbol{x})) \left\{ \nabla g_1(\boldsymbol{x}) \nabla g_1(\boldsymbol{x})^\top \right\} + h'(g_1(\boldsymbol{x})) \left\{ \nabla^2 g_1(\boldsymbol{x}) \right\} \\
&\succeq \boldsymbol{O}
\end{aligned}$$

- E.g., $\log \sum_i \exp(g_i)$ is convex if $g_i$ is

# Operations That Preserve Convexity (3)

- Let $g(x) = x^2$, $h(y) = y^2$ for $y \geqslant 0$, and $f(x) = h \circ g(x) = x^4$
- To show that $epi(f)$ is convex, observe first that $f(x) \leqslant z$ in is equivalent to the existence of $y$ such that $h(y) \leqslant z$ and $g(x) \leqslant y$
- The above conditions ensure that the set $\{(x, y, z) : h(y) \leqslant z, g(x) \leqslant y\}$ in the space of $(x, y, z)$-variables is convex
- Hence, $epi(f)$, the projection of that convex set onto the space of $(x, z)$-variables, is convex



# Outline


1. Optimization Problems
   - Standard Forms and Terminology
   - Problem Classes
2. Convexity
   - Convex Sets
   - Convex Functions
3. Convex Optimization
   - Optimality
   - Disciplined Convex Programming and CVX
   - LP and QP
4. Algorithms
   - Unconstrained Problems
   - Constrained Problems
   - Large-Scale Problems**
5. Duality
   - Weak Duality
   - Strong Duality

# Problem Revisited

- Form:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$
$$\text{subject to } g_i(\boldsymbol{x}) \leqslant 0, h_j(\boldsymbol{x}) = 0, i = 1, \cdots, m, j = 1, \cdots, p$$

  where $f$ is a ***convex function***, $g_i$ are ***convex*** functions, and $h_j$ are ***affine*** functions
- $epi(f)$ is a convex set
- $C = \{\boldsymbol{x} : g_i(\boldsymbol{x}) \leqslant 0, h_j(\boldsymbol{x}) = 0, i = 1, \cdots, m, j = 1, \cdots, p\}$ is a convex set
  - $g_i$'s are convex implies that the 0-sublevel sets $\{\boldsymbol{x} : g_i(\boldsymbol{x}) \leqslant 0\}$ are convex sets
  - $C$ is the intersection of convex sublevel sets and hyperplanes
- The problem amounts to finding the "lowest" point in the set $epi(f) \cap \{(\boldsymbol{x}, t) : \boldsymbol{x} \in C, t \in \mathbb{R}\}$, which is convex
  - Local optimal points are also global optima

# Global vs. Local Optima in Convex Optimization

## Theorem

*For convex problems with objective $f : \mathcal{D} \to \mathbb{R}$, any locally optimal point is globally optimal. In addition, the optimal set is convex.*

## Proof.

Let $\boldsymbol{y}$ and $\boldsymbol{x}^*$ be a point and a local minimizer of $f$ on the intersection of feasible set $C$ and $\mathcal{D}$. We need to prove that $f(\boldsymbol{y}) \geqslant f(\boldsymbol{x}^*) = p^*$. By convexity of $f$ and $C$, we have $\boldsymbol{x}_\theta := \theta \boldsymbol{y} + (1-\theta)\boldsymbol{x}^*$, and:

$$f(\boldsymbol{x}_\theta) - f(\boldsymbol{x}^*) \leqslant \theta f(\boldsymbol{y}) + (1-\theta) f(\boldsymbol{x}^*) - f(\boldsymbol{x}^*) = \theta(f(\boldsymbol{y}) - f(\boldsymbol{x}^*)).$$

Since $\boldsymbol{x}^*$ is a local minimizer, the left-hand side in this inequality is nonnegative for all small enough values of $\theta > 0$. We conclude that the right hand side is nonnegative, i.e., $f(\boldsymbol{y}) \geqslant f(\boldsymbol{x}^*) = p^*$ as claimed.
Also, the optimal set is convex, since it can be written as
$X^* = \{\boldsymbol{x} \in C \cap \mathcal{D} : f(\boldsymbol{x}^*) \leqslant p^*\}$. This ends our proof. $\square$

# Outline


1. Optimization Problems
   - Standard Forms and Terminology
   - Problem Classes
2. Convexity
   - Convex Sets
   - Convex Functions
3. Convex Optimization
   - Optimality
   - Disciplined Convex Programming and CVX
   - LP and QP
4. Algorithms
   - Unconstrained Problems
   - Constrained Problems
   - Large-Scale Problems**
5. Duality
   - Weak Duality
   - Strong Duality

# Disciplined Convex Programming and CVX

- A convex optimization software can solve a convex optimization problem efficiently
  - E.g., CVX, optimization toolbox in Matlab (for LP and QP)
- But it cannot identify whether a problem, in an arbitrary form, is convex or not
  - Don't expect it to accept any problem you give, and tell you the problem is not convex
- Discipline convex optimization defines
  - A library of convex functions
  - The rule sets corresponding to operations that preserve convexity. E.g., sum, affine composition, pointwise maximum, partial minimization, composition with monotone convex functions, etc.

# Outline


1. Optimization Problems
   - Standard Forms and Terminology
   - Problem Classes
2. Convexity
   - Convex Sets
   - Convex Functions
3. Convex Optimization
   - Optimality
   - Disciplined Convex Programming and CVX
   - LP and QP
4. Algorithms
   - Unconstrained Problems
   - Constrained Problems
   - Large-Scale Problems**
5. Duality
   - Weak Duality
   - Strong Duality

# Outline


1. Optimization Problems
   - Standard Forms and Terminology
   - Problem Classes
2. Convexity
   - Convex Sets
   - Convex Functions
3. Convex Optimization
   - Optimality
   - Disciplined Convex Programming and CVX
   - LP and QP
4. Algorithms
   - Unconstrained Problems
   - Constrained Problems
   - Large-Scale Problems**
5. Duality
   - Weak Duality
   - Strong Duality

# Unconstrained Problems

- Form:

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$

  where $f$ is convex
- For simplicity, here we assume $f \in \mathcal{C}^1$
- Optimality condition: $\boldsymbol{x}^*$ is optimal iff $\nabla f(\boldsymbol{x}^*) = 0$
- For general $f$ (other than affine or quadratic), we may not be able to solve $\boldsymbol{x}^*$ in a close form
- In practice, suboptimal solutions may be acceptable
- There exist iterative algorithms that yield suboptimal points much faster

# Iterative Algorithms

- Assumption: the problem is attained (i.e., $C \neq \emptyset$ and $p^*$ is attained)

**Algorithm 4.1:** General Descent Method

**Input**: $\boldsymbol{x}^{(0)}$, an initial guess from $\mathcal{D}$

1 **repeat**

2 Determine a ***search direction*** $\boldsymbol{d}^{(t)} \in \mathbb{R}^n$ ;

3 ***Line search***: Choose a ***step size*** $\eta^{(t)}$ such that $f(\boldsymbol{x}^{(t)} + \eta^{(t)}\boldsymbol{d}^{(t)}) < f(\boldsymbol{x}^{(t)})$;

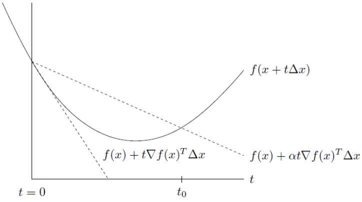
4 ***Update rule***: $\boldsymbol{x}^{(t+1)} \leftarrow \boldsymbol{x}^{(t)} + \eta^{(t)}\boldsymbol{d}^{(t)}$ ;

5 **until** *convergence criterion is satisfied*;

- Convergence criterion: $\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\| \leqslant \epsilon$, $\|\nabla f(\boldsymbol{x}^{(t+1)})\| \leqslant \epsilon$, etc.
- Line search could be exact: $\eta^{(t)} \leftarrow \arg\min_{\eta>0} \phi(\eta) := f(\boldsymbol{x}^{(t)} + \eta\boldsymbol{d}^{(t)})$, which minimizes $f$ along the ray $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \eta\boldsymbol{d}^{(t)}$, $\forall \eta \in \mathbb{R} > 0$

# Backtracking Line Search

- In practice, $\eta^{(t)}$ is usually obtained by another iterations called ***backtracking linear search***




($\eta = t$ here)

**Algorithm 4.2:** Backtracking Line Search

**Input**: $\alpha \in (0, 0.5)$, $\beta \in (0,1)$

1. $\eta \leftarrow 1$;
2. **while** $\boldsymbol{x}^{(t)} + \eta \boldsymbol{d}^{(t)} \notin \mathcal{D}$ **do**
3. $\quad \eta \leftarrow \beta\eta$;
4. **end**
5. **while** $f(\boldsymbol{x}^{(t)} + \eta \boldsymbol{d}^{(t)}) = \phi(\eta) > \phi(0) + \alpha\phi'(0)\eta = f(\boldsymbol{x}^{(t)}) + \alpha \nabla f(\boldsymbol{x}^{(t)})^\top \boldsymbol{d}^{(t)}\eta$ **do**
6. $\quad \eta \leftarrow \beta\eta$;
7. **end**

- $\alpha$, typically in $[0.01, 0.3]$, indicates how much relaxation we accept to the descent direction predicted by the linear extrapolation
- $\beta$, typically in $[0.1, 0.8]$, determines how fine-grained the search is

# Newton's Method (1)

- Recall that when $f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x}$ is quadratic and $\boldsymbol{Q} \succeq \mathbf{O}$, we cab obtain $\boldsymbol{x}^*$ by solving $\boldsymbol{Q}\boldsymbol{x}^* = -\boldsymbol{c}$
  - No solution if $\boldsymbol{c} \notin \mathcal{R}(\boldsymbol{Q})$; otherwise $X^* = \{-\boldsymbol{Q}^\dagger \boldsymbol{c} + \boldsymbol{z} : \boldsymbol{z} \in \mathcal{N}(\boldsymbol{Q})\}$ (remember how to solve linear equations using SVD?)
  - When $\boldsymbol{Q} \succ \mathbf{O}$, $\boldsymbol{x}^* = -\boldsymbol{Q}^{-1}\boldsymbol{c}$ is unique
  - Complexity?

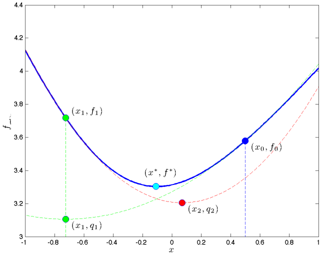
# Newton's Method (1)

- Recall that when $f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x}$ is quadratic and $\boldsymbol{Q} \succeq \mathbf{O}$, we cab obtain $\boldsymbol{x}^*$ by solving $\boldsymbol{Q}\boldsymbol{x}^* = -\boldsymbol{c}$
    - No solution if $\boldsymbol{c} \notin \mathcal{R}(\boldsymbol{Q})$; otherwise $X^* = \{-\boldsymbol{Q}^\dagger \boldsymbol{c} + \boldsymbol{z} : \boldsymbol{z} \in \mathcal{N}(\boldsymbol{Q})\}$ (remember how to solve linear equations using SVD?)
    - When $\boldsymbol{Q} \succ \mathbf{O}$, $\boldsymbol{x}^* = -\boldsymbol{Q}^{-1}\boldsymbol{c}$ is unique
    - Complexity? $O(n^3)$
- We can leverage the quadratic approximation of a general $f$ to give an iterative algorithm

# Newton's Method (2)

- Assumption: $f \in \mathcal{C}^2$ and is strictly convex (i.e., $\nabla^2 f(\boldsymbol{x}) \succ \boldsymbol{O}$ everywhere)

Update rule: $\boldsymbol{x}^{(t+1)} \leftarrow \boldsymbol{x}^{(t)} - (\nabla^2 f(\boldsymbol{x}^{(t)}))^{-1} \nabla f(\boldsymbol{x}^{(t)})$;



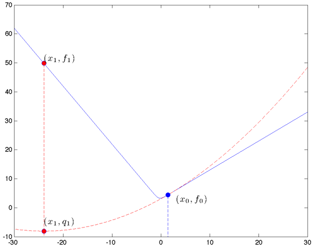

- Based on a ***local quadratic approximation*** of the the function at the current point $\boldsymbol{x}_t$:
  $\tilde{f}(\boldsymbol{x}) := f(\boldsymbol{x}^{(t)}) + \nabla f(\boldsymbol{x}^{(t)})(\boldsymbol{x} - \boldsymbol{x}^{(t)}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^{(t)})^\top \nabla^2 f(\boldsymbol{x}^{(t)})(\boldsymbol{x} - \boldsymbol{x}^{(t)})$
- $\boldsymbol{x}^{(t+1)}$ is set to be a solution to the problem of minimizing $\tilde{f}$

# Remarks (1)

- Pros:
  - No need for line search (although in practice, we often set $\boldsymbol{d}^{(n)} = -(\nabla^2 f(\boldsymbol{x}^{(t)}))^{-1}\nabla f(\boldsymbol{x}^{(t)})$ and perform linear search)
  - Converges fast (1 iteration for quadratic $f$)
- Cons:
  - Computing $(\nabla^2 f(\boldsymbol{x}_t))^{-1}$ may be too costly for large-scale problems
  - $\nabla^2 f(\boldsymbol{x}_t)$ may be singular or ill-conditioned (try $\boldsymbol{d}^{(n)} = -[\nabla^2 f(\boldsymbol{x}^{(t)}) + \mu \boldsymbol{I}]^{-1}\nabla f(\boldsymbol{x}^{(t)})$ instead)

# Remarks (2)

- Might fail to converge for some convex functions
  - Works best for self-concordant functions, whose the Hessians do not vary too fast




- Failure of the Newton method. $\boldsymbol{x}_0$ is chosen in a region where the function is almost linear. As a result, the quadratic approximation is almost a straight line, and the Hessian is close to zero, sending $\boldsymbol{x}_1$ to a relatively large negative value. The method quickly diverges in this case

# Gradient Descent (1)

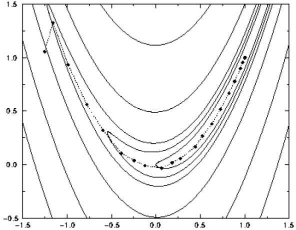
- Assumption: $f \in \mathcal{C}^1$
- Recall that at a given point $\boldsymbol{x}$, $\nabla f(\boldsymbol{x})$ points to the steepest ascend direction

$$\boxed{\text{Search direction: } \boldsymbol{d}^{(t)} = -\nabla f(\boldsymbol{x}^{(t)});}$$

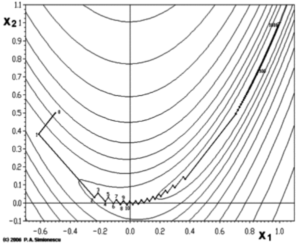
- Since $\nabla f(\boldsymbol{x}^{(t)} + \eta^{(t)} \boldsymbol{d}^{(t)})^\top \boldsymbol{d}^{(t)} = 0$, the next gradient $\nabla f(\boldsymbol{x}^{(t+1)})$ is orthogonal to the current descent direction $\boldsymbol{d}^{(t)} = -\nabla f(\boldsymbol{x}^{(t)})$

# Remarks

- Pros:
  - Easy to implement
  - Requires only the first order information on $f$ (computing each iteration is cheap)
- Cons:
  - Much more iterations (as compared to the Newton's method) to convergence
  - "Zig-zagging" around a narrow valley with flat bottom
    - E.g., Rosenbrock's banana: $f(\boldsymbol{x}) = 100(x_2 - x_1^2) + (1 - x_1^2)$




(Newton vs. Gradient)

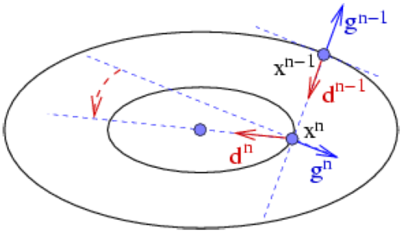


# Conjugate Gradient Descent (1)

- A simple variation of the gradient descent
  - Line search and update rule are the same
  - But tilt the next search direction to better aim at the minimum of the Hessian of $f$

Search direction: $\boldsymbol{d}^{(t)} = -\nabla f(\boldsymbol{x}^{(t)}) + c^{(t)} \boldsymbol{d}^{(t-1)}$ for some constant $c^{(t)}$;

- $c^{(t)}$ can be $\frac{\|\nabla f(\boldsymbol{x}^{(t)})\|^2}{\|\nabla f(\boldsymbol{x}^{(t-1)})\|^2}$, $\frac{(\nabla f(\boldsymbol{x}^{(t)}) - \nabla f(\boldsymbol{x}^{(t-1)}))^\top \nabla f(\boldsymbol{x}^{(t)})}{\|\nabla f(\boldsymbol{x}^{(t-1)})\|^2}$, etc.
- Designed to perform well on quadratic functions




$(\boldsymbol{g}^{(t)} := \nabla f(\boldsymbol{x}^{(t)}))$

# Conjugate Gradient Descent (2)

- Suppose $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x}$ is quadratic (so that $\nabla f(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$)
- Idea: instead of searching for $\boldsymbol{x}^{(t+1)}$ minimizing $f$ along $\boldsymbol{x}^{(t)} - \eta \nabla f(\boldsymbol{x}^{(t)})$, seek for $\boldsymbol{x}^{(t+1)}$ minimizing $f$ ***in the affine space*** $\mathcal{W}^{(t+1)} := \boldsymbol{x}^{(0)} + span(\boldsymbol{d}^{(0)}, \boldsymbol{d}^{(1)}, \cdots, \boldsymbol{d}^{(t-1)}, \nabla f(\boldsymbol{x}^{(t)}))$

**Lemma**

*If $\boldsymbol{x}^{(t+1)}$ is the minimizer of $f$ in $\mathcal{W}^{(t+1)}$, then $\nabla f(\boldsymbol{x}^{(t+1)}) \perp \mathcal{W}^{(t+1)}$.*

**Proof.**

Otherwise, we can decrease $f$ along the projection of $\nabla f(\boldsymbol{x}^{(t+1)})$ onto $\mathcal{W}^{(t+1)}$, contradicting to that $\boldsymbol{x}^{(t+1)}$ is the minimizer. $\square$

# Conjugate Gradient Descent (3)

**Lemma**

*Let $\boldsymbol{x}^{(t)}$ be the minimizer of $f$ in $\mathcal{W}^{(t)}$. From $\boldsymbol{x}^{(t)}$, the direction $\boldsymbol{d}^{(t)}$ points to the minimizer $\boldsymbol{x}^{(t+1)}$ in $\mathcal{W}^{(t+1)}$ iff $\boldsymbol{d}^{(t)\top}\boldsymbol{A}\boldsymbol{d}^{(i)} = 0$ for $0 \leqslant i \leqslant t-1$. The direction $\boldsymbol{d}^{(t)}$ is said to be **conjugate** to all previous $\boldsymbol{d}^{(i)}$.*

**Proof.**

By definition, we have $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)} + \eta\boldsymbol{d}^{(t)}$ and

$$\nabla f(\boldsymbol{x}^{(t+1)}) = \boldsymbol{A}\boldsymbol{x}^{(t+1)} + \boldsymbol{b} = \nabla f(\boldsymbol{x}^{(t)}) + \eta\boldsymbol{A}\boldsymbol{d}^{(t)}.$$

From the above lemma $\nabla f(\boldsymbol{x}^{(t+1)}) \perp \mathcal{W}^{(t+1)}$ and $\nabla f(\boldsymbol{x}^{(t)}) \perp \mathcal{W}^{(t)}$, we have

$$0 = \nabla f(\boldsymbol{x}^{(t+1)})^\top \nabla f(\boldsymbol{x}^{(t)}) = \|\nabla f(\boldsymbol{x}^{(t)})\|^2 + \eta\boldsymbol{d}^{(t)\top}\boldsymbol{A}\nabla f(\boldsymbol{x}^{(t)}),$$

implying $\eta \neq 0$. Furthermore,

$$0 = \nabla f(\boldsymbol{x}^{(t+1)})^\top \boldsymbol{d}^{(i)} = \nabla f(\boldsymbol{x}^{(t)})\boldsymbol{d}^{(i)} + \eta\boldsymbol{d}^{(t)\top}\boldsymbol{A}\boldsymbol{d}^{(i)} = \eta\boldsymbol{d}^{(t)\top}\boldsymbol{A}\boldsymbol{d}^{(i)},$$

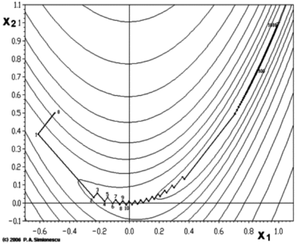
implying $\boldsymbol{d}^{(t)\top}\boldsymbol{A}\boldsymbol{d}^{(i)} = 0$ for all $i$. □

# Conjugate Gradient Descent (4)

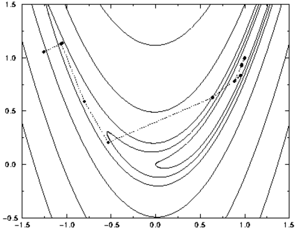
- How to find $\boldsymbol{d}^{(t)}$ such that it is conjugate to all $\boldsymbol{d}^{(i)}$?
- Notice that $\nabla f(\boldsymbol{x}^{(t+1)}) - \nabla f(\boldsymbol{x}^{(t)}) = \boldsymbol{A}(\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}) = \eta \boldsymbol{A}\boldsymbol{d}^{(t)}$ (see the proof of the above lemma).
- So, $\boldsymbol{d}^{(t)\top}\boldsymbol{A}\boldsymbol{d}^{(i)} = 0 \Rightarrow \boldsymbol{d}^{(t)\top}(\nabla f(\boldsymbol{x}^{(t+1)}) - \nabla f(\boldsymbol{x}^{(t)})) = 0 \Rightarrow$ $\boldsymbol{d}^{(t)\top}\nabla f(\boldsymbol{x}^{(t+1)}) = \boldsymbol{d}^{(t)\top}\nabla f(\boldsymbol{x}^{(t)}) = \text{some constant}$
- Since $\nabla f(\boldsymbol{x}^{(i)})$ forms an orthogonal family, we have $\boldsymbol{d}^{(t)}$ a scaling of $\sum_{i=0}^{t} \frac{\nabla f(\boldsymbol{x}^{(i)})}{\|\nabla f(\boldsymbol{x}^{(i)})\|^2}$
- Apply the above to $\boldsymbol{d}^{(0)}, \boldsymbol{d}^{(1)}, \cdots, \boldsymbol{d}^{(t)}$, we have $\boldsymbol{d}^{(t)} = -\nabla f(\boldsymbol{x}^{(t)}) + c^{(t)}\boldsymbol{d}^{(t-1)}$
    - You can easily verify that $c^{(t)} = \frac{\|\nabla f(\boldsymbol{x}^{(t)})\|^2}{\|\nabla f(\boldsymbol{x}^{(t-1)})\|^2}$ makes the equation holds

# Remarks

- Pros:
  - Easy to implement
  - Still a first order method (same cheap iterations as in gradient descent)
  - Converges fast (at most $n$ iterations for quadratic function $f : \mathbb{R}^n \to \mathbb{R}$)
  - Can be applied to non-quadratic $f$, by replacing $\boldsymbol{A}$ with the Hessian of $f$
    - Works well if $\nabla^2 f(\boldsymbol{x}^{(t+1)})$ and $\nabla^2 f(\boldsymbol{x}^{(t)})$ do not vary too much
- Caution:
  - For general $f$, $\boldsymbol{d}^n$ may not be a descent direction. Set it to $-\nabla f(\boldsymbol{x}^{(t)})$ in this case




(Gradient vs. Conjugate Gradient)



# Outline

1. Optimization Problems
   - Standard Forms and Terminology
   - Problem Classes
2. Convexity
   - Convex Sets
   - Convex Functions
3. Convex Optimization
   - Optimality
   - Disciplined Convex Programming and CVX
   - LP and QP
4. **Algorithms**
   - Unconstrained Problems
   - **Constrained Problems**
   - Large-Scale Problems**
5. Duality
   - Weak Duality
   - Strong Duality

# Constrained Problems

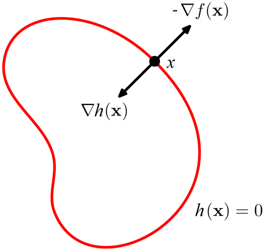
- Form:

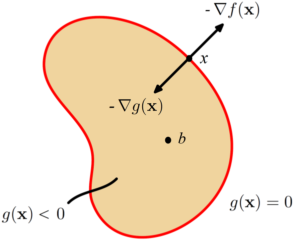
$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$
$$\text{subject to } \boldsymbol{x} \in C = \{\boldsymbol{x} : g_i(\boldsymbol{x}) \leqslant 0, h_j(\boldsymbol{x}) = 0, i = 1, \cdots, m, j = 1, \cdots, p\}$$

  where $f$ and $g_i$ are convex, $h_j$ are affine
- For simplicity, here we assume $f \in \mathcal{C}^1$
- Optimality condition: $\boldsymbol{x}^*$ is optimal iff $\nabla f(\boldsymbol{x}^*)^\top (\boldsymbol{x} - \boldsymbol{x}^*) \geqslant 0, \forall \boldsymbol{x} \in C$, as $f(\boldsymbol{x}) \geqslant f(\boldsymbol{x}^*) + \nabla f(\boldsymbol{x}^*)^\top (\boldsymbol{x} - \boldsymbol{x}^*)$

# Active Sets

- Define the ***active set*** $\mathcal{A}(\boldsymbol{x})$ at a point $\boldsymbol{x}$ as the set of constrains $\theta$'s such that $\theta(\boldsymbol{x}) = 0$, i.e., $\mathcal{A}(\boldsymbol{x}) := \{\theta : \theta(\boldsymbol{x}) = 0\}$
  - Equality constrains $h_j$'s are always active
- Recall for any constrain $\theta$, the gradient $\nabla\theta(\boldsymbol{x})$ is orthogonal to a tangent line/space passing through the level set at $\boldsymbol{x}$
- $\boldsymbol{x}^*$ occurs when
  - $\forall j$, $\nabla h_j(\boldsymbol{x}^*)$ and $-\nabla f(\boldsymbol{x}^*)$ are ***parallel*** (i.e., $-\nabla f(\boldsymbol{x}^*) = \nu_j \nabla h_j(\boldsymbol{x}^*)$ for some $\nu_j \neq 0$)
  - $\forall i$ such that $g_i$ is active, $-\nabla g_i(\boldsymbol{x}^*)$ and $-\nabla f(\boldsymbol{x}^*)$ are ***opposite*** (i.e., $-\nabla f(\boldsymbol{x}^*) = \lambda_i \nabla g_i(\boldsymbol{x}^*)$ for some $\lambda_i > 0$)



# Iterative Algorithms

- Assumption: the problem is attained (i.e., $C \neq \emptyset$ and $p^*$ is attained)
- Iterative algorithms in the presence of constrains?

# Iterative Algorithms

- Assumption: the problem is attained (i.e., $C \neq \emptyset$ and $p^*$ is attained)
- Iterative algorithms in the presence of constrains?

1. Transform the constrained problem into a unconstrained one, or
2. Make sure that $\boldsymbol{x}^{(t+1)}$ falls inside the feasible set during each iteration

# Exterior-Point Methods

- For equality constrains $h_j(\boldsymbol{x}) = 0$
- Idea: penalize non-admissible solutions
- Create "barrier functions" $\psi_j(\boldsymbol{x})$ such that $\psi_j(\boldsymbol{x}) = 0$ if $h_j(\boldsymbol{x}) = 0$; $\psi_j(\boldsymbol{x}) \gg 0$ otherwise
  - E.g., $\psi_j(\boldsymbol{x}) = \mu \| h_j(\boldsymbol{x}) \|^2$ for some large $\mu$
- Solve the unconstrained problem: $\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \mu \sum_{j=1}^{p} \psi_j(\boldsymbol{x})$
  - Objective is still convex
- A solution falls outside the feasible set, an "exterior point"

# Interior-Point Methods

- For inequality constrains $g_i(\boldsymbol{x}) \leqslant 0$
- Assumption: the original problem is ***strictly*** feasible (i.e., there exists $\boldsymbol{x} \in X^*$ such that $g_i(\boldsymbol{x}) < 0$ for all $i$)
- Idea: penalize non-admissible solutions
- Create barrier functions $\psi_i(\boldsymbol{x})$ such that $\psi_i(\boldsymbol{x}) = 0$ if $g_i(\boldsymbol{x}) \leqslant 0$; $\psi_i(\boldsymbol{x}) \gg 0$ otherwise
  - E.g., the ***logarithmic barrier*** $\psi_i(\boldsymbol{x}) = -\mu \log(-g_i(\boldsymbol{x}))$ for some $\mu$
- Solve the unconstrained problem (still convex): $\min_{\boldsymbol{x}} f(\boldsymbol{x}) - \mu \sum_{i=1}^{m} \log(-g_i(\boldsymbol{x}))$
- A solution falls inside the feasible set, an "interior point"

# Remarks

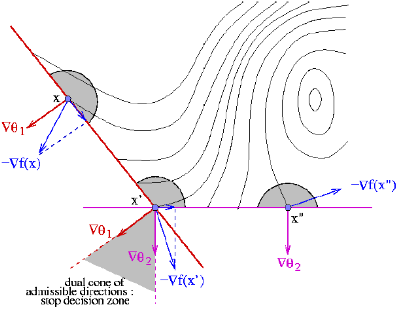
- For $\mu$ large, solving the above problem results in a point well aligned/inside the feasible set
- As $\mu \to 0$ the solution converges to a global minimizer for the original, constrained problem
  - In fact, the theory of convex optimization says that if we set $\mu = m/\epsilon$ (or $\mu = p/\epsilon$ for equality constrains), then the minimizer is $\epsilon$-suboptimal.
- In practice, we solve the unconstrained problem several times, with $\mu$ from large to small

# Projected Gradient Descent (1)

- $\mathbf{x}^{(t+1)}$ may fall outside $C$ during an iteration
- Idea: if so, project $\mathbf{x}^{(t+1)}$ onto the boundary of $C$

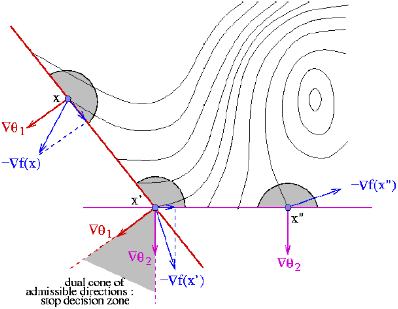
Update rule: $\mathbf{x}^{(t+1)} \leftarrow \boldsymbol{P}(\mathbf{x}^{(t)} - \eta^{(t)} \nabla f(\mathbf{x}^{(t)}))$ for some projector $\boldsymbol{P}$;

- For simplicity, we consider only the affine constrains here
- Suppose $\mathbf{x}^{(t)}$ is already on the boundary of $C$
- We can identify the active set $\mathcal{A}(\mathbf{x}^{(t)})$ at $\mathbf{x}^{(t)}$
- Define the ***tangent space of active constrains*** at $\mathbf{x}^{(t)}$: $\bigcap_{\theta \in \mathcal{A}(\mathbf{x}^{(t)})} \{\mathbf{x} : \nabla\theta(\mathbf{x}^{(t)})^\top (\mathbf{x} - \mathbf{x}^{(t)}) = 0\}$
- We seek for the projection of $\mathbf{x}^{(t+1)}$ onto that tangent space



# Projected Gradient Descent (2)

- Since $\boldsymbol{x}^{(t)}$ is already in the tangent space, the update rule can be written as
  $\boldsymbol{x}^{(t+1)} \leftarrow (\boldsymbol{x}^{(t)} - \eta^{(t)} \boldsymbol{P} \nabla f(\boldsymbol{x}^{(t)}))$
  (recall $\boldsymbol{P}^2 = \boldsymbol{P}$)
- $\nabla\theta(\boldsymbol{x}^{(t)})^\top (\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}) = 0$ implies
  $\nabla\theta(\boldsymbol{x}^{(t)})^\top (-\eta^{(t)} \boldsymbol{P} \nabla f(\boldsymbol{x}^{(t)})) = 0$
- Let
  $\boldsymbol{\Theta} = \left[\nabla\theta_1(\boldsymbol{x}^{(t)}), \cdots, \nabla\theta_a(\boldsymbol{x}^{(t)})\right] \in \mathbb{R}^{n \times a}$, where $a = |\mathcal{A}(\boldsymbol{x}^{(t)})|$
- We instead seek for the projection of $-\nabla f(\boldsymbol{x}^{(t)})$ onto $\{\boldsymbol{x} : \boldsymbol{\Theta}^\top \boldsymbol{x} = \boldsymbol{0}\}$



# Projected Gradient Descent (3)

- Target: $-\boldsymbol{P}\nabla f(\boldsymbol{x}^{(t)}) \in \{\boldsymbol{x} : \boldsymbol{\Theta}^\top \boldsymbol{x} = \boldsymbol{0}\}$. How to find $\boldsymbol{P}$?

# Projected Gradient Descent (3)

- Target: $-\boldsymbol{P}\nabla f(\boldsymbol{x}^{(t)}) \in \{\boldsymbol{x} : \boldsymbol{\Theta}^\top \boldsymbol{x} = \boldsymbol{0}\}$. How to find $\boldsymbol{P}$?
- Recall from the fundamental theorem of linear algebra that $\{\boldsymbol{x} : \boldsymbol{\Theta}^\top \boldsymbol{x} = \boldsymbol{0}\} = \mathcal{R}(\boldsymbol{\Theta})^\perp = span(\nabla\theta_1(\boldsymbol{x}^{(t)}), \cdots, \nabla\theta_a(\boldsymbol{x}^{(t)}))^\perp$
- Also, recall that the projection of any point $\boldsymbol{y}$ onto $\mathcal{R}(\boldsymbol{\Theta})$ is $\boldsymbol{\Theta}\boldsymbol{x}^*$, where $\boldsymbol{x}^* = (\boldsymbol{\Theta}^\top\boldsymbol{\Theta})^{-1}\boldsymbol{\Theta}^\top\boldsymbol{y}$ is the solution to the least square problem

$$\arg\min_{\boldsymbol{x}} \|\boldsymbol{\Theta}\boldsymbol{x} - \boldsymbol{y}\|^2$$

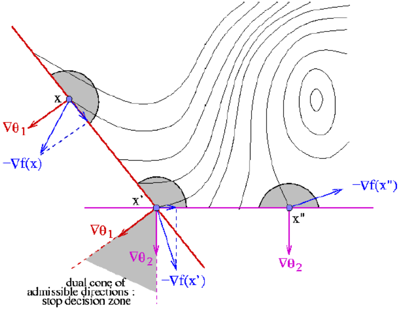
- Let $\boldsymbol{Q} = \boldsymbol{\Theta}(\boldsymbol{\Theta}^\top\boldsymbol{\Theta})^{-1}\boldsymbol{\Theta}^\top$, the projection of $\boldsymbol{y}$ onto $\mathcal{R}(\boldsymbol{\Theta})^\perp$ is $\boldsymbol{y} - \boldsymbol{Q}\boldsymbol{y} = (\boldsymbol{I} - \boldsymbol{Q})\boldsymbol{y}$, so $\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{Q}$

# The Changing Active Sets

- We may encounter $-\boldsymbol{P}\nabla f(\boldsymbol{x}^{(t)}) = \boldsymbol{0}$ during an iteration. Should we stop?

# The Changing Active Sets

- We may encounter $-\boldsymbol{P}\nabla f(\boldsymbol{x}^{(t)}) = \boldsymbol{0}$ during an iteration. Should we stop?
- No, some constrains $\theta$ in $\mathcal{A}(\boldsymbol{x}^{(t)})$ may be "unnecessary," i.e., we cannot find $\eta > 0$ such that $\boldsymbol{x}^{(t)} - \eta \boldsymbol{P}_\theta \nabla f(\boldsymbol{x}^{(t)})$ is on the boundary of $C$,
    - $\boldsymbol{P}_\theta$ projects $\boldsymbol{d}^{(t)}$ onto $\{\boldsymbol{x} : \nabla\theta(\boldsymbol{y}^{(t)})^\top \boldsymbol{x} = 0\}$
    - We can obtain $\eta$ by first solving $g(\boldsymbol{x}^{(t)} - \eta \boldsymbol{P}_\theta \nabla f(\boldsymbol{x}^{(t)})) = 0$ for each another constrain $g \in C$, and then take the minimum of the solutions that are in $(0, \infty)$




- Remove all such constrains $\theta$'s in $\mathcal{A}(\boldsymbol{x}^{(t)})$. Stop only if $\mathcal{A}(\boldsymbol{x}^{(t)}) = \emptyset$

# Algorithm

**Algorithm 4.3:** Projected Gradient Descent Method

**Input**: $\boldsymbol{x}^{(0)}$, an initial guess from $\mathcal{D} \cap C$

**repeat**

  $\boldsymbol{d}^{(t)} \leftarrow -\nabla f(\boldsymbol{x}^{(t)})$;

  Determine $\eta^{(t)}$;

  $\boldsymbol{x}^{(t+1)} \leftarrow \boldsymbol{x}^{(t)} + \eta^{(t)} \boldsymbol{d}^{(t)}$;

  **if** $\boldsymbol{x}^{(t+1)} \notin C$ **then**

    $\boldsymbol{y}^{(t)} \leftarrow \boldsymbol{x}^{(t)} + \eta' \boldsymbol{d}^{(t)}$ is the intersect between $\{\boldsymbol{x}^{(t)} + \eta \boldsymbol{d}^{(t)} : \eta > 0\}$ and the boundary of $C$;

    $\mathcal{A}(\boldsymbol{y}^{(t)}) \leftarrow$set of active constrains at $\boldsymbol{y}^{(t)}$, excluding those $\theta$'s such that there is no intersect between $\{\boldsymbol{x}^{(t)} + \eta \boldsymbol{P}_\theta \boldsymbol{d}^{(t)} : \eta > 0\}$ and the boundary of $C$;

    **if** $\mathcal{A}(\boldsymbol{y}^{(t)}) \neq \emptyset$ **then** $\boldsymbol{x}^{(t+1)} \leftarrow \boldsymbol{y}^{(t)} + (\eta^{(t)} - \eta') \boldsymbol{P} \boldsymbol{d}^{(t)}$ **else** $\boldsymbol{x}^{(t+1)} \leftarrow \boldsymbol{y}^{(t)}$;

  **end**

**until** *convergence criterion is satisfied*;

# Outline


1. Optimization Problems
   - Standard Forms and Terminology
   - Problem Classes
2. Convexity
   - Convex Sets
   - Convex Functions
3. Convex Optimization
   - Optimality
   - Disciplined Convex Programming and CVX
   - LP and QP
4. **Algorithms**
   - Unconstrained Problems
   - Constrained Problems
   - **Large-Scale Problems****
5. Duality
   - Weak Duality
   - Strong Duality

# Decomposition Methods

TBA

# Weak and Strong Duality

- Next, we shows how the notion of ***weak duality*** allows to develop, in a systematic way, approximations of non-convex problems based on convex optimization.
- Starting with any given minimization problem, which we call the ***primal problem***, we can form a ***dual problem***, which
  - Is always convex (specifically, a concave maximization problem)
  - Provides a lower bound on the values of the primal
- When the primal is convex, the ***strong duality*** holds—the dual problem shares the same optimal value as that of the primal
  - Gives more insights to the optimality conditions

# Outline

1. Optimization Problems
   - Standard Forms and Terminology
   - Problem Classes
2. Convexity
   - Convex Sets
   - Convex Functions
3. Convex Optimization
   - Optimality
   - Disciplined Convex Programming and CVX
   - LP and QP
4. Algorithms
   - Unconstrained Problems
   - Constrained Problems
   - Large-Scale Problems**
5. **Duality**
   - **Weak Duality**
   - Strong Duality

# Primal Problem

- Consider a primal problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$$
$$\text{subject to } g_i(\boldsymbol{x}) \leqslant 0, h_j(\boldsymbol{x}) = 0, i = 1, \cdots, m, j = 1, \cdots, p$$

$g_i$ & $h_j$ are constraints

  - $f$, $g_i$, and $h_j$ can be arbitrary (need not be convex or affine)

- For simplicity, let $f(\boldsymbol{x}) = \infty$ (同理 resp., $g_i(\boldsymbol{x})$ and $h_j(\boldsymbol{x})$) if $\boldsymbol{x}$ is not in the domain of $f$ (resp., $g_i$ and $h_j$)
- $p^* := \inf_{\boldsymbol{x}: g_i(\boldsymbol{x}) \leqslant 0, h_j(\boldsymbol{x})=0} f(\boldsymbol{x})$ and $\boldsymbol{x}$ are call ***primal value*** and ***variables*** respectively

inf : infimum，最大下界

小於或等於這個子集的所有其他元素的不一定在這個子集內的最大元素

sup : supremum，最小上界

$\inf\{1, 2, 3\} = 1.$

$\inf\{x \in \mathbb{R} : 0 < x < 1\} = 0.$

$\inf\{x \in \mathbb{R} : x^3 > 2\} = 2^{1/3}.$

$\inf\{(-1)^n + 1/n : n = 1, 2, 3, \ldots\} = -1.$

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

subject to $g_i(\mathbf{x}) \leqslant 0, h_j(\mathbf{x}) = 0, i = 1, \cdots, m, j = 1, \cdots, p$

- Define a ***Lagrange function*** (or simply ***Lagrangian***) $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ with values

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) := f(\mathbf{x}) + \sum_{i=1}^{m} \alpha_i g_i(\mathbf{x}) + \sum_{j=1}^{p} \beta_j h_j(\mathbf{x})$$

- Then the primal problem can be written as

先固定x找α,β，使後面這坨最大

sup：最小上界，可以想像成在有限集合中取最大值

$$\min_{\mathbf{x}\in\mathbb{R}^n} \sup_{\boldsymbol{\alpha}\geqslant\mathbf{0}} \mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

用剛剛找到的α,β，解x

  - $p^* = \inf_{\mathbf{x}\in\mathbb{R}^n} \sup_{\boldsymbol{\alpha}\geqslant\mathbf{0}} \mathcal{L}(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$
  - This creates "barriers" that penalize $g_i(\mathbf{x}) > 0$ and $h_j(\mathbf{x}) \neq 0$
  - The constrains $\boldsymbol{\alpha} \geqslant \mathbf{0}$ are essential

如果$g^i(x)>0$，因為sup，$\alpha^i g^i(x)$會變成無限大，那麼就不會被接下來的min選到

如果$g^i(x)>0$，因為$\alpha^i \geq 0$，因此$\alpha^i g^i(x)$最大就是0。所以它會找到「合理的**x**」

# Dual Problem

- Given a primal problem $\min_{\boldsymbol{x}\in\mathbb{R}^n}\sup_{\boldsymbol{\alpha}\geqslant\boldsymbol{0}}\mathcal{L}(\boldsymbol{x},\boldsymbol{\alpha},\boldsymbol{\beta})$, define its ***dual problem*** as
$$\max_{\boldsymbol{\alpha}\geqslant\boldsymbol{0}}\inf_{\boldsymbol{x}\in\mathbb{R}^n}\mathcal{L}(\boldsymbol{x},\boldsymbol{\alpha},\boldsymbol{\beta})$$
  - $d^* := \sup_{\boldsymbol{\alpha}\geqslant\boldsymbol{0}}\inf_{\boldsymbol{x}\in\mathbb{R}^n}\mathcal{L}(\boldsymbol{x},\boldsymbol{\alpha},\boldsymbol{\beta})$ is called the ***dual value***
- It can be easily shown that
$d^* = \sup_{\boldsymbol{\alpha}\geqslant\boldsymbol{0}}\inf_{\boldsymbol{x}\in\mathbb{R}^n}\mathcal{L}(\boldsymbol{x},\boldsymbol{\alpha},\boldsymbol{\beta}) \leqslant \inf_{\boldsymbol{x}\in\mathbb{R}^n}\sup_{\boldsymbol{\alpha}\geqslant\boldsymbol{0}}\mathcal{L}(\boldsymbol{x},\boldsymbol{\alpha},\boldsymbol{\beta}) = p^*$
(called max-min inequality) [Homework] 一開始就inf了，就只能在inf裡找sup，最後就會比較小
  - $d^*$ is a lower bound of $p^*$
  - $p^* - d^*$ is called the ***duality gap*** duality gap 來自max-min 不等式
- $dual(\boldsymbol{\alpha},\boldsymbol{\beta};\boldsymbol{x}) := \inf_{\boldsymbol{x}\in\mathbb{R}^n}\mathcal{L}(\boldsymbol{x},\boldsymbol{\alpha},\boldsymbol{\beta})$ is called the ***dual function***
  - Defined as a point-wise minimum (in $\boldsymbol{x}$), therefore concave
- The dual problem $\max_{\boldsymbol{\alpha}\geqslant\boldsymbol{0}} dual(\boldsymbol{\alpha},\boldsymbol{\beta})$ is always a concave-maximization problem (convex)

固定α,β找**x**，因為1次函數 $\sum_{i=1}^{m}\alpha_i g_i(\boldsymbol{x})$ 是convex也是concave
min(concave)是concave

# Example

- Consider a primal problem:

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{x}\|^2$$
$$\text{subject to } \boldsymbol{A}\boldsymbol{x} \leqslant \boldsymbol{b}$$

- $dual(\boldsymbol{\alpha};\boldsymbol{x}) = \min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{x}\|^2 + \boldsymbol{\alpha}^\top(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}) = -\frac{1}{2}\|\boldsymbol{A}^\top\boldsymbol{\alpha}\|^2 - \boldsymbol{b}^\top\boldsymbol{\alpha}$ [Proof]
  - $\boldsymbol{x}^* = \boldsymbol{A}^\top\boldsymbol{\alpha}$

Ax-b>0的「損害」會被α放大，
導致繼續縮小||x||²帶來的效益低於此損害

- Dual problem:

$$\max_{\boldsymbol{\alpha}} -\frac{1}{2}\|\boldsymbol{A}^\top\boldsymbol{\alpha}\|^2 - \boldsymbol{b}^\top\boldsymbol{\alpha}$$
$$\text{subject to } \boldsymbol{\alpha} \geqslant \boldsymbol{0}$$

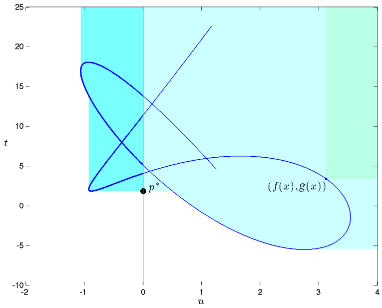
  - Equivalent to $\min_{\boldsymbol{\alpha}\geqslant\boldsymbol{0}} \frac{1}{2}\|\boldsymbol{A}^\top\boldsymbol{\alpha}\|^2 + \boldsymbol{b}^\top\boldsymbol{\alpha}$

找一個正確的α，使得Ax-b>0
的「損害」可以被正確地懲罰

# Geometric Interpretation (1)

找最低點，在深藍區域

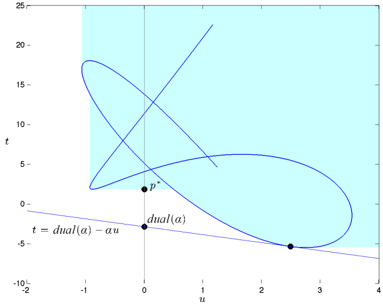
- Consider a primal problem: $\min_{\boldsymbol{x}} f(\boldsymbol{x})$ subject to $g(\boldsymbol{x}) \leqslant 0$
- Dual problem: $\max_{\alpha \geqslant 0} dual(\alpha) = \max_{\alpha \geqslant 0} \inf_{\boldsymbol{x}} f(\boldsymbol{x}) + \alpha g(\boldsymbol{x})$
- Let $A := \{(u,t) : u \geqslant g(\boldsymbol{x}), t \geqslant f(\boldsymbol{x})\}$, the blue area

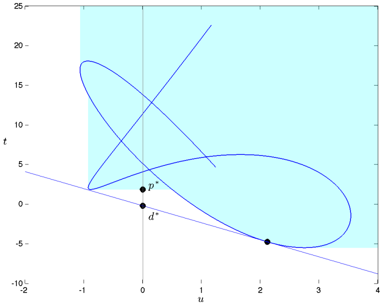



- The solutions are feasible only in the dark blue area

# Geometric Interpretation (2)

- $\inf_{\boldsymbol{x}} f(\boldsymbol{x}) + \alpha g(\boldsymbol{x})$ is attained, so we can rewrite the dual function as $dual(\alpha) = \min_{(u,t)\in A} t + \alpha u = t^* + \alpha u^*$
- Given any fixed $\alpha \geqslant 0$, $\{(u,t) : t = dual(\alpha) - \alpha u\}$ is a line with slop $-\alpha$ intercepting $A$ at $(t^*, u^*)$
  - The line intercepts $\{(u,t) : u = 0\}$ at $(0, dual(\alpha))$
- The dual problem is to find the best line intercepting $A$ that produce the highest intercept with $\{(u,t) : u = 0\}$



# Remarks

- The dual function *dual* may not be easy to compute: it is itself an optimization problem!
  - Duality works best when *dual* can be computed in closed form
- Even if it is possible to compute *dual*, it might not be easy to maximize: convex problems are not always easy to solve
- A lower bound might not be of great practical interest: often we need a sub-optimal solution
  - Duality does not seem at first to offer a way to compute such a primal point
- However, duality is a powerful tool in understanding the problem

# Outline


1. Optimization Problems
   - Standard Forms and Terminology
   - Problem Classes
2. Convexity
   - Convex Sets
   - Convex Functions
3. Convex Optimization
   - Optimality
   - Disciplined Convex Programming and CVX
   - LP and QP
4. Algorithms
   - Unconstrained Problems
   - Constrained Problems
   - Large-Scale Problems**
5. Duality
   - Weak Duality
   - Strong Duality

# Strong Duality

- Primal problem:

$$\begin{gathered}\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})\\ \text{subject to } g_i(\boldsymbol{x}) \leqslant 0, h_j(\boldsymbol{x}) = 0, i = 1,\cdots,m, j = 1,\cdots,p\end{gathered}$$

  - $p^* := \inf_{\boldsymbol{x}: g_i(\boldsymbol{x})\leqslant 0, h_j(\boldsymbol{x})=0} f(\boldsymbol{x})$

- Dual problem:

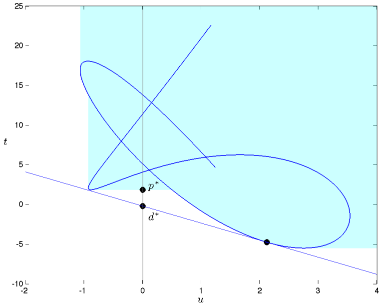
$$\max_{\boldsymbol{\alpha}\geqslant\mathbf{0}} \inf_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) + \sum_{i=1}^{m} \alpha_i g_i(\boldsymbol{x}) + \sum_{j=1}^{p} \beta_j h_j(\boldsymbol{x})$$

  - $d^* := \sup_{\boldsymbol{\alpha}\geqslant\mathbf{0}} \inf_{\boldsymbol{x}\in\mathbb{R}^n} \mathcal{L}(\boldsymbol{x},\boldsymbol{\alpha},\boldsymbol{\beta})$

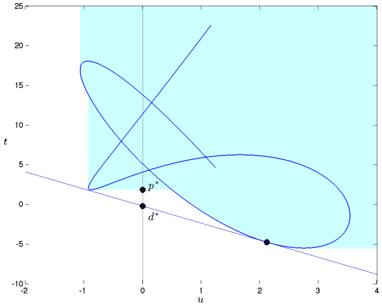
- We say that ***strong duality*** holds if the duality gap is zero: $d^* = p^*$

# Slater's Sufficient Condition for Strong Duality (1)

- How to make $(0, d^*) = (0, p^*)$?



- How to make $(0, d^*) = (0, p^*)$?
- One sufficient condition:
  1. $A = \{(u, t) : u \geqslant g(\boldsymbol{x}), t \geqslant f(\boldsymbol{x})\}$ (the blue area) is a convex set
  2. The line $\{(u, t) : t = dual(\alpha) - \alpha u\}$ is not vertical (so $d^*$ is attained)



# Slater's Sufficient Condition for Strong Duality (2)

- The above two points imply:

1. The primal problem is ***convex***
   - Since $\{u : u \geqslant g(\boldsymbol{x})\}$ and $\{t : t \geqslant f(\boldsymbol{x})\}$ are convex, so does $A$ [Proof]
2. ***Slater condition***: the primal problem is ***strictly feasible***: $\exists \boldsymbol{x} : g_i(\boldsymbol{x}) < 0, h_j(\boldsymbol{x}) = 0$
   - The interior points of $A = \{(u,t) : u \geqslant g(\boldsymbol{x}), t \geqslant f(\boldsymbol{x})\}$ (the blue area) cut into the area $\{(u,t) : u < 0\}$
   - If $g_i(\boldsymbol{x})$ is affine, we can relax the feasibility above by $g_i(\boldsymbol{x}) \leqslant 0$

- Sufficient condition for strong duality, but ***not*** necessary

# Solving the Dual Problem

- Suppose the strong duality holds, then by solving the dual problem, we obtain:
  - The primal value $p^* = d^*$
  - Furthermore, $\boldsymbol{x}^*$ if we can write $\boldsymbol{x}^*$ in a close form with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in $dual(\boldsymbol{\alpha}, \boldsymbol{\beta}; \boldsymbol{x}) := \inf_{\boldsymbol{x} \in \mathbb{R}^n} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$
- Why solving the dual problem instead?

# Solving the Dual Problem

- Suppose the strong duality holds, then by solving the dual problem, we obtain:
  - The primal value $p^* = d^*$
  - Furthermore, $\boldsymbol{x}^*$ if we can write $\boldsymbol{x}^*$ in a close form with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in $dual(\boldsymbol{\alpha}, \boldsymbol{\beta}; \boldsymbol{x}) := \inf_{\boldsymbol{x} \in \mathbb{R}^n} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$
- Why solving the dual problem instead?
  - We gain insights to the primal problem

# Karush-Kuhn-Tucker (KKT) Conditions

## Theorem

*Suppose $f$, $g_i$, and $h_j$ are continuously differentiable at $\boldsymbol{x}^*$, and the primal problem is attained, convex, and satisfies the Slate condition. Then a primal variable $\boldsymbol{x}^*$ is optimal iff there exists $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ such that the following conditions, called* ***Karush-Kuhn-Tucker (KKT) conditions*** *are satisfied:*

***Lagrangian stationarity***:
$\nabla f(\boldsymbol{x}^*) + \sum_{i=1}^{m} \alpha_i^* \nabla g_i(\boldsymbol{x}^*) + \sum_{j=1}^{p} \beta_j^* \nabla h_j(\boldsymbol{x}^*) = 0$
***Primal feasibility***: $g_i(\boldsymbol{x}^*) \leqslant 0$ *and* $h_j(\boldsymbol{x}^*) = 0$ *for all* $i = 1, \cdots, m$ *and* $j = 1, \cdots, p$
***Dual feasibility***: $\alpha_i^* \geqslant 0$ *for all* $i = 1, \cdots, m$
***Complementary slackness***: $\alpha_i^* g_i(\boldsymbol{x}^*) = 0$ *for all* $i = 1, \cdots, m$

# Complementary Slackness

- Why $\alpha_i^* g_i(\boldsymbol{x}^*) = 0$ for all $i = 1, \cdots, m$?
- When strong duality holds and both primal and dual problems are attained, by $(\boldsymbol{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$, we have

$$f(\boldsymbol{x}^*) + \sum_{i=1}^{m} \alpha_i^* g_i(\boldsymbol{x}^*) + \sum_{j=1}^{p} \beta_j^* h_j(\boldsymbol{x}^*) = dual(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*; \boldsymbol{x}^*) = d^* = p^* = f(\boldsymbol{x}^*)$$

- Since $\boldsymbol{\alpha}^* \geqslant 0$, each term in $\sum_{i=1}^{m} \alpha_i^* g_i(\boldsymbol{x}^*)$ must be 0
- So what?

# Complementary Slackness

- Why $\alpha_i^* g_i(\boldsymbol{x}^*) = 0$ for all $i = 1, \cdots, m$?
- When strong duality holds and both primal and dual problems are attained, by $(\boldsymbol{x}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$, we have

$$f(\boldsymbol{x}^*) + \sum_{i=1}^{m} \alpha_i^* g_i(\boldsymbol{x}^*) + \sum_{j=1}^{p} \beta_j^* h_j(\boldsymbol{x}^*) = dual(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*; \boldsymbol{x}^*) = d^* = p^* = f(\boldsymbol{x}^*)$$

- Since $\boldsymbol{\alpha}^* \geqslant 0$, each term in $\sum_{i=1}^{m} \alpha_i^* g_i(\boldsymbol{x}^*)$ must be 0
- So what? If $\alpha_i^* > 0$, then $g_i(\boldsymbol{x}^*) = 0$
- We can tell from the values of $\alpha_i^*$'s which inequality constraint is ***active***

# Example

- Suppose $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, in the primal problem:

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \frac{1}{2} \|\boldsymbol{x}\|^2 \\ \text{subject to } \boldsymbol{A}\boldsymbol{x} \leqslant \boldsymbol{b}$$

- Dual problem:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^m} \frac{1}{2} \|\boldsymbol{A}^\top \boldsymbol{\alpha}\|^2 + \boldsymbol{b}^\top \boldsymbol{\alpha} \\ \text{subject to } \boldsymbol{\alpha} \geqslant \boldsymbol{0}$$

  - $\boldsymbol{x}^* = \boldsymbol{A}^\top \boldsymbol{\alpha}$
- We now solve $m$ instead of $n$ variables
  - If $n \gg m$, solving the dual problem takes less time
- Furthermore, by complementary slackness, $\boldsymbol{\alpha}^\top (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) = \boldsymbol{0}$
  - We can tell that the $j$-th constraint is active (i.e., $\boldsymbol{A}_{j,:}\boldsymbol{x} = b_j$) iff $\alpha_j \neq 0$