ECE 236B
Convex Optimization
Note

Patricia Xiao (Zhiping)

Winter 2019

Contents

1 Introduction

What makes learning Convex Optimization useful?
This course also includes code implementation, a package called cvx in matlab, equivalent package called cvxpy in Python (`https://www.cvxpy.org/`). Some useful examples on `https://www.cvxpy.org/examples/index.html`. Such as the basic functions at `https://www.cvxpy.org/tutorial/functions/index.html`.
Stanford online course (`https://lagunita.stanford.edu/courses/Engineering/CVX101/Winter2014/course/`) is also pretty helpful.

1.1 Mathematical Optimization

Minimize a function, subject to some constraints.
Not necessarily convex.
Variables are not actions, they are parameters of the model.
General optimization problems could be hard to solve (and not always finding a solution). But some special cases, like least square, is easy to solve. Solve means finding the global minimum - and it is guaranteed to exist.

1.2 Least Square and Linear Programming

Their common parent is the convex optimization problem.
Least square problems has analytical solutions. There are analytical solutions that are impossible to actually compute, but in this course, we are focusing on some problems that **don't have analytical solution but are easy to compute**. Convex optimization problems normally don't have analytical solutions, but have reliable and efficient algorithms.
Linear Programming doesn't have analytical solution. The cool thing about it is that there are many problems in other forms, **could be transformed into a Linear Programming problem**. We'd have to know what kinds of questions can be transformed and how to transform. This extends the range of problems that linear programming applies to.
Both models are mature techniques, tracing back to Gauss and Fourier time. Both of them are special cases of convex optimization.

1.3 Examples

Lamp-illuminating problem, optimizing the mini-max percentage error; constraint the range (of the light power) between 0 and max; with regularization towards (max / 2).
Constraints might look the same but totally different. Untrained intuitions are not reliable.
It is important to get trained and find "the correct thing to estimate" properly.
In this course, not to replicate, but to demystify.

1.4 Nonlinear Optimization (Non-Linear Programming)

Traditional method of non-convex problems.

Back then (50's, 60's etc.) scientists are categorizing problems as, LP (Linear Programming), NLP (Non-Linear Programming), ILP (I for integer), etc.

For Non-Linear Programming, localized solution starts from an initial guess and go gradient descent, global solution could be exponential time in the worst case. Almost every global optimization is based on convex optimization as a subroutine.

1.5 A Brief History

Convex analysis 1900-1970 roughly.

The algorithms involved includes (according to the timeline):

- simplex (an algorithm, very simple) for linear programming
- early interior-point method
- ellipsoid method and other subgradient methods
- polynomial-time interior-point method for linear programming
- polynomial-time interior-point method for non-linear convex optimization

Generally, the application in engineering fields increased dramatically.

2 Math Basis

2.1 Numerical linear algebra background

2.1.1 Introduction

Going deep into the solver.

2.1.2 Matrix structure and algorithm complexity

Time complexity solving $Ax = b$ with $A \in \mathbb{R}^{n \times n}$, in terms of the flop (floating-point operation, a rough estimation)[1] count:

- For general methods, grows as n^3;
- Less if A well-structured: banded[2], sparse, Toeplitz, etc.

[1] Just the standard method we use to estimate time complexity.

[2] The larger difference between i and j, the closer to 0 a_{ij} is.

Vector operations with $x, y \in \mathbb{R}^n$:

- inner product $x^T y$: $2n - 1$ flops
- sum $x + y$: n flops
- scalar multiplication αx: n flops

matrix-vector product $y = Ax$ with $A \in \mathbb{R}^{m \times n}$:

- $m(2n - 1)$ flops
- if A is sparse with N non-zero elements, $2N$ flops
- if $A = UV^T$ ($U \in \mathbb{R}^{m \times p}$, $V \in \mathbb{R}^{n \times p}$): $2p(n + m)$ flops.

matrix-matrix product $C = AB$ with $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{n \times p}$:

- $mp(2n - 1)$ flops
- less if A or B is sparse
- if C symmetric, and $m = p$: $\frac{1}{2}(m + 1)(2n - 1) \approx m^2 n$ flops

a norm $\|Ax - b\|_1^2 + \lambda \|x\|_1^2$ with $A \in \mathbb{R}^{m \times n}$, $\lambda > 0$:

- mn^2 flops if $m \geq n$
- $m^2 n$ flops if $n \geq m$
- After solving one value of λ, cost of solving k other λ value versions is:$n^2 k$ when $m \geq n$, $m^2 k$ when $n \geq m$

The performance could be also greatly influenced by the order you compute them.

2.1.3 Solving linear equations with factored matrices

$x = A^{-1} b$ is easy to solve when A is:

- **diagonal matrix** (when $i \neq j$, $a_{ij} = 0$): n flops

$$x = (b_1/a_{11}, b_2/a_{22}, \ldots, b_n/a_{nn})$$

- **lower triangular matrix** (when $i < j$, $a_{ij} = 0$): n^2 flops

$$
\begin{aligned}
x_1 &= b_1/a_{11} \\
x_2 &= (b_2 - a_{21}x_1)/a_{22} \\
x_3 &= (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33} \\
&\dots \\
x_n &= (b_n - a_{n1}x_1 - a_{n2}x_2 - \dots - a_{n,n-1}x_{n-1})/a_{nn}
\end{aligned}
$$

  The flop count for computing $A^{-1}b$ is the same order as the flop count for computing Ab.

- **upper triangular matrix** (when $i > j$, $a_{ij} = 0$): n^2 flops, similar with the lower triangular one

- **orthogonal matrix** ($A^T = A^{-1}$):

  1. For general orthogonal A, $2n^2$ flops
  2. If further structured could be less, e.g. if $A = I - 2uu^T$, $\|u\|_2 = 1$, only $4n$ flops. $A^{-1}b = A^Tb = b - 2u^Tbu$
  3. If it is special, permutation matrix:

$$
a_{ij} = \begin{cases} 1 & j = \pi_j \\ 0 & \text{otherwise} \end{cases}
$$

     where $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ is a permutation (re-order) of $(1, 2, \dots, n)$.

$$
Ax = (x_{\pi_1}, \dots, x_{\pi_n})
$$

     costs 0 flops.

The factor-solve methods for solving $Ax = b$: cost one factorization plus k solves, the **factorization** always dominant the **cost**

1. Factor A into simple matrices mentioned before (those kinds above, usually $k = 2$ or 3):
$$
A = A_1 A_2 \dots A_k
$$
2. Solving $A_1x_1 = b$, $A_2x_2 = x_1$, $\dots$, $A_kx_k = x_{k-1}$
3. Now we have:
$$
x = A^{-1}b = A_k A_{k-1} \dots A_1 b
$$

Equations with multiple right hand sides:

$$
Ax_i = b_i, \; i = 1, 2, \dots, m
$$

cost is one factorization plus m solves.

2.1.4 LU factorization

Every nonsingular matrix A (invertible) could be factored as:

$$A = PLU$$

where P is permutation matrix, L lower-triangular and U upper-triangular.
Factorization cost: $(2/3)n^3$ flops.
Solving linear equation with LU costs: factorization plus solvers $= (2/3)n^3 + 0 + n^2 + n^2 \approx (2/3)n^3$.

2.1.5 sparse LU factorization

$$A = P_1 L U P_2$$

Another permutation matrix P_2 makes it **possible** (heuristic) for L, U being sparse. Cost is usually less that $(2/3)n^3$, but the exact value depends, and is really complex.

$$P_1^T A P_2^T = LU$$

2.1.6 Cholesky factorization

Every **positive definite** A ($A \succ 0$) could be factored as:

$$A = LL^T$$

where L is lower-triangular.
It costs $(1/3)n^3$ flops to do this factorization.
People often require L's entries to be positive.
The computation cost of a linear equation is then $(1/3)n^3 + n^2 + n^2 \approx (1/3)n^3$.

2.1.7 sparse Cholesky factorization

$$A = PLL^T P^T$$

with permutation matrix P that makes L **hopefully** sparse (heuristic). Complexity is usually less that $(1/3)n^3$, depends only on the sparsity features of A, but still complex.

$$P^T A P = LL^T$$

2.1.8 LDL^T factorization

Any nonsingular symmetric matrix A can be factored as:

$$A = PLDL^T P^T$$

P, L has similar meanings as before, D is block-diagonal with 1×1 or 2×2 diagonal blocks.
General solving time: $(1/3)n^3$.
With sparse A, could have sparse L, and the cost will be significantly smaller.

2.1.9 Equations with structured sub-blocks

The first topic under **Block elimination and the matrix inversion lemma**.

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

with $x_i \in \mathbb{R}^{n_i}$, $A_{i,j} \in \mathbb{R}^{n_i \times n_j}$.
If A_{11} is nonsingular:

$$\begin{aligned} & x_1 = A_{11}^{-1}(b_1 - A_{12}x_2) \\ & (A_{22} - A_{21}A_{11}^{-1}A_{12})x_2 = b_2 - A_{21}A_{11}^{-1}b_1 \end{aligned}$$

Compute them little by little, x_2 first and then x_1.
In the middle of the process, **Schur complement**:

$$S = A_{22} - A_{21}A_{11}^{-1}A_{12}$$

is computed.
Cost of computing the Schur complement is:

$$f + n_2 s + n_2^2 n_1$$

if simply compute the Schur complement of A_{11}.
The total cost is:

$$f + n_2 s + 2n_2^2 n_1 + (2/3)n_2^3$$

where f is the factorization cost, s is the cost of solving the actual problem.

2.1.10 Structured matrix with low-rank term

The first topic under **Block elimination and the matrix inversion lemma**.

$$(A + BC)x = b$$

with $A \in \mathbb{R}^{n\times n}$ and structured, $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{p\times n}$. B, C low-rank.

$$\begin{bmatrix} A & B \\ C & -I \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

y is introduced, called **in-elimination**.
Thus we have:

$$\begin{aligned} & (I + CA^{-1}B)y = CA^{-1}b \\ & Ax = b - By \end{aligned}$$

this proves the **block-inversion lemma**: with A and $A + BC$ nonsingular,

$$\begin{aligned} & (A + BC)^{-1} \\ =& A^{-1} - A^{-1}B(I + CA^{-1}B)^{-1}CA^{-1} \end{aligned}$$

A beautiful **example**: A diagonal and B, C dense: with this block method its cost is as low as:

$$2p^2 n + (2/3)p^3 \approx 2p^2 n$$

2.2 Linear Algebra

2.2.1 Dimension & Rank and Determinants

Refer to `http://www.math.drexel.edu/~jwd25/LA_FALL_06/lectures/lecture4B.html`.

2.2.2 Norm

- $\|x\|_1 = \sum_{i=1}^{n} x_i$ for $x \in \mathbb{R}^n$

2.2.3 Dual Norm

The dual norm of $\|x\|$ is:

$$\|x\|_* = \sup_{\|u\|\leq 1} u^T x$$

A trick:

$$\|x\| - A^T x \geq 0 \Longrightarrow \|A\|_* \leq 1$$
$$\inf_x \left(\|x\| + A^T x\right) \Longrightarrow 0 \; (\|A\|_* \leq 1)$$

2.2.4 Null Space (a.k.a. Kernel), Range (a.k.a. Image)

Null space:

Given:

- Linear transformation on $\mathbb{R}^n$: T

then the set of vector X, who makes:

$$T(X) = 0$$

is called the null space ($Null(X)$), or kernel ($Ker(X)$).

In other words,

$$\mathcal{N}(A) = \{x | Ax = 0\}$$

To find the nullspace of a matrix, we can apply procedures:

- QR factorization
- Singular value decomposition

Range: sometimes also called "image" and expressed as $Im(A)$ etc.

$$\mathcal{R}(A) = \{Ax | x \in \mathbb{R}^n\}$$

They are connected in various ways (Let's assume $\mathcal{N}(A) \in \mathbb{R}^n$, $\mathcal{R}(A) \in \mathbb{R}^m$):

$$\mathbf{dim}\,\mathcal{N}(A)^{\perp} = \mathbf{dim}\,\mathcal{R}(A^T) = \mathbf{dim}\,\mathcal{R}(A) = r$$

$$\mathbf{dim}\,\mathcal{N}(A) = n - r$$

$$\mathbf{dim}\,\mathcal{R}(A)^{\perp} = \mathbf{dim}\,\mathcal{N}(A^T) = m - r$$

$$x \in \mathcal{N}(A)^{\perp} \Longrightarrow Ax \in \mathcal{R}(A)$$

$$\mathcal{R}(A)^{\perp} \iff \mathcal{N}(A^T) \qquad \mathcal{N}(A)^{\perp} \iff \mathcal{R}(A^T)$$

$$x_1 \in \mathcal{R}(A^T) = \mathcal{N}(A)^{\perp}, x_2 \in A \Longrightarrow x_1 x_2 \in \mathcal{R}(A)$$

$$x_1 \in \mathcal{N}(A), x_2 \in A \Longrightarrow x_1 x_2 \in \{0\}$$

Besides we have:

$$\mathcal{R}(AB) \subseteq \mathcal{R}(A) \qquad \mathcal{N}(AB) \supseteq \mathcal{N}(B) \qquad \mathcal{R}((AB)^T) \subseteq \mathcal{R}(B^T) \qquad \mathcal{N}((AB)^T) \supseteq \mathcal{N}(A^T)$$

$$\mathcal{R}(A) = \mathcal{R}(AA^T) \qquad \mathcal{N}(A) = \mathcal{N}(AA^T)$$

Could view this problem in such a way that: thinking about 3D space, x, y-plane's Null Space is those who parallel with the z-axis, etc.

Could refer to mathworld at `http://mathworld.wolfram.com/NullSpace.html` and `https://math.stackexchange.com/questions/2037602/what-is-range-of-a-matrix`.

2.2.5 Pseudo-Inverse

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 $A^T A$ is nonsingular, and thus:

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

2.2.6 Vectorization

$$\mathbf{vec}(A) = [a_{1,1}, \ldots, a_{m,1}, a_{1,2}, \ldots, a_{m,2}, \ldots, a_{1,n}, \ldots, a_{m,n}]^T$$

Resulting in $A \in \mathbb{R}^{m \times n}$, **column** matrix.

2.2.7 Singular Matrix

A **square** matrix that **does not have a inverse**.

A singular matrix A has $\mathbf{det}(A) = 0$.

Could refer to mathworld at `http://mathworld.wolfram.com/SingularMatrix.html`.

2.2.8 More on Determinant

It is calculated via:

$$\mathbf{det}(A) = \sum_{i=1}^{n} a_{ij}(-1)^{i+j} A_{-(i,j)}$$

where $A_{-(i,j)}$ is A without the i^{th} row and j^{th} column.

Basically when $n = 2$, $\mathbf{det}(A) = a_{11}a_{22} - a_{12}a_{21}$

Useful Property 1:

$$\nabla_X \log \mathbf{det}\, X = \frac{\partial \log \mathbf{det}\, X}{\partial X} = X^{-1}$$

$$\begin{aligned}
&\mathbf{det}(A) = \mathbf{det}(A^T) \\
&\mathbf{det}(aA) = a^n \,\mathbf{det}(A) \\
&\mathbf{det}(AB) = \mathbf{det}(A)\,\mathbf{det}(B) \\
&\mathbf{det}(I) = \mathbf{det}(AA^{-1}) = \mathbf{det}(A)\,\mathbf{det}(A^{-1}) = 1 \\
&\mathbf{det}(A) = \frac{1}{\mathbf{det}(A^{-1})} \\
&\mathbf{det}(BAB^{-1}) = \mathbf{det}(A) \\
&\mathbf{det}(B^{-1}AB - \lambda I) = \mathbf{det}(A - \lambda I) \\
&\mathbf{det}(\overline{A}) = \overline{\mathbf{det}(A)} \\
&\mathbf{det}(A + \epsilon I) = 1 + \epsilon\,\mathbf{det}(A) + \mathcal{O}(\epsilon^2)
\end{aligned}$$

where $\overline{a}$ refers to complex conjugate of a (no matter a number or matrix).

Could refer to mathworld at `http://mathworld.wolfram.com/Determinant.html`.

2.2.9 Matrix Trace

$$\mathbf{tr}(A) = \sum_{i=1}^{n} a_{ii}$$

- $\mathbf{tr}(A) = \mathbf{tr}(A^T)$
- $\mathbf{tr}(A + B) = \mathbf{tr}(A) + \mathbf{tr}(B)$

- $\mathbf{tr}(\alpha A) = \alpha\,\mathbf{tr}(A)$
- $\mathbf{tr}(BAB^{-1}) = \mathbf{tr}(A)$
- $\mathbf{tr}(AB) = \mathbf{tr}(BA)$

Useful Property 0:
Inner product of two symmetric matrices is the trace of the product:

$$\begin{cases} A \in \mathbb{S}^n \\ B \in \mathbb{S}^n \end{cases} \implies AB = \mathbf{tr}(AB)$$

Useful Property 1:

$$\mathbf{tr}(X^T Y) = \mathbf{tr}(XY) = \sum_{i,j} X_{ij} Y_{ij} = \mathbf{vec}(X)^T\,\mathbf{vec}(Y)$$

Useful Property 2:

$$\nabla_X\,\mathbf{tr}(AX) = A$$

Could refer to mathworld at `http://mathworld.wolfram.com/MatrixTrace.html`.

2.2.10 Eigen Decomposition

Could refer to mathworld at `http://mathworld.wolfram.com/EigenDecomposition.html`. Another reference at `https://www.utdallas.edu/~herve/Abdi-EVD2007-pretty.pdf`.

2.2.11 Positive Semi-Definite Matrix

$$X \in \mathbb{S}^n_+,\ \forall y \in \mathbb{R}^n,\ y^T X y \geq 0 \iff X \succeq 0$$

Basically:

- The diagonal values of X (X_{ii}) have to be ≥ 0.
- The element with largest modulus lies on the main diagonal.
- The determinant of X, $\mathbf{det}(X) \geq 0$.
- $X_{ii} + X_{jj} \geq 2|\mathcal{R}[X_{ij}]|$, $\forall i \neq j$, where $\mathcal{R}[\cdots]$ means the **real** part of a value.

Besides, when:

$n = 1$	$X_{11} \geq 0$
$n = 2$	$X_{ii} \geq 0,\ X_{11}X_{22} - X_{12}^2 \geq 0$

And iterative, for $n+1$ size, the conditions for size n still hold, by emitting any one row & any one column. That is, their submatr

2.2.12 Positive Definite Matrix

Could refer to mathworld at http://mathworld.wolfram.com/PositiveDefiniteMatrix.html.

2.2.13 Schur complement

$$X = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \in \mathbb{R}^{(p+q)\times(p+q)}$$

we have the Schur complement for A as:

$$S = X/A = D - CA^{-1}B$$

and the Schur complement for D is:

$$S = X/D = A - BD^{-1}C$$

Could refer to wikipedia at https://en.wikipedia.org/wiki/Schur_complement.
If we only care about the symmetric cases there's many interesting properties.
$X \in \mathbb{S}^n$ partitioned as:

$$X = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$$

where $A \in \mathbb{S}^k$.
Then the Schur (pronounced like "sure") complement of A in X is:

$$S = C - B^T A^{-1} B$$

The time complexity computing it is in section 2.1.9.
Could refer to textbook page 650, A.5.5.

2.3 Calculus and Analysis

2.3.1 Differentiate

Some basic functions:

$f(x)$	$f'(x)$
x^a	ax^{a-1}
e^x	e^x
a^x	$a^x \ln(a)$
$\ln(x)$	$\frac{1}{x}\ x > 0$
$\log_a(x)$	$\frac{1}{x\ln(a)}$
$\sin(x)$	$\cos(x)$
$\cos(x)$	$-\sin(x)$
$\tan(x)$	$\sec^2(x) = \frac{1}{\cos^2(x)} = 1 + \tan^2(x)$
$\arcsin(x)$	$\frac{1}{\sqrt{1-x^2}}$
$\arccos(x)$	$-\frac{1}{\sqrt{1-x^2}}$
$\arctan(x)$	$\frac{1}{1+x^2}$

- If $f(x) = g(x)h(x)$:
$$f'(x) = g'(x)h(x) + g(x)h'(x)$$
- If $f(x) = h(g(x))$:
$$f'(x) = g'(x)h'(g(x))$$
$$f''(x) = g'(x)^2 h''(g(x)) + h'(g(x))g''(x)$$

2.3.2 Conjugation

$$\phi_x(g) = xgx^{-1}$$
$$\phi_x(g)\phi_x(h) = \phi_x(gh)$$

Could refer to mathworld at `http://mathworld.wolfram.com/Conjugation.html`.

2.3.3 Complex Conjugate

The complex conjugate of a complex number $z = a + bi$ is defined to be:

$$\overline{z} = a - bi$$

Could refer to mathworld at `http://mathworld.wolfram.com/ComplexConjugate.html`.

3 Course Concepts

3.1 Introduction

3.1.1 Mathematical Optimization and Convex Optimization

Optimization Problem:

$$\text{minimize } f_0(x)$$
$$\text{subject to } f_i(x) \le b_i \ (i = 1, 2, \ldots, m)$$

where:

- $x = (x_1, \ldots, x_n)$ is optimization variables (a.k.a. decision variables)
- f_0 is objective function ($\mathbb{R}^n \to \mathbb{R}$)
- f_i are constraint functions ($\mathbb{R}^n \to \mathbb{R}$)
- x^* optimal solution

Convex Optimization Problem:

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

where:

- Objective and constraint functions are convex
- Convexity:
  $$f_i(\alpha x + \beta y) \le \alpha f_i(x) + \beta f_i(y)$$
  with $\alpha + \beta = 1$, $\alpha \ge 0$, $\beta \ge 0$.
  "The cord lies above the graph".
  - Affine functions[3] satisfies this for always. Have 0 curvature.
  - Have non-negative curvature for always.
- Generally speaking, **no analytical solution**.
- Reliable & efficient algorithms. Time complexity: $\max\{n^3, n^2m, F\}$. F is the cost of evaluating f_i and their first and second derivatives
- **Almost** a technology
- Many problem could be transformed into a convex form (though might be hard to recognize), and solve via convex optimization.
- does not always have an unique solution though

3.1.2 Least Square and Linear Programming

- most famous, most widely-used optimization problems
- **both** convex-optimization problems

Least square:

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

where $A \in \mathbb{R}^{k \times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^k$.

- The square norm of an affine function of x
- Affine function: linear plus constant, e.g. $A^T x - b$
- Analytical solution: $x^* = (A^T A)^{-1} A^T b$

[3] Affine function: linear plus constant.

- Time complexity n^2k in the worst case
- A mature technology (Gauss)

Linear Programming (LP):

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{subject to } a_i^T x \leq b_i, \ i = 1, 2, \ldots, m \end{aligned}$$

where $x, c, a_i \in \mathbb{R}^n$, $b \in \mathbb{R}^m$.

- No analytical formula of the solution
- Time complexity n^2m if $m \geq n$
- A mature technology (Fourier)

3.2 Convex Set

3.2.1 Empty Set

Empty set is regarded as convex set.

3.2.2 Affine Sets and Convex Sets

Affine Set:

Line through x_1, x_2: $x = \theta x_1 + (1 - \theta)x_2$ where $\theta \in \mathbb{R}$.

The affine set is the set that contains the **line** through **any** two **distinct** points in the set.

The solution set of linear equations, $\{x | Ax - b = 0\}$ is an affine set, and, conversely, **any** affine set could be expressed as solution set of set of linear equations.

Convex Set:

Same idea as affine set except that instead of *line*, we talk about *line segment*.

Line segment through x_1, x_2: $x = \theta x_1 + (1 - \theta)x_2$ where $\theta \in [0, 1]$.

Convex set contains the **line segment** through **any** two **distinct** points in the set.

$$x_1, x_2 \in C, \theta \in [0, 1] \Longrightarrow x = \theta x_1 + (1 - \theta)x_2 \in C$$

> *"any point in the set can see any other point in the set, through a clear line of sight path in between"*

3.2.3 Convex Combination and Convex Hull

Convex Combination: We have k points $x_1, x_2, \ldots, x_k$, convex combination refers to any x of the form:

$$x = \theta_1 x_1 + \theta_2 x_2 + \ldots \theta_k x_k$$

where $\theta_1 + \theta_2 + \cdots + \theta_k = 1$, and $\theta_i \geq 0$.

One view: mixture coefficients θ, x the mixture.

Convex Hull:

$conv(S)$: Set of all convex combinations of points in S.

3.2.4 Conic combination and Convex Cone

Conic (non-negative) combination (of x_1, x_2):

Any point of the form $x = \theta_1 x_1 + \theta_2 x_2$ where $\theta \succeq 0$ ($\theta_1, \theta_2 \geq 0$).

Convex Cone:

Set that contains all conic combinations of points in the set.

3.2.5 Hyperplane and Halfspace (example 1 of convex set)

Hyperplane: **Affine** and **Convex**!

Set of the form:

$$\{x | a^T x = b\} \ (a \neq 0)$$

The shape looks like: **a normal** to the hyperplane is **a**, and then **x** is pointing to "right-hand side" along the boundary. When **b** is changed, the hyperplane moves up and down, parallel.

The **distance** between two parallel line with parameter b_1 and b_2 respectively, is $\frac{|b_1 - b_2|}{\|a\|_2}$

Halfspace: **Convex**!

$$\{x | a^T x \leq b\} \ (a \neq 0)$$

The half of the space that **a** is pointing to is $a^T x > b$, the half where $-$**a** will point to is $a^T x < b$. Boundary is the hyperplane. **a** is an **outward normal**.

3.2.6 Euclidean Balls and Ellipsoids (example 2 of convex set)

Euclidean Ball with center x_c and radius r:

$$\begin{aligned} B(x_c, r) &= \{x \mid \|x - x_c\|_2 \leq r\} \\ &= \{x_c + ru \mid \|u\|_2 \leq 1\} \end{aligned}$$

where u is a vector of norm with length less than or equal to 1.

Euclidean refers to the ℓ_2 norm.

Ellipsoid: a generalization of ball.

$$\begin{aligned} &\{x \mid \|(x - x_c)^T P^{-1}(x - x_c)\|_2 \leq 1\} \\ =&\{x_c + Au \mid \|u\|_2 \leq 1\} \end{aligned}$$

with $P \in \mathbb{S}^n_{++}$, symmetric positive definite. A: **Non**-square and **Non**-singular (linear transformation A of a ball plus the offset x_c).

The semiaxes of the ellipsoid are given by the eigenvectors of P. The semiaxis lengths are the square roots of the eigenvalues of P.

When $P = r^2 I$ it is a ball.

The representation of P is **unique**, that is to say, when two sets are equal the P has to be the same.

But the representation of A is **not unique**, multiple As might give the same set. For example, if Q is orthogonal, which means that $Q^T Q = I$, then A could be replaced by AQ.

3.2.7 Norm Balls and Norm Cones (example 3 of convex set)

Norm: $\|\cdot\|$, nothing but a function, just like absolute value in multiple dimensions.

- Positive Definiteness:
  - Positive: $\|\mathbf{x}\| \geq 0$
  - Definiteness: $\|\mathbf{x}\| = 0 \iff x = 0$
- $\forall t \in \mathbb{R}$, $\|t\mathbf{x}\| = |t|\ \|\mathbf{x}\|$
- **Triangle Inequality**: $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$
- $\|\cdot\|$ is general, unspecified norm
- $\|\cdot\|_{symb}$ is particular norm

$$\|\mathbf{x}\|_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}}$$

- $\|\mathbf{x}\|_\infty = \max_{i=1}^n |x_i|$

Norm Ball: with center x_c and radius r:

$$\{x \mid \|x - x_c\| \leq r\}$$

Convex.

Norm Cone:

$$\{(x,t) \mid \|x\| \leq t\}$$

Convex.

Euclidean Norm Cone (Second-Order Cone):

$$\{(x,t) \mid \|x\|_2 \leq t\}$$

Special case of norm cone, definitely convex.

3.2.8 Polyhedra (a.k.a. Polytopes) (example 4 of convex set)

Solution set of **finite** many inequalities and equalities.

$$Ax \preceq b, \qquad Cx = d$$

$A \in \mathbb{R}^{m\times n}$, $C \in \mathbb{R}^{p\times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $d \in \mathbb{R}^p$.

It is the intersection of finite number of halfspaces and hyperplanes. Defined by each **row** of A or C.

$\preceq$ in this 1-D case is equivalent with componentwise inequality.

3.2.9 Positive Semidefinite Cone (example 5 of convex set)

Denote $\mathbb{S}^n$ an $n\times n$ symmetric matrix, $\mathbb{S}^n_+$ positive semi-definite matrix ($\{X \in \mathbb{S}^n | X \succeq 0\}$), $\mathbb{S}^n_{++}$ positive definite matrix ($\{X \in \mathbb{S}^n | X \succ 0\}$).

$\mathbb{S}^n_+$ is a convex cone.

For example:

$$\begin{bmatrix} x & y \\ y & z \end{bmatrix} \in \mathbb{R}^2_+$$

is a positive semidefinite cone. In fact, it is just a rotated second-order cone.

3.2.10 Operations that Preserve Convexity

How to prove that a set is convex?

Option 1 Fall back to the definition, proving that for $\forall x_1, x_2 \in C$ and $\forall \theta \in [0, 1]$, $\theta x_1 + (1-\theta)x_2 \in C$.

> In fact, it is good enough to show that the set is closed under averaging, that is to say, let $\theta = 0.5$ all the times and prove it will be good enough to show that the set is convex.

Option 2 Show that C is obtained from simple convex sets by **operations that preserve convexity**. (Imagine an expression tree this way.)

Operations that preserve convexity:

- **Intersection**: The intersection of **any number of** (finite or infinite) convex sets is convex.

  > For example, if a set satisfies something for t in a range $t \in [a, b]$, then we could separate this problem into infinite sets each with a fixed t.

- Affine functions: $f : \mathbb{R}^n \to \mathbb{R}^m$

  $$f(x) = Ax + b$$

  with $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$. Then:

  - The image of a convex set under f is convex:

    $$S \subseteq \mathbb{R}^n \Longrightarrow f(S) = \{f(x) | x \in S\} \text{ is convex}$$

  - The inverse image ($f^{-1}(C)$) of a convex set under f is convex:

    $$C \subseteq \mathbb{R}^m \Longrightarrow f^{-1}(C) = \{x \in \mathbb{R}^n | f(x) \in C\} \text{ is convex}$$

  Sometimes f^{-1} does not even exist, that is to say, f is not invertable, but **this operation still works**, and the conclusion still holds.

  > Some examples include:
  >
  > - Scaling, Transaction (move to another position), Rotation, Projection (e.g. simply reduce the dimension by ignoring some dimensions, that is geographically like a shadow)
  > - **Linear matrix inequality**'s solution set:
  >
  > $$\{x \mid x_1 A_1 + \cdots + x_m A_m \preceq B\}$$
  >
  >   with $A_i, B \in \mathbb{S}^p$. By having $f : \mathbb{R}^m \to \mathbb{S}^p$:
  >
  > $$f(x) = B - \sum_{i=1}^{m} x_i A_i$$
  >
  >   and use its inverse set.

- Hyperbolic cone:

$$\{x \mid x^T P x \le (c^T x)^2,\ c^T x \ge 0\}$$

with $P \in \mathbb{S}^n_+$. It is an inverse image of a standard cone under an affine function.

- Perspective (a.k.a. projective mapping) functions P: $\mathbb{R}^{n+1} \to \mathbb{R}^n$

$$P(x,t) = x/t, \qquad \textbf{dom } P = \{(x,t) | t > 0\}$$

basically, it is dividing the first n element with the last one.

Images and inverse images under perspective are convex.

- Linear-fractional functions $f : \mathbb{R}^n \to \mathbb{R}^m$

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

It is just like the image put into your camera or your retina.

Images and inverse images under linear-fractional functions are convex.

A quick example:

$$f(x) = \frac{x}{x_1 + x_2 + 1}$$

where $x \in \mathbb{R}^2$.

3.2.11 Generalized Inequalities

A convex cone $K \subseteq \mathbb{R}^n$ is a **proper cone** if:

- K is closed (contains its boundary)
- K is solid (has no empty interior, a shape not a line segment)
- K is pointed (contains no line, can't have a ray and its negative together in a proper cone)

Some examples include: nonnegative orthant $K = \mathbb{R}^n_+$, positive semidefinite cone $K = \mathbb{S}^n_+$, non-negative polynomials on $[0,1]$:

$$K = \{x \in \mathbb{R}^n \mid x_1 + x_2 t + x_3 t^2 + \cdots + x_n t^{n-1}\ (t \in [0,1])\}$$

Generalized inequality (based on a proper cone K):

$$x \preceq_K y \iff y - x \in K$$
$$x \prec_K y \iff y - x \in \textbf{int } K$$

int here refers to interior, that is the inner part, not the boundary.

Some way-too-common (so that we drop the K when expressing) examples:

- $K = \mathbb{R}_+$ (not really generalized, just the basis, used for comparison):

$$x \preceq_K y \iff y - x \in K$$

$$x \preceq_{\mathbb{R}_+} y \iff y - x \in \mathbb{R}_+$$
$$x \leq y \iff y - x \geq 0$$

- $K = \mathbb{R}^n_+$:
$$x \preceq_K y \iff y - x \in K$$
$$x \preceq_{\mathbb{R}^n_+} y \iff y - x \in \mathbb{R}^n_+$$
$$x \preceq y \iff y_i - x_i \geq 0 \ (i = 1, 2, \ldots n)$$

- $K = \mathbb{S}^n_+$:
$$x \preceq_K y \iff y - x \in K$$
$$x \preceq_{\mathbb{S}^n_+} y \iff y - x \in \mathbb{S}^n_+$$
$$x \preceq y \iff y - x \text{ positive semidefinite}$$

Some properties that are kept similar with that on $\mathbb{R}$:

- $x \preceq_K y, a \preceq b \Longrightarrow (x + a) \preceq (y + b)$

Some properties are not the same with $\mathbb{R}$:

- It is not generally a linear ordering, we can have $x \npreceq_K y$ and $y \npreceq_K x$ at the same time.

3.2.12 Minimum and Minimal elements of Generalized Inequality

Minimum Element:

$x \in S$ is **the** minimum element of S with respect to $\preceq_K$ if:

$$y \in S \Longrightarrow x \preceq_K y$$

Minimal Element:

$x \in S$ is **a** minimal element of S with respect to $\preceq_K$ if:

$$y \in S, y \preceq x \Longrightarrow y = x$$

3.2.13 Separating Hyperplanes and Supporting Hyperplanes

Separating Hyperplane Theorem:

Theorem. *If C and D are disjoint convex sets, then there must exists $a \neq 0$, b such that $a^T x \leq b$ for $x \in C$, and $a^T x \geq b$ for $x \in D$.*

The division line $a^T x = b$ has a norm a pointing to D side.

Strict separation requires additional assumptions. (e.g. C is closed, D is singleton)

It means that, there's a linear classifier that will distinguish C and D.

Supporting Hyperplane Theorem:

The basic idea is if you have a hyperplane that touches the set but not going into it, as if the hyperplane is holding the set, that is supporting.

Supporting hyperplane to set C at point x_0:

$$\{x \mid a^T x = a^T x_0\}$$

Theorem. *If C is convex, then there exists a supporting hyperplane at **every** boundary point of C.*

But the supporting hyperplane of C at x_0 is **not** necessarily uniquely defined - imagine a corner of a square, for example.

3.2.14 Dual Cones and Generalized Inequalities

Dual cone of a cone K:

$$K^* = \{y \mid y^T x \geq 0, \ \forall x \in K\}$$

Some examples:

K	**K***
$\mathbb{R}_+^*$	$\mathbb{R}_+^*$
$\mathbb{S}_+^*$	$\mathbb{S}_+^*$
$\{(x,t) \mid \|x\|_2 \leq t\}$	$\{(x,t) \mid \|x\|_2 \leq t\}$
$\{(x,t) \mid \|x\|_1 \leq t\}$	$\{(x,t) \mid \|x\|_\infty \leq t\}$

The first three are **self-dual** cones.

Dual cones for proper cones are proper. Hence define generalized inequalities:

$$y \succeq_{K^*} 0 \iff y^T x \geq 0 \text{ for all } x \succeq_K 0$$

The dual cone, geographically, is: starting from the boundary, the "original point" (o) of the set, draw a hyperplane, with norm y, if the set is fully included in the halfspace where y points to, then $y \in K^*$.

If K is thin, then K^* is thick. Normally, if K is "90 degrees", then it is self-dual. Also normally, $(K^*)^*$ is K. For example, $x \in \mathbb{R}$, then for $K = \{(x,t) \mid |x| \leq t\}$, $K^* = K$.

3.2.15 Minimum and Minimal Elements via Dual Inequalities

minimum element w.r.t. $\preceq_K$

$$x \text{ is the minimum element of } S \iff \forall \lambda \succ_{K^*} 0,\ x \text{ is the } \textbf{unique} \text{ minimizer of } \lambda^T z \text{ over } S$$

minimal element w.r.t. $\preceq_K$

$$x \text{ is minimal of } S \iff \exists \lambda \succ_{K^*} 0,\ x \text{ minimizes } \lambda^T z \text{ over } S$$

That is, geographically, all points where, from that point, the line segments (in right angle) to any of the axis, doesn't have intersection with the rest of the set.

Optimal Production Frontier

Different production methods use different amount of resources $x \in \mathbb{R}^n$.

P: The production set. Resource vector x for all possible production methods.

Efficient (**Pareto optimal**) methods correspond to resource vectors x that are **minimal** w.r.t. $\mathbb{R}^n_+$.

3.3 Convex Function

3.3.1 Definition

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

- $\textbf{dom}\, f$ is a convex set
- $\forall x, y \in \textbf{dom}\, f, \theta \in [0, 1]$:

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

- And **strictly-convex** is defined as:

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

  for $\forall x, y \in \textbf{dom}\, f, \theta \in (0, 1)$.

We also know that:

- f is concave if $-f$ is convex, f is convex if $-f$ is concave.
- The line segment in the graph space (from x to y) is sometimes called **chord**. An old way of talking about convexity is saying that f is below its chord.

3.3.2 Convex and Concave Examples

Some examples on $\mathbb{R}$ we have for convex and concave (in respect to x):

name	**expression**	restrictions	**convex**	**concave**
affine	$ax + b$	$x, a, b \in \mathbb{R}$	✓	✓
exponential	e^{ax}	$x, a \in \mathbb{R}$	✓	
powers	x^{α}	$x \in \mathbb{R}_{++}$ $(x > 0)$, $\alpha \in (-\infty, 0] \cup [1, \infty)$	✓	
powers	x^{α}	$x \in \mathbb{R}_{++}$ $(x > 0)$, $\alpha \in [0, 1]$		✓
power of $\lvert\cdot\rvert$	$\lvert x\rvert^{\alpha}$	$x \in \mathbb{R}$, $\alpha \geq 1$	✓	
negative entropy	$x \log x$	$x \in \mathbb{R}_{++}$ $(x > 0)$	✓	
logarithm	$\log x$	$x \log x$ $x \in \mathbb{R}_{++}$ $(x > 0)$		✓

Some examples on $x \in \mathbb{R}^n$ we have for convex and concave:

name	**expression**	restrictions	**convex**	**concave**
affine	$f(x) = a^T x + b$	$x, a, b \in \mathbb{R}^n$	✓	✓
norms	$f(x) = \|x\|_p$	$x \in \mathbb{R}^n$, $p \geq 1$	✓	

Some examples on $x \in \mathbb{R}^{m \times n}$ we have for convex and concave:

name	**expression**	restrict	**convex**	**concave**
affine	$f(X) = \mathbf{tr}(A^T X) + b = \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij} X ij + b$	$X, A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}$	✓	✓
spectral norm	$f(X) = \|X\|_2 = \sigma_{max}(X) = (\lambda_{max}(X^T X))^{1/2}$	$X \in \mathbb{R}^{m \times n}$	✓	

Spectral norm is also known as maximum singular value norm.

3.3.3 Trick 1: Restriction of a convex function to a line

f is convex only g is convex:

$$g(t) = f(x + tv), \qquad \mathbf{dom}\, g = \{t \mid x + tv \in \mathbf{dom}\, f,\ x \in \mathbf{dom}\, f,\ v \in \mathbb{R}^n\}$$

with $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R} \to \mathbb{R}$.

It is also true statement for concave cases.

If, for any choice of x and v, g is always convex (/concave) in t, then f must be convex (/concave) as well.

It is useful in high dimensions when n is large, and you can check the convexity of f by checking convexity of a line. If you can find a sample that does not hold, then it is **definitely not convex**.

An example: $\mathbf{dom}\, f = \mathbb{S}^n_{++}$, $f : \mathbb{S}^n \to \mathbb{R}$, and it is the log-determinant:

$$f(X) = \mathbf{log}\,\mathbf{det}\, X$$

We could have:

$$\begin{aligned} g(t) &= \mathbf{log}\,\mathbf{det}(X + tV) \\ &= \mathbf{log}\,\mathbf{det}\left(X^{1/2}(I + tX^{-1/2}VX^{-1/2})X^{1/2}\right) \\ &= \mathbf{log}\,\mathbf{det}(X) + \mathbf{log}\,\mathbf{det}(I + tX^{-1/2}VX^{-1/2}) \\ &= \mathbf{log}\,\mathbf{det}(X) + \sum_{i=1}^{n}(1 + t\lambda_i) \end{aligned}$$

where λ_i are the eigenvalues of $(X^{-1/2}VX^{-1/2})$.

For any choice of $X \succ 0$, and V, g is always concave on t, thus f is concave as well.

3.3.4 Trick 2: Extended-value extension

An extended-value extension of f is denoted as $\tilde{f}$, defined as:

$$\tilde{f}(x) = \begin{cases} f(x) & x \in \mathbf{dom}\, f \\ \infty & x \notin \mathbf{dom}\, f \end{cases}$$

This extension often simplifies the notation. Most of the operations on infinity work out fine.

3.3.5 Trick 3.1: First-Order Condition

f is **differentiable** iff:

- $\mathbf{dom}\, f$ is open (end-points not included)
- There exists $\nabla f(x)$ at each $x \in \mathbf{dom}\, f$ such that:

$$\nabla f(x) = (\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_n})$$

The **first-order condition** refers to that:

> If f is first-order differentiable, then it is convex iff:
>
> $$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$
>
> holds for all $x, y \in \mathbf{dom}\, f$.

In other words, **a convex function's first-order approximation is a global underestimator**.

3.3.6 Trick 3.2: Second-Order Conditions

f is **twice differentiable** iff:

- $\textbf{dom}\, f$ is open (end-points not included)
- There exists the Hessian $\nabla^2 f(x) \in \mathbb{S}^n$ at each $x \in \textbf{dom}\, f$ such that:

$$\nabla^2 f(x)_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

for $i, j = 1, 2, \ldots, n$.

The **second-order conditions** refer to that:

If f is second-order differentiable, then it is convex iff:

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

holds for all $x \in \textbf{dom}\, f$.

And if it is $\nabla^2 f(x) \succ 0$ in stead of $\nabla^2 f(x) \succeq 0$, then f is **strictly** convex. **But a function could be strictly convex without having $\nabla^2 f(x) \succeq 0$.**

Some examples:

name	$f(x)$	$\nabla f(x)$	$\nabla^2 f(x)$	convex condition
quadratic	$\frac{1}{2}x^T Px + q^T x + r$ $(P \in \mathbb{S}^n)$	$Px + q$	P	$P \succeq 0$
least-squares objective	$\|Ax - b\|_2^2$	$2A^T(Ax - b)$	$2A^T A$	**always** convex
quadratic ove linear	$f(x, y) = \frac{x^2}{y}$ $(y > 0)$		$\nabla^2 f(x, y) = \frac{2}{y^3} \begin{bmatrix} y \\ -x \end{bmatrix} \begin{bmatrix} y \\ -x \end{bmatrix}^T$	$\nabla^2 f(x, y) \succeq 0$
log-sum-exp	$\log \sum_{i=1}^n \exp x_i$		$\frac{1}{\mathbb{1}^T z\, \textbf{diag}\, z} - \frac{1}{(\mathbb{1}^T z)^2} zz^T$ $(z_k = \exp x_k)$	**always** convex
geometric mean	$(\prod_{i=1}^n x_i)^{1/n}$, $x \in \mathbb{R}_{++}^n$			**always** concave

Proof of that the log-sum-exp is convex:

To show that:

$$\nabla^2 f(x) = \frac{1}{\mathbb{1}^T z\, \textbf{diag}\, z} - \frac{1}{(\mathbb{1}^T z)^2} zz^T \succeq 0$$

We need to prove that, $\forall v$:

$$v^T(\nabla^2 f(x))v = \frac{(\sum_k z_k v_k^2)(\sum_k z_k) - (\sum_k z_k v_k)^2}{(\sum_k z_k)^2} \geq 0$$

According to the CauchySchwarz Inequality (See 4.2.1), we have:

$$(\sum_k z_k v_k^2)(\sum_k z_k) \geq (\sum_k z_k v_k)^2$$

So it always holds.

And the proof of that the geometric mean is concave is alike.

The **log-sum-exp** is very important because:

- It has physics meaning of DB combining
- It is the smooth approximation of a maximum, **a.k.a. soft-max**!

3.3.7 Epigraph and Sublevel Set

Epigraph is what connects the convex sets and the convex functions.

α-sublevel set of $f : \mathbb{R}^n \to \mathbb{R}$:

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

epigraph of $f : \mathbb{R}^n \to \mathbb{R}$:

$$\textbf{epi}\, f = \{(x, t) \in \mathbb{R}^{n+1} \mid x \in \textbf{dom}\, f, f(x) \leq t\}$$

f is convex $\iff$ epi f is a convex set.

Plus, there are some other conclusions:

- **epi f is a halfspace $\iff$ f is affine.**
- **epi f is a convex cone $\iff$ f is convex and positively homogeneous, i.e., $f(\alpha x) = \alpha f(x)$ for any $\alpha \geq 0$ and any x.**
- **epi f is a polyhedron $\iff$ f is convex and piecewise-affine.**

If you have a convex function, the sublevel sets are all convex. **But** if you have a convex sublevel set, you can't say for sure whether or not the function is convex.

3.3.8 Jensen's Inequality

Refer to subsection 4.2.2 for more details.

Note that, the basic inequality (which is exactly the definition of convexity, and it is called **Jensen's Inequality**):

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

is a special case of:

$$f(\mathbb{E}z) \leq \mathbb{E}f(z)$$

3.3.9 Brief Summary: How to prove convexity?

- Verify definition. If the function is complex, simplify it by restricting it to a line.
- If the function f is twice-differentiable, examine $\nabla^2 f$.
- Show that f is obtained from some known convex functions by **certain operations** (the operations that preserve convexity).

3.3.10 Operations that Preserve Convexity

Operations **know to preserve convexity** are:

1. **Nonnegative weighted sum**:

   $\sum \alpha_i f_i$ is convex if $\alpha_i \geq 0$ and f_i are convex.

   Proof:

   - nonnegative multiple: $\alpha \geq 0$, f is convex, then αf is convex
   - sum: $\sum f_i$ is convex if all f_i are convex

   Example: log barrier for linear inequalities:

   $$f(x) = -\sum_{i=1}^{m} \log(b_i - a_i^T x)$$

   where $\mathbf{dom}\, f = \{x \mid a_i^T x < b, i = 1, 2, \ldots, m\}$

2. **Composition with affine function**:

   $f(Ax + b)$ is convex when f is convex and $Ax + b \in \mathbf{dom}\, f$.

   Example: **any** norm of affine function is convex.

3. **Pointwise maximum and pointwise supremum**:

   Pointwise maximum: if $f_1, \ldots, f_m$ are convex, then:

$$f(x) = \max\{f_1(x), f_2(x), \ldots, f_m(x)\}$$

   is convex.

   Examples:

   - piecewise-linear function is convex:

$$f(x) = \max_{i=1,\ldots m} (a_i^T x + b_i)$$

   - sum of r largest components of $x \in \mathbb{R}^n$ $(r \leq n)$ is convex:

$$f(x) = \sum_{i=1}^{r} x_{[i]}$$

     where $x_{[i]}$ is the i^{th} largest component in x.
     (Proved by viewing f as the maximum of the sum of random r elements in x.)

   Note that the pointwise maximum **does not preserve differenciability**.

   Actually in convex optimization, differenciability does not play a very important role. This is not gonna be a calculus class. Rather, linear algebra is more important.

   Pointwise supremum: if $f(x, y)$ is convex in x for $\forall y \in \mathcal{A}$, then:

$$g(x) = \sup_{y \in \mathcal{A}} f(x, y)$$

   is convex.

   Examples:

   - support function of a set C is convex:

$$S_C(x) = \sup_{y \in C} y^T x$$

   - distance to farthest point in a set C is convex:

$$f(x) = \sup_{y \in C} \|x - y\|$$

   - maximum eigenvalue of symmetric matrix, for $X \in \mathbb{S}^n$:

$$\lambda_{max}(X) = \sup_{\|y\|_2 = 1} y^T X y$$

4. **Composition with scalar functions**:

   f is the composition of $g : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R} \to \mathbb{R}$:

$$f(x) = h(g(x))$$

   f is convex if:

$$\begin{cases} g \text{ convex, } h \text{ convex, } \tilde{h} \text{ nondecreasing} \\ g \text{ concave, } h \text{ convex, } \tilde{h} \text{ nonincreasing} \end{cases}$$

   where $\tilde{h}$ is the extended-value extension of h (see 3.3.4).

   To get the intuitive idea we might have a look at the case where $n = 1$, see 2.3.1 for more details:

$$f''(x) = g'(x)^2 h''(g(x)) + h'(g(x))g''(x)$$

   Note that **monotonicity** (nondecreasing or nonincreasing) must hold for $\tilde{h}$.

   Examples:

   - $\exp g(x)$ is convex if g is convex
   - $\frac{1}{g(x)}$ is convex if g is positive and concave

5. **Vector Composition**:

   f is the composition of $g : \mathbb{R}^n \to \mathbb{R}^k$ and $h : \mathbb{R}^k \to \mathbb{R}$:

$$f(x) = h(g(x)) = h(g_1(x), g_2(x), \ldots, g_k(x))$$

   f is convex if:

$$\begin{cases} g_i \text{ convex, } h \text{ convex, } \tilde{h} \text{ nondecreasing in each argument} \\ g_i \text{ concave, } h \text{ convex, } \tilde{h} \text{ nonincreasing in each argument} \end{cases}$$

   where $\tilde{h}$ is the extended-value extension of h (see 3.3.4).

   Similarly we have when $n = 1$ and g, h both differentiable:

$$f''(x) = g'(x)^T(\nabla^2 h(g(x)))g'(x) + \nabla h(g(x))^T g''(x)$$

   Examples:

   - $\log \sum_{i=1}^m \exp g_i(x)$ is convex if g_i are convex (choose h to be the log-sum-exp function)
   - $\sum_{i=1}^m \log g_i(x)$ is concave if g_i are concave and positive

6. **Minimization (a.k.a. Partial Minimization)**:

   if $f(x, y)$ is convex in (x, y) and C is a convex set, then:

$$g(x) = \inf_{y \in C} f(x, y)$$

   is convex.

   Note that the conditions are stronger (more restrictions) than the maximum and supremum case. It has to be **jointly-convex** in minimum.

   Examples:

   - $f(x) = x^T Ax + 2x^T By + y^T Cy$, with:

$$C \succ 0, \qquad \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succeq 0$$

     Minimizing over y gets:

$$g(x) = \inf_{y} f(x, y) = x^T (A - BC^{-1} B^T) x$$

     g is convex, hence the Schur complement $(A - BC^{-1} B^T) \succeq 0$.

   - distance to a set:

$$\mathbf{dist}(x, S) = \inf_{y \in S} \|x - y\|$$

     is convex if S is convex.

7. **Perspective**:

   The **perspective** $g : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ of a function $f : \mathbb{R}^n \to \mathbb{R}$,

$$g(x, t) = tf(x/t), \qquad \mathbf{dom}\, g = \{(x, t) \mid t > 0, x/t \in \mathbf{dom}\, f\}$$

   is convex if f is convex.

   Examples:

   - $f(x) = x^T x$ is convex, so for $t > 0$, $g(x, t) = \frac{x^T x}{t}$ is convex
   - negative logarithm $f(x) = -\log x$ is convex, hence the *relative entropy*:

$$g(x, t) = t \log t - t \log x$$

     is convex on $\mathbb{R}^2_{++}$.

   - for any convex f,

$$g(x) = (c^T x + d) f(\frac{Ax + b}{c^T x + d})$$

     is convex on domain:

$$\mathbf{dom}\, g = \{x \mid c^T x + d > 0, \frac{Ax + b}{c^T x + d} \in \mathbf{dom}\, f\}$$

3.3.11 The Conjugate Function

The **conjugate** of a function f is defined as:

$$f^*(y) = \sup_{x \in \textbf{dom} f} (y^T x - f(x))$$

The conjugate function f^* of any function f is convex, even if f is not.

On a 2D plane it looks like: the value of a function's conjugate at a certain y_0 is the negative of b (the intersection with y axis) for line $y = y_0^T x + b$, when the line serves as a **tangent line**.

Examples:

1. negative logarithm:

$$f(x) = -\log x$$

$$\begin{aligned} f^*(x) &= \sup_{x>0}(xy + \log x) \\ &= \begin{cases} -1 - \log(-y) & y < 0 \\ \infty & otherwise \end{cases} \end{aligned}$$

2. strictly-convex quadratic with $Q \in \mathbb{S}^n_{++}$ (easily calculate by that it is differentiable):

$$f(x) = \frac{1}{2} x^T Q x$$

$$\begin{aligned} f^*(x) &= \sup_x (y^T x - \frac{1}{2} x^T Q x) \\ &= \frac{1}{2} y^T Q^{-1} y \end{aligned}$$

Reference: `https://www.encyclopediaofmath.org/index.php/Conjugate_function`

3.3.12 Quasiconvex Functions (1/3 Generalization)

Definition: $f : \mathbb{R}^n \to \mathbb{R}$ is quasiconvex if:

- $\textbf{dom} f$ is convex set;
- the sublevel sets:

$$S_\alpha = \{x \in \textbf{dom} f \mid f(x) \le \alpha\}$$

  are convex for all α.

There are some properties:

- f is quasiconcave if $-f$ is quasiconvex
- f is quasilinear if it is quasiconvex and quasiconcave

It justifies the idea of unimodal, like, we can't have two "bumps".

function	domain	quasiconvex	quasiconcave	quasilinear
$\sqrt{\lvert x \rvert}$	$\mathbb{R}$	✓		
$ceil(x)$	$\mathbb{R}$	✓	✓	✓
$\log x$	$\mathbb{R}_{++}$	✓	✓	✓
$f(x) = x_1 x_2$	$\mathbb{R}^2_{++}$		✓	
$f(x) = \frac{1}{x_1 x_2}$	$\mathbb{R}^2_{++}$	✓ (**convex**)		
$f(x) = \frac{x_1}{x_2}$	$\mathbb{R}^2_{++}$	✓	✓	✓
linear fractional function $f(x) = \frac{a^T x + b}{c^T x + d}$	$\textbf{dom} f = \{x \mid c^T x + d > 0\}$	✓	✓	✓
distance ratio $f(x) = \frac{\lVert x-a \rVert_2}{\lVert x-b \rVert_2}$	$\textbf{dom} f = \{x \mid \lVert x - a \rVert_2 \leq \lVert x - b \rVert_2\}$	✓		

A **real**-life example: **internal rate of return**[4].

Important **properties** of quasiconvex functions:

1. **modified Jensen inequality**: f is quasiconvex, then,

$$\theta \in [0,1] \Longrightarrow f(\theta x + (1-\theta)y) \leq \max\{f(x), f(y)\}$$

   which means that, instead of "below the chord", it goes below the "box".

2. **first-order condition**:
   if f is
   - differentiable
   - $\textbf{dom} f$ is convex set

   then it is quasiconvex iff

$$f(y) \leq f(x) \Longrightarrow \nabla f(x)^T (y - x) \leq 0$$

   in other words, it is like another way of describing the general idea that "it is a collection of centered, cascade shapes" or something alike.

3. **sums** of quasiconvex functions are **not necessarily** quasiconvex

Do not care too much about the function f, think about $f \geq \alpha$ or $f \leq \alpha$ directly.

[4] See the lecture note for *convex functions* for more details.

3.3.13 Log-concave and Log-convex Functions (2/3 Generalization)

Log-concave

A positive function f is log-concave $\Longrightarrow$ $\log f$ is concave:

$$\log f(\theta x + (1-\theta)y) \geq \theta \log f(x) + (1-\theta)\log f(y)$$

$$f(\theta x + (1-\theta)y) \geq f(x)^{\theta} f(y)^{1-\theta}$$

Log-convex

A positive function f is log-convex $\Longrightarrow$ $\log f$ is convex:

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

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

It is not a simple extension. And it happens to be that most of the interesting cases are log-concave.

Examples:

- **powers**: x^{α} on $\mathbb{R}_{++}$, when $\alpha \geq 0$ it is log-concave, when $\alpha \leq 0$ it is log-convex.
- **normal** (and many other common probability densities):

$$f(x) = \frac{1}{\sqrt{(2\pi)^n \operatorname{\mathbf{det}} \Sigma}} e^{-\frac{1}{2}(x-\overline{x})^T \Sigma^{-1}(x-\overline{x})}$$

- **cumulative Gaussian distribution function (Φ)**:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{u^2}{2}} du$$

Properties of Log-concave Functions

1. If f is:
   - twice-differentiable
   - with convex domain

   then it is log-concave iff:

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

   for all $x \in \operatorname{\mathbf{dom}} f$.

2. **product of log-concave functions is log-concave**:

   equivalent with that, the sum of concave functions is concave function.

3. sum of log-concave functions is **not always** log-concave:

   it means that the mixture distribution, with log-concave distributions, is not always log-concave.

4. **integration**: if $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is log-concave, then,

$$g(x) = \int f(x, y)dy$$

   is log-concave.[5]

Important Consequences of the Integration Property:

1. **convolution** of log-concave functions is log-concave:

   convolution of functions f and g is $h = f * g$:

$$h(x) = (f * g)(x) = \int f(x - y)g(y)dy$$

   then if f, g are log-concave functions, h is log-concave.

2. (before taking the **log**, sometimes called "the yield function" etc.) if:

   - $C \in \mathbb{R}^n$ is a convex set
   - y is a random variable with log-concave pdf, [6]

   then,

$$f(x) = \mathbf{prob}(x + y \in C)$$

   is log-concave.

   Could be proved by (denote the pdf of y as $p(y)$):

$$f(x) = \int g(x + y)p(y)dy, \qquad g(u) = \begin{cases} 1 & u \in C \\ 0 & u \notin C \end{cases}$$

   More concretely, the yield function:

$$Y(x) = \mathbf{prob}(x + w \in S)$$

   - $S \in \mathbb{R}^n$ is a convex set of acceptable values
   - $x \in \mathbb{R}^n$ are the nominal parameter values for product
   - $w \in \mathbb{R}^n$ is a random variable with log-concave pdf, random variations of parameters in manufactured product

   then:

   - Y is log-concave
   - yield regions $\{x | Y(x) \geq \alpha\}$ are **convex**

Note some strong asymmery here in these cases. e.g. You want to maximize something when don't care about minimization at all.

[5] And this is hard to show.

[6] pdf = Probability Distribution Function

3.3.14 Convexity with respect to Generalized Inequalities (3/3 Generalization)

This is the idea of generalizing it to the vector values.

That is, with a cone K, and $f : \mathbb{R}^n \to \mathbb{R}^m$, we define the conditions of being K-**convex** as:

- **dom** f is convex set
- $\forall x, y \in \textbf{dom}\, f$, $\theta \in [0, 1]$,

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

An example is that, $f : \mathbb{R}^m \to \mathbb{R}^m$, $f(X) = X^2$ is $\mathbb{S}^m_+$-convex.

3.4 Convex Optimization Problems

3.4.1 Optimization Problem in Standard Form

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

where the optimal value p^* (the **inferior** [7] value of $f_0(x)$ under all those constraints) is an ordinary value if the problem is bounded and feasible, is ∞ when not feasible (no x satisfies the constraints, the feasible set is empty), $-\infty$ if the problem is unbounded below (can reach any smaller number).

Some explanation:

element	range	meaning
x	$\mathbb{R}^n$	optimization variable / decision variable
f_0	$\mathbb{R}^n \to \mathbb{R}$	objective function / cost function
f_i	$\mathbb{R}^n \to \mathbb{R}$	the **inequality** constraint functions
h_i	$\mathbb{R}^n \to \mathbb{R}$	the **equality** constraint functions

Some concepts:

- x is **feasible** if:
  - $x \in \textbf{dom}\, f_0$
  - x satisfies the constraints
- x is **optimal** if:

[7] It is inferior, not minimum or minimal.

  - x is feasible
  - $f_0(x) = p^*$

  and we denote the set of optimal points as X_{opt}.

- x is **locally optimal** if, $\exists R > 0$ such that x is optimal for the problem:

$$\begin{aligned} \text{minimize } & f_0(x) \\ \text{subject to } & f_i(x) \leq 0 \ (i = 1, 2, \ldots, m) \\ & h_i(x) = 0 \ (i = 1, 2, \ldots, p) \\ & \|z - x\|_2 \leq R \end{aligned}$$

And quite naturally, we have some **implicit constraints** for the standard form optimization problem:

$$x \in \mathcal{D} = \bigcap_{i=1}^{m} \textbf{dom}\, f_i \bigcap_{i=1}^{p} \textbf{dom}\, h_i$$

and $\mathcal{D}$ is called **the domain of the problem**.

By the way, $f_i(x) \leq 0$ and $h_i(x) = 0$ are **explicit constraints**. And if there's no **explicit** constraint, the problem is **unconstrained**.

Note that:

- $\inf(0, 1] = 0$, $\sup[0, 1) = 1$
- $\inf \varnothing = -\infty$, $\sup \varnothing = \infty$
- **minimize** (part of the optimization problem, keyword that introduces an attribute of the optimization object) is not minimum (min, a function, who takes a bunch of variables and compare them, select the smallest value), they have totally different meanings.

We have **feasibility problems**, defined as:

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

and it is equivalent with a **special case** of the general optimization problem where $f_0(x) = 0$, then $p^* = 0$ when feasible, $p^* = \infty$ when infeasible. (Any x is optimal in this case.)

3.4.2 Convex Optimization Problems

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

where f_0 and **all** f_i (inequality constraints) are **convex** functions; and **all** h_i are **affine** (in the form of $a_i^T x - b_i = 0$).

The problem is quasiconvex if:

- f_0 is quasiconvex;
- f_i $(i = 1, 2, \ldots, m)$ are convex.

It is often written as:

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } f_i(x) \le 0 \ (i = 1, 2, \ldots, m) \\ &\qquad\qquad Ax = b \ (A \in \mathbb{R}^{p \times n}) \end{aligned}$$

Note that there's an important property: the **feasible set** of a **convex optimization problem** is **convex**. A convex optimization problem might be viewed as minimizing a convex function over a convex set in some abstract cases, but not strictly this way.

Also note that sometimes even if a problem **is not** a convex optimization problem, it could have an **equivalent expression** that is convex optimization problem.

For example, an optimization problem that is significantly not convex:

$$\begin{aligned} &\text{minimize } x_1^2 + x_2^2 \\ &\text{subject to } x_1/(1 + x_2)^2 \le 0 \\ &\qquad\qquad (x_1 + x_2)^2 = 0 \end{aligned}$$

is equivalent (although not identical[8]) to convex optimization problem:

$$\begin{aligned} &\text{minimize } x_1^2 + x_2^2 \\ &\text{subject to } x_1 \le 0 \\ &\qquad\qquad x_1 + x_2 = 0 \end{aligned}$$

Any local optimal point of a **convex** problem **(globally) optimal**.

It basically tells us that, if you find an optimal point and you know it is a convex optimization problem, then this point you found is globally optimal.

[8] Identical means exactly itself.

It could be proved by the definition of local optimal with radius R and optimal point x, neighbor feasible point z within the distance of R; having another y that might locates far away and together with:

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

and let y be a **global optimal point** with $f_0(y)$ smaller than $f_0(x)$, then there's a conflict in comparing z's f_0 value with x's.

Optimality Criterion for **Differentiable** f_0:

x is optimal when f_0 is differentiable $\iff$:

- x is feasible
- $\forall y$, if y is feasible:
$$\nabla f_0(x)^T(y-x) \geq 0$$

If **nonzero**, $\nabla f_0(x)$ defines a supporting hyperplane for the feasible set X at point x, with $-\nabla f_0(x)$ being the **outward** norm. In general:

- **unconstrained problem**:
$$\text{minimize } f_0(x)$$
then:
$$x \text{ is optimal} \iff \begin{cases} x \in \mathbf{dom}\, f_0 \\ \nabla f_0(x) = 0 \end{cases}$$

- **equality-constrained problem**:
$$\begin{array}{c} \text{minimize } f_0(x) \\ \text{subject to } Ax = b \end{array}$$
then:
$$x \text{ is optimal} \iff \begin{cases} x \in \mathbf{dom}\, f_0 \\ Ax = b \\ \nabla f_0(x) + A^T v = 0 \end{cases}$$

- **minimization over nonnegative orthant**:
$$\begin{array}{c} \text{minimize } f_0(x) \\ \text{subject to } x \succeq 0 \end{array}$$
then:
$$x \text{ is optimal} \iff \begin{cases} x \in \mathbf{dom}\, f_0 \\ x \succeq 0 \\ \begin{cases} \nabla f_0(x)_i \geq 0 & x_i = 0 \\ \nabla f_0(x)_i = 0 & x_i > 0 \end{cases} \end{cases}$$

Here naturally comes the definition of the **Equivalent Convex Problems**.

> Two problems are (informally) **equivalent** if the **solution** of one is readily obtained from the other, and vice versa.

For example:

- **eliminating the equality constraints** of:

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } f_i(x) \le 0, \ (i = 1, 2, \ldots, m) \\ &\qquad Ax = b \end{aligned}$$

  we have:

$$\begin{aligned} &\text{minimize } f_0(Fz + x_0) \\ &\text{subject to } f_i(Fz + x_0) \le 0, \ (i = 1, 2, \ldots, m) \end{aligned}$$

  where we have:

$$x = Fz + x_0 \iff Ax = b \ \text{ for some } z$$

  is the solution of $Ax = b$.

- **introducing equality constraints** to:

$$\begin{aligned} &\text{minimize } f_0(A_0 x + b_0) \\ &\text{subject to } f_i(A_i x + b_i) \le 0, \ (i = 1, 2, \ldots, m) \end{aligned}$$

  we have that:

$$\begin{aligned} &\text{minimize } f_0(y_0) \\ &\text{subject to } f_i(y_i) \le 0, \ (i = 1, 2, \ldots, m) \\ &\qquad y_i = A_i x + b_i, \ (i = 0, 1, 2, \ldots, m) \end{aligned}$$

- **introducing slack variables for linear inequations** to:

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } a_i^T x \le b_i, \ (i = 1, 2, \ldots, m) \end{aligned}$$

  we have that:

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } a_i^T x + s_i = b_i, \ (i = 1, 2, \ldots, m) \\ &\qquad s_i \ge 0, \ (i = 0, 1, 2, \ldots, m) \end{aligned}$$

- **epigraph form** *(very important!)*: from any standard form convex problem, we have it equivalent with:

$$\begin{aligned} &\text{minimize } t \text{ (minimizing over } x, \, t) \\ &\text{subject to } f_0(x) - t \le 0 \\ &\qquad f_i(x) \le 0 \ (i = 1, 2, \ldots, m) \\ &\qquad Ax = b \end{aligned}$$

- **partial minimization** (minimizing over some variables):

$$\begin{aligned} \text{minimize } & f_0(x_1, x_2) \\ \text{subject to } & f_i(x) \le 0 \ (i = 1, 2, \ldots, m) \end{aligned}$$

  is equivalent to:

$$\begin{aligned} \text{minimize } & \tilde{f}_0(x_1) \\ \text{subject to } & f_i(x) \le 0 \ (i = 1, 2, \ldots, m) \end{aligned}$$

  where we have:

$$\tilde{f}_0(x_1) = \inf_{x_2} f_0(x_1, x_2)$$

3.4.3 Quasiconvex Optimization

$$\begin{aligned} \text{minimize } & f_0(x) \\ \text{subject to } & f_i(x) \le 0 \\ & Ax = b \end{aligned}$$

with $f_0 : \mathbb{R}^n \to \mathbb{R}$ **quasiconvex**, $f_1, \ldots, f_m$ **convex**.

Feature: **can** have locally optimal points that are not (globally) optimal.

Recall the definition of quasiconvex functions, we have that $f_0(x) \le t$ is t-sublevel set of f_0, and its domain is convex set. Having t-sublevel set if f_0 denoted as 0-sublevel set of ϕ_t, which means:

$$f_0(x) \le t \iff \phi_t(x) \le 0$$

and for **each** fixed t, $\phi_t(x)$ is convex in x.

We define the family of function ϕ_t such that:

- $\phi_t(x)$ is convex in x for fixed t
- t-sublevel set if f_0 is 0-sublevel set of ϕ_t

A quasiconvex optimization problem could be solved via **convex feasibility** problem**s** (recall that convex feasibility problems are convex optimization problems as well): [9]

$$\phi_t(x) \le 0, \ f_0(x) \le 0, \ Ax = b$$

for any fixed t, if it is **feasible**, then $p^* \le t$, if infeasible, on the other hand, $p^* \ge t$. Using binary search method, and set the allowed epsilon (difference between upper, lower bounds) being ϵ, requires $\lceil \log_2(\frac{u-l}{\epsilon}) \rceil$ iterations to converge, where u and l are the upper and lower bound at the very beginning. This process is called **<u>bisection</u>**.

[9] Recall that there's code implementation in homework for this sort of problems.

3.4.4 Linear Optimization (LP)

Every convex function can be well approximated by a piecewise-linear function.

A **linear program (LP)** could have:

- none
- one
- infinite Many

solutions.

Linear programs with **fewer** constraints than variables are **NOT** guaranteed to be feasible.

Linear Program (LP)

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

it is convex problem with **affine objective and constraint** functions.

Its **feasible set** is always **polyhedron**.

In fact, d could be removed without influencing the solutions. x^* is always at an **angle** of the feasible set, and $-c$ points outward from x^*.

LP could be solved extremely fast, with total reliability.

Examples:

- piecewise linear minimization $\Longrightarrow$ LP:

  $$\text{minimize } \max_{i=1,2,\ldots m} (a_i^T x + b_i)$$

  is equivalent with,

  $$\begin{aligned} \text{minimize } & t \\ \text{subject to } & a_i^T x + b_i \leq t \ (i = 1, 2, \ldots, m) \end{aligned}$$

- **Chebyshev center** of a polyhedron:

  $$\mathcal{P} = \{x | a_i^T x \leq b_i, \ i = 1, 2, \ldots, m\}$$

  which is the center of the largest **inscribed** ball: [10]

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

[10] It means that the ball is the largest ball **inside** the shape of the set.

and we have:

$$\begin{aligned} & a_i x \le b_i \ (\forall x \in \mathcal{B}) \\ \iff & \sup\{a_i^T(x_c+u) \mid \|u\|_2 \le r\} \\ & = a_i^T x_c + r\|a_i\|_2 \\ & \le b_i \end{aligned}$$

thus this whole problem could be solved by solving the LP:

$$\begin{aligned} & \text{minimize } r \\ & \text{subject to } a_i^T x_c + r\|a_i\|_2 \le b_i \ (i = 1, 2, \ldots, m) \end{aligned}$$

Linear-fractional Program

$$\begin{aligned} \text{minimize } & f_0(x) = \frac{c^T x + d}{e^T x + f} \\ \text{subject to } & Gx \preceq h \\ & Ax = b \end{aligned}$$

where

$$\textbf{dom}\, f_0 = \{x \mid e^T x + f > 0\}$$

It is a significant sort of generalization of LP. A kind of **quasiconvex** problem. It is also equal to LP with variables y, z:

$$\begin{aligned} \text{minimize } & c^T y + dz \\ \text{subject to } & Gy \preceq hz \\ & Ay = bz \\ & e^T y + fz = 1 \\ & z \ge 0 \end{aligned}$$

Generalized Linear-fractional Program

$$\begin{aligned} \text{minimize } & f_0(x) = \max_{i=1,2,\ldots m} \frac{c_i^T x + d_i}{e_i^T x + f_i} \\ \text{subject to } & Gx \preceq h \\ & Ax = b \end{aligned}$$

where

$$\textbf{dom}\, f_0 = \{x \mid e_i^T x + f_i > 0 \ (i = 1, 2, \ldots, r)\}$$

is also a quasiconvex problem solvable by bisection.

Robust Linear Programming is described in **QP** (3.4.5) part.

3.4.5 Quadratic Optimization (QP, SOCP, QCQP)

Non-affine problems.

Quadratic Program (QP)

$$\begin{aligned} \text{minimize } & \frac{1}{2}x^T P x + q^T x + r \\ \text{subject to } & Gx \preceq h \\ & Ax = b \end{aligned}$$

where $P \in \mathbb{S}^n_+$, **thus** objective function is convex quadratic.

Note that $\frac{1}{2}x^T P x + q^T x + r$ is guaranteed convex only when $P \in \mathbb{S}^n_+$.

In the polyhedron feasible set, the x^* always lies on an **edge**, with $-\nabla f(x^*)$ be a norm of the corresponding edge pointing outward of the feasible set. Level sets (α-sublevel sets) are ellipsoids.

Examples:

- **Least-Squares**:

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

  with:

  - analytical solution $x^* = A^\dagger b$ ($A^\dagger$ is the pseudo-inverse);
  - optional linear constraints $l \preceq x \preceq u$.

- **Linear Program (LP) with Random Cost**:

$$\begin{aligned} \text{minimize } & \bar{c}^T x + \gamma x^T \Sigma x \\ & = \mathbb{E} c^T x + \gamma \mathbf{var}(C^T x) \\ \text{subject to } & Gx \preceq h \\ & Ax = b \end{aligned}$$

  with:

  - random vector c has mean $\bar{c}$ and covariance Σ (hence $c^T x$ has mean $\bar{c}^T x$ and covariance $x^T \Sigma x$)
  - aversion parameter $\gamma > 0$, controls the trade-off between the expected cost and risk (variation)

Quadratically Constrained Quadratic Program (QCQP)

$$\begin{aligned} \text{minimize } & \frac{1}{2}x^T P_0 x + q_0^T x + r \\ \text{subject to } & \frac{1}{2}x^T P_i x + q_i^T x + r \ (i = 1, 2, \ldots m) \\ & Ax = b \end{aligned}$$

where:

- $P_i \in \mathbb{S}^n_+$, thus objective and constraints are convex quadratic;
- If $P_1, P_2, \ldots, P_m \in \mathbb{S}^n_{++}$, then the feasible region is **intersection** of m **ellipsoids** and an affine set.

Note that, $LP \subseteq QP \subseteq QCQP \subseteq SOCP$.

Second-order Cone Programming (SOCP)

$$\begin{aligned} &\text{minimize } f^T x \\ &\text{subject to } \|A_i x + b_i\|_2 \leq c_i^T x + d_i \ (i = 1, 2, \ldots m) \\ &\qquad\qquad Fx = g \end{aligned}$$

where:

- $A_i \in \mathbb{R}^{n_i \times n}$, $F \in \mathbb{R}^{p \times n}$;
- $\|A_i x + b_i\|_2 \leq c_i^T x + d_i$ are called **second-order cone (SOC) constraints**;
- $(A_i x + b_i, c_i^T x + d_i)$ is in **a** second-order cone in $\mathbb{R}^{n_i+1}$
- **If $n_i = 0$, it reduces to LP, if $c_i = 0$, reduces to QCQP;**
- $LP \subseteq QP \subseteq QCQP \subseteq SOCP$.

Robust Linear Programming

$$\begin{aligned} &\text{minimize } c^T x + d \\ &\text{subject to } a_i x \leq b_i, \quad i = 1, 2, \ldots, m \end{aligned}$$

How to handle uncertainty in c, a_i, b_i? Take a_i as an example. Two common approaches:

- **deterministic model**: constraints must hold for all $\forall a_i \in \mathcal{E}_i$:

  $$\begin{aligned} &\text{minimize } c^T x + d \\ &\text{subject to } a_i x \leq b_i, \ \forall a_i \in \mathcal{E}_i, \quad i = 1, 2, \ldots, m \end{aligned}$$

  a.k.a. "worst-case" model, to guarantee that it works in the worst cases.

  Could be done via **SOCP**:

  $$\mathcal{E}_i = \{\overline{a}_i + P_i u \mid \|u\|_2 \leq 1\}$$

  where $\overline{a}_i \in \mathbb{R}^n$ and $P_i \in \mathbb{R}^{n \times n}$, then,

  $$\sup_{\|u\|_2 \leq 1} (\overline{a}_i + P_i u)^T x = \overline{a}_i x + \|P_i^T x\|_2$$

  thus the original problem is equivalent with:

  $$\begin{aligned} &\text{minimize } c^T x + d \\ &\text{subject to } \overline{a}_i x + \|P_i^T x\|_2 \leq b_i, \quad i = 1, 2, \ldots, m \end{aligned}$$

  And, in this case, the term $\|P_i^T x\|_2$ could be treated as the amount of **margin** to leave (and we could get rid of it in some cases this way).

- **stochastic model**: for random variable a_i, constraints hold with probability η:

$$\begin{aligned} &\text{minimize } c^T x + d \\ &\text{subject to } \textbf{prob}(a_i x \le b_i) \ge \eta, \quad i = 1, 2, \ldots, m \end{aligned}$$

  It is suitable if you are pretty sure about the distribution.

  Could be solved via **SOCP** as well:

  Assuming a_i's distribution be Gaussian random variable with mean $\overline{a}_i$ and covariance Σ_i:

$$a_i \sim \mathcal{N}(\overline{a}_i, \Sigma_i)$$

  hence,

$$\textbf{prob}(a_i^T x \le b_i) = \Phi\Big(\frac{b_i - \overline{a}_i^T x}{\|\Sigma_i^{\frac{1}{2}} x\|_2}\Big)$$

  where $\Phi(x)$, the CDF (Cumulative Distribution Function) of $\mathcal{N}(0,1)$ is:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt$$

  thus when $\eta \ge \frac{1}{2}$:

$$\begin{aligned} &\text{minimize } c^T x + d \\ &\text{subject to } \overline{a}_i^T + \Phi^{-1}(\eta) \|\Sigma_i^{\frac{1}{2}} x\|_2 \le b_i, \quad i = 1, 2, \ldots, m \end{aligned}$$

These models have meanings, they originally came from some practical problems in real life.

3.4.6 Geometric Programming (GP)

Generally speaking, it is not convex problem, but you may change the variables and make it convex.

Geometric Program (GP)

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

where:

- f_i is **posynomial**:

  A **posynomial** function $f(x)$ is:

$$f(x) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \ldots x_n^{a_{nk}}$$

- h_i is **monomial**:

  A **monomial** function $f(x)$ is:

$$f(x) = cx_1^{a_1} x_2^{a_2} \dots x_n^{a_n}, \qquad \mathbf{dom}\, f = \mathbb{R}^n_{++},\ c > 0,\ a_i \in \mathbb{R}$$

  Sometimes also called a "scaling law".

It is **obviously not convex**.

The **product**, **ratio**, **square root**, etc. of **monomial** functions is **monomial**; the **sum** or **square** of **posynomial** functions is **posynomial**.

But we could change it into convex form by defining:

- $y_i = \log x_i$
- take **logarithm** of cost and constraints

such that:

- **monomial** functions:

$$\begin{aligned} f(x) &= cx_1^{a_1} x_2^{a_2} \dots x_n^{a_n} \\ \iff \log f(x) &= \sum_{i=1}^{n} a_i \log x_i \\ &= a^T y + b,\ (x_i = e^{y_i}, b = \log c) \end{aligned}$$

- **posynomial** functions:

$$\begin{aligned} f(x) &= \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \dots x_n^{a_{nk}} \\ \iff \log f(x) &= \log \Big(\sum_{k=1}^{K} e^{a_k^T y + b_k} \Big),\ (x_i = e^{y_i}, b_i = \log c_i) \end{aligned}$$

and thus the **equivalent convex form** of **GP**:

$$\begin{aligned} \text{minimize }\ & \log \big(\sum_{k=1}^{K} e^{a_{0k}^T y + b_{0k}} \big) \\ \text{subject to }\ & \log \big(\sum_{k=1}^{K} e^{a_{ik}^T y + b_{ik}} \big) \le 0,\ i = 1, 2, \dots, m \\ & Gy + d = 0 \end{aligned}$$

An GP example: the design of the cantilever beam. (Multiple segments, from the wall to the end, becoming smaller.) See slides or textbook for details.

Forming a problem as a standard GP will already enable libraries such as the cvx (or cvxpy) automatically handle the rest for you.

Note that two posynomial functions: f, g, for positive variable z, that $f(z) \leq g(z)$ **doesn't** make a posynomial constraint, and couldn't be automatically handled by library.

An **arithmetic mean** of the entries of a variable **divided by** the **geometric mean**, is **posynomial**.

3.4.7 Generalized Inequality Constraints

Convex Problems with Generalized Inequality Constraints:

$$\begin{aligned} \text{minimize } & f_0(x) \\ \text{subject to } & f_i(x) \preceq_{K_i} 0, \; i = 1, 2, \ldots, m \\ & Ax = b \end{aligned}$$

where:

- $f_0 : \mathbb{R}^n \to \mathbb{R}$ is convex;
- $f_i : \mathbb{R}^n \to \mathbb{R}^{k_i}$ is K_i convex, with respect to **proper** cone K_i.

It has the **same properties** as standard convex problems. For example, convex feasible set, and the local optimum is global optimum.

Examples:

- **conic form problem**: a special case with affine cost and affine constraints,

$$\begin{aligned} \text{minimize } & c^T x \\ \text{subject to } & Fx + g \preceq_K 0 \\ & Ax = b \end{aligned}$$

Conic form problems are generalizations of linear programs (LP).

3.4.8 Semidefinite Program (SDP)

$$\begin{aligned} \text{minimize } & c^T x \\ \text{subject to } & x_1 F_1 + x_2 F_2 + \cdots + x_n F_n + G \preceq 0 \\ & Ax = b \end{aligned}$$

where:

- $F_i, G \in \mathbb{S}^k$

- the inequality constraint is called **linear matrix inequality (LMI)**
- it includes problems with **multiple** LMI constraints, since the sum of two LMI is also LMI:

$$\because x_1F_{11} + x_2F_{12} + \cdots + x_nF_{1n} + G_1 \preceq 0$$
$$x_1F_{21} + x_2F_{22} + \cdots + x_nF_{2n} + G_2 \preceq 0$$
$$\therefore x_1 \begin{bmatrix} F_{11} & 0 \\ 0 & F_{21} \end{bmatrix} + x_2 \begin{bmatrix} F_{12} & 0 \\ 0 & F_{22} \end{bmatrix} + \cdots + x_n \begin{bmatrix} F_{1n} & 0 \\ 0 & F_{2n} \end{bmatrix} + \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix} \preceq 0$$

It generalizes **most** problems so far (**except GP**).

From LP to SDP:

LP:

$$\text{minimize } c^T x$$
$$\text{subject to } Ax \preceq b$$

is equivalent with SDP:

$$\text{minimize } c^T x$$
$$\text{subject to } \mathbf{diag}(Ax - b) \preceq 0$$

From SOCP to SDP:

SOCP:

$$\text{minimize } f^T x$$
$$\text{subject to } \|A_i x + b_i\|_2 \leq c_i^T x + d_i, \ i = 1, 2, \ldots, m$$

is equivalent with SDP:

$$\text{minimize } f^T x$$
$$\text{subject to } \begin{bmatrix} (c_i^T x + d_i)I & A_i x + b_i \\ (A_i x + b_i)^T & c_i^T x + d_i \end{bmatrix} \succeq 0, \ i = 1, 2, \ldots, m$$

From Eigenvalue Minimization to SDP:

$$\text{minimize } \lambda_{max}(A(x))$$

where $A(x) = A_0 + x_1A_1 + \cdots + x_nA_n$, with given $A_i \in \mathbb{S}^k$.

Since:

$$\lambda_{max}(A) \leq t \iff A \preceq tI$$

we have that, it is equivalent with the SDP:

$$\text{minimize } t$$
$$\text{subject to } A(x) \preceq tI$$

Matrix Norm Minimization to SDP:

$$\text{minimize } \|A(x)\|_2 = (\lambda_{max}(A(x)^T A(x)))^{\frac{1}{2}}$$

where $A(x) = A_0 + x_1 A_1 + \cdots + x_n A_n$, with given $A_i \in \mathbb{R}^{p\times q}$, and $t \in \mathbb{R}$, $x \in \mathbb{R}^n$.
Since:

$$\begin{aligned} \|A\|_2 \le t &\iff A^T A \preceq t^2 I \ (t \ge 0) \\ &\iff \begin{bmatrix} tI & A \\ A^T & tI \end{bmatrix} \succeq 0 \end{aligned}$$

we have that, it is equivalent with the SDP:

$$\begin{aligned} &\text{minimize } t \\ &\text{subject to } \begin{bmatrix} tI & A(x) \\ A(x)^T & tI \end{bmatrix} \succeq 0 \end{aligned}$$

3.4.9 Vector Optimization

The final generalization of optimization problem. A bit complicated.

General Vector Optimization Problem:

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

where $f_0(x) : \mathbb{R}^n \to \mathbb{R}^q$ is minimized with respect to proper cone $K \in \mathbb{R}^q$.

Convex Vector Optimization Problem:

$$\begin{aligned} \text{minimize } & f_0(x) \\ \text{subject to } & f_i(x) \le 0, \ i = 1, 2, \ldots, m \\ & Ax = b \end{aligned}$$

where $f_0(x) : \mathbb{R}^n \to \mathbb{R}^q$ is minimized with respect to proper cone $K \in \mathbb{R}^q$; f_0 is K-convex, and $f_1, \ldots f_m$ are convex.

Optimal and Pareto Optimal Points:

Define the set of achievable objective values:

$$\mathcal{O} = \{f_0(x) | x \text{ is feasible}\}$$

then a **feasible** x is:

- **optimal**, if $f_0(x)$ is **the** minimum value (x^*) of $\mathcal{O}$, where minimum means that:

$$y \text{ is feasible} \Longrightarrow f_0(x^*) \preceq f_0(y)$$

- **Pareto optimal**, if $f_0(x)$ is **a** minimal value (x^{po}) of $\mathcal{O}$, where minimal means that:

$$y \text{ is feasible}, f_0(y) \preceq f_0(x^{po}) \Longrightarrow f_0(x^{po}) = f_0(y)$$

An **example** of the optimal and Pareto optimal points:
Multicriterion Optimization:

$$f_0(x) = (F_1(x), F_2(x), \dots, F_q(x))$$

with q different objectives, $F_1, \dots F_q$, and $K = \mathbb{R}^q_+$.
Examples of the **multicriterion optimization**:

- **Regularized Least-Squares**:

  $$\text{minimize (w.r.t. } \mathbb{R}^2_+) \ (\|Ax - b\|_2^2, \|x\|_2^2)$$

  we could draw a curve of $F_2 \sim F_1$, and find the Pareto optimal points to solve it.

- **Risk Return Trade-Off in Portfolio Optimization**:

  $$\begin{aligned} \text{minimize (w.r.t. } \mathbb{R}^2_+) \ & (-\overline{p}^T x, x^T \Sigma x) \\ \text{subject to } & \mathbb{1}^T x = 1, \\ & x \succeq 0 \end{aligned}$$

  it is, minimizing the vector of risk of return and return variance. $x, p \in \mathbb{R}^n$, x is the investment portfolio, p is the asset price changes.

How to find Pareto optimal points? By scalarization.
Scalarization:

1. Choose $\lambda \succ_{K^*} 0$, where K^* is the dual cone;

2. Solve the optimization problem, called **scalar problem**:

   $$\begin{aligned} \text{minimize (w.r.t. } K) \ & \lambda^T f_0(x) \\ \text{subject to } & f_i(x) \leq 0, \ i = 1, 2, \dots m \\ & h_i(x) = 0, \ i = 1, 2, \dots p \end{aligned}$$

3. By varying λ, we could find almost all **Pareto optimal points** in the original problem. They are **optimal points** in the scalar problem.

4. If x_i is Pareto optimal, then the corresponding λ_i should be the inward norm of the tangent line at point $f_0(x_i)$.

5. For **multicriterion problems**:

   $$\lambda^T f_0(x) = \lambda_1 F_1(x) + \cdots + \lambda_q F_q(x)$$

   for the case where $f_0(x) = (\|Ax - b\|_2^2, \|x\|_2^2)$, or *the risk return trade-off* case, we could have $\lambda = (1, \gamma)$ and thus we have a form similar with what we frequently observe in ML, lost with regularization.

Suppose x minimizes $f(x) + g(x)$. Then x is Pareto optimal for the bi-criterion optimization problem minimize $(f(x), g(x))$.

- $f_0(x_1, x_2) = x_1 + x_2$ solution is the minimal point of the intersection of the feasible set and $x_1 + x_2 = b$;
- $f_0(x_1, x_2) = \max\{x_1, x_2\}$ solution is the minimal point of the intersection of the feasible set and $x_1 = x_2$ (consider the 3D shape of $f_0(x_1, x_2) \sim x_1 \times x_2$);
- $f_0(x_1, x_2) = -x_1 - x_2$ solution is the minimal point of the intersection of the feasible set and $-x_1 - x_2 = b$;
- $f_0(x_1, x_2) = x_1^2 + 9x_2^2$ solution is equivalent with considering minimizing $(x_1^2, 9x_2^2)$ and find the optimal point, that is equivalent with finding the optimal point for minimizing $(x_1, 3x_2)$;

3.5 Duality

It is an organized way of forming highly nontrivial bounds on convex optimization problems. It works even for hard problems that are not necessarily convex (but the functions, the problems involved must be convex to make it work).

3.5.1 Lagrangian

Standard form problem that is **not necessarily** convex:

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

with $\mathbf{dom}\, f = \mathcal{D}$, $x \in \mathbb{R}^n$, optimal value denoted as p^*.
Its **Lagrangian**, denoted as $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$, 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, with λ and ν being the Lagrange multiplier associated with the constraints.

3.5.2 Lagrange dual function

Lagrange dual function is denoted as $g : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$,

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

g is always **concave**, and could be $-\infty$ for some λ, ν.

There's an important property, called the **lower-bound property**:

If $\lambda \succeq 0$, then $g(\lambda, \nu) \leq p^*$.

its proof goes: since x is feasible and $\lambda \succeq 0$:

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

minimizing all feasible $\tilde{x}$ gives $p^* \geq g(\lambda, \nu)$. It holds for all λ and all ν.

For **differentiable** L, how we naturally **calculate** g is to calculate x that makes $\nabla_x L(x, \lambda, \nu) = 0$, and plug it in L to obtain g.

However, if L is **unbounded** in x (e.g. affine), in the form of:

$$L(x, \lambda, \nu) = L_1(\lambda, \nu) + L_2(\lambda, \nu) L_3(x)$$

where L_3 entirely relies on x, it is like x or x^2, an *informal* expression here, then we have:

$$g(\lambda, \nu) = \begin{cases} L_1(\lambda, \nu) & L_2(\lambda, \nu) = 0 \\ -\infty & \text{otherwise} \end{cases}$$

then we have g **concave on domain** $\{(\lambda, \nu) | L_2(\lambda, \nu) = 0\}$, and that $p^* \geq L_1(\lambda, \nu)$ if $L_2(\lambda, \nu) = 0$.

Or sometimes the condition is different, e.g. $L_2 \geq 0$, when $L_3 = x^3$.

3.5.3 Lagrange dual and Conjugate Function

Original problem:

$$\begin{aligned} \text{minimize } & f_0(x) \\ \text{subject to } & Ax \preceq b \\ & Cx = d \end{aligned}$$

Then the dual function is:

$$\begin{aligned} g(\lambda, \nu) &= \inf_{x \in \mathbf{dom}\, f_0} \left(f_0(x) + (A^T\lambda + C^T\nu)^T x - (b^T\lambda + d^T\nu) \right) \\ &= -f_0^*(-A^T\lambda - C^T\nu) - (b^T\lambda + d^T\nu) \end{aligned}$$

where, as defined before, the conjugate function,

$$f^*(y) = \sup_{x \in \mathbf{dom}\, f} (y^T x - f(x))$$

It simplifies the derivation of dual if f_0 is known.

3.5.4 (Lagrange) dual problem

It is finding the best (tight) lower bound.

$$\begin{aligned} &\text{maximize } g(\lambda, \nu) \\ &\text{subject to } \lambda \succeq 0 \end{aligned}$$

this is obviously a convex optimization problem, where the optimal value is d^*.

(λ, ν) is feasible if: (1) $\lambda \succeq 0$, and (2) $(\lambda, \nu) \in \mathbf{dom}\, g$. The dual problem is often simplified by making the implicit constraint of $(\lambda, \nu) \in \mathbf{dom}\, g$ explicit.

The dual function provides a lower bound on the optimal value for both **convex optimization problems** and **non-convex problems**.

3.5.5 Weak and strong duality

That is to say, if the **primal** (original) and the **dual** problem have the same optimal values.

Weak Duality: $d^* \leq p^*$

- Always holds, for both convex and nonconvex problems;
- d^* is not always computable, although in most cases it is);

Strong Duality: $d^* = p^*$

- Does not hold in general;
- **Usually** holds for convex problems;
- How to decide if the strong duality holds? By using conditions that are called **constraint qualifications**.

3.5.6 Slater's Constraint Qualifications

It is **just one type of** constraint qualifications. Other types exist as well.

Satisfying the constraint qualifications **doesn't necessarily** means that the **dual problem** is not unbounded.

Slater's Constraint Qualifications:

Strong duality will hold for a **convex** problem:

$$\begin{aligned} &\text{maximize } f_0(x) \\ &\text{subject to } f_i(x) \leq 0 \;\; i = 1, 2, \ldots, m \\ &\qquad Ax = b \end{aligned}$$

if it is **strictly feasible**, in other words,

$$\exists x \in \mathbf{int}\, \mathcal{D} : f_i(x) < 0 \;\; i = 1, 2, \ldots m, \;\; Ax = b$$

- The conditions also guarantee the dual optimum is attained if $p^* > -\infty$;
- Can be sharpened (strengthened):
  - $\textbf{int}\,\mathcal{D} \Longrightarrow \textbf{relint}\mathcal{D}$ (interior relative to affine hull);
  - linear inequalities **do not** need to hold with strict inequality.

It always apply to LP, QP [11]

Strong duality may occur in cases where convexity isn't satisfied.

When this occurs, it means that the **original problem** could be solved by **convex optimization**.

Every problem involving **two quadratic functions** (e.g. f_0 being quadratic, and subject to a quadratic function) can be solved exactly, **convex or not**. Because, for all those problems, **the dual is an SDP**. This guarantees the zero duality gap (strong duality).

For example,

$$\begin{aligned} &\text{minimizing } x^T P x \\ &\text{subject to } \|x\|_2 = 1 \end{aligned}$$

It is not convex but it is solvable by convex optimization. So as:

$$\begin{aligned} &\text{minimizing } x^T P x + 2b^T x \\ &\text{subject to } x^T x \leq 1 \end{aligned}$$

etc. We say they are not convex problem because, note that P can be **anything**, not guaranteed symmetric.

In details,

$$\begin{aligned} &\text{minimizing } x^T P x + 2b^T x \\ &\text{subject to } x^T x \leq 1 \end{aligned}$$

has dual function:

$$\begin{aligned} \text{maximize } & -b^T(A+\lambda I)\dagger b - \lambda \\ \text{subject to } & A + \lambda I \succeq 0 \\ & b \in \mathcal{R}(A+\lambda I) \end{aligned}$$

and this dual function has an **equivalent SDP**:

$$\begin{aligned} \text{maximize } & -t - \lambda \\ \text{subject to } & \begin{bmatrix} A+\lambda I & b \\ b^T & t \end{bmatrix} \succeq 0 \end{aligned}$$

Applicational examples include PCA and SVD.

[11] For proof see the textbook. Or we can walk through the process of writing down their dual problem from raw.

3.5.7 Geometric interpretation

The duality gap somehow represents the "best line" that is tangent to the set $\mathcal{G} = \{(f_1(x), f_0(x))|x \in \mathcal{D}\}$. An illustration is in the lecture note.

It is equivalent with understanding it with epigraph:

$$\mathcal{A} = \{(u,t)|f_1(x) \leq u, f_0(x) \leq t \text{ for some } x \in \mathcal{D}\}$$

Remember that, when the feasible set of an optimization problem is not convex, then the duality gap could be nonzero or zero.

3.5.8 Optimality Conditions: Complementary Slackness

$$\text{Assume: } \begin{cases} \text{strong duality holds} \\ x^* \text{ is primal optimal} \\ (\lambda^*, \nu^*) \text{ is dual optimal} \end{cases} \Longrightarrow g(\lambda^*, \nu^*) = \inf_x L(x, \lambda^*, \nu^*)$$

$$\begin{aligned} &= \inf_x \Big(f_0(x) + \sum_{i=1}^m \lambda_i^* f_i(x) + \sum_{i=1}^p \nu_i^* h_i(x) \Big) \\ &\leq f_0(x^*) + \sum_{i=1}^m \lambda_i f_i(x^*) + \sum_{i=1}^p \nu_i h_i(x^*) \\ &\leq f_0(x^*) \end{aligned}$$

When $f_0(x^*) = g(\lambda^*, \nu^*)$, the two inequalities hold with equality, and thus:

- x^* minimizes $L(x, \lambda^*, \nu^*)$
- **Complementary slackness**: $\lambda_i^* f_i(x^*) = 0$ for $\forall i \in \{1, 2, \ldots, m\}$, therefore,

$$\begin{cases} \lambda_i^* > 0 \Longrightarrow f_i(x^*) = 0 \\ f_i(x^*) < 0 \Longrightarrow \lambda_i^* = 0 \end{cases}$$

> "if one of them holds with some slack, the other must hold without slack; you can't have slack on both."

3.5.9 Optimality Conditions: Karush-Kuhn-Tucker (KKT) Conditions

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

- **Primal Constraints**:

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

- **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):

$$\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + \sum_{i=1}^{p} \nu_i \nabla h_i(x) = 0$$

If strong duality holds and x, λ, ν are optimal, then KKT condition must be satisfied.

More details:

- 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 section 3.5.6, implies strong duality) is satisfied, x is optimal $\iff$ $\exists \lambda, \nu$ that satisfy KKT conditions.

3.5.10 Perturbation and sensitivity analysis

(unperturbed) optimization problem and its dual:

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \le 0 & (i = 1, 2, \dots, m) \\ & h_i(x) = 0 & (i = 1, 2, \dots, p) \end{array} \qquad \begin{array}{ll} \text{maximize} & g(\lambda, \nu) \\ \text{subject to} & \lambda \succeq 0 \end{array}$$

perturbed problem and its dual:

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \le u_i & (i = 1, 2, \dots, m) \\ & h_i(x) = v_i & (i = 1, 2, \dots, p) \end{array} \qquad \begin{array}{ll} \text{maximize} & g(\lambda, \nu) - u^T \lambda - v^T \nu \\ \text{subject to} & \lambda \succeq 0 \end{array}$$

We name the new optimal value $p^*(u, v)$, as a function of parameters u, v. And we can get some information from the unperturbed version that is helpful on it.

global sensitivity result:

$$\text{Assume for the unperturbed problem, } \begin{cases} \text{Strong duality holds} \\ \lambda^*, \nu^* \text{ are dual optimal} \end{cases}$$

$$\begin{aligned} \text{then } p^*(u, v) &\ge g(\lambda^*, \nu^*) - u^T \lambda^* - v^T \nu^* \\ &= p^*(0, 0) - u^T \lambda^* - v^T \nu^* \end{aligned}$$

where $u_i > 0$ means **loosen** constraint, $u_i < 0$ means **tighten** constraint.
We could gain some interpretations in respect with λ^*, ν^* influence on $p^*(u, v)$, under certain conditions of u, v (> 0 or < 0 in specific).
local sensitivity:
If, in addition, $p^*(u, v)$ is differentiable on u and on v, then we have:

$$\lambda_i^* = -\frac{\partial p^*(0,0)}{\partial u_i} \qquad \nu_i^* = -\frac{\partial p^*(0,0)}{\partial v_i}$$

there's proof in lecture note and in textbook.
These sensitivity provides us theory bases if we want to tighten or loosen some constraints.

3.5.11 Generalized inequalities

Problems with generalized inequalities:

$$\begin{aligned} &\text{minimizing } f_0(x) \\ &\text{subject to } f_i(x) \preceq_{K_i} 0 \ i = 1, 2, \ldots, m \\ &\qquad\qquad h_i(x) = 0 \ i = 1, 2, \ldots, p \end{aligned}$$

Changes:

- Lagrange multiplier λ_i is no longer $\mathbb{R}$, it becomes $\mathbb{R}^{k_i}$;
- Lagrangian $L : \mathbb{R}^n \times \mathbb{R}^{k_1} \times \mathbb{R}^{k_2} \times \cdots \times \mathbb{R}^{k_m} \times \mathbb{R}^p \to \mathbb{R}$ is defined as:

  $$L(x, \lambda_1, \lambda_2, \ldots, \lambda_m, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i^T f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

  it'll be exactly the same if we replace ν_i with ν_i^T, since $h_i(x)$ suppose to be zero.
- Dual function is defined as:

  $$g(\lambda_1, \lambda_2, \ldots, \lambda_m, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda_1, \lambda_2, \ldots, \lambda_m, \nu)$$

- Lower-bound property: If $\lambda_i \succeq_{K_i^*} 0$, then $g(\lambda_1, \lambda_2, \ldots, \lambda_m, \nu) \le p^*$;
-

An example: **SDP** (3.4.8):

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{subject to } x_1 F_1 + x_2 F_2 + \cdots + x_n F_n + G \preceq 0 \\ &\qquad\qquad Ax = b \end{aligned}$$

in this illustration simplified as

$$\begin{aligned} &\text{minimize } c^T x \\ &\text{subject to } x_1 F_1 + x_2 F_2 + \cdots + x_n F_n \preceq G \end{aligned}$$

- Lagrange multiplier: $Z \in \mathbb{S}^k$
- Lagrangian:

$$L(x, Z) = c^T x + \mathbf{tr}\left(Z(x_1 F_1 + x_2 F_2 + \cdots + x_n F_n - G)\right)$$

- dual function:

$$g(Z) = \inf_x L(x, Z) = \begin{cases} -\mathbf{tr}(GZ) & \mathbf{tr}(F_i Z) + c_i = 0,\ i = 1, \dots, n \\ -\infty & \text{otherwise} \end{cases}$$

- the dual SDP:

$$\begin{array}{ll} \text{minimize} & -\mathbf{tr}(GZ) \\ \text{subject to} & \mathbf{tr}(F_i Z) + c_i = 0,\ i = 1, \dots, n \\ & Z \succeq 0 \end{array}$$

- $p^* = d^*$ if primal SDP strictly feasible (exists "inside" feasible x)

3.6 Approximation and fitting

3.6.1 Norm approximation

$$\text{minimize } \|Ax - b\|$$

where $A \in \mathbb{R}^{m \times n}$, $m \geq n$. This is a norm on $\mathbb{R}^m$.

There are multiple ways of interpreting the solution of this problem. See lecture notes. (e.g. geometric, estimation, optimization problem.)

- least-square approximation:

$$\text{minimize } \|Ax - b\|_2$$

  the approximate solution must satisfies:

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

  The following conditions are equivalent:

  1. $Ax = b$ has a unique least-square solution
  2. The columns of A are linearly independent
  3. $A^T A$ is invertible
  4. $\mathbf{rank}\, A = n$

  and this condition leads to $x^* = (A^T A)^{-1} A^T b$.

- Chebyshev approximation:

$$\text{minimize } \|Ax - b\|_\infty$$

can be solved as an LP:

$$\begin{aligned} &\text{minimize } t \\ &\text{subject to } -t\mathbb{1} \preceq Ax - b \preceq t\mathbb{1} \end{aligned}$$

- sum of absolute residuals approximation:

$$\text{minimize } \|Ax - b\|_1$$

can be solved as an LP:

$$\begin{aligned} &\text{minimize } \mathbb{1}^T y \\ &\text{subject to } -y \preceq Ax - b \preceq y\mathbb{1} \end{aligned}$$

Could refer to `https://textbooks.math.gatech.edu/ila/least-squares.html`.

Penalty function approximation

$$\begin{aligned} &\text{minimize } \phi(r_1) + \cdots + \phi(r_m) \\ &\text{subject to } r = Ax - b \end{aligned}$$

where $A \in \mathbb{R}^{m\times n}$ and $\phi : \mathbb{R} \to \mathbb{R}$ is a convex penalty function.
Valid examples of ϕ:

- quadratic: $\phi(u) = u^2$
- deadzone-linear with width width a: $\phi(u) = \max\{0, |u| - a\}$
- log-barrier with limit a:

$$\phi(u) = \begin{cases} -a^2 \log(1 - (u/a)^2) & |u| < a \\ \infty & \text{otherwise} \end{cases}$$

  This kind of extended functions are also (**strictly**) convex.

- Huber penalty function with parameter M:

$$\phi(u) = \begin{cases} u^2 & |u| \le M \\ M(2|u| - M) & |u| > M \end{cases}$$

  it is a robust estimator that works unbelievably well, it allows a few outliers.

3.6.2 Least-norm problems

$$\begin{aligned} &\text{minimize } \|x\| \\ &\text{subject to } Ax = b \end{aligned}$$

Interpretations of the solutions could be geometric, estimation (x is the smallest variable satisfies $Ax = b$), design (x are design variables).

Examples:

- least-square approximation:

$$\begin{aligned} &\text{minimize } \|x\|_2 \\ &\text{subject to } Ax = b \end{aligned}$$

  the approximate solution must satisfies:

$$2x + A^T\nu = 0, \qquad Ax = b$$

- minimum sum of absolute values:

$$\begin{aligned} &\text{minimize } \|x\|_1 \\ &\text{subject to } Ax = b \end{aligned}$$

  can be solved as an LP:

$$\begin{aligned} &\text{minimize } \mathbb{1}^T y \\ &\text{subject to } -y \preceq x \preceq y, \\ &\qquad\qquad\quad Ax = b \end{aligned}$$

  It is expected that **most** components of the x^* will turn out to be **zero**.

  By the way, if we have $\|\cdot\|_\infty$:

$$\begin{aligned} &\text{minimize } \|x\|_\infty \\ &\text{subject to } Ax = b \end{aligned}$$

  then we could expect x^* having **most** components being $\pm\|x\|_\infty$.

- extension: least-penalty problem

$$\begin{aligned} &\text{minimize } \phi(x_1) + \cdots + \phi(x_n) \\ &\text{subject to } Ax = b \end{aligned}$$

  where $\phi : \mathbb{R} \to \mathbb{R}$ is a convex penalty function.

3.6.3 Regularized approximation

The parent of the previous two approximations. More general.

$$\text{minimize (w.r.t. } \mathbb{R}^2_+) \ (\|Ax - b\|, \|x\|)$$

$A \in \mathbb{R}^{m \times n}$, the norms on $\mathbb{R}^m$ and that on $\mathbb{R}^n$ could be different.

Interpretations could also vary from estimation, optimal design, to robust estimation. Generally, the idea is: find a **good** estimation of $Ax \approx b$ with **small** x.

Generally, a **smaller** x is **less sensitive for errors** in A.

Besides, small x can represent physical quantities that are **easier to realize** in real systems.

Examples:

- extension: least-penalty problem

  $$\text{minimize } \|Ax - b\| + \gamma\|x\|$$

  solution for $\gamma > 0$ traces out the optimal trade-off curve.

- **Other** common methods such as minimize $\|Ax - b\|^2 + \delta\|x\|^2$ with $\delta > 0$. e.g. **Tikhonov regularization (ridge regression)**:

  $$\begin{aligned} &\text{minimize } \|Ax - b\|_2^2 + \delta\|x\|_2^2 \ (\delta > 0) \\ &\implies \text{minimize } \left\| \begin{bmatrix} A \\ \sqrt{\sigma} I \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|_2^2 \\ &\implies x^* = (A^T A + \delta I)^{-1} A^T b \end{aligned}$$

- Optimal input design (see lecture notes, it is basically a convolutional kernel), signal reconstruction (see lecture notes, instead of having two norms, its $\|x\|$ is replaced by $\phi(x)$, a regularization function / smoothing signal).

3.6.4 Robust approximation

$$\text{minimize } \|Ax - b\|$$

with uncertain A.

General idea of two ways of solving it:

- **stochastic**: assume A is random, minimize $\mathbb{E}(\|Ax - b\|)$;

  An example:

  $$A = \overline{A} + U, \ U \text{ random}, \ \mathbb{E}U = 0, \ \mathbb{E}U^T U = P$$

$$
\begin{aligned}
\mathbb{E}(\|Ax - b\|_2^2) &= \mathbb{E}(\|\overline{A}x - b + Ux\|_2^2) \\
&= \|\overline{A}x - b\|_2^2 + \mathbb{E}(x^T U^T U x) \\
&= \|\overline{A}x - b\|_2^2 + x^T P x \\
&= \|\overline{A}x - b\|_2^2 + \|P^{1/2}x\|_2^2
\end{aligned}
$$

and the robust LS problem is transformed into an LS problem this way.

It is equivalent to a Tikhonov regularized nominal problem if the entries of A are i.i.d.

- **worst case**: minimize $\sup_A \|Ax - b\|$; refer to the lecture notes for some special-cases.

tractable only in some cases (certain distribution, certain norm, etc.).

The safe way of checking if the robust model works in practical, is to try some valid A and see if the result is approximately stable.

3.7 Statistical estimation

3.7.1 Maximum likelihood estimation (MLE)

It is under the category of parametric distribution estimation. basically, just maximize (over x) $\log p_x(y)$.

$$\ell(x) = \log p_x(y)$$

is called the log-likelihood function. $x \in C$ can be added easily by defining $\log p_x(y) = 0$ when $x \notin C$.

It is convex optimization problem if $\log p_x(y)$ is concave in x for fixed y.

Examples include, linear measurement with IID noise, with different distributions of noise (see lecture note); logistic regression.

3.7.2 Optimal detector design

Binary hypothesis testing: knowing that $x \in X$ is generated from one of two distributions, figure out which.

See lecture notes.

3.7.3 Experiment design

General, standard, ML routine.

See lecture notes.

3.8 Geometric problems

3.8.1 Extremal volume ellipsoids

Minimum volume ellipsoid around a set (C), also known as **Löwner – John** Ellipsoid, denoted as $\mathcal{E}$:

$$\mathcal{E} = \{v \mid \|Ax + b\|_2 \leq 1\}, \qquad A \in \mathbb{S}^n_{++}$$

volume of $\mathcal{E}$ is proportion to $\mathbf{det}\, A^{-1}$, the question of finding $\mathcal{E}$ is equivalent with:

$$\begin{array}{rl} \text{minimize (over } A,\, b) & \mathbf{log\,det}\, A^{-1} \\ \text{subject to} & \sup\limits_{v \in C} \|Ax + b\|_2 \leq 1 \end{array}$$

it is convex, but for general C it is hard to solve, because evaluating the constraints is hard. Solvable for **finite set**:

$$\begin{array}{rl} \text{minimize (over } A,\, b) & \mathbf{log\,det}\, A^{-1} \\ \text{subject to} & \|Ax_i + b\|_2 \leq 1, \;\; i = 1, 2, \ldots, m \end{array}$$

it gives a *Löwner – John* Ellipsoid for a polyhedron, the convex hull formed by all possible x points. (This is not obvious but it's true.)

Maximum volume inscribed (inside) ellipsoid of (C), denoted as $\mathcal{E}$:

$$\mathcal{E} = \{v \mid \|Bu + d\|_2 \leq 1\}, \qquad B \in \mathbb{S}^n_{++}$$

volume of $\mathcal{E}$ is proportion to $\mathbf{det}\, B$, the question of finding $\mathcal{E}$ is equivalent with:

$$\begin{array}{rl} \text{maximize} & \mathbf{log\,det}\, B \\ \text{subject to} & \sup\limits_{\|u\|_2 \leq 1} I_C(Bu + d) \leq 0 \end{array}$$

$$\left(I_C(x) = \begin{cases} 0 & x \in C \\ \infty & x \notin C \end{cases} \right)$$

it is convex, but for general C it is hard to solve, because evaluating the constraints is hard.

Solvable for polyhedron $\{x | a_i^T x \leq b_i, \; i = 1, 2, \ldots, m\}$

$$\begin{array}{rl} \text{maximize} & \mathbf{log\,det}\, B \\ \text{subject to} & \|Ba_i\|_2 + a_i^T d \leq b_i, \;\; i = 1, \ldots, m \end{array}$$

since we have $\sup_{\|u\|_2 \leq 1} a_i^T (Bu + d) = \|Ba_i\|_2 + a_i^T d$.

Efficiency, with $C \subseteq \mathbb{R}^n$ convex and bounded with nonempty interior:

Ellipsoid Type	$L\ddot{o}wner - John$ Ellipsoid	Maximum volume inscribed Ellipsoid
Original	Covers C	Lies inside C
Behavior	Shrunk by a factor n	Expanded by a factor n
Result	Lies inside C	Covers C

If C is symmetric, then n could be reduced to $\sqrt{n}$.

Any norm on set of $\mathbb{R}^n$ could be projected into the quadratic norm within the factor of $\frac{1}{4}$.

The smallest norm from infinite set $\mathcal{P}$ to a point a is always easy, while for the largest norm, Euclidean is not easy but ℓ_∞ is.

3.8.2 Centering

Definitions of center of C: Center of the Maximum volume inscribed Ellipsoid (MVE): called **Chebyshev center** when it is ball instead of ellipsoid. **Chebyshev center** could be found using LP.

MVE center is invariant under affine coordinate transforms.

The **analytic center** of convex inequalities and linear equations:

$$\begin{aligned} &f_i(x) \le 0,\ \ i = 1, 2, \dots, m \\ &Fx = g \end{aligned}$$

is defined as the optimal point of:

$$\begin{aligned} \text{minimize}\ \ & -\sum_{i=1}^{m} \log(-f_i(x)) \\ \text{subject to}\ & Fx = g \end{aligned}$$

is easier to compute than ellipsoid centers, and two convex optimization problems could have the **same** feasible set but **different centers**.

It is a function of the description of the set; not the feasible set itself.

3.8.3 Classification

This part is basically the idea that give rise to algorithms such as SVM. Very standard and nothing new.

Just why it works, how to estimate the values, soft margin (called slack in this case), hinge loss, non-linear SVM, etc. Focusing on why they are solvable, they could be transformed into convex optimization problems.

See the lecture note.

3.9 Unconstrained minimization

3.9.1 Terminology and assumptions

$$\text{minimize } f(x)$$

with no constraint, and f is convex, twice continuously differentiable (hence $\mathbf{dom}\, f$ is open), assuming that $p^* = \inf_x f(x)$ is attained and finite.

unconstrained minimization methods:

- produce sequence of points $x^{(k)} \in \mathbf{dom}\, f$, such that $f(x^{(k)}) \to p^*$;
- interpretation: this is to solve the optimality condition iteratively:

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

- **requirement** on $x^{(0)}$: $x^{(0)} \in \mathbf{dom}\, f$ and the sublevel set $S = \{x | f(x) \leq f(x^{(0)})\}$ is closed.
- it is hard to verify that a certain sublevel set is closed, but in certain conditions **all sublevel sets** are closed:
  1. equivalent to that $\mathbf{epi}\, f$ is closed
  2. true if $\mathbf{dom}\, f = \mathbb{R}^n$
  3. true if when x approaches the boundary, $f(x) \to \infty$

strong convexity and implications:

If $\exists m > 0$ such that, $\forall x \in S$:

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

then f **mush** be **strongly** convex in S.

Implications:

- for $x, y \in S$:

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

  hence, S is **bounded**.

- $p^* > -\infty$ and $\forall x \in S$:

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

  if m is known then this is a useful stopping criterion.

3.9.2 Gradient descent method

Descent Methods:

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

where $\Delta x^{(k)}$ is called step, and t the step size (learning rate).

From **convexity**, it implies that:

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

Line Search Types:

1. **Exact** Line Search:
$$t = \text{argmin}_{t>0} f(x + t\Delta x)$$

2. **Backtracking** Line Search (with parameters $\alpha \in (0, \frac{1}{2})$, $\beta \in (0, 1)$):
   Staring with $t = 1$, repeatedly having $t = \beta t$ at each iteration, until
$$f(x + t\Delta x) < f(x) + \alpha t \nabla f(x)^T \Delta x$$

In descent methods, the particular choice of line search does not matter so much, but the particular choice of search direction matters a lot.

Gradient Descent Method:

General descent method with $\Delta x = -\nabla f(x)$

- Usually stopping criteria $\|\nabla f(x)\|_2 \leq \epsilon$
- Very simple but **slow**, thus **not** frequently used in practice.

Starting at a point near the optimal point **doesn't necessarily means** it'll converge fast in gradient descent.

3.9.3 Steepest descent method

normalized steepest descent direction (at x, for norm $\|\cdot\|$):

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

interpretation: for small v,

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

Δx_{nsd} is the unit-norm step with the most (/ the most significant) negative-direction derivative.

unnormalized steepest descent direction

$$\Delta x_{sd} = \|\nabla f(x)\|_* \Delta x_{usd}$$

where $\|\cdot\|_*$ simply means the (whatever it is) norm.

And it satisfies:

$$\nabla f(x)^T \Delta x_{sd} = -\|\nabla f(x)\|_*^2$$

steepest descent method

General descent method with $\Delta x = \Delta x_{sd}$.

Examples:

$f(x)$	name	Δx_{sd}
$\|x\|_2$	Euclidean Norm	$-\nabla f(x)$
$\|x\|_P = (x^T P x)^{1/2}$	Quadratic Norm	$-P^{-1}\nabla f(x)$
$\|x\|_1$	ℓ_1-Norm	$-(\frac{\partial f(x)}{\partial x_i})e_i$

by the way in the ℓ_1-norm,

$$|\frac{\partial f(x)}{\partial x_i}| = \|\nabla f(x)\|_\infty$$

it basically means updating one entity at a time.

Choice of the matrix P (for norm $\|\cdot\|_P$) has strong effect on **the speed of converge**.

3.9.4 Newton's method gradient

The Newton's method comes from the idea of "doing steepest descent method, but let's change the P each iteration to the temporary best guess".

$$\Delta x_{nt} = -\frac{\nabla f(x)}{\nabla^2 f(x)}$$

Interpretations includes:

- $x + \Delta x_{nt}$ 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**.

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

3.9.5 Newton decrement

$$\lambda(x) = (\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x))^{1/2}$$

which is a **measurement of proximity** of x to x^*.

properties:

- equal to the **quadratic Hessian norm** of Δx_{nt}:

$$\lambda(x) = (\Delta x_{nt}^T \nabla^2 f(x) \Delta x_{nt})^{1/2}$$

- **directional derivative**:

$$\nabla f(x)^T \Delta x_{nt} = -\lambda(x)^2$$

- **affine invariant** (unlike $\|\nabla f(x)\|_2$)

$$\tilde{f}(y) = f(Tx),\ y^{(0)} = T^{-1}x^{(0)} \Longrightarrow y^{(k)} = T^{-1}x^{(k)}$$

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

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

3.9.6 Newton's method

1. **given** starting point $x \in \mathbf{dom}\, f$, tolerance $\epsilon > 0$
2. **repeat** the following steps:
   (a) Compute the Newton step and decrement:

$$\Delta x_{nt} = -\frac{\nabla f(x)}{\nabla^2 f(x)}$$
$$\lambda^2 = (\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x))$$

   (b) (**stopping criterion**) quit if $\frac{\lambda^2}{2} \leq \epsilon$
   (c) Line search: choose t by backtracking line search
   (d) Update $x = x + t\Delta x_{nt}$

3.9.7 Classical Convergence Analysis for Newton method

Assume:

- f strongly convex on S with constant m
- $\nabla^2 f$ is Lipschitz continuous on S, with constant $L > 0$

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

(L measures how well f could be approximated by a quadratic function; in other words, the third derivation is small)

Outline: there exist **constants** $\eta \in (0, \frac{m^2}{L})$, $\gamma > 0$, such that

$$\begin{cases} f(x^{(k+1)}) - f(x^{(k)}) \leq -\gamma & \|\nabla f(x)\|_2 \geq \eta \\ \frac{L}{2m^2}\|\nabla f(x^{(k+1)})\|_2 \leq \big(\frac{L}{2m^2}\|\nabla f(x^{(k)})\|_2\big)^2 & \|\nabla f(x)\|_2 < \eta \end{cases}$$

1. **damped Newton phase** ($\|\nabla f(x)\|_2 \geq \eta$):
   - most iterations require backtracking steps
   - function value decreases by at least γ
   - if $p^* > -\infty$, **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 $t = 1$
   - $\|\nabla f(x)\|_2$ converges to zero quadratically

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

Therefore, the number of iterations in total is bounded by

$$\frac{f(x^{(0)}) - p^*}{\gamma} + \log_2 \log_2(\frac{\epsilon_0}{\epsilon})$$

where γ, ϵ_0 are constants depending on m, L, $x^{(0)}$, but all of them are **usually unknown**. The second term is small (in terms of 6) and almost constant in practice.

It provides insights and explanations into the two phases, and why Newton method **converges super fast**.

When Newton's method is started from a point near the solution, it will converge very quickly. It is fast even if it's not close to the optimal at the beginning.

3.9.8 Newton Method and Cholesky factorization

Evaluate derivatives and solve Newton system (the main task each step):

$$H\Delta x = g$$

where $g = -\nabla f(x)$, the Hessian $H = \nabla^2 f(x)$.

could be implemented via **Cholesky factorization**:

$$H = LL^T \qquad \Delta x_{nt} = L^{-T}L^{-1}g \qquad \lambda(x) = \|L^{-1}g\|_2$$

and it costs $\frac{n^3}{3}$ flops for unstructured problem; much less if H is sparse, banded.

Newton's method is **seldom used** in machine learning practice, even though Newton's method does **work well on noisy data**, and common loss functions **are** self-concordant, is because:

- Hessian inversion may be a numerically expensive operation;
- it is generally not practical to form or store the Hessian in such problems, due to large problem size.

However, under **proper factorization**, it might be much less costly.

3.10 Equality constrained minimization

Stepping towards solving the generalized convex optimization problem, by adding constraints. Equality constraints are the easiest.

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } Ax = b \end{aligned}$$

where we assume that: f is a twice continuously differentiable convex function; $A \in \mathbb{R}^{p\times n}$ with $\textbf{rank}\, A = p$; p^* is finite and attained.

The optimality conditions (for x^* being optimal, iff), according to KKT (see section 3.5), is that: there exists ν^*:

$$\nabla f(x^*) + A^T\nu = 0 \qquad Ax^* = b$$

they come from the dual residual and the primal residual.

The equality-constrained quadratic minimization with $P \in \mathbb{S}^n_+$:

$$\begin{aligned} &\text{minimize } \frac{1}{2}x^T Px + q^T x + r \\ &\text{subject to } Ax = b \end{aligned}$$

optimality conditions are:

$$\begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ v^* \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix}$$

the **coefficient matrix** is also called the **KKT matrix**; this matrix is **nonsingular** iff:

$$Ax = 0, x \neq 0 \Longrightarrow x^T P x > 0$$

this nonsingular-condition is **equivalent** with

$$P + A^T A \succ 0$$

3.10.1 Eliminating equality constraints

$$\{x|Ax = b\} = \{Fz + \hat{x}|z \in \mathbb{R}^{n-p}\}$$

$\hat{x}$ is **any**, particular solution; $F \in \mathbb{R}^{n\times(n-p)}$; $\mathcal{R}(F) = \mathcal{N}(A)$, thus $\mathbf{rank}\, F = n - p$, $AF = 0$.

Therefore, the eliminated problem:

$$\text{minimize } f(Fz + \hat{x})$$

becomes an unconstrained problem of z. And once we have z^*:

$$x^* = fz^* + \hat{x}$$
$$\nu^* = -(AA^T)^{-1}A\nabla f(x^*)$$

3.10.2 Newton's method with equality constraints

Newton's step Δx_{nt} of f at feasible x is given by solution v of:

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} -\nabla f(x) \\ 0 \end{bmatrix}$$

Interpretations:

1. v solves second order approximation

$$\begin{aligned} &\text{minimize } \hat{f}(x+v) = f(x) + \nabla f(x)^T v + (1/2) v^T \nabla^2 f(x) v \\ &\text{subject to } A(x+v) = b \end{aligned}$$

2. it follows from linearizing optimality conditions

$$\nabla f(x+v) + A^T w \approx \nabla f(x) + \nabla^2 f(x) w + A^T w = 0 \qquad A(x+v) = b$$

Newton's decrement $\lambda(x)$:

$$\lambda(x) = (\Delta x_{nt}^T \nabla^2 f(x) \Delta x_{nt})^{1/2} = (-\nabla f(x)^T \Delta x_{nt})^{1/2}$$

with **properties**:

- gives an estimation:

$$f(x) - p^* \approx f(x) - \inf_{Ay=b} \hat{f}(y) = \frac{\lambda(x)^2}{2}$$

- directional derivative in Newton derivative:

$$\frac{d}{dt} f(x + t\Delta x_{nt})\Big|_{t=0} = -\lambda(x)^2$$

- **in general**,

$$\lambda(x) \neq (\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x))^{1/2}$$

  it is **not equivalent with** the value without constraints.

The Newton's method (the procedure) is **the same** with without constraints, and it is also affine invariant.

A **feasible** descent method: $x^{(k)}$ feasible and $f(x^{(k+1)}) < f(x^{(k)})$.

$$\text{minimize } \tilde{f}(z) = f(Fz + \hat{x})$$

where:

- $z \in \mathbb{R}^{n-p}$
- $A\hat{x} = b$, $\textbf{rank}\, F = n - p$ and $AF = 0$
- $x^{(0)} = Fz^{(0)} + \hat{x} \Longrightarrow x^{(k)} = Fz^{(k)} + \hat{x}$

Don't need separate convergence analysis. It is about the same with the original one.

3.10.3 Infeasible start Newton method

With infeasible x:

$$x \in \textbf{dom}\, f, \qquad Ax \neq b$$

we have that

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = - \begin{bmatrix} \nabla f(x) \\ Ax - b \end{bmatrix}$$

where still, v's solution is Δx_{nt}.

There are similar interpretations, for details see the lecture notes.

The infeasible start Newton method has different process than that of the feasible one, for details see the lecture notes. **Not a descent method**!

3.10.4 Implementation of the Newton Method

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} g \\ h \end{bmatrix}$$

solution methods:

- LDL^T factorization
- If H is **nonsingular**, we can do the elimination:

$$AH^{-1}A^Tw = h - AH^{-1}g \qquad Hv = -(g + A^Tw)$$

- If H is **singular**, we can do the elimination **for**:

$$\begin{bmatrix} H + A^TQA & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} g + A^TQh \\ h \end{bmatrix}$$

  with $Q \succeq 0$ for which $H + A^TQA \succ 0$.

Equality-constrained analytic centering:

- **primal problem**

$$\begin{aligned} \text{minimize } & -\sum_{i=1}^{n} \log x_i \\ \text{subject to } & Ax = b \end{aligned}$$

- **dual problem**

$$\text{maximize } -b^T\nu + \sum_{i=1}^{n} \log(A^T\nu)_i + n$$

Complexity per iteration of three methods is identical, each case end up solving:

$$ADA^Tw = h$$

with D positive diagonal.

1. block elimination to solve the KKT system:

$$\begin{bmatrix} \mathbf{diag}(x)^{-2} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ w \end{bmatrix} = \begin{bmatrix} \mathbf{diag}(x)^{-1}\mathbb{1} \\ 0 \end{bmatrix}$$

   reduces to solving

$$A\,\mathbf{diag}(x)^2A^Tw = b$$

2. solving Newton system:

$$A\,\mathbf{diag}(A^T\nu)^{-2}A^T\Delta\nu = -b + A\,\mathbf{diag}(A^T\nu)^{-1}\mathbb{1}$$

3. block elimination to solve the KKT system:

$$\begin{bmatrix} \mathbf{diag}(x)^{-2} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \nu \end{bmatrix} = \begin{bmatrix} \mathbf{diag}(x)^{-1}\mathbb{1} \\ Ax - b \end{bmatrix}$$

reduces to solving

$$A\,\mathbf{diag}(x)^2 A^T w = 2Ax - b$$

Example: network flow optimization (see lecture notes).

3.11 Interior-point methods

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

3.11.1 Inequality constrained minimization

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } f_i(x) \leq 0 \; i = 1, 2, \dots m \\ &\qquad\qquad Ax = b \end{aligned}$$

assume that:

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

3.11.2 Logarithmic barrier function and central path

First, **reformulate** the original problem via indicator function of $\mathbb{R}_-$ (outputs 0 if input ≤ 0, otherwise ∞) as I_-.

$$\begin{aligned} &\text{minimize } f_0(x) + \sum_{i=1}^{m} I_-(f_i(x)) \\ &\text{subject to } Ax = b \end{aligned}$$

Then, approximate via **logarithm barrier**:

$$\begin{aligned} &\text{minimize } f_0(x) - \frac{1}{t} \sum_{i=1}^{m} \log(f_i(x)) \\ &\text{subject to } Ax = b \end{aligned}$$

for $t > 0$, $-\frac{1}{t}\log(-u)$ is a **smooth approximation** of $I_-(u)$, and the approximation improves as $t \to \infty$.

Logarithm barrier function:

$$\phi(x) = -\sum_{i=1}^{m} \log(-f_i(x)),\ \ \mathbf{dom}\,\phi = \{x | f_i(x) < 0,\ i = 1, 2, \ldots, m\}$$

and we know that,

- it is **convex**
- it is **twice continuously differentiable**:

$$\nabla\phi(x) = \sum_{i=1}^{m} \frac{1}{-f_i(x)} \nabla f_i(x)$$
$$\nabla^2\phi(x) = \sum_{i=1}^{m} \frac{1}{f_i(x)^2} \nabla f_i(x) \nabla f_i(x)^T + \sum_{i=1}^{m} \frac{1}{-f_i(x)} \nabla^2 f_i(x)$$

Central Path:

For $t > 0$, define $x^*(t)$ as the solution of

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

assuming that $x^*(t)$ exists and is unique for each $t > 0$, then the **central path** is defined as

$$\{x^*(t) | t > 0\}$$

Example: for a LP problem ($f_0(x) = c^T x$), hyperplane $c^T x = c^T x^*(t)$ is tangent to level curve of ϕ through $x^*(t)$.

Dual points on central path

$x = x^*(t)$ if there exists a w such that

$$t\nabla f_0(x) + \sum_{i=1}^{m} \frac{1}{-f_i(x)} \nabla f_i(x) + A^T w = 0, \qquad Ax = b$$

therefore, $x^*(t)$ minimizes the Lagrangian:

$$L(x, \lambda^*(t), \nu^*(t)) = f_0(x) + \sum_{i=1}^{m} \lambda_i^*(t) f_i(x) + \nu^*(t)^T (Ax - b)$$

where we have:

$$\begin{cases} \lambda_i^*(t) = \frac{1}{-t f_i(x^*(t))} \\ \nu^*(t) = \frac{w}{t} \end{cases}$$

this **confirms** the intuitive idea that

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

proof:

$$\begin{aligned} p^* &\geq g(\lambda^*(t), \nu^*(t)) \\ &= L(x^*(t), \lambda^*(t), \nu^*(t)) \\ &= f_0(x^*(t)) - \frac{m}{t} \end{aligned}$$

$\frac{m}{t}$ is called a **duality gap**.

Interpretations via **KKT conditions**:

$x = x^*(t)$, $\lambda = \lambda^*(t)$, $\nu = \nu^*(t)$ satisfy:

1. **primal constraints**: $f_i(x) \leq 0,\ i = 1, 2, \dots m,\ Ax = b$
2. **dual constraints**: $\lambda \succeq 0$
3. **approximate complementary slackness** [12]:

$$-\lambda_i f_i(x) = \frac{1}{t},\ i = 1, 2, \dots, m$$

4. gradient of Lagrangian with respect to x vanishes:

$$\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + A^T \nu = 0$$

An example of central path is **force field interpretation**, it is an example from control theory, see the lecture notes for details.

3.11.3 Barrier method

A.k.a. "path-following", "SUMT (sequential unconstrained minimization technique)", etc.

The process:

1. **given** strictly feasible $x, t = t^{(0)} > 0, \mu > 1$, tolerance $\epsilon > 0$.
2. **repeat**:
   (a) Centering step: Compute $x^*(t)$ by minimizing $tf_0 + \phi$, subject to $Ax = b$;
   (b) Update: $x = x^*(t)$;
   (c) Stopping criterion: **quit** if $\frac{m}{t} < \epsilon$

[12] Difference with KKT is that here it was $\lambda_i f_i(x) = 0$ in the KKT.

(d) Increase t: $t = \mu t$

where we have:

- terminate when $f_0(x) - p^* \leq \epsilon$, follows from $f_0(x^*(t)) - p^* \leq \frac{m}{t}$
- centering is usually done using Newton's method; inner-outer iterations trade-off at μ

For the convergence analysis, see lecture notes or the textbook.

What happens in general: as you approach the target, the problem gets harder.

3.11.4 Feasibility and phase I methods

Phase I is to compute the **strictly feasible starting point** for barrier method.

Basic phase I method:

$$\begin{aligned} &\text{minimize (over } x,\, s)\ s \\ &\qquad\text{subject to } f_i(x) \leq s,\ i = 1, 2, \ldots, m \\ &\qquad\qquad Ax = b \end{aligned}$$

For more details see lecture notes or textbook.

3.11.5 Complexity analysis via self-concordance

Analyzes the time complexity. e.g. Newton's method iterations per centering step.

3.11.6 Generalized inequalities

$$\begin{aligned} \text{minimize } & f_0(x) \\ \text{subject to } & f_i(x) \preceq_{K_i} 0,\ i = 1, 2, \ldots, m \\ & Ax = b \end{aligned}$$

where

- f_0 convex
- $f_i : \mathbb{R}^n \to \mathbb{R}^{k_i}$, $i = 1, 2, \ldots, m$ convex with respect to proper cone $K_i \in \mathbb{R}^{k_i}$
- f_i twice continuously differentiable
- $A \in \mathbb{R}^{p \times n}$, $\textbf{rank}\, A = p$
- assume that p^* is finite and attained
- assume strict feasibility $\Longrightarrow$ (1) strong duality; (2) dual optimum attained

Examples of greatest interest: SOCP, SDP.

Generalized logarithm for proper cone:

$\psi : \mathbb{R}^n \to \mathbb{R}$ is called the **generalized logarithm for proper cone** $K \subseteq \mathbb{R}^q$ if:

- $\mathbf{dom}\,\psi = \int K$
- $\nabla^2\psi(y) \prec 0$ for $\forall y \succ_K 0$
- $\psi(sy) = \psi(y) + \theta \log s$, for $\forall s > 0, y \succ_K 0$ (θ is the degree of ψ)

Examples:

cone name	**cone**	**generalized logarithm**	**degree**
nonnegative orthant	$K = \mathbb{R}^n_+$	$\psi(y) = \sum_{i=1}^n \log y_i$	$\theta = n$
positive semidefinite cone	$K = \mathbb{S}^n_+$	$\psi(Y) = \mathbf{log\,det}\,Y$	$\theta = n$
second-order cone	$K = \{y \in \mathbb{R}^{n+1} \| (y_1^2 + y_2^2 + \cdots + y_n^2)^{1/2} \le y_{n+1}\}$	$\psi(y) = \log(y_{n+1}^2 - y_1^2 - \cdots - y_n^2)$	$\theta = 2$

Properties: for $y \succ_K 0$,

$$\nabla\psi(y) \succeq_{K^*} 0, \qquad y^T\nabla\psi(y) = \theta$$

Examples:

- **nonnegative orthant** ($K = \mathbb{R}^n_+$, $\psi(y) = \sum_{i=1}^n \log y_i$, $\theta = n$):

$$\nabla\psi(y) = (\frac{1}{y_1}, \frac{1}{y_2}, \ldots, \frac{1}{y_n}) \qquad y^T\nabla\psi(y) = n$$

- **positive semidefinite cone** ($K = \mathbb{S}^n_+$, $\psi(Y) = \mathbf{log\,det}\,Y$, $\theta = n$):

$$\nabla\psi(Y) = Y^{-1} \qquad \mathbf{tr}(Y\nabla\psi(Y)) = n$$

- **second-order cone** ($K = \{y \in \mathbb{R}^{n+1} | (y_1^2 + y_2^2 + \cdots + y_n^2)^{1/2} \le y_{n+1}\}$, $\psi(y) = \log(y_{n+1}^2 - y_1^2 - \cdots - y_n^2)$, $\theta = 2$):

$$\nabla\psi(y) = \frac{2}{y_{n+1}^2 - y_1^2 - y_2^2 - \cdots - y_n^2}\begin{bmatrix} -y_1 \\ -y_2 \\ \vdots \\ -y_n \\ y_{n+1} \end{bmatrix} \qquad \mathbf{tr}(y^T\nabla\psi(y)) = 2$$

Logarithm barrier for $f_i(x) \preceq_{K_i} 0$ ($i = 1, 2, \ldots m$):

$$\phi(x) = -\sum_{i=1}^m \psi_i(-f_i(x))$$

with $\mathbf{dom}\,\phi = \{x | f_i(x) \prec_{K_i} 0,\ i = 1, 2, \ldots, m\}$.

- ψ_i is the generalized logarithm for K_i, with degree θ_i;
- ϕ is convex and twice continuously differentiable

Central path: $\{x^*(t)|t>0\}$ where $x^*(t)$ solves:

$$\begin{aligned}\text{minimize } & tf_0(x)+\phi(x)\\ \text{subject to } & Ax=b\end{aligned}$$

Dual points on central path

$x=x^*(t)$ if there exists a $w\in\mathbb{R}^p$ such that

$$t\nabla f_0(x)+\sum_{i=1}^m Df_i(x)^T\nabla f_i(x)+A^Tw=0,\qquad Ax=b$$

where $Df_i(x)^T\in\mathbb{R}^{k_i\times n}$ is derivative matrix of f_i; therefore, $x^*(t)$ minimizes the Lagrangian:

$$L(x,\lambda^*(t),\nu^*(t))=f_0(x)+\sum_{i=1}^m\lambda_i^*(t)f_i(x)+\nu^*(t)^T(Ax-b)$$

where we have:

$$\begin{cases}\lambda_i^*(t)=\frac{1}{t}\nabla\psi_i(-f_i(x^*(t)))\\ \nu^*(t)=\frac{w}{t}\end{cases}$$

besides, the properties of ψ_i implies that

$$\lambda_i^*(t)\succ_{K_i^*}0$$

with dual gap

$$f_0(x^*(t))-g(\lambda^*(t),\nu^*(t))=\frac{1}{t}\sum_{i=1}^m\theta_i$$

Example: **SDP**, $F_i\in\mathbb{S}^p$

$$\begin{aligned}\text{minimize } & c^Tx\\ \text{subject to } & F(x)=\sum_{i=1}^n x_iF_i+G\preceq 0\end{aligned}$$

- **logarithm barrier**:

$$\phi(x)=\mathbf{log\,det}(-F(x)^{-1})$$

- **central path**: $x^*(t)$ minimizes

$$tc^Tx-\mathbf{log\,det}(-F(x))$$

  hence,

$$tc_i-\mathbf{tr}(F_iF(x^*(t))^{-1})=0,\ i=1,2,\dots,n$$

- **dual point on central path**: $Z^*(t) = -\frac{1}{t}F(x^*(t))^{-1}$ is feasible for

$$\begin{aligned} \text{maximize} \quad & \mathbf{tr}(GZ) \\ \text{subject to} \quad & \mathbf{tr}(F_iZ) + c_i = 0, \; i = 1, 2, \ldots, n \\ & Z \succeq 0 \end{aligned}$$

- **duality gap on central path**:

$$c^T x^*(t) - \mathbf{tr}(GZ^*(t)) = \frac{p}{t}$$

- The barrier method's only difference is that stopping condition $\frac{m}{t} < \epsilon$ is replaced by $\frac{\sum_{i=1}^m \theta_i}{t}$.
- number of outer iterations

$$\left\lceil \frac{\log(\frac{\sum_{i=1}^m \theta_i}{\epsilon t^{(0)}})}{\log \mu} \right\rceil$$

3.11.7 Primal-dual interior-point methods

Generally speaking, the **primal-dual interior-point methods** are:

- more efficient when high accuracy is required
- can start from infeasible points
- cost each iteration same as barrier method
- update primal and dual variables at each iteration
- no distinct between inner and outer iterations
- often exhibits superlinear asymptotic convergence
- search direction can be interpreted as Newton directions for modified KKT conditions

it is more commonly used in practice, than the barrier method.

4 Exercise and Examples Conclusions

4.1 Basic Settings

4.1.1 About Vectors

All vectors of form $\mathbb{R}^n$ in this course are, by default, vertical, instead of horizontal.

4.2 Math Tools

4.2.1 Cauchy-Schwarz Inequality

Abbreviated as "Cauchy Inequality" in most Chinese materials.
Generally, it means:

$$(\mathbf{u}^T\mathbf{v})^2 \leq (\mathbf{u}^T\mathbf{u})(\mathbf{v}^T\mathbf{v})$$

for $\forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^T$.

4.2.2 Jensen's Inequality

The **basic** inequality goes:

$$\text{If } \begin{cases} f \text{ is convex} \\ \theta \in [0,1] \end{cases} \implies f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$$

And its **extension** goes:

$$\text{If } \begin{cases} f \text{ is convex} \\ z \text{ is any random variable} \end{cases} \implies f(\mathbb{E}z) \leq \mathbb{E}f(z)$$

The basic inequality is simply a special case of the extension with discrete distribution with:

$$\mathbf{prob}(z = x) = \theta, \qquad \mathbf{prob}(z = y) = (1-\theta)$$

4.3 Proof Routines

4.3.1 To prove convex set

Geometrically:

1. Any $x_1, x_2 \in C$
2. Is the line segment between (x_1, x_2) always in C? If yes then it is convex set.

Algebraically:

1. Assume $x_1, x_2 \in C$
2. If $x = \theta x_1 + (1-\theta)x_2 \in C$ then the set C is convex.

Technically: The operations that preserve convexity (see 3.2.10).
Some examples:

- The set of points closer to a given point than another given point is convex. (ball or halfspace)
- The set of points closer to a given point than a given set is convex. (operation preserve convexity)
- The set of points closer to one set than another is **not** convex, where distance is defined by the closest point's line segment in between.

4.3.2 To prove convex function

- Verify definition. If the function is complex, simplify it by restricting it to a line.
- If the function f is twice-differentiable, examine $\nabla^2 f$.
- Show that f is obtained from some known convex functions by certain operations (the operations that preserve convexity). Refer to part 3.3.10 for more details.
- **Prove that the epigraph of f is a convex set.**

$$\begin{aligned} \textbf{epi}\, f &= \{(x,t) | f(x) \leq t\} \\ &= \{(x,t) | g(x,t) \leq 0\} \end{aligned}$$

where in $g(x,t)$, t is shown as $\frac{\cdots}{t}$. Now proving that $g(x,t)$ is convex is enough.

An example:

$$f(x) = e^{-a\sqrt{x}}, \qquad \textbf{dom}\, f = \mathbb{R}_+$$

when will f be convex?

Solution: let $f(x) = h(g(x))$, then we have

$$g(x) = -a\sqrt{x}, \qquad h(x) = e^x$$

$h(x)$ is obviously convex and non-decreasing, thus to make f convex, g need to be convex. Therefore, $a \geq 0$.

4.3.3 To prove quasiconvex / quasiconcave / quasilinear

- quasiconvex:
  - the sublevel sets ($f(x) \leq \alpha$) are convex
  - $-f$ is quasiconcave
- quasiconcave:
  - the superlevel sets ($f(x) \geq \alpha$) are convex
  - $-f$ is quasiconvex
- quasilinear:
  - both quasiconvex and quasiconcave

4.3.4 Relaxations on Constraints' Expressions

Sometimes abbreviated. Such as, we have two lines of constraints:

$$\begin{aligned} Ax &\succeq b \\ x &\succeq 0 \end{aligned}$$

and indeed in standard form it means:

$$\begin{aligned} -Ax + b &\preceq 0 \\ -x &\preceq 0 \end{aligned}$$

and the formal standard form actually looks like:

$$-\begin{bmatrix} A \\ I \end{bmatrix} x + \begin{bmatrix} b \\ 0 \end{bmatrix} \preceq 0$$

4.3.5 Lagrangian, Dual, KKT

All routine follows what is discussed in section 3.5, **Duality** part.
The **key point** in solving these problems, is to get the **equivalent form** of some **primal** problems.
The equivalent primal problem could be obtained; but, there **isn't necessarily linkage** between the original problem's dual and the equivalent problem's dual. It could lead to **completely different duals**. It means that, we could have an original problem with an unsolvable dual, with equivalent questions with readily-solvable duals.
Some common reformulations:

- (introducing) new variables / new constraints
- explicit constraints $\longleftrightarrow$ implicit constraints
  e.g. The explicit constraint:
  $$\begin{aligned} \text{minimize } & f_0(x) \\ \text{subject to } & constraint_{partA} \\ & constraint_{partB} \end{aligned}$$
  is equivalent with implicit constraint:
  $$\begin{aligned} \text{minimize } & \tilde{f}_0(x) = \begin{cases} f_0(x) & constraint_{partA} \\ \infty & otherwise \end{cases} \\ \text{subject to } & constraint_{partB} \end{aligned}$$
- transform the objective / constraints
  $$e.g.\ f_0(x) \rightarrow \phi(f_0(x)),\ \phi \text{ is } \textbf{convex} \text{ and } \textbf{increasing}$$

Examples:

- minimize $f_0(\phi(x)) \longleftrightarrow$ minimize $f_0(y)$ & $s.t.\ \phi(x) - y = 0$

4.4 Matrix Tricks

4.4.1 Dealing with $\mathbb{S}^n_+$ or $\mathbb{S}^n_{++}$

- Generally, if the conclusion is not obvious, try on fixed z:

$$z^T M z \leq z^T N z \Longrightarrow M \preceq N$$

4.4.2 A useful fact

$$\begin{aligned} &\inf_{X \succ 0} \left(\log\det X^{-1} + \mathbf{tr}(XY) \right) \\ = &\begin{cases} \log\det Y + n & Y \succ 0 \\ -\infty & \text{otherwise} \end{cases} \end{aligned}$$

and the minimizer when $Y \succ 0$ is $X = Y^{-1}$.

To prove it we could assume that $Y \not\succ 0$ and then there exists $a \neq 0$ s.t. $a^T Y a \preceq 0$. Choosing $X = I + taa^T$, then $\det X = 1 + t\|a\|_1^2$ and:

$$\log\det X^{-1} + \mathbf{tr}(XY) = -\log(1 + ta^T a) + \mathbf{tr}\, Y + ta^T Y a$$

goes to $-\infty$ as $t \to \infty$.

Appeared in:

- *(A7.1 (b))*

4.4.3 Some Useful Equations about Trace

With $Q \in \mathbb{S}^n_{++}$,

$$\mathbf{tr}(Q a_i a_i^T) = a_i^T Q a_i$$

4.4.4 Eigenvalue Decomposition

For any symmetric matrix $X \in \mathcal{S}^n$, we can do eigenvalue decomposition as:

$$X = Q \Lambda Q^{-1}, \qquad \Lambda = \begin{bmatrix} \lambda_1 & \dots & \\ \vdots & & \vdots \\ & \dots & \lambda_n \end{bmatrix}$$

where Q's columns are independent eigenvectors.

$$XQ = Q\Lambda$$

And we also have:

$$
\begin{aligned}
X &= Q\Lambda Q^{-1} \\
X^2 &= (Q\Lambda Q^{-1})(Q\Lambda Q^{-1}) \\
&= Q\Lambda Q^{-1}Q\Lambda Q^{-1} \\
&= Q\Lambda^2 Q^{-1} \\
X^n &= Q\Lambda^n Q^{-1} \\
X^{-1} &= Q\Lambda^{-1} Q^{-1} \\
\Lambda^p &= \begin{bmatrix} \lambda_1^p & \dots & \\ \vdots & & \vdots \\ & \dots & \lambda_n^p \end{bmatrix}
\end{aligned}
$$

For positive semidefinite matrix X, we have $\lambda_i \geq 0$; for positive definite matrix X, we have $\lambda_i > 0$.

Could refer to `http://mathworld.wolfram.com/EigenDecomposition.html`.