Lecture 1:
Introduction, Review of Linear Algebra, Convex Analysis

Prof. Krishna R. Pattipati
Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu; (860) 486-2890

January 21, 2016

Outline

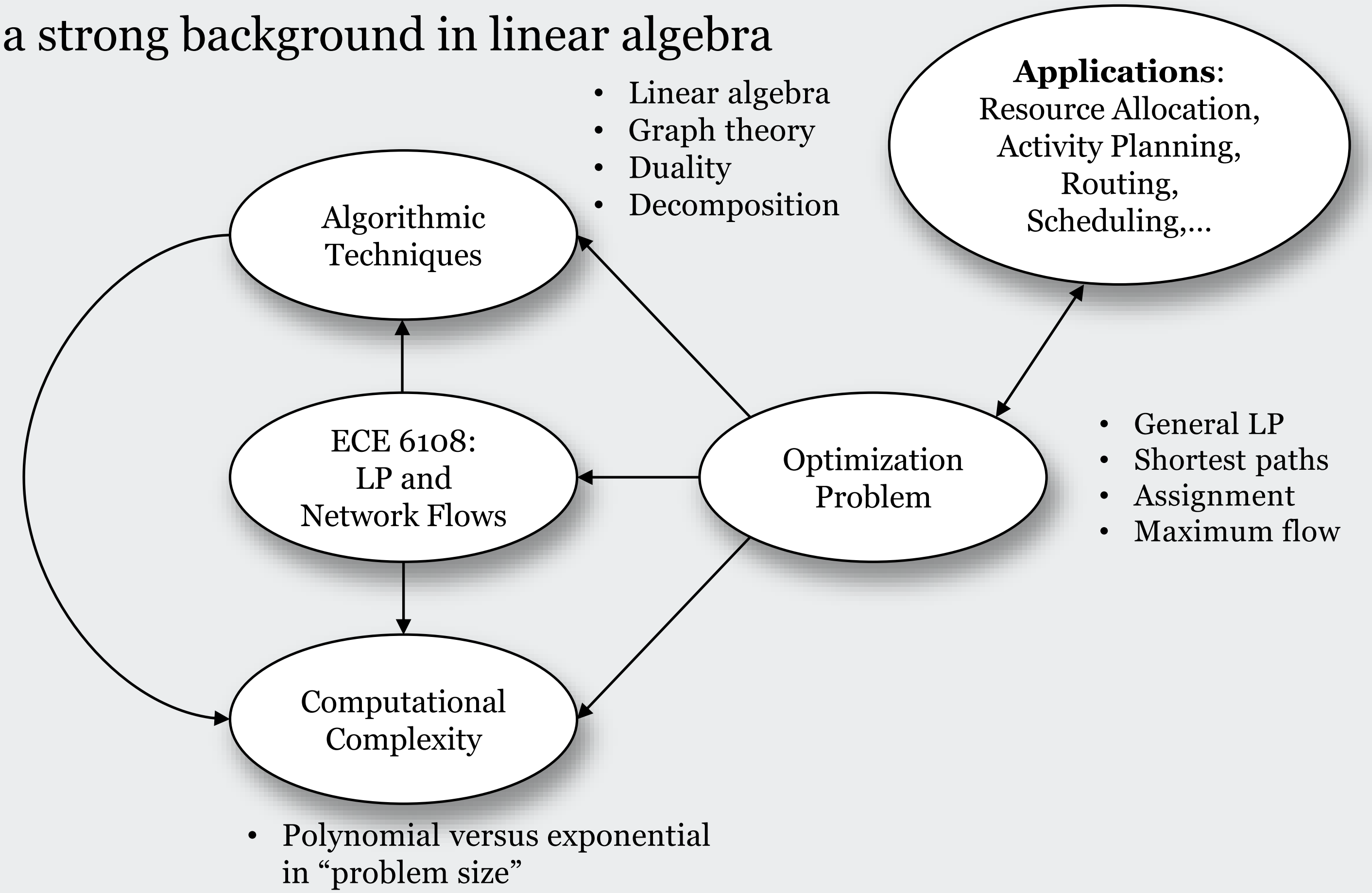
- Course Objectives
- Optimization problems
  - Classification
  - Measures of complexity of algorithms
- Background on Matrix Algebra
  - Matrix-vector notation
  - Matrix-vector product
  - Linear subspaces associated with an $m \times n$ matrix A
  - LU and QR decompositions to solve $A\underline{x} = \underline{b}$, A is $n \times n$
- Convex analysis
  - Convex sets
  - Convex functions
  - Convex programming problem
- LP is a special case of convex programming problem
  - Local optimum ≡ global optimum

Reading List

- Dantzig and Thapa, Foreword to Linear Programming: Volume 1
- Papadimitrou and Steiglitz, Chapter 1
- Bertsimas and Tsitsiklis, Chapter 1 & Sections 2.1 and 2.2
- Ahuja, Magnanti and Orlin, Chapter 1

Course objectives

- Provide systems analysts with central concepts of widely used and elegant optimization techniques used to solve LP and Network Flow problems
- Requires skills from both Math and CS
- Need a strong background in linear algebra



Three Recurrent Themes

1. Mathematically formulate the optimization problem
2. Design an algorithm to solve the problem
   - Algorithm ≡ a step-by-step solution process
3. Computational complexity as a function of "size" of the problem

- **What is an optimization problem?**
  - Arise in mathematics, engineering, applied sciences, economics, medicine and statistics
  - Have been investigated at least since 825 A.D.
    - Persian author Abu Ja'far Mohammed ibn musa al khowarizmi wrote the first book on Math
  - Since the 1950s, a hierarchy of optimization problems have emerged under the general heading of "mathematical programming"

What is an optimization problem?

- **Has three attributes**
  - Independent variables or parameters $(x_1, x_2, \ldots, x_n)$
    - Parameter vector: $\underline{x} = [x_1, x_2, \ldots, x_n]^T$
  - Conditions or restriction on the acceptable values of the variables

    $\Rightarrow$ Constraints of the problem
    - Constraint set: $\underline{x} \in \Omega$ ($e.g.$, $\Omega = \{\underline{x} : x_i \geq 0\}$)
  - A single measure of goodness, termed the objective (utility) function or cost function or goal, which depends on the parameter vector $\underline{x}$:
    - Cost function: $f(x_1, x_2, \ldots, x_n) = f(\underline{x})$

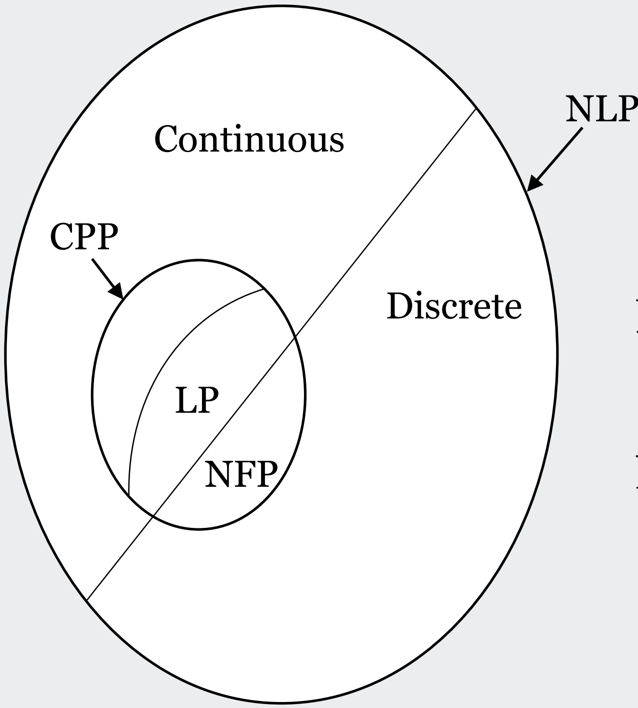
Typical cost functions

	$\underline{x} \in R^n$	$\underline{x} \in Z^n$	$\underline{x} \in \{0,1\}^n$
$f \in R$	* $f : R^n \to R$	* $f : Z^n \to R$	* $f : \{0,1\}^n \to R$
$f \in Z$	# $f : R^n \to Z$	* $f : Z^n \to Z$	* $f : \{0,1\}^n \to Z$
$f \in \{0,1\}$	# $f : R^n \to \{0,1\}$	# $f : Z^n \to \{0,1\}$	* $f : \{0,1\}^n \to \{0,1\}$

R = set of reals; Z = set of integers
* denotes the most common optimization cost functions
# Typically a problem of mapping features to categories

- Abstract formulation
  - "Minimize $f(x)$ where $\underline{x} \in \Omega$"
- The solution approach is algorithmic in nature
  - Construct a sequence $\underline{x}_0 \to \underline{x}_1 \to \ldots \to \underline{x}^*$ where $\underline{x}^*$ minimizes $f(\underline{x})$ subject to $\underline{x} \in \Omega$

Classification of mathematical programming problems



LP: Linear Programming

NFP: Network Flow Problems

CPP: Convex Programming Problems

NLP: Nonlinear Programming Problems

- Unconstrained continuous NLP
  - $\Omega = R^n$, i.e., no constraints on $\underline{x}$
  - Algorithmic techniques . . . ECE 6437
    - Steepest descent
    - Conjugate gradient
    - Newton
    - Gauss-Newton
    - Quasi-Newton

Classification of mathematical programming problems

- Constrained continuous NLP
  - Ω defined by:
    - Set of equality constraints, E
      - $h_i(\underline{x}) = 0;\ i = 1, 2, \ldots, m;\ \ m < n$
    - Set of inequality constraints, I
      - $g_i(\underline{x}) \geq 0;\ i = 1, 2, \ldots, p$
    - Simple bound constraints
      - $x_i^{LB} \leq x \leq x_i^{UB};\ i = 1, 2, \ldots, n$
  - Algorithmic techniques . . . ECE 6437
    - Penalty and barrier function methods
    - Reduced gradient method
    - Augmented Lagrangian (multiplier) methods
    - Recursive quadratic programming
- Convex programming problems (CPP)
  - Characterized by:
    - $f(\underline{x})$ is convex . . . will define shortly!
    - $g_i(\underline{x})$ is concave or $-g_i(\underline{x})$ is convex
    - $h_i(\underline{x})$ linear $\Rightarrow A\underline{x} = \underline{b}$; A an $m \times n$ matrix
  - **Key**: local minimum ≡ global minimum
  - Necessary conditions are also sufficient (the so-called Karush-Kuhn-Tucker (KKT) conditions (1951))

$$A\underline{x} = \underline{b}$$

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

Linear programming (LP) problems

- LP is characterized by:
  - $f(\underline{x})$ linear $\Rightarrow f(\underline{x}) = c_1x_1 + c_2x_2 + \cdots + c_nx_n = \underline{c}^T\underline{x}$
  - $g_i(\underline{x})$ linear $\Rightarrow \underline{a}_i^T\underline{x} \geq b_i;\ i \in I$
  - $h_i(\underline{x})$ linear $\Rightarrow \underline{a}_i^T\underline{x} = b_i;\ i \in E$
  - $x_i \geq 0;\ i \in P$
  - x_i unconstrained; $i \in U \Rightarrow$ unconstrained if $i \in U$
- An important subclass of convex programming problems
  - Widely used model in production planning, allocation, routing and scheduling,….
  - Examples of Industries using LP
    - Petroleum: extraction, refining, blending and distribution
    - Food: economical mixture of ingredients; shipping from plants to warehouses
    - Iron and steel industry: pelletization of low-grade ores, shop loading, blending of iron ore and scrap to produce steel,…
    - Paper mills: minimize trim loss
    - Communication networks: routing of messages
    - Ship and aircraft routing, Finance,…

Search Space of LP is Finite

- A key property of LP
  - Number of possible solutions, N is finite
    - If n variables, m equality constraints, p inequality constraints and q unconstrained variables

$$N = \binom{n+p+q}{m+p+q}$$

$n = 100,\ m = p = q = 10$

$\Rightarrow N = 1.6975 \times 10^{28}$ possible solutions!

- Lies on the border of **combinatorial** or **discrete optimization and continuous optimization problems**
  - Also called **enumeration problems**, since can theoretically count the number of different solutions
- Counting the number of solutions is a **laborious process**
  - Even if each solution takes 10^{-12} seconds (terahertz machine !!), it takes 538 million years to search for an optimal solution.

A Brief History of LP

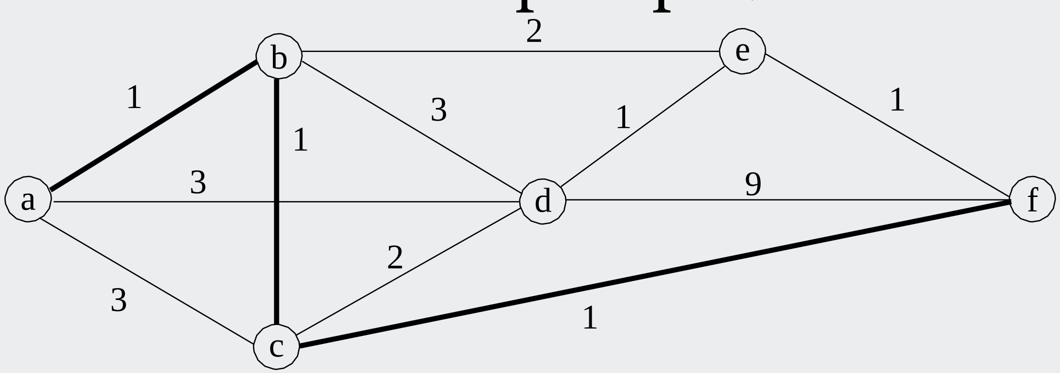
- Chronology and Pioneers
  - Fourier : System of Linear Inequalities (1826)
  - de la Vallee Poussin: Minimax Estimation (1911)
  - Von Neumann: Game Theory (1928) and Steady Economic Growth (1937)
  - Leontief: Input-output Model of the Economy (1932); Nobel Prize: 1973
  - Kantorovich: Math. Methods in Organization and Planning Production (1939); Nobel Prize: 1975
  - Koopmans: Economics of Cargo Routing (1942); Nobel Prize: 1975
  - Dantzig: Activity Analysis for Air Force Planning and Simplex Method (1947)
  - Charnes, Cooper and Mellon: Commercial applications in Petroleum industry (1952)
  - Orchard-Hays: First successful LP software (1954)
  - Merrill Flood, Ford and Fulkerson: **Network Flows** (1950, 1954)
  - Dantzig, Wets, Birge, Beale, Charnes and Cooper: Stochastic Programming (1955-1960, 1980's)
  - Gomory: Cutting plane methods for Integer Programming (1958)
  - Dantzig-Wolfe and Benders: **Decomposition Methods** (1961-62)
  - Bartels-Golub-Reid (1969, 1982) & Forrest-Tomlin (1972): **Sparse LU methods**
  - Klee & Minty: Exponential Complexity of Simplex (1972)
  - Bland: Avoid Cycling in Simplex (1977)
  - Khachian: LP has Polynomial Complexity (1979)
  - Karmarkar: Projective Interior Point Algorithm (1984)
  - **Primal-dual/Path Following** (1984-2000)
  - **Modern implementations (XMP, OSL, CPLEX, Gurobi, Mosek, Xpress,…)**

Two Main Methods for LP

- Fortunately, there exist efficient methods
  - Revised simplex (Dantzig: 1947-1949)
  - Ellipsoid method (Khachian: 1979)
  - Interior point algorithms (Karmarkar: 1984)
- Dantzig's Revised simplex (●)
  - In theory, can have exponential complexity, but works well in practice
  - Number of iterations **grows** with problem **size**
- Khachian's Ellipsoid method (●)
  - Polynomial complexity of LP, but not competitive with the Simplex method ⇒ not practical
- Karmarkar's (or interior point) algorithms (●)
  - Polynomial complexity
  - Number of iterations is relatively constant (≈ 20-50) with the size of the problem
  - Need efficient matrix decomposition techniques

Network flow problems (NFP)

- Subclass of LP problems defined on graphs
  - Simpler than general LP
  - One of the most elegant set of optimization problems
- Examples of network flow problems
  - Shortest path on a graph
  - Maximum flow problem
  - Minimum cost flow problem
  - Transportation problem
  - Assignment problem (also known as weighted bipartite matching problem)
- Illustration of shortest path problem



- Shortest path from a to f is: $a \rightarrow b \rightarrow c \rightarrow f$

  shortest path length = 3

Integer programming (IP) problems

- Hard intractable problems
- NP-complete problems (exponential time complexity)
- Examples of IP problems
  - Travelling salesperson problem
  - VLSI routing
  - Test sequencing & test pattern generation
  - Multi-processor scheduling to minimize makespan
  - Bin-packing and covering problems
  - Knapsack problems
  - Inference in graphical models
  - Multicommodity flow problems
  - Max cut problem
- Illustration of traveling salesperson problem
  - Given a set of cities $C = \{c_1, c_2, \ldots, c_n\}$
  - For each pair (c_i, c_j), the distance $d(c_i, c_j) = d_{ij}$
  - Problem is to find an ordering $\langle c_{\pi(1)}, c_{\pi(2)}, \ldots, c_{\pi(n)} \rangle$ such that

$$\sum_{i=1}^{n-1} d\left(c_{\pi(i)}, c_{\pi(i+1)}\right) + d\left(c_{\pi(n)}, c_{\pi(1)}\right)$$

is a minimum

⇒ Shortest closed path that visits every node once (Hamiltonian path)

Want efficient algorithms

- How to measure problem size?
  - In LP, the problem size is measured in one of two ways:
    - Crude way:

$$n+m+p+q$$

    - Correct way: (size depends on the base used)

$$\sum_{i=1}^{m+p}\left[\left(\log_2|b_i|\right)^+ + \sum_{j=1}^{n}\left(\log_2|a_{ij}|\right)^+\right] + \sum_{j=1}^{n}\left(\log_2|c_j|\right)^+$$

  - For network flow problems, the size is measured in terms of the number of nodes and arcs in the graph and the largest arc weight
- How to measure efficiency of an algorithm ?
  - The time requirements of # of operations as a function of the problem size
  - Time complexity measured using big "O" notation
    - A function $h(n)=O(g(n))$ (read as $h(n)$ equals "big oh" of $g(n)$) iff $\exists$ constants $c, n_0>0$ such that $|h(n)| \leq c|g(n)|, \forall n>n_0$

Polynomial versus Exponential Complexity

- Polynomial versus exponential complexity
  - An algorithm has polynomial time complexity if $h(n) = O(p(n))$ for some polynomial function
    - Crude Examples: $O(n)$, $O(n^2)$, $O(n^3)$, …
  - Some algorithms have exponential time complexity
    - Examples: $O(2^n)$, $O(3^n)$, etc.
- Significance of polynomial vs. exponential complexity
  - Time complexity versus problem size (1 *ns/op*)

Complexity	*Problem size n*			
	10	20	30	40
n	10^{-8}	2.10^{-8}	3.10^{-8}	4.10^{-8}
n^2	10^{-7}	4.10^{-7}	9.10^{-7}	16.10^{-7}
n^3	10^{-6}	8.10^{-6}	27.10^{-6}	64.10^{-6}
2^n	10^{-6}	10^{-3}	1.07	18.3 min
3^n	6×10^{-5}	3.48	2.37 days	385.5 years

  - Last two rows are **inherently intractable**
  - NP-hard; must go for suboptimal heuristics
  - Certain problems, although intractable, are optimally solvable in practice (e.g., knapsack for as many as 10,000 variables)

Background on matrix algebra

- Vector – Matrix Notation

$$\underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{a column vector}$$

- $x_i \in R; \quad x_i \in [-\infty, \infty] \Rightarrow \underline{x} \in R^n$
- $\underline{x} \in Z^n$ for integers
- $x_i \in \{0, 1\}$ for binary
- $A = [a_{ij}]$ an $m \times n$ matrix $\in R^{mn}$
- $A^T = [a_{ji}]$ an $n \times m$ matrix $\in R^{nm}$
- $m = n \Rightarrow A$ is a square matrix
- A square $n \times n$ matrix is symmetric if $a_{ij} = a_{ji}$

$$\begin{bmatrix} 2 & 4 \\ 4 & 11 \end{bmatrix} \quad \text{symmetric}$$

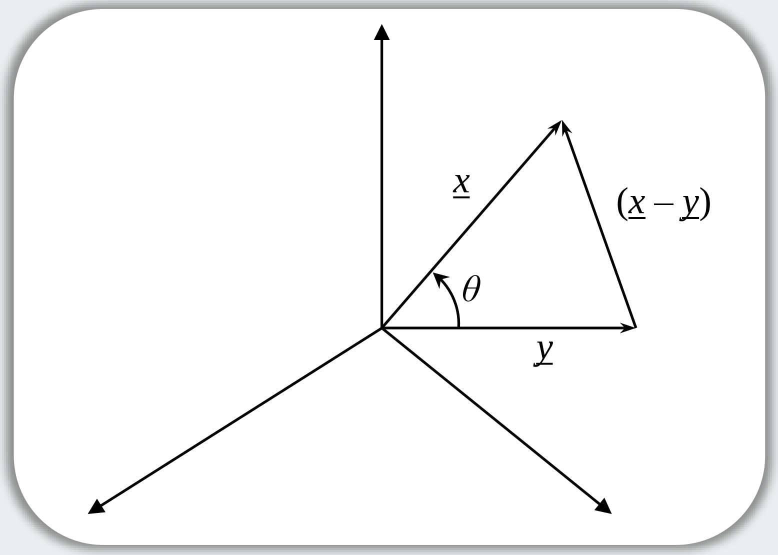
- Diagonal matrix: $A = \begin{bmatrix} d_1 & & 0 \\ & d_2 & \\ 0 & & d_n \end{bmatrix} = Diag(d_1, d_2, \ldots, d_n)$

Matrix-vector notation

- Identity matrix: $I_n = Diag(1,1,\ldots,1)$
- A matrix is PD if $\underline{x}^T A \underline{x} > 0, \forall \underline{x} \neq \underline{0}$
- A matrix is PSD if $\underline{x}^T A \underline{x} \geq 0, \forall \underline{x} \neq \underline{0}$
- Note: $\underline{x}^T A \underline{x} = \underline{x}^T A^T \underline{x} \Rightarrow \underline{x}^T A \underline{x} = \underline{x}^T [(A + A^T)/2] \underline{x}$
  $\left(\frac{A+A^T}{2}\right)$ is called the *symmetrized* part of A
- If A is skew symmetric, $A^T = -A \Rightarrow \underline{x}^T A \underline{x} = 0 \;\forall \underline{x}$
- $A = Diag(d_i) \Rightarrow \underline{x}^T A \underline{x} = \sum_{i=1}^{n} d_i x_i^2$
- Vector $\underline{x}$ is an $n \times 1$ matrix
- $\underline{x}^T \underline{y} =$ inner (dot, scalar) product $= \sum_{i=1}^{n} x_i y_i$ (a scalar)

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_{1n} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Inner product and cosine relationship



- Know (n=3 case):

$$\|\underline{x}-\underline{y}\|^2 = (x_1-y_1)^2+(x_2-y_2)^2+(x_3-y_3)^2$$
$$= (x_1^2+x_2^2+x_3^2)+(y_1^2+y_2^2+y_3^2)-2(x_1y_1+x_2y_2+x_3y_3)$$

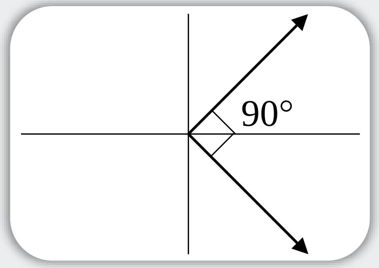
- Also know:

$$\|\underline{x}-\underline{y}\|^2 = (\underline{x}^T\underline{x})+(\underline{y}^T\underline{y})-2\sqrt{(\underline{x}^T\underline{x})(\underline{y}^T\underline{y})}\cos\theta$$
$$\Rightarrow \cos\theta = \frac{\underline{x}^T\underline{y}}{\sqrt{(\underline{x}^T\underline{x})(\underline{y}^T\underline{y})}} = \frac{\underline{x}^T\underline{y}}{\|\underline{x}\|_2\|\underline{y}\|_2}$$

Vector norms

$$\begin{matrix} \underline{x} & \underline{y} \end{matrix}$$
$$\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \Rightarrow \frac{4}{5} = \cos\theta = \cos^{-1} 0.8 \approx 36.9^\circ$$

- $\theta = 90^\circ \Rightarrow \underline{x}$ and $\underline{y}$ are perpendicular to each other
  $\Rightarrow$ ORTHOGONAL $\Rightarrow \underline{x}^T \underline{y} = 0$, e.g.,
  $$\underline{x} = [.8 \quad -.6]^T, \quad \underline{y} = [.6 \quad .8]^T$$



- Vector norms
  - Norms generalize the concept of absolute value of a real number to vectors (and matrices) (measure of "SIZE" of a vector (and matrix))
  - $\|\underline{x}\|_p$ = Holder or p-norm = $[|x_1|^p + |x_2|^p + \ldots + |x_n|^p]^{1/p} = \left[\sum_{i=1}^{n} |x_i|^p\right]^{1/p}$ ~ "size"
  - Most important: $\begin{cases} p = 1 & \Rightarrow \|\underline{x}\|_1 = \sum_{i=1}^{n} |x_i| \\ p = 2 & \Rightarrow \|\underline{x}\|_2 = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2} \quad \text{(RSS)} \\ p = \infty & \Rightarrow \|\underline{x}\|_\infty = \max_i |x_i| \end{cases}$
  - All norms convey approximately the same information
  - Only thing is some are more convenient to use than others

Matrix-vector product

- $\underline{\hat{x}}$ approx. to $\underline{x} \Rightarrow$ absolute error $\|\underline{x} - \underline{\hat{x}}\|$

  Relative error $\|\underline{x} - \underline{\hat{x}}\| / \|\underline{x}\|$

  ∞-norm $\Rightarrow$ # of correct significant digits in $\underline{\hat{x}}$

  Relative error = $10^{-p} \Rightarrow p$ significant digits of accuracy
- Matrix-vector product

$$A\underline{x} = \begin{bmatrix} 2 & 4 & 5 \\ 1 & 2 & 6 \\ 3 & 1 & 2 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 3 \\ 4 \end{bmatrix} x_1 + \begin{bmatrix} 4 \\ 2 \\ 1 \\ 5 \end{bmatrix} x_2 + \begin{bmatrix} 5 \\ 6 \\ 2 \\ 6 \end{bmatrix} x_3$$

  $\Rightarrow A\underline{x} = \sum_{i=1}^{n} \underline{a}_i x_i$; $A\underline{x} \Rightarrow$ linear combinations of columns of A

  $\Rightarrow A\underline{x} : R^n \rightarrow R^m$ transformation from an n-dimensional space to an m-dimensional space
- Characterization of subspaces associated with a matrix A
  - A subspace is what you get by taking **all** linear combinations of n vectors
  - **Q**: Can we talk about the dimension of a subspace? Yes!
  - **Q**: Can we characterize the subspace such that it is representable by a finite minimal set of vectors $\Rightarrow$ "basis of a subspace," yes!

Independence and rank of a matrix

- Suppose we have a set of vectors $\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_r$

  $\{a_1, a_2, \ldots, a_r\}$ are <u>dependent</u> iff $\exists$ scalars $x_1, x_2, \ldots, x_r$ s.t.

  $\sum_{i=1}^{r} \underline{a}_i x_i = \underline{0}$ and at least one $x_i \neq 0$

  they are <u>independent</u> if $\sum_{i=1}^{r} \underline{a}_i x_i = \underline{0} \implies x_i = 0$

  $\implies \nexists x_i \neq 0$ such that $\sum_{i=1}^{r} \underline{a}_i x_i = \underline{0}$
- Rank of a matrix

$$\begin{aligned}\text{rank}(A) &= \#\text{ of linearly independent columns}\\ &= \#\text{ of linearly independent rows}\\ &= \text{rank}(A^T)\\ &= \dim[\text{range}(A)] \leq \min(m,n)\end{aligned}$$

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix} \Rightarrow \text{indep. columns} = \text{indep. rows} = \text{rank} = 2$$

Row 1+ Row 2=Row 3
Column 1 + Column 2 = - Column 3

Linear subspaces associated with a matrix

- Linear spaces associated with $A\underline{x} = \underline{b}$
  - range(A) = $R(A)$ =$\{\underline{y} \in R^m | \underline{y} = \sum_{i=1}^{n} \underline{a}_i x_i$ for vectors $\underline{x} \in R^n\}$

    = column space of A
  - dim($R(A)$) = r, rank of (A)
  - $A\underline{x} = \underline{b}$ has a solution if $\underline{b}$ can be expressed as a linear combination of the columns of $A \Longrightarrow \underline{b} \in R(A)$
  - Null space of $A = N(A) = \left\{\underline{x} \in R^n \middle| A\underline{x} = \underline{0}\right\}$

    $\Longrightarrow$ also called kernel of A or ker (A)
  - Note that $\underline{x} = [000]^T$ always satisfies $A\underline{x} = \underline{0}$
  - Key: dim($N(A)$) = $n - r = n -$ rank(A)
  - If rank(A) = n, then $A\underline{x} = \underline{0} \Rightarrow \underline{x} = \underline{0} \Rightarrow N(A)$ is the origin
  - $R(A^T) = \{\underline{z} \in R^n | A^T \underline{y} = \underline{z}, \forall \underline{y} \in R^m\}$

    $\Rightarrow$ For a solution to exist, $\underline{z}$ should be in the column space of A^T or row space of A
  - $N(A^T) = \{\underline{y} \in R^m | A^T \underline{y} = \underline{0}\}$ = null space of A^T

Column, row and null spaces

	column space	null space of A
$A\underline{x} = \underline{b}$	$_{(m)}R(A)$	$_{(n)}N(A)$
$A^T\underline{y} = \underline{z}$	$_{(n)}R(A^T)$	$_{(m)}N(A^T)$
	row space of A	null space of A^T

- KEY:
  - $\dim[R(A^T)] + \dim[N(A)] = r + n - r = n$
  - $\dim[R(A)] + \dim[N(A^T)] = r + m - r = m$
  - rank of A = rank of A^T = r
  - Linearly ind. col. of A = linearly ind. rows of A

<u>Example</u>:

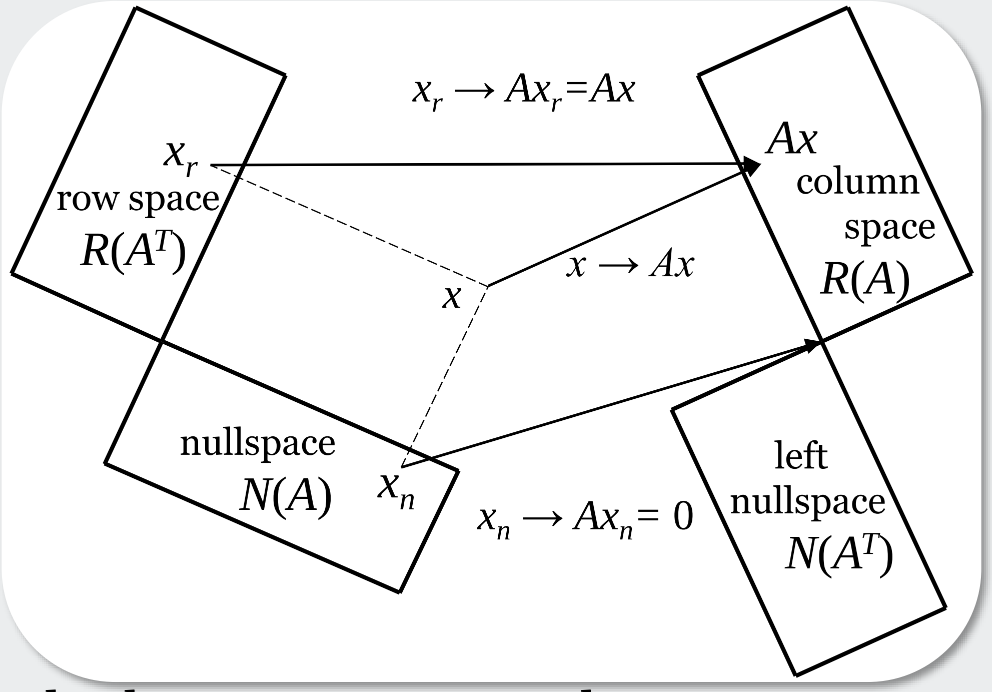
$$A^T\underline{y} = \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \\ 0 & -1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = \underline{0} \Rightarrow \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \in N(A^T) \qquad \Rightarrow \begin{bmatrix} 1 & -1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & -1 \end{bmatrix} \text{ are linearly independent}$$

$$\begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 1 \\ 0 & -1 & 1 \end{bmatrix} \text{ are linearly independent,} \qquad \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \in N(A^T)$$

indep. col of A^T = indep. Rows of A = indep. col of A = indep. Rows of A^T

Geometric insight

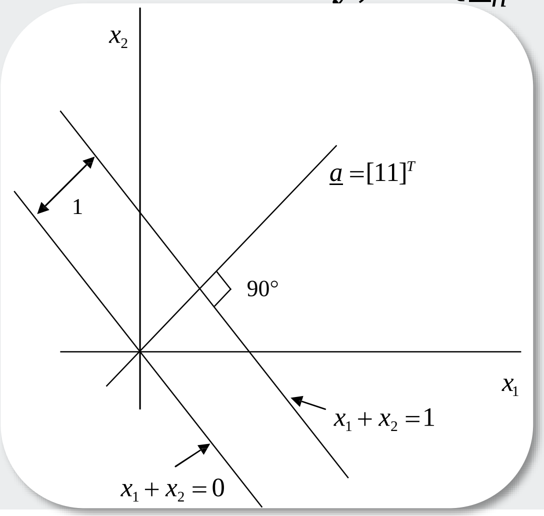
- Every $\underline{x} \in N(A) \perp^r$ to every $\underline{z} \in R(A^T)$
  $\Rightarrow$ if $A^T \underline{y} = \underline{z}$ and $A\underline{x} = \underline{0} \Rightarrow \underline{x}^T \underline{z} = \underline{x}^T A^T \underline{y} = \underline{0}^T \underline{y} = 0$
- Every $\underline{y} \in N(A^T) \perp^r$ to every $\underline{z} \in R(A^T) \Rightarrow \underline{y}^T \underline{b} = \underline{y}^T A^T \underline{x} = 0$

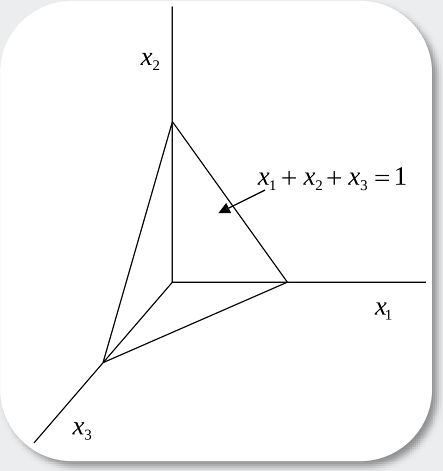


- In general, when rank$(A) = r < n$, then $\dim(N(A)) = n - r$
  - Suppose we have $\underline{x}_r \Rightarrow A\underline{x}_r = \underline{b}$
  - Then all $\underline{x} = \underline{x}_r + \underline{x}_n$ are also solutions to $A\underline{x} = \underline{b}$
    $\Rightarrow$ infinite # of solutions

Hyperplanes model equality constraints

- Can we give a geometric meaning to all this? Yes!
  - Consider a single equation $\underline{a}^T \underline{x} = b$ (a scalar)
  - $\underline{a}b/\underline{a}^T \underline{a}$ is a solution to $\underline{x}$
  - Since $\underline{x}_n \in N(\underline{a}^T)$, we have $\underline{a}^T \underline{x} = 0$ and $\dim(N(\underline{a}^T)) = n - 1$
  - But, what is $\underline{a}^T \underline{x}_n = 0 \Rightarrow$ it is a hyperplane passing through the origin
  - $\underline{a}^T \underline{x} = b \Rightarrow$ it is a hyperplane at a distance b from the origin
  - So, $H = \{\underline{x} \in R^n \mid \underline{a}^T \underline{x} = b\}$ is a hyperplane
  - $\dim(H) = n - 1$ since we can find $n - 1$ independent vectors that are orthogonal to $\underline{a}$
  - Or alternately, $H = \{\underline{x}_n \in N(\underline{a}^T) \mid \underline{a}^T (\underline{x}_r + \underline{x}_n) = b\}$



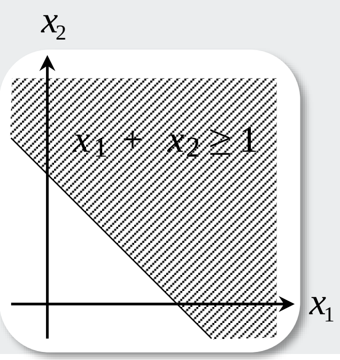


Half spaces model inequality constraints

- If we have m equations in $A\underline{x} = \underline{b}$, each equation is a hyperplane
- Then $\{\underline{x} \in R^n \mid A\underline{x} = \underline{b}\}$ is the intersection of m hyperplanes and this subspace has dimension equal to $(n - m)$
- Note: intersection of n nonparallel hyperplanes in R^n is a point $\underline{x} = A^{-1}\underline{b} \Rightarrow$ solution to $A\underline{x} = \underline{b}$
- For every hyperplane $H = \{\underline{x} \mid \underline{a}^T \underline{x} = b\}$, we can define negative and positive closed and open half spaces

closed	open
$H_{c+} = \{\underline{a}^T\underline{x} \geq b\}$	$H_{o+} = \{\underline{a}^T\underline{x} > b\}$
$H_{c-} = \{\underline{a}^T\underline{x} \leq b\}$	$H_{o-} = \{\underline{a}^T\underline{x} < b\}$

- Half spaces model inequality constraints
- Example: $x_1 + x_2 \geq 1$



Partitioning and transformations

- Partitioned matrices
  - Horizontal partition . . . useful in developing revised simplex method

$$A\underline{x} = \begin{bmatrix} B & N \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = Bx_1 + Nx_2$$

  - Horizontal and vertical partition

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} A_1 x_1 + A_2 x_2 \\ A_3 x_1 + A_4 x_2 \end{bmatrix}$$

- Elementary transformations
  - Column j of $A = \underline{a}_j = A\underline{e}_j$; $\underline{e}_j = j^{th}$ unit vector with 1 in the j^{th} component and 0, elsewhere
  - Row i of $A = \underline{e}_i^T A \Rightarrow$ element $a_{ij} = \underline{e}_i^T A \underline{e}_j$
  - $A\underline{e}_j\underline{e}_j^T = \underline{a}_j\underline{e}_j^T = [\underline{0}, \underline{0}, \ldots, \underline{a}_j, \ldots, \underline{0}] \Rightarrow j^{th}$ column is $\underline{a}_j$ and the rest are zero vectors

Deleting and inserting columns

$$I + A\underline{e}_j\underline{e}_j^T = I + \underline{a}_j\underline{e}_j^T = \begin{bmatrix} 1 & 0 & a_{1j} & 0 \\ 0 & 1 & a_{2j} & \vdots \\ \vdots & \cdots & 1+a_{jj} & \vdots \\ 0 & \cdots & a_{mj} & 1 \end{bmatrix}$$

- Suppose we have an $n \times n$ matrix A and we want to delete the j^{th} column of A and insert a new column $\underline{b}$ in its place

$$\begin{aligned} A_{new} &= A - A\underline{e}_j\underline{e}_j^T + \underline{b}\underline{e}_j^T \\ &= A\left[I - \underline{e}_j\underline{e}_j^T + A^{-1}\underline{b}\underline{e}_j^T\right] \\ &= A\begin{bmatrix} 1 & 0 & \uparrow & 0 \\ 0 & 1 & \vdots & \vdots \\ \vdots & \cdots & A^{-1}\underline{b} & \vdots \\ 0 & \cdots & \downarrow & 1 \end{bmatrix} \end{aligned}$$

- Sherman-Morrison-Woodbury formula:

$$\overline{A} = A + \underline{a}\underline{b}^T$$

$$\overline{A}^{-1} = A^{-1} - \frac{A^{-1}\underline{a}\underline{b}^T A^{-1}}{1 + \underline{b}^T A^{-1}\underline{a}}$$

Application:

$$A_{new} = A[I + (\underline{\alpha} - \underline{e}_j)\underline{e}_j^T]; \underline{\alpha} = A^{-1}\underline{b}$$

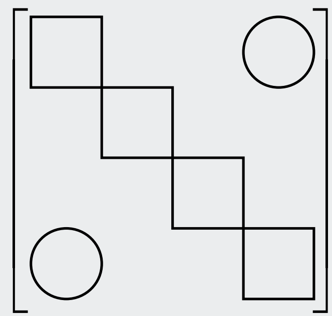
$$A_{new}^{-1} = \underbrace{[I - \frac{(\underline{\alpha} - \underline{e}_j)\underline{e}_j^T}{\alpha_j}]}_{E} A^{-1} = EA^{-1}$$

"Product Form of the Inverse (PFI)"

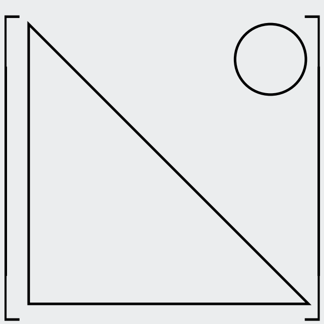
- Modern: LU and QR decomposition

Decomposition

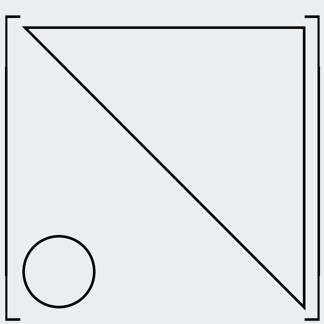
- Special matrices
  - Block diagonal – useful in modeling large loosely-connected systems



  - Orthogonal ⇒ $Q^{-1}=Q^T$
    - ❖ $\underline{q}_i^T \underline{q}_j = 0,\ \forall\, i \neq j,\ \ \underline{q}_j^T \underline{q}_j = 1$
    - ❖ Very useful in solving linear systems and in solving LP via revised simplex method
  - Lower triangular



  - Upper triangular



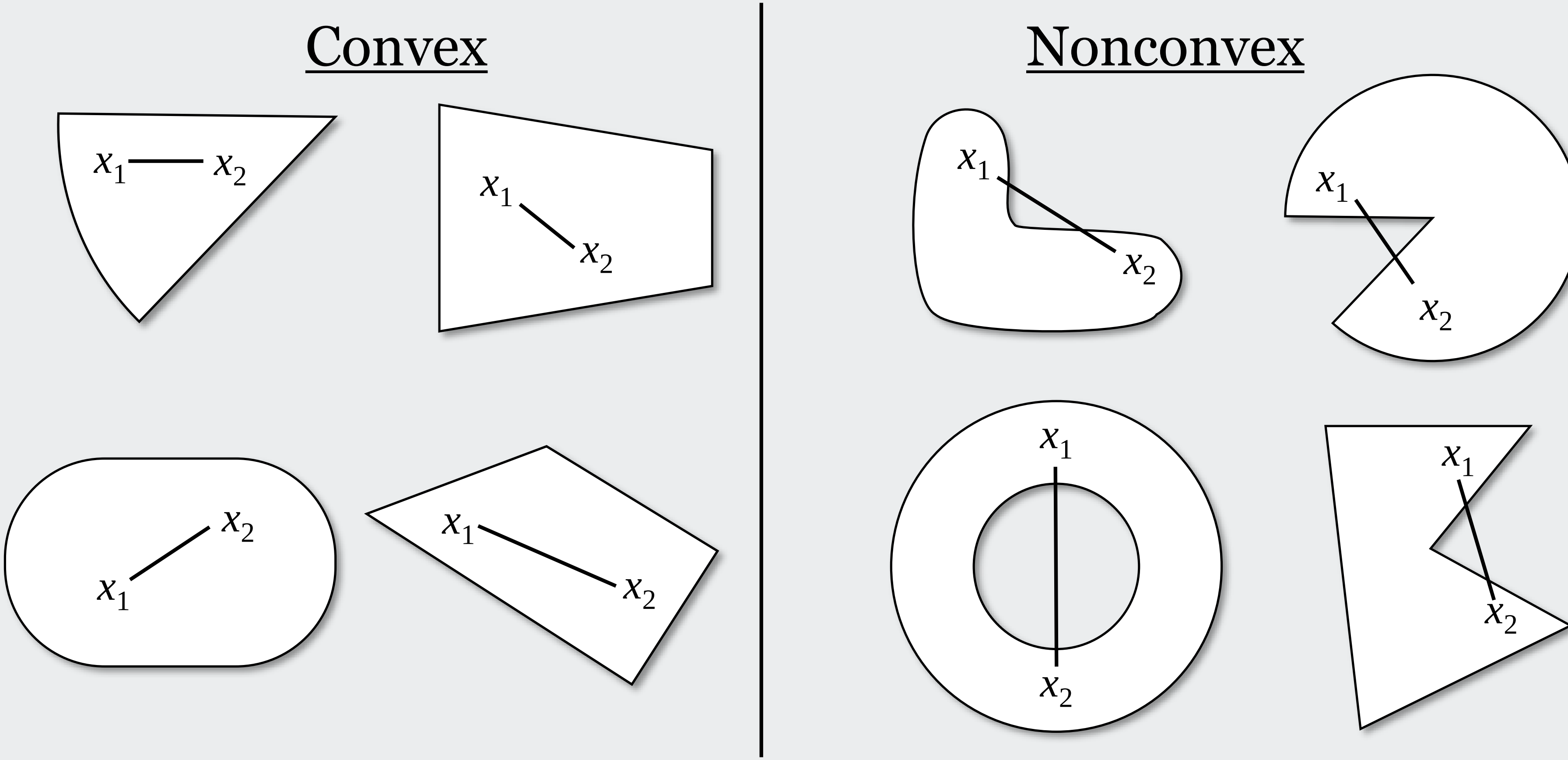
LU *and* QR decomposition

- Solution of $A\underline{x} = \underline{b}$ when A is square and has full rank
  - LU decomposition ⇒ write $A = LU$
    - Solve $L\underline{y} = \underline{b}$ via **Forward Elimination**
    - Solve $U\underline{x} = \underline{y}$ via **Backward Substitution**
  - QR decomposition ⇒ $A = QR$ where R is upper triangular
    - Solve $R\underline{x} = Q^T \underline{b}$ via **Backward Substitution**

- In Lecture 3, we will discuss how to update L and U (or Q and R) when the matrix is modified by removing a column and inserting a new one in its place when we talk about basis updates

Convex analysis – Convex sets

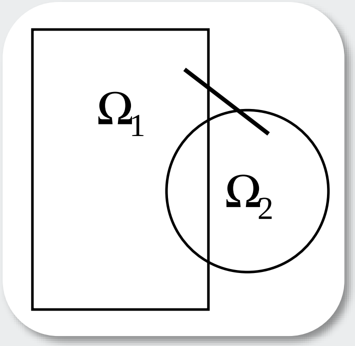
- A set $\Omega \in R^n$ is convex if for any two points x_1 and x_2 in the set Ω, the line segment joining x_1 and x_2 is also in Ω

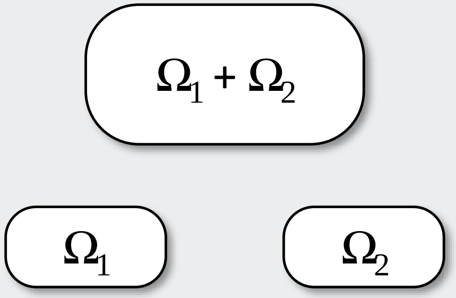


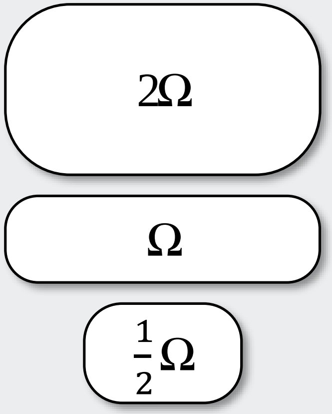
- **A convex set is one whose boundaries do not bulge inward or do not have indentations**

Examples of convex sets

- Examples:
  - A hyperplane $\underline{a}^T \underline{x} = b$ is a convex set
  - A closed half space
    - $H_{c+} = \{ \underline{x} \mid a^T \underline{x} \geq b \}$
    - $H_{c-} = \{ \underline{x} \mid a^T \underline{x} \leq b \}$
  - $\cap \Omega_i$ is convex
  - $\cup \Omega_i$ need not be convex
  - Sums and differences of convex sets are convex
  - Expansions or contractions of convex sets are convex







  - Empty set is convex

Convex cone and convex combination

- Useful results:
  - Intersection of hyperplanes is convex
  - Intersection of halfspaces is convex
    - e.g., $x_1 + x_2 \leq 1;\ x_1 \geq 0,\ x_2 \geq 0$
- Set of intersection of m closed halfspaces is called a **convex polytope** $\Rightarrow$ set of solutions to $A\underline{x} \leq \underline{b}$ or $A\underline{x} \geq \underline{b}$ is a convex polytope
- A bounded polytope is called a **polyhedron**
- **Convex cone:** $\underline{x} \in \text{cone} \Rightarrow \lambda \underline{x} \in \text{cone}\ \forall \lambda \geq 0$
- **Convex combination:** given a set of points $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_k$, $\underline{x} = \alpha_1 \underline{x}_1 + \alpha_2 \underline{x}_2 + \ldots + \alpha_k \underline{x}_k$ such that $\alpha_1 + \alpha_2 + \ldots + \alpha_k = 1$, $\alpha_i \geq 0$ is termed the convex combination of $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_k$
- A point $\underline{x}$ in a convex set Ω is an extreme point (corner) if there are no two points $x_1, x_2 \in \Omega$ such that $\underline{x} = \alpha \underline{x}_1 + (1 - \alpha)\underline{x}_2$ for any $0 < \alpha < 1$

Convex hull and convex polyhedron

- A closed convex hull C is a convex set such that every point in C is a convex combination of its extreme points, i.e.,

$$\underline{x} = \sum_{i=1}^{k} \alpha_i \underline{x}_i$$

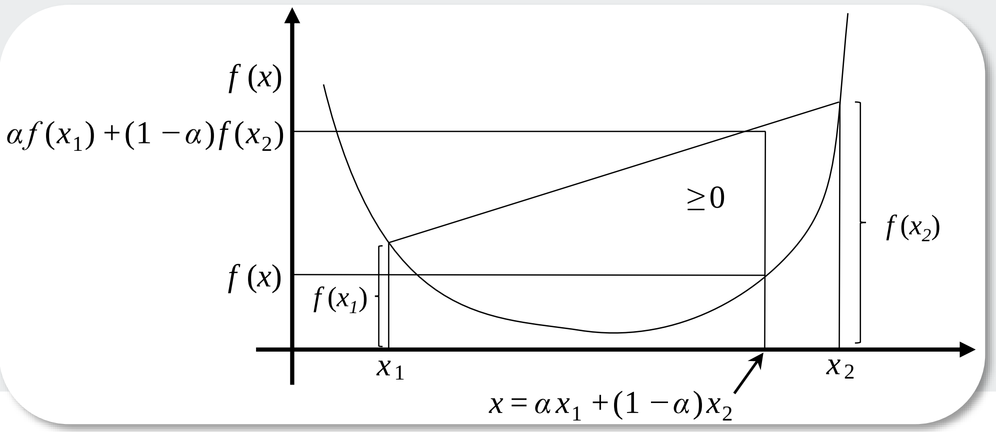
- In particular, a convex polyhedron can be thought of as:
  - The intersection of a finite number of closed half spaces
  - (or) as the convex hull of its extreme points
- Convex polyhedrons play an important role in LP
  - We will see that we need to look at only a finite number of extreme points
  - This is what makes LP lie on the border of continuous and discrete optimization problems

Convex functions

- Consider $f(\underline{x}): \Omega \rightarrow R$, $f(\underline{x})$ a scalar function
- $f(\underline{x})$ is a convex function on the convex set Ω if for any two points $\underline{x}_1, \underline{x}_2 \in \Omega$

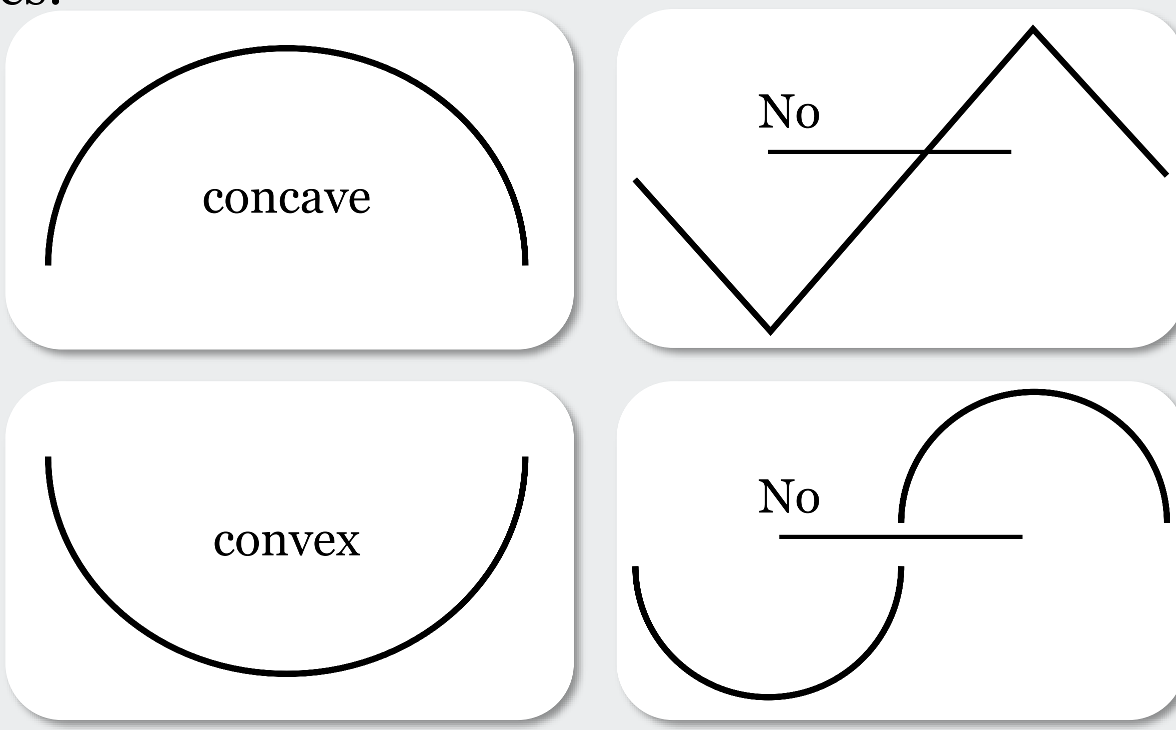
$$f\left(\alpha \underline{x}_1 + (1-\alpha)\underline{x}_2\right) \leq \alpha f\left(\underline{x}_1\right) + (1-\alpha) f\left(\underline{x}_2\right); \; 0 \leq \alpha \leq 1$$

- A convex function bends up
- A line segment (chord, secant) between any two points never lies below the graph
- Linear interpolation between any two points $\underline{x}_1$ and $\underline{x}_2$ overestimates the function



Examples of convex functions

- Concave if $-f(\underline{x})$ is convex
- Examples:



- **Proof**: $f(\underline{x}) = \underline{c}^T \underline{x}$, a linear function is convex
- $f(\alpha \underline{c}^T \underline{x}_1 + (1-\alpha)\underline{c}^T \underline{x}_2) = \underline{c}^T \underline{x}$ holds with equality
- $f(\underline{x}) = \underline{x}^T Q \underline{x}$ is convex if Q is PD . . . HW problem

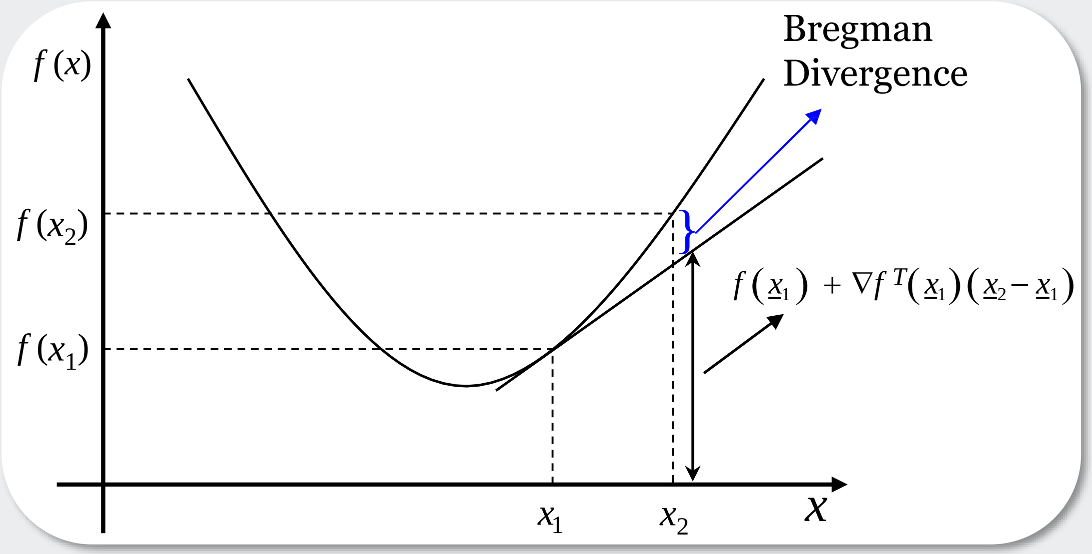
Properties of convex functions

- In general,

$$f(\alpha_1 \underline{x}_1 + \alpha_2 \underline{x}_2 + \cdots + \alpha_n \underline{x}_n) = f\left(\sum_i \alpha_i \underline{x}_i\right) \le \sum_i \alpha_i f(\underline{x}_i)$$

where $\sum_i \alpha_i = 1; \alpha_i \geq 0$… **Jensen's inequality**

  - Linear extrapolation underestimates the function



  - Hessian, the matrix of second partials, $H = \left[\frac{\partial^2 f}{\partial x_i \partial x_j}\right]$ is a positive semi-definite (PSD) or positive definite (PD) matrix

Level sets of convex functions

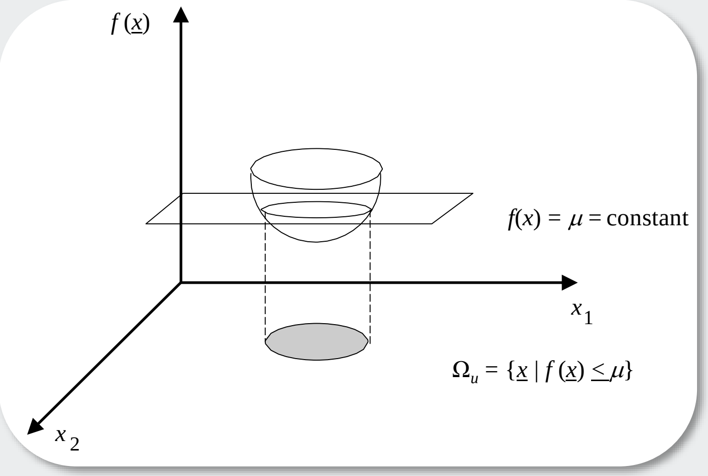
- Sum of convex functions is convex
- The epigraph or level set $\Omega_\mu = \{\underline{x} \mid f(\underline{x}) \leq \mu\}$ is convex, $\forall \mu$, if $f(\underline{x})$ is convex
  - Proof:

$$\text{If } \underline{x}_1, \underline{x}_2 \in \Omega_\mu \Rightarrow f(\underline{x}_1),\ f(\underline{x}_2) \leq \mu$$

$$\text{Consider } \underline{x} = \alpha \underline{x}_1 + (1-\alpha)\underline{x}_2$$

$$f(\alpha \underline{x}_1 + (1-\alpha)\underline{x}_2) \leq \alpha f(\underline{x}_1) + (1-\alpha) f(\underline{x}_2) \leq \mu$$

$$\Rightarrow \underline{x} \in \Omega_\mu$$

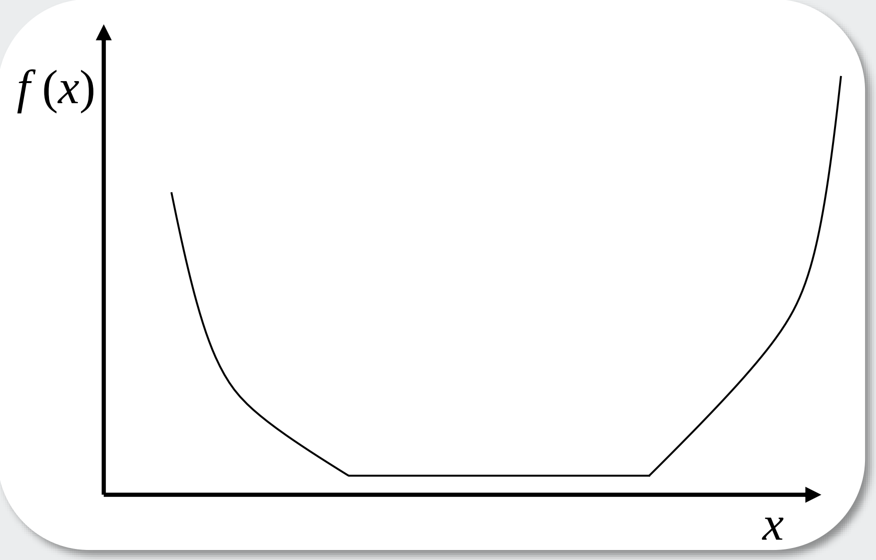


Convex programming problem (CPP)

- $\min\ f(\underline{x}) \ldots f$ is convex, such that $A\underline{x} = \underline{b}$, $g_i(\underline{x}) \geq 0$;
  $i = 1, 2, \ldots, p$; g_i concave $\Rightarrow -g_i$ convex
- $\Omega_i = \{\underline{x} \mid -g_i(\underline{x}) \leq 0\} = \{\underline{x} \mid g_i(\underline{x}) \geq 0\} \Rightarrow$ convex
- $\Omega_\mu = \{\underline{x} \mid f(\underline{x}) \leq \mu\}$ is convex
- $A\underline{x} = \underline{b} \Rightarrow$ intersection of hyperplanes $\Rightarrow$ convex set $\Omega_A \Rightarrow$
  $$\Omega = \bigcap \Omega_i \bigcap \Omega_\mu \bigcap \Omega_A \text{ is convex}$$
- Key property of CPP: **local optimum ⇔ global optimum**
- Suppose $\underline{x}^*$ is a local minimum, but $\underline{y}$ is a global minimum
- Consider $\underline{x} = \alpha\underline{x}^* + (1-\alpha)\underline{y} \in \Omega_\mu$
- Convexity $\Rightarrow f(\alpha\underline{x}^* + (1-\alpha)\underline{y}) \leq \alpha f(\underline{x}^*) + (1-\alpha)f(\underline{y}) \leq f(\underline{x}^*)$
  $\Rightarrow x^*$ is not a local optimum $\Rightarrow$ a contradiction

LP = special case of CPP

- Local optima must be bunched together as shown



- General LP problem is a special case of CPP

$$\begin{aligned} &\min \underline{c}^T \underline{x} \\ &\text{s.t. } \underline{a}_i^T \underline{x} = b_i, i \in E \\ &\quad \underline{a}_i^T \underline{x} \geq b_i, i \in I \\ &\quad x_i \geq 0, i \in P \end{aligned}$$

⇒ Local optimum and global optimum must be the <u>same</u>

Summary

- Course Objectives
- Optimization problems
  - Classification
  - Measures of complexity of algorithms
- Background on Matrix Algebra
  - Matrix-vector notation
  - Matrix-vector product
  - Linear subspaces associated with an $m \times n$ matrix A
  - LU and QR decompositions to solve $A\underline{x} = \underline{b}$, A is $n \times n$
- Convex analysis
  - Convex sets
  - Convex functions
  - Convex programming problem
- LP is a special case of convex programming problem
  - Local optimum ≡ global optimum