Carnegie Mellon

18-660: Numerical Methods for Engineering Design and Optimization

Xin Li
Department of ECE
Carnegie Mellon University
Pittsburgh, PA 15213

Electrical & Computer ENGINEERING

Overview

- **Convex Analysis**
  - Convex function
  - Convex set
  - Convex optimization

Ordinary Least-Squares Regression

- Solve over-determined linear equation by optimization

$$\text{M samples}\left\{\underbrace{\begin{bmatrix} \\ A \\ \\ \end{bmatrix}}_{\text{N coefficients}} \cdot X = \begin{bmatrix} \\ B \\ \\ \end{bmatrix}\right. \qquad (M > N)$$

⇩

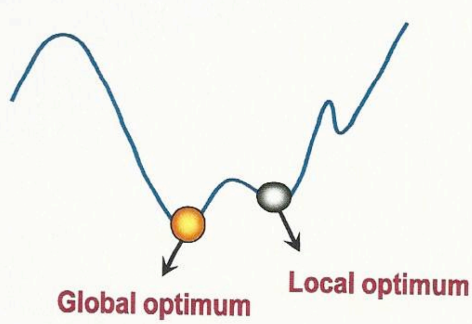
$$\boxed{\min_X \quad \|A \cdot X - B\|_2^2}$$

Unconstrained Nonlinear Programming

- Nonlinear cost function without constraints

$$\min_X \quad \|A \cdot X - B\|_2^2$$

- General nonlinear optimization is difficult to solve



Unconstrained Quadratic Programming

$$\min_{X} \quad \|A \cdot X - B\|_2^2$$

- **However, ordinary least-squares regression is different from general nonlinear programming**

1

- 2 **Optimization cost is a quadratic function of X and it is always non-negative for any given X**
  - This is a unique property that enables us to solve least-squares regression efficiently

Positive Semi-Definite

- **If a quadratic function $X^T A_Q X$ is always non-negative, the quadratic coefficient matrix A_Q is positive semi-definite**
  - Assume that A_Q is symmetric so that its eigenvalues are real
  - Any asymmetric A_Q can be converted to a symmetric one

$$[x]^T [A_Q][x]$$

$$x^T A_Q x = (x^T A_Q x)^T = x^T A_Q^T x$$

$$= \frac{1}{2} x^T A_Q x + \frac{1}{2} x^T A_Q^T x = x^T \underbrace{\left[\frac{1}{2}(A_Q + A_Q^T)\right]}_{\text{symmetric}} x$$

Positive Semi-Definite

- **Simple example:**

$$A_Q = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$x^T A_Q x = [x_1, x_2] \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = [0 \;\; x_1] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1 x_2$$

$$\frac{1}{2}(A_Q + A_Q^T) = \frac{1}{2}\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{bmatrix}$$

$$x^T\left[\frac{1}{2}(A_Q + A_Q^T)\right]x = [x_1, x_2] \begin{bmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$= \left[\frac{1}{2}x_2 \;\; \frac{1}{2}x_1\right] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{1}{2}x_2 x_1 + \frac{1}{2}x_1 x_2 = x_1 x_2$$

Positive Semi-Definite

- **A_Q is positive semi-definite if and only if all eigenvalues of A_Q are non-negative (necessary and sufficient condition)**

- **Eigenvalue decomposition**

$$A_Q \cdot \underset{\text{Eigenvector}}{V_i} = V_i \underset{\text{eigenvalue}}{\lambda_i}$$

A_Q is symmetric

$\Rightarrow$ All eigenvalues are real

$\Rightarrow$ All eigenvectors are real and orthogonal

Positive Semi-Definite

- **Eigenvalue decomposition**

$$A_Q \cdot V_i = V_i \cdot \lambda_i \qquad V = [V_1 \quad V_2 \quad \cdots] \qquad \Sigma = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \end{bmatrix}$$

$$A_Q \cdot V = A_Q \cdot [V_1\ V_2 \ldots] = [A_Q V_1\ A_Q V_2 \ldots]$$

$$= [V_1 \lambda_1\ V_2 \lambda_2 \ldots]$$

$$= \underbrace{[V_1\ V_2 \ldots]}_{V} \underbrace{\begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \end{bmatrix}}_{\Sigma}$$

$$A_Q \cdot V = V \cdot \Sigma \qquad V^T V = I$$

$$A_Q \cdot V \cdot V^T = V \cdot \Sigma V^T \qquad V V^T = I$$

Positive Semi-Definite

- **If one of the eigenvalues of A_Q is negative**

$$A_Q = V \cdot \Sigma \cdot V^T \quad where \quad V^T V = I$$

$$x^T A_Q \cdot x = x^T \cdot V \cdot \Sigma \cdot V^T x = (V^T x)^T \cdot \Sigma \cdot (V^T x)$$

$$\{[(V^T x)^T] \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \end{bmatrix} [V^T x$$

Positive Semi-Definite

- If one of the eigenvalues of A_Q is negative

$$(V^TX)^T \cdot \begin{bmatrix} \times & & & \\ & \times & & \\ & & \ddots & \\ & & & -\varepsilon \end{bmatrix} \cdot (V^TX)$$

$V^Tx = [0 \quad 0 \ldots \quad 0]^T$

$[0 \ldots 0 \quad 1] \cdot \begin{bmatrix} x & & \\ & \ddots & \\ & & x & \\ & & & -\varepsilon \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} = -\varepsilon \geq 0$

x^TA_Qx is negative

Positive Semi-Definite

- **If a quadratic function $X^TA_QX + B_Q^TX + C_Q$ is always non-negative (for any X), all eigenvalues of A_Q are non-negative**
  - I.e., A_Q is positive semi-definite
  - Why? (You can prove this conclusion by following the steps of eigenvalue decomposition)

- **The quadratic coefficient matrix for the least-squared error $err(X) = ||AX-B||_2^2$ is positive semi-definite**

Positive Semi-Definite

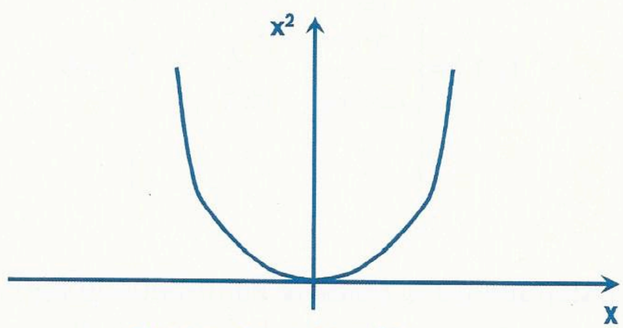
- The reverse statement is NOT true
- Even if A_Q is positive semi-definite, $f(X) = X^T A_Q X + B_Q^T X + C_Q$ can be either positive or negative

$f(x) = x^2 - 1 \qquad A_Q = 1$

$f(x=0) = 0 - 1 = -1 < 0$

Convex Function

- Positive semi-definite quadratic functions have special properties that can be utilized by nonlinear optimization
- A simple one-dimensional example $f(x) = x^2$

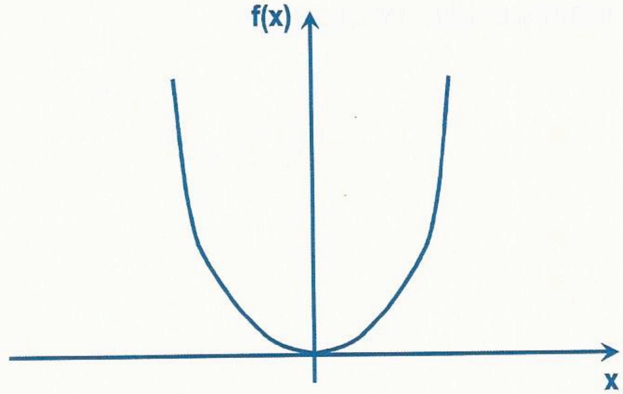


The function $f(X) = x^2$ is convex

Convex Function

- f(X) is convex, if for all *vectors* X_1, X_2 and $0 \le \alpha \le 1$, we have

$$f[\alpha \cdot X_1 + (1-\alpha)\cdot X_2] \le \alpha \cdot f(X_1) + (1-\alpha)\cdot f(X_2)$$



A one-dimensional convex example

Convex Function

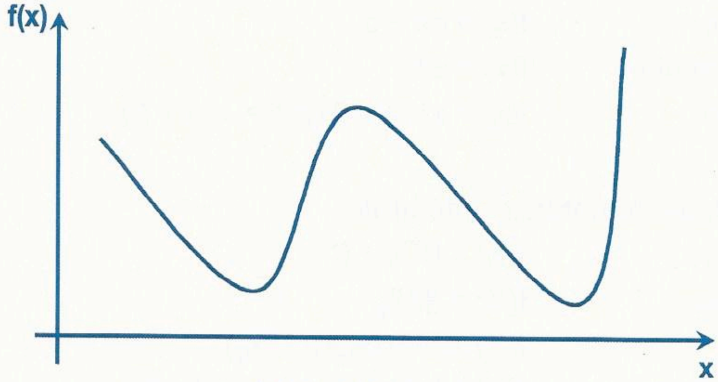
- **Second-order sufficient condition for convexity**
  - Not a necessary condition – convex function might not be smooth, and Hessian matrix might not exist

$$\nabla^2 f(X) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} & \cdots \\ \vdots & \vdots & \end{bmatrix} \succ= 0$$

Hessian matrix is positive semi-definite for ALL X
(Hessian matrix depends on X)

Convex Function

- To guarantee convexity, Hessian matrix must be positive semi-definite for **ALL** X

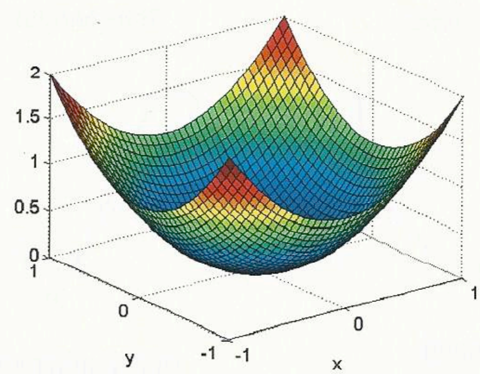


f(x) is NOT convex!

Convex Function

- A quadratic function $f(X) = X^T A_Q X + B_Q^T X + C_Q$ is convex if and only if A_Q is positive semi-definite

$$\nabla^2 f(X) = 2A_Q \quad \text{Constant}$$



$f(x, y) = x^2 + y^2$

Convex Function

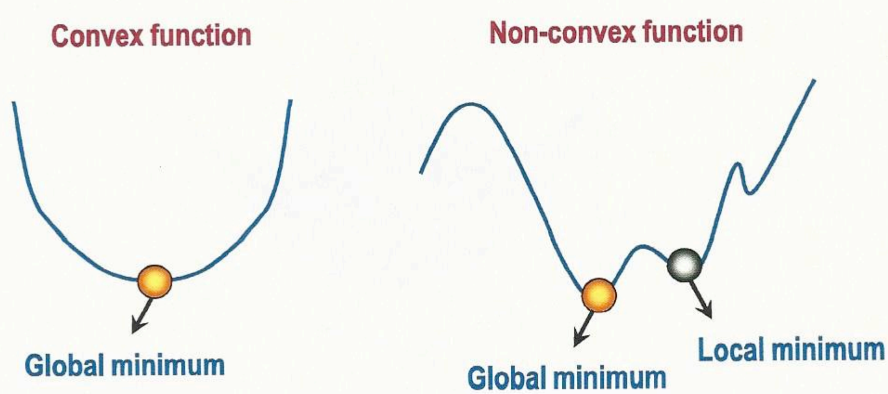
- **Several popular examples of convex functions**

- **One dimensional convex functions**
  - Linear: $f(x) = bx + c$
  - Exponential: $f(x) = e^{ax}$
  - Power: $f(x) = x^a$ ($a < 0$ or $a > 1$, $x > 0$)

- **N-dimensional convex functions**
  - Linear: $f(X) = B^T X + C$
  - L_2-norm: $f(X) = ||X||_2$
  - Max: $f(X) = \max(x_1, x_2, \ldots, x_N)$
  - Log-sum-exp: $f(X) = \log(e^{x1} + e^{x2} + \ldots + e^{xN})$
  - Log-determinant: $f(X) = -\log[\det(X)]$ (X is positive definite)

Convex Function

- **Minimizing a convex function is much easier than a general nonlinear programming**
  - Convex functions do not contain local minima



Constrained Nonlinear Optimization

- **The least-squares problem attempts to minimize a convex cost function without any constraints**

- **Many practical optimization problems contain both a cost function and a number of constraints**
  - E.g., minimax optimization for regression

$$\min_{X,t} \quad t \quad \longrightarrow \text{Cost function}$$

$$\text{S.T.} \quad \begin{cases} -t \le A(1,:) \cdot X - B_1 \le t \\ -t \le A(2,:) \cdot X - B_2 \le t \\ \vdots \\ -t \le A(M,:) \cdot X - B_M \le t \end{cases} \longrightarrow \text{Constraints}$$

Constrained Nonlinear Optimization

- **A general nonlinear programming problem has the form of:**

$$\min_{X} \quad f(X) \leftarrow$$

$$\text{S.T.} \quad \begin{cases} g_1(X) \le 0 \\ g_2(X) \le 0 \\ \vdots \end{cases}$$

Constraints

- **Equality constraints can be expressed in this general form**

$g_1(x) \ge 0 \Rightarrow -g_1(x) \le 0$

$$g(x) = 0 \quad \Rightarrow \begin{cases} g(x) \le 0 \\ g(x) \ge 0 \Leftrightarrow -g(x) \le 0 \end{cases}$$

Constrained Nonlinear Optimization

- A point X is **feasible** if it satisfies all constraints:

$$\begin{cases} g_1(X) \le 0 \\ g_2(X) \le 0 \\ \vdots \end{cases}$$

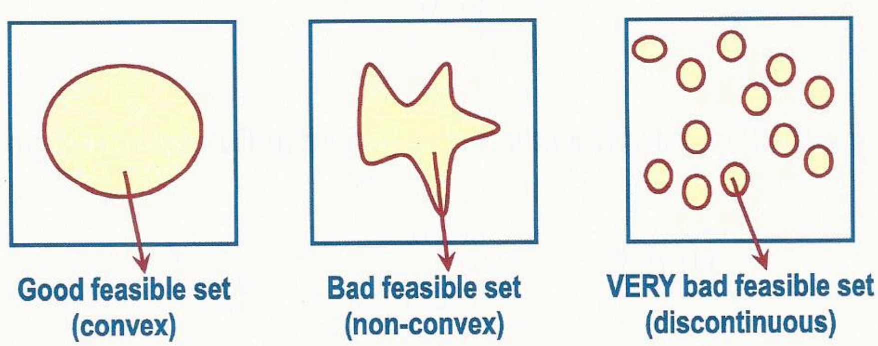
- The set of all feasible points is called the **feasible set**, or the **constraint set**

- An optimization is said to be feasible, if the corresponding feasible set is non-empty

Constrained Nonlinear Optimization

- Feasible set plays an important role in nonlinear optimization

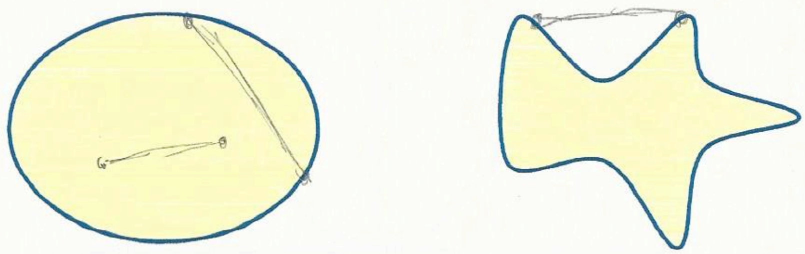
- Even if the cost function is convex, nonlinear optimization can still be difficult given a "bad" feasible set



Convex Set

- **A set D is convex, if for all $X_1, X_2 \in D$ and $0 \le \alpha \le 1$, we have**

$$\alpha \cdot X_1 + (1-\alpha) \cdot X_2 \in D$$



Convex **Non-convex**

Convex Set

- **Several popular examples of convex sets**
  - Hyperplane: $\{X \mid B^T X = C\}$
  - Polytope: $\{X \mid B^T X \le C\}$
  - Ball: $\|X\|_2 \le C$
  - Positive semi-definite matrices (a non-trivial example):

$$\{X \mid X \in R^{N \times N}, X = X^T, X \succ= 0\}$$

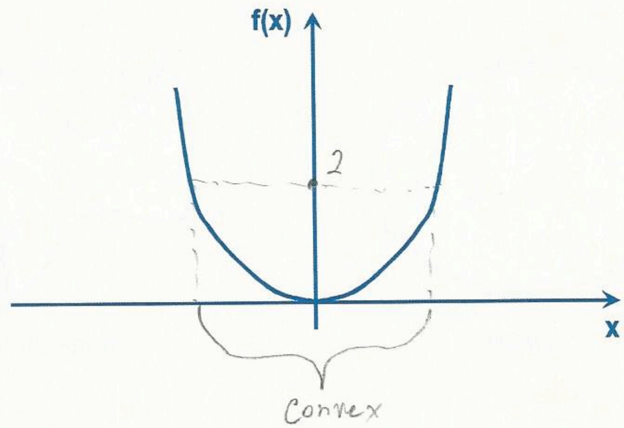
  - If X_1 and X_2 are positive semi-definite, their positive combination is also positive semi-definite

Convex Set

- Given a function f(X), the **α-sublevel set** is defined as:

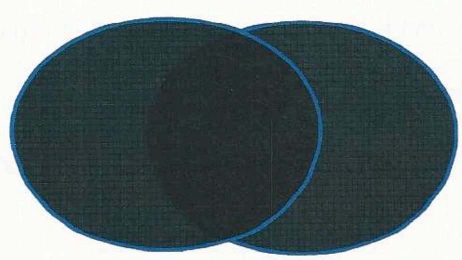
$$\{X \mid f(X) \le \alpha\}$$

- If f(X) is convex, its sublevel sets are convex



Convex Set

- Set convexity is preserved under intersection
  - If D_1 and D_2 are convex then $D_1 \cap D_2$ is convex



Convex Optimization

$$\min_{X} \; f(X) \rightarrow \text{Convex}$$

$$\text{S.T.} \begin{cases} g_1(X) \le 0 \\ g_2(X) \le 0 \\ \vdots \end{cases}$$

- **If all $g_i(X)$'s are convex, the constraint set is convex**
  - Constraint set is the intersection of all convex 0-sublevel sets
- **The minimization of a convex cost function over a convex constraint set is called convex optimization**

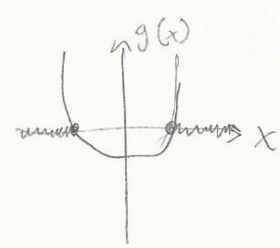
Convex Optimization

- **The following optimizations are NOT convex, even if f(X) and g(X) are both convex**

$$\max_{x} f(x)$$

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

$$\text{s.t. } g(x) \ge 0$$



Convex Optimization

- **Linear programming is a special case of convex optimization**
- **Most convex optimization with smooth cost function and constraints can be efficiently and robustly solved**
  - Decide if the optimization is feasible or infeasible
  - If feasible, provide the optimal solution
- **Several good convex solvers**
  - MOSEK (www.mosek.com)
  - CVX (www.stanford.edu/~boyd/cvx/)
  - More details on convex solver in future lectures...

Summary

- **Convex analysis**
  - Convex function
  - Convex set
  - Convex optimization

Carnegie Mellon

18-660: Numerical Methods for Engineering Design and Optimization

Xin Li
Department of ECE
Carnegie Mellon University
Pittsburgh, PA 15213

Electrical & Computer ENGINEERING

Overview

- Over-determined Linear Equation Solver
  - Pseudo-inverse
  - QR decomposition

Over-Determined Linear Equation

- **Fit an approximate function f(x) from sampling points**

$$f(x) \approx \alpha_1 \cdot b_1(x) + \alpha_2 \cdot b_2(x) + \cdots$$

$$\text{M samples} \left\{ \underbrace{\begin{bmatrix} \\ \quad A \quad \\ \\ \end{bmatrix}}_{\text{N coefficients}} \cdot \alpha = \begin{bmatrix} \\ B \\ \\ \end{bmatrix} \right.$$

(M > N)
Over-determined linear equation

Over-Determined Linear Equation

- **Solve over-determined linear equation**
  - No exact solution to satisfy all equations, but we can find the least-squares solution:

$$A \cdot \alpha = B$$

$$\min_{\alpha} \quad \|A \cdot \alpha - B\|_2^2$$

(Least-squares solution)

Over-Determined Linear Equation

- **There are two popular approaches to solve over-determined linear equations**

$$\text{M samples} \left\{ \underbrace{\begin{bmatrix} \\ \quad A \quad \\ \\ \end{bmatrix}}_{\text{N coefficients}} \right. \cdot \alpha = \begin{bmatrix} \\ B \\ \\ \end{bmatrix} \qquad (M > N)$$

Over-Determined Linear Equation

- **Solution 1**

$$A \cdot \alpha = B$$

$$\underbrace{\begin{bmatrix} A^T \end{bmatrix}}_{N \times M} \underbrace{\begin{bmatrix} \\ A \\ \\ \end{bmatrix}}_{M \times N} \underbrace{\begin{bmatrix} \alpha \end{bmatrix}}_{N \times 1} = \underbrace{\begin{bmatrix} A^T \end{bmatrix}}_{N \times M} \underbrace{\begin{bmatrix} \\ B \\ \\ \end{bmatrix}}_{M \times 1}$$

Over-Determined Linear Equation

- **Solution 1**

$$\begin{bmatrix} A^T \\ \text{N x M} \end{bmatrix} \cdot \begin{bmatrix} A \\ \text{M x N} \end{bmatrix} \cdot \alpha = \begin{bmatrix} A^T \\ \text{N x M} \end{bmatrix} \cdot \underset{\text{M x 1}}{\begin{bmatrix} B \end{bmatrix}}$$

$$\underbrace{(A^T A)}_{N \times N} \cdot \alpha = \underbrace{(A^T B)}_{N \times 1}$$

$$\alpha = (A^T A)^{-1} \cdot (A^T B)$$

Over-Determined Linear Equation

- **Proof of optimality**

$$\min_{\alpha} \quad \|A \cdot \alpha - B\|_2^2$$

$$F(\alpha) = \|A \cdot \alpha - B\|_2^2 = (A \cdot \alpha - B)^T \cdot (A \cdot \alpha - B)$$

$$F(\alpha) = \|A \cdot \alpha - B\|_2^2 = (\alpha^T \cdot A^T - B^T) \cdot (A \cdot \alpha - B)$$

$$F(\alpha) = \alpha^T A^T A \alpha - \alpha^T A^T B - B^T A \alpha + B^T B$$

Over-Determined Linear Equation

- **Proof of optimality**

$$F(\alpha)=\alpha^T A^T A\alpha-\alpha^T A^T B-B^T A\alpha+B^T B$$

$$\alpha^T A^T B=\left(\alpha^T A^T B\right)^T=B^T A\alpha$$

$$F(\alpha)=\alpha^T A^T A\alpha-2B^T A\alpha+B^T B$$

$$\frac{\partial}{\partial\alpha}F(\alpha)=\frac{\partial}{\partial\alpha}\left(\alpha^T A^T A\alpha\right)-\frac{\partial}{\partial\alpha}\left(2B^T A\alpha\right)=0$$

Over-Determined Linear Equation

- **Proof of optimality**

$$\frac{\partial}{\partial\alpha}F(\alpha)=\frac{\partial}{\partial\alpha}\left(\alpha^T \underbrace{A^T A}_{\mathbf{W}}\alpha\right)-\frac{\partial}{\partial\alpha}\left(2\underbrace{B^T A}_{\mathbf{P^T}}\alpha\right)=0$$

$$W=A^T A \qquad \alpha^T W\alpha=\sum_i\sum_j w_{ij}\alpha_i\alpha_j$$

$$\frac{\partial}{\partial\alpha_n}\left(\alpha^T W\alpha\right)=\frac{\partial}{\partial\alpha_n}\left(w_{nn}\alpha_n^2+\sum_{i\neq n}w_{in}\alpha_i\alpha_n+\sum_{j\neq n}w_{nj}\alpha_n\alpha_j\right)$$

$$=2w_{nn}\alpha_n+\sum_{i\neq n}w_{in}\alpha_i+\sum_{j\neq n}w_{nj}\alpha_j$$

$$=2w_{nn}\alpha_n+\sum_{j\neq n}2w_{nj}\alpha_j$$

$$=\sum_j 2w_{nj}\alpha_j$$

Over-Determined Linear Equation

- **Proof of optimality**

$$\frac{\partial}{\partial \alpha}F(\alpha)=\frac{\partial}{\partial \alpha}\left(\alpha^T \underbrace{A^T A}_{W}\alpha\right)-\frac{\partial}{\partial \alpha}\left(2\underbrace{B^T A}_{P^T}\alpha\right)=0$$

$$W = A^T A \qquad \frac{\partial}{\partial \alpha_n}\left(\alpha^T W \alpha\right)=\sum_j 2w_{nj}\alpha_j$$

$$\frac{\partial}{\partial \alpha}\left(\alpha^T W \alpha\right)=\begin{bmatrix}\frac{\partial}{\partial \alpha_1}\left(\alpha^T W \alpha\right)\\ \frac{\partial}{\partial \alpha_2}\left(\alpha^T W \alpha\right)\\ \vdots\end{bmatrix}=\begin{bmatrix}\sum_j 2w_{1j}\alpha_j\\ \sum_j 2w_{2j}\alpha_j\\ \vdots\end{bmatrix}=2W\alpha=2A^T A\alpha$$

Over-Determined Linear Equation

- **Proof of optimality**

$$\frac{\partial}{\partial \alpha}F(\alpha)=\frac{\partial}{\partial \alpha}\left(\alpha^T \underbrace{A^T A}_{W}\alpha\right)-\frac{\partial}{\partial \alpha}\left(2\underbrace{B^T A}_{P^T}\alpha\right)=0$$

$$P=\left(B^T A\right)^T = A^T B \qquad 2P^T\alpha=\sum_i 2p_i\alpha_i$$

$$\frac{\partial}{\partial \alpha_n}\left(2P^T\alpha\right)=\frac{\partial}{\partial \alpha_n}\left(\sum_i 2p_i\alpha_i\right)=2p_i$$

$$\frac{\partial}{\partial \alpha}\left(2P^T\alpha\right)=\begin{bmatrix}\frac{\partial}{\partial \alpha_1}\left(2P^T\alpha\right)\\ \frac{\partial}{\partial \alpha_2}\left(2P^T\alpha\right)\\ \vdots\end{bmatrix}=\begin{bmatrix}2p_1\\ 2p_2\\ \vdots\end{bmatrix}=2P=2A^T B$$

Over-Determined Linear Equation

- **Proof of optimality**

$$\frac{\partial}{\partial\alpha}F(\alpha)=\frac{\partial}{\partial\alpha}\big(\alpha^T \underbrace{A^T A}_{\mathbf{W}}\alpha\big)-\frac{\partial}{\partial\alpha}\big(2\underbrace{B^T A}_{\mathbf{P^T}}\alpha\big)=0$$

$$\frac{\partial}{\partial\alpha}\left(\alpha^T W\alpha\right)=2A^T A\alpha \qquad \frac{\partial}{\partial\alpha}\left(2P^T\alpha\right)=2A^T B$$

$$2A^T A\alpha-2A^T B=0$$

$$\alpha=\left(A^T A\right)^{-1}\cdot\left(A^T B\right)$$

Over-Determined Linear Equation

- **Solution 2**

$$A\cdot\alpha=B$$

$$\underset{M\times N}{\begin{bmatrix} A \end{bmatrix}}=\underset{M\times N}{\begin{bmatrix} Q \end{bmatrix}}\underset{N\times N}{\begin{bmatrix} R \end{bmatrix}}$$

$$Q=\begin{bmatrix} Q_1 & Q_2 & \cdots \end{bmatrix}$$

Q: orthogonal matrix $Q^T Q=I$

$$Q_i^T Q_j=\begin{cases} 0 & (i\neq j) \\ 1 & (i=j) \end{cases}$$

R: upper triangular matrix

$$R=\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ 0 & & \times & \times \\ & & & \times \end{bmatrix}$$

QR decomposition

QR Decomposition

$$\begin{bmatrix} A_1 & A_2 & \cdots \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 & \cdots \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & \cdots \\ & r_{22} & \cdots \\ & & \ddots \end{bmatrix}$$

$A_1 = Q_1 \cdot r_{11}$ $\qquad Q_1^T Q_1 = 1$

$\|A_1\|_2 = \|Q_1 r_{11}\|_2$

$= \|Q_1\|_2 \cdot |r_{11}|$ $\qquad Q_1 = \dfrac{A_1}{r_{11}}$

$= |r_{11}|$

$r_{11} = \|A_1\|_2$

QR Decomposition

$$\begin{bmatrix} A_1 & A_2 & \cdots \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 & \cdots \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & \cdots \\ & r_{22} & \cdots \\ & & \ddots \end{bmatrix}$$

$A_2 = Q_1 r_{12} + Q_2 r_{22}$ $\qquad Q_1^T Q_2 = 0$ $\qquad Q_2^T Q_2 = 1$

$Q_1^T A_2 = Q_1^T (Q_1 r_{12} + Q_2 r_{22})$

$= Q_1^T Q_1 r_{12} + Q_1^T Q_2 r_{22}$

$= 1 \cdot r_{12} + 0$

$= r_{12}$

QR Decomposition

$$\begin{bmatrix} A_1 & A_2 & \cdots \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 & \cdots \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & \cdots \\ & r_{22} & \cdots \\ & & \ddots \end{bmatrix}$$

$A_2 = Q_1 r_{12} + Q_2 r_{22}$ $\quad Q_1^T Q_2 = 0 \quad Q_2^T Q_2 = 1$

$A_2 - Q_1 r_{12} = Q_2 r_{22}$

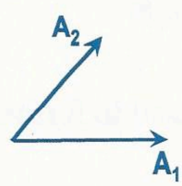
$\| A_2 - Q_1 r_{12} \|_2 = \| Q_2 \cdot r_{22} \|_2 \qquad Q_2 = \dfrac{A_2 - Q_1 r_{12}}{r_{22}}$

$= \| Q_2 \|_2 \cdot | r_{22} | = | r_{22} |$

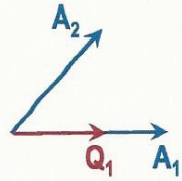
$r_{22} = \| A_2 - Q_1 r_{12} \|_2$

QR Decomposition

- **Geometrical interpretation** $\quad A = \begin{bmatrix} A_1 & \vdots & A_2 \end{bmatrix}$



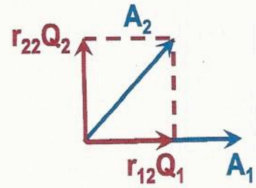
Start from two vectors A_1 and A_2



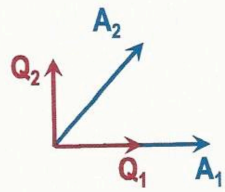
Normalized A_1 to determine Q_1

QR Decomposition

- **Geometrical interpretation**



Decompose A_2 into $r_{12}Q_1$ and $r_{22}Q_2$



Normalized $r_{22}Q_2$ to determine Q_2

QR Decomposition

- **It is referred to as classical Gram-Schmidt algorithm**
  - Q may not be orthogonal due to numerical errors

- **Modified Gram-Schmidt algorithm was proposed to further improve numerical stability**

- **More details can be found at**

Numerical Recipes: The Art of Scientific Computing, 2007

Over-Determined Linear Equation

- **Solution 2**

$$A \cdot \alpha = B \quad A = Q \cdot R$$

$Q \cdot R \cdot \alpha = B$

$\underbrace{Q^T Q}_{I} R \alpha = Q^T B$

$R \cdot \alpha = Q^T B$

$\alpha = R^{-1} \cdot (Q^T B)$

$\begin{bmatrix} A \\ M \times N \end{bmatrix}$ $\begin{bmatrix} Q \\ M \times N \end{bmatrix}$

$\begin{bmatrix} R \\ N \times N \end{bmatrix}$ $\begin{bmatrix} B \\ M \times 1 \end{bmatrix}$

Over-Determined Linear Equation

- **In theory, these two approaches yield identical results**

$$A \cdot \alpha = B \quad A = Q \cdot R$$

$\alpha = (A^T A)^{-1} \cdot (A^T B)$

$\alpha = R^{-1} (Q^T B)$

$A^T A = (QR)^T \cdot (QR) = R^T \underbrace{Q^T Q}_{I} R = R^T R$

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

$(A^T B) = (QR)^T B = R^T Q^T B$

$\alpha = R^{-1} \cdot \underbrace{R^{-T} \cdot R^T}_{I} \cdot Q^T B = R^{-1} Q^T B$

$(A^T A)^{-1} A^T$: pseudo-inverse of A

Over-Determined Linear Equation

- **In practice, Solution 2 is preferred since it is more robust to numerical noise**
  - MATLAB uses a modified version of solution 2 to solve over-determined linear equations
  - You can use a simple command "α = A\B" to solve over-determined linear equations in MATLAB

- **To understand the numerical difference, we check the condition number of both approaches**

Over-Determined Linear Equation

- **A simple example based on L_1 matrix norm**

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 10^{-5} \\ 10^{-10} & 10^{-10} \end{bmatrix}$$

$A \cdot \alpha = B$

$\alpha = (A^T A)^{-1} \cdot (A^T B)$

$\alpha = R^{-1} \cdot (Q^T B)$

$$A^T A = \begin{bmatrix} 1 & 10^{-20} \\ 10^{-20} & 10^{-10} \end{bmatrix} \qquad R = \begin{bmatrix} 1 & 10^{-20} \\ 0 & 10^{-5} \end{bmatrix}$$

$$k(A^T A) = 10^{10} \qquad k(R) = 10^{5}$$

Solution 1 | **Solution 2**

Over-Determined Linear Equation

- **Comparison on two solutions**

Solution 1 $\alpha = \left(A^T A\right)^{-1} \cdot \left(A^T B\right)$

Solution 2 $\alpha = R^{-1} \cdot \left(Q^T B\right)$

$$k\left(A^T A\right) >> k(R)$$

Solution 2 is more numerically robust

Summary

- **Over-determined linear equation solver**
  - Pseudo-inverse
  - QR decomposition