Math Notes

Linear Algebra and Optimisation

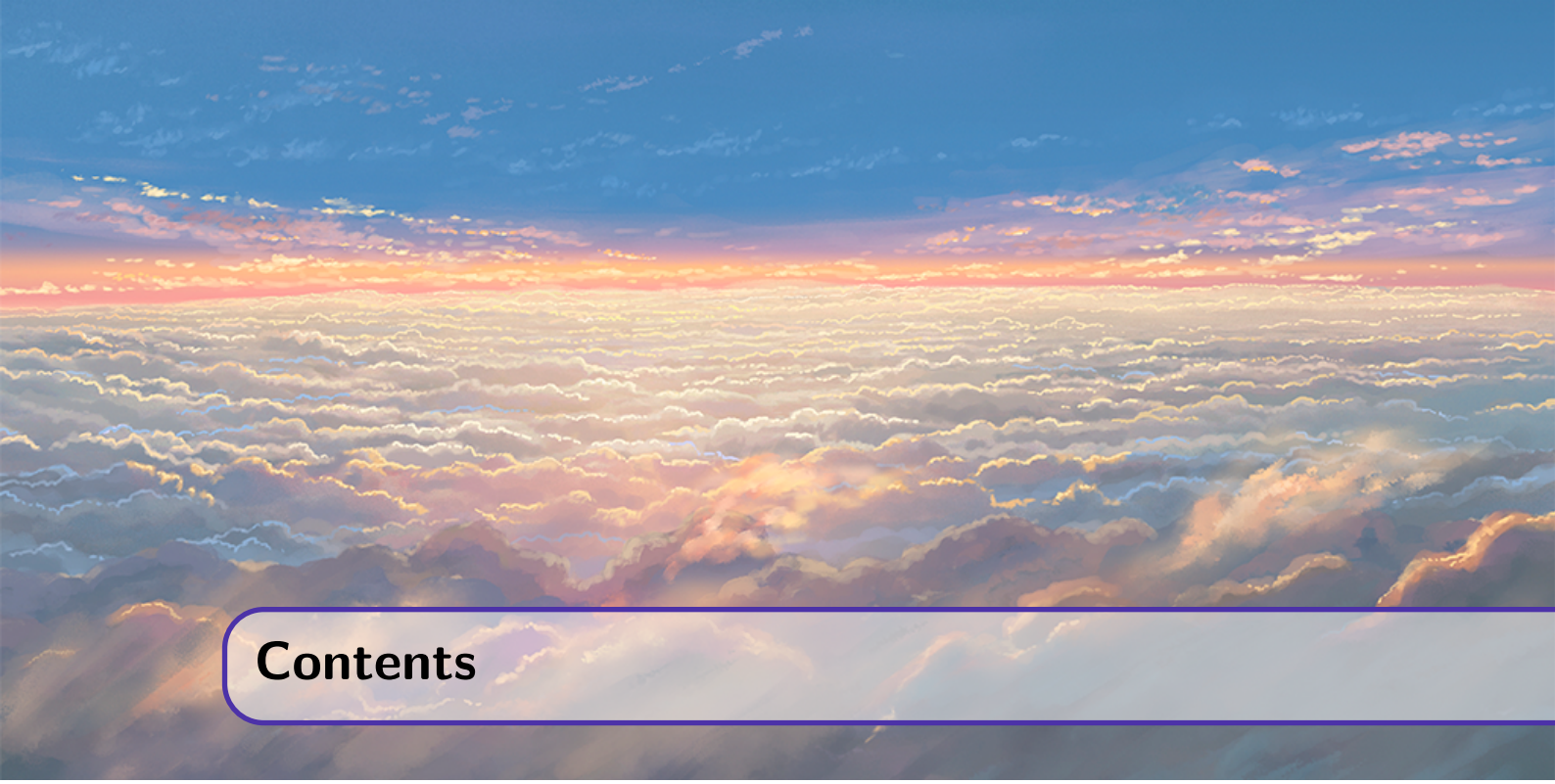
Xia Wenxuan

Written by Xia Wenxuan, 2021

https://github.com/gegeji

All the material in this document is taken from my textbook or video material, and some material uses optical character recognition (OCR) to aid input. It may contains typographical or content inaccuracies. This document is for my personal study only. I am not responsible for the accuracy of the content of the text.

Edited and Revised on November 19, 2021



Contents

Vectors

1. 对向量的介绍

1.1 Vector

Definition 1.1.1 — Vector. 一个有序的数字列表.

$$\begin{bmatrix} -1.1 \\ 0.0 \\ 3.6 \\ -7.2 \end{bmatrix} \text{或者} \quad \begin{pmatrix} -1.1 \\ 0.0 \\ 3.6 \\ -7.2 \end{pmatrix} \text{或者} \quad (-1.1, 0, 3.6, -7.2)$$

表中的数字是元素 (项、系数、分量). 元素的数量是向量的大小 (维数, 长度). 大小为 n 的向量称为n 维向量. 向量中的数字通常被称作标量.

用符号来表示向量, 比如 a , b, 一般小写字母表示. 其它表示形式 $\boldsymbol{g}, \vec{a}$

Definition 1.1.2 — n 维向量 a 的第 i 元素. n 维向量 a 的第 i i 元素表示为 a_i.
有时 i 指的是向量列表中的第 i 个向量.

Definition 1.1.3 — $a = b$. 对于所有 i, 如果有 $a_i = b_i$, 则称两个相同大小的向量 a 和 b 是相等的, 可写成 $a = b$

Definition 1.1.4 — stacked vector. 假设 b、c、d 是大小为 m、n、p 的向量

$$a = \begin{bmatrix} b \\ c \\ d \end{bmatrix}$$

$$a = (b_1, b_2, \ldots, b_m, c_1, c_2, \ldots, c_n, d_1, d_2, \ldots, d_p)$$

Definition 1.1.5 — 零向量. 所有项为 0 的 n 维向量表示为 0_n 或者 0.

Definition 1.1.6 — 全一向量. 所有项为 1 的 n 维向量表示为 $\mathbf{1}_n$ 或者 1.

Definition 1.1.7 — 单位向量. 当第 i 项为 1, 其余项为 0 时表示为 e_i

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Definition 1.1.8 — 稀疏向量. 如果一个向量的许多项都是 0, 该向量为稀疏 (Sparse) 的. 稀疏向量能在计算机上高效地存储和操作.

$\mathrm{nnz}(x)$ 是指向量 x 中非零的项数 (number of non-zeros), 有时用 ℓ_0 表示.

向量 $x = (x_1, x_2)$ 可以在二维中表示一个位置或一个位移、图像、单词统计、颜色等.

1.2 Vector Space

Definition 1.2.1 — 向量空间 V. 设 V 是非空子集, P 是一数域, 向量空间 V 满足:

1. 向量加法: $V + V \to V$, 记作 $\forall x, y \in V$, 则 $x + y \in V$ (加法封闭)
2. 标量乘法: $F \times V \to V$, 记作 $\forall x \in V, \lambda \in P$, 则 $\lambda x \in V$ (乘法封闭)

上述两个运算满足下列八条规则 ($\forall x, y, z \in V, \lambda, \mu \in P$)

1. $x + y = y + x$ (交换律)
2. $x + (y + z) = (x + y) + z$ (结合律)
3. V 存在一个零元素, 记作 0, $x + 0 = x$
4. 存在 x 的负元素, 记作 $-x$, 满足 $x + (-x) = 0$
5. $\forall x \in V$, 都有 $1x = x, 1 \in P$
6. $\lambda(\mu x) = (\lambda\mu)x$
7. $(\lambda + \mu)x = \lambda x + \mu x$
8. $\lambda(x + y) = \lambda x + \lambda y$

Corollary 1.2.1 向量空间也称为线性空间.

Corollary 1.2.2 如果 $x, y \in \mathbb{R}^2$, 则 $x + y \in \mathbb{R}^2, \lambda x \in \mathbb{R}^2 (\lambda \in \mathbb{R})$.

Definition 1.2.2 — 数域. 数的非空集合 P, 且其中任意两个数的和、差、积、商 (除数不为零) 仍属于该集合, 则称数集 P 为一个数域.

- **Example 1.1** 有理数 $\mathbb{Q}$ ∎
- **Example 1.2** 实数 $\mathbb{R}$

  $x, y \in \mathbb{R}, x = 1, y = 2\ x + y \in \mathbb{R}, x \times y \in \mathbb{R}$ ∎
- **Example 1.3** 复数 $\mathbb{C}$ ∎

1.3 向量运算

Definition 1.3.1 — 向量加法. n 维向量 a 和 b 可以相加, 求和形式表示为 $a + b$.

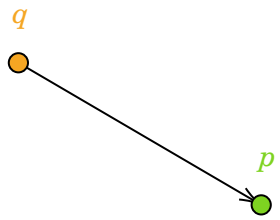
设向量 a, b, C 是向量空间 V 的元素, 即 $a, b, c \in V$.

1. 交换律: $a + b = b + a$
2. 结合律: $(a + b) + c = a + (b + c)$ (因此可写成 $a + b + c$)
3. $a + 0 = 0 + a = a$
4. $a - a = 0$

Corollary 1.3.1 — 向量位移相加. 如果二维向量 a 和 b 都表示位移, 则它们的位移之和为 $a+b$

■ **Example 1.4** 点 q 到点 p 的位移是 $p-q$.

Figure 1.1: The translation from q to p



■

Definition 1.3.2 — 标量与向量的乘法.

$$\beta a = \begin{bmatrix} \beta a_1 \\ \vdots \\ \beta a_n \end{bmatrix}$$

标量 β, γ 与向量 a、b 进行乘法, 有如下性质:

1. 结合律: $(\beta\gamma)a = \beta(\gamma a)$
2. 左分配律: $(\beta+\gamma)a = \beta a + \gamma a$
3. 右分配律: $\beta(a+b) = \beta a + \beta b$

Definition 1.3.3 — 线性组合. 对于向量 $a_1, \ldots, a_m$ 和标量 $\beta_1, \ldots, \beta_m$,

$$\beta_1 a_1 + \cdots + \beta_m a_m$$

是向量的线性组合. $\beta_1, \ldots, \beta_m$ 是该向量的系数.

■ **Example 1.5** 对于任何向量 $b \in \mathbb{R}^n$, 有如下等式

$$b = b_1 e_1 + \cdots + b_n e_n, b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

■

1.4 内积

Definition 1.4.1 — 内积. 在数域 $\mathbb{R}$ 上的向量空间 V, 定义函数 $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$, 满足:

1. $\langle a, a \rangle \geq 0, \forall a \in V$, 当且仅当 $a = 0$ 时 $\langle a, a \rangle = 0$
2. $\langle \alpha a + \beta b, c \rangle = \alpha \langle a, c \rangle + \beta \langle b, c \rangle, \forall \alpha, \beta \in \mathbb{R}$, 且 $a, b, c \in V$
3. $\langle a, b \rangle = \langle b, a \rangle, \forall a, b \in V$

函数 $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{R}$ 成为内积.

■ **Example 1.6** 在向量空间 $\mathbb{R}^n$ 上, 计算两个向量对应项相乘之后求和函数

$$\langle a, b \rangle = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n = a_b^T$$

where $a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \in \mathbb{R}^n$. ▪

Proof. $\langle a, a \rangle = a_1 a_1 + a_2 a_2 + \cdots + a_n a_n = \sum_{i=1}^n a_i^2 \geq 0, \langle a, a \rangle = 0$, 则 $a = 0$

$$\begin{aligned} \langle \alpha a + \beta b, c \rangle &= (\alpha a_1 + \beta b_1) c_1 + (\alpha a_2 + \beta b_2) c_2 + \cdots + (\alpha a_n + \beta b_n) c_n \\ &= \alpha \sum_{i=1}^n a_i c_i + \beta \sum_{i=1}^n b_i c_i \\ &= \alpha \langle a, c \rangle + \beta \langle b, c \rangle \end{aligned}$$

$$\langle a, b \rangle = a^T b = b^T a = \langle b, a \rangle$$

■

内积的性质：交换律、结合律、分配律.
交换律：$a^T b = b^T a$
结合律：$(\gamma a)^T b = \gamma \left(a^T b\right)$
分配律：$(a + b)^T c = a^T c + b^T c$

1.4.1 常用的内积等式

Corollary 1.4.1 — 选出第 i 项.

$$e_i^T a = a_i$$

Corollary 1.4.2 — 向量每一项之和.

$$\mathbf{1}^T a = a_1 + \cdots + a_n$$

Corollary 1.4.3 — 向量每一项的平方和.

$$a^T a = a_1^2 + \cdots + a_n^2$$

Corollary 1.4.4 — 向量元素的平均值.

$$(\mathbf{1}/n)^T a = (a_1 + \cdots + a_n)/n$$

Corollary 1.4.5 — Selective sum. Let b be a vector all of whose entries are either 0 or 1. Then

$$b^T a$$

is the sum of the elements in a for which $b_i = 1$.

Definition 1.4.2 — The sum of block vectors. If the vectors a and b are block vectors, and the corresponding blocks have the same sizes (in which case we say they *conform*), then

$$a^T b = \begin{bmatrix} a_1 \\ \vdots \\ a_k \end{bmatrix}^T \begin{bmatrix} b_1 \\ \vdots \\ b_k \end{bmatrix} = a_1^T b_1 + \cdots + a_k^T b_k$$

内积用途很广.

■ **Example 1.7 — 计算同时出现的项目数.**

$$a = (0,1,1,1,1,1,1), \quad b = (1,0,1,0,1,0,0)$$

Here we have $a^T b = 2$, which is the number of objects in both A and B (i.e., objects 3 and 5). ■

■ **Example 1.8 — Weights, features, and score.** When the vector f represents a set of *features* of an object, and w is a vector of the same size (often called a *weight vector*), the inner product $w^T f$ is the sum of the feature values, scaled (or weighted) by the weights, and is sometimes called a *score*. ■

■ **Example 1.9**

$$p(x) = c_1 + c_2 x + \cdots + c_{n-1} x^{n-2} + c_n x^{n-1}$$

Let t be a number, $z = \left(1, t, t^2, \ldots, t^{n-1}\right)$ be the n-vector of powers of t. Then

$$c^T z = p(t)$$

■

1.5 Cauchy-Schwartz Inequality

Theorem 1.5.1 — Cauchy-Schwartz Inequality. 设 $\langle \cdot, \cdot \rangle$ 是向量空间 V 上的内积, $\forall x, y \in V$, 则有

$$|\langle x, y \rangle|^2 \leq \langle x, x \rangle \langle y, y \rangle$$

当 $x = -\lambda y$ 时，有 $|\langle x, y \rangle|^2 = \langle x, x \rangle \langle y, y \rangle$。

Proof. 令 $\lambda \in \mathbb{R}$, 则有

$$0 \leq \langle x + \lambda y, x + \lambda y \rangle = \langle x, x \rangle + \lambda \langle y, x \rangle + \lambda \langle x, y \rangle + \lambda^2 \langle y, y \rangle = \langle x, x \rangle + 2\lambda \langle y, x \rangle + \lambda^2 \langle y, y \rangle$$

则

$$\lambda^2 \langle y, y \rangle + 2\lambda \langle y, x \rangle + \langle x, x \rangle \geq 0, \forall \lambda \in \mathbb{R}$$

所以

$$\nabla = (2\langle y, x \rangle)^2 - 4\langle y, y \rangle \langle x, x \rangle \leq 0$$

$$|\langle x, y \rangle|^2 \leq \langle x, x \rangle \langle y, y \rangle$$

当 $|\langle x, y \rangle|^2 = \langle x, x \rangle \langle y, y \rangle$ 时, 有

$$\langle x, x \rangle^2 + 2\lambda \langle y, x \rangle + \lambda^2 \langle y, y \rangle = 0$$

也即

$$\langle x + \lambda y, x + \lambda y \rangle = 0$$

因此 $x + \lambda y = 0$, 即 $x = -\lambda y$。 ■

Theorem 1.5.2 — Cauchy-Schwarz 不等式的矩阵元素形式.

$$\left(\sum_{i=1}^{n} u_i v_i\right)^2 \leq \left(\sum_{i=1}^{n} u_i^2\right)\left(\sum_{i=1}^{n} v_i^2\right)$$

Proof. The Cauchy-Schwarz inequality can be proved using only ideas from elementary algebra in this case. Consider the following quadratic polynomial in x

$$0 \leq (u_1 x + v_1)^2 + \cdots + (u_n x + v_n)^2 = \left(\sum_i u_i^2\right) x^2 + 2\left(\sum_i u_i v_i\right) x + \sum_i v_i^2$$

Since it is nonnegative, it has at most one real root for x, hence its discriminant is less than or equal to zero. That is,

$$\left(\sum_i u_i v_i\right)^2 - \left(\sum_i u_i^2\right)\left(\sum_i v_i^2\right) \leq 0$$

which yields the Cauchy-Schwarz inequality. ■

Corollary 1.5.3 — Cauchy-Schwarz 不等式变体. 由 Cauchy-Schwarz 不等式

$$|\langle x, y\rangle|^2 \leq \langle x, x\rangle\langle y, y\rangle$$

可以推得

$$|\langle a, b\rangle| \leq \|a\|_2 \|b\|_2$$
$$\langle a, b\rangle \geq -\|a\|_2 \|b\|_2$$

1.6 浮点运算

计算机以浮点格式存储 (实) 数值.

基本的算术运算 (加法, 乘法等) 被称为浮点运算 (flop).

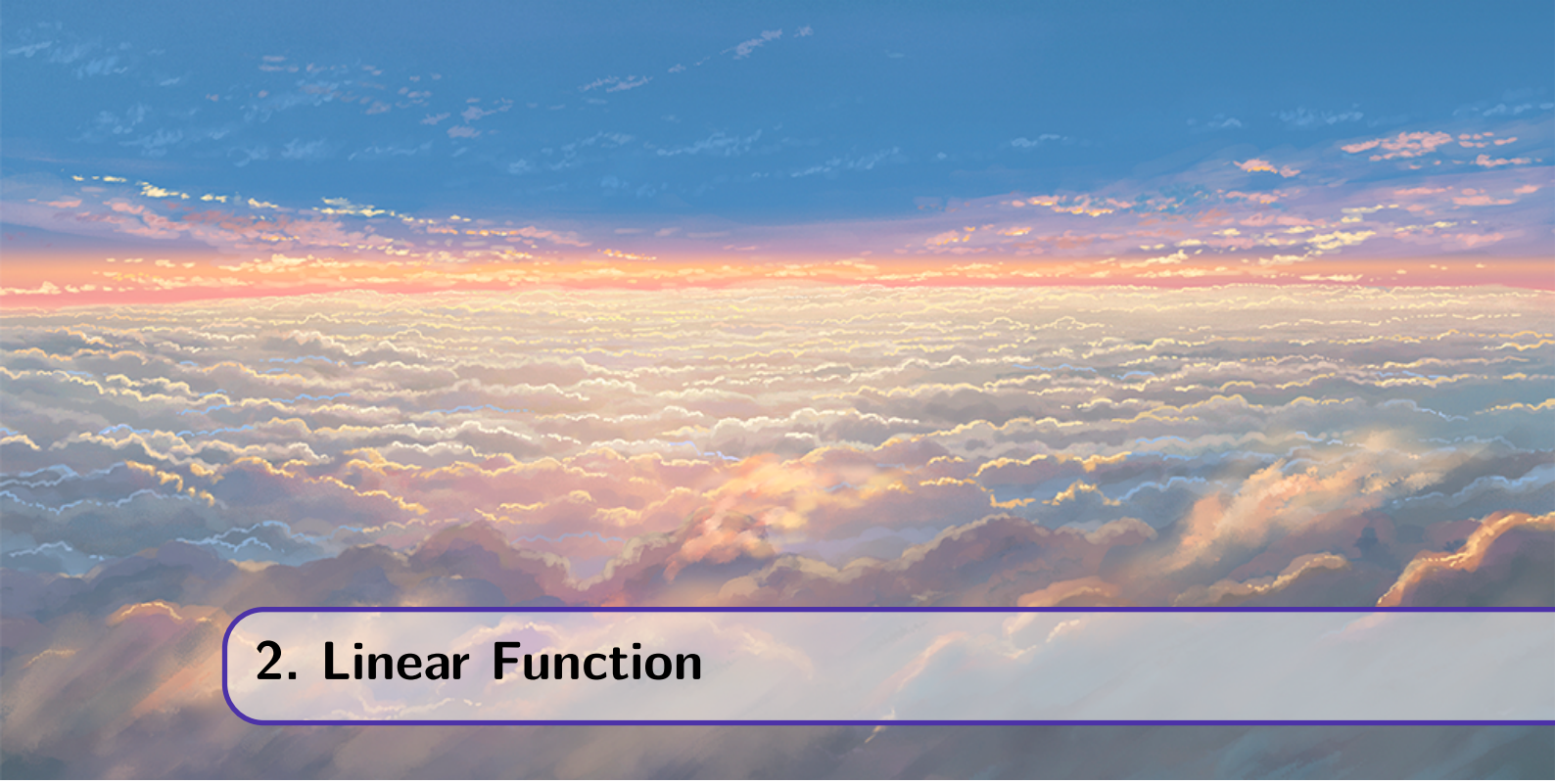
算法或操作的时间复杂度：作为输入维数的函数所需要的浮点运算总数.

算法复杂度通常以非常粗略地近似估算.

（程序）执行时间的粗略估计：计算机速度/flops，目前的计算机大约是 1Gflops/秒 (10^9flops/秒)。

Corollary 1.6.1 假设有 n 维向量 x 和 y:

- $x + y$ 需要 n 次加法, 所以时间复杂度为 (n)flops.
- $x^T y$ 需要 n 次乘法和 $n-1$ 次加法, 所以时间复杂度为 $(2n-1)$flops.
- 对于 $x^T y$, 通常将其时间复杂度简化为 $2n$, 甚至为 n.
- 当 x 或 y 是稀疏的时候, 算法的实际运算时间会比理论时间更少.



2. Linear Function

2.1 Linear Function

Definition 2.1.1 — Linear Function. f 是一个将 n 维向量映射成数的函数.

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

线性函数 f 满足以下两个性质 $(k \in \mathbb{R}, x, y \in \mathbb{R}^n)$:

- 齐次性 (homogeneity): $f(kx) = kf(x)$
- 叠加性 (Additivity): $f(x+y) = f(x) + f(y)$

▪ **Example 2.1** 求平均值: $f(x) = \frac{1}{n}\sum_{i=1}^{n} x_i$ 为线性函数. ▪

▪ **Example 2.2** 求最大值: $f(x) = \max\{x_1, x_2, \ldots, x_n\}$ 并不是线性函数. ▪

Proof. 令 $x = (1, -1), y = (-1, 1), \alpha = 0.5, \beta = 0.5$, 有

$$f(\alpha x + \beta y) = 0 \neq \alpha f(x) + \beta f(y) = 1$$

.

$$\begin{aligned} f(x+y) &= \max\{x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n\} \\ &\leq \max\{x_1, x_2, \ldots, x_n\} + \max\{y_1, y_2, \ldots, y_n\} \\ &\leq f(x) + f(y) \end{aligned}$$

■

Theorem 2.1.1 设 $\alpha_{1,\ldots,}\alpha_m \in \mathbb{R}, u_1, \ldots, u_m \in \mathbb{R}^n$, 则线性函数 f 满足

$$\begin{aligned} f\left(\alpha_1 u_1+\alpha_2 u_2+\ldots+\alpha_m u_m\right) &= f\left(\alpha_1 u_1\right)+f\left(\alpha_2 u_2+\ldots+\alpha_m u_m\right) \\ &= \alpha_1 f\left(u_1\right)+f\left(\alpha_2 u_2+\ldots+\alpha_m u_m\right) \\ &= \alpha_1 f\left(u_1\right)+\alpha_2 f\left(u_2\right)+\ldots+\alpha_m f\left(u_m\right) \end{aligned}$$

Definition 2.1.2 — 内积函数 (inner product function). 对于 n 维向量 a, 满足以下形式的函数被称为内积函数

$$f(x)=a^T x=a_1 x_1+a_2 x_2+\ldots+a_n x_n$$

上述 $f(x)$ 可以看作是每项 x_{i} 的加权之和.

Corollary 2.1.2 内积函数都是线性的.

Proof.

$$\begin{aligned} f(\alpha x+\beta y) &= a^T(\alpha x+\beta y) \\ &= a^T(\alpha x)+a^T(\beta y) \\ &= \alpha\left(a^T x\right)+\beta\left(a^T y\right) \\ &= \alpha f(x)+\beta f(y) \end{aligned}$$

■

Corollary 2.1.3 所有线性函数都是内积.

$$\begin{aligned} f(x) &= f\left(x_1 e_1+x_2 e_2+\ldots+x_n e_n\right) \\ &= x_1 f\left(e_1\right)+x_2 f\left(e_2\right)+\ldots+x_n f\left(e_n\right) \end{aligned}$$

Proof. 假设 $f:\mathbb{R}^n\to\mathbb{R}$ 是线性函数, 那么可用 $f(x)=a^T x$ 来表示, a 为常量.

$$\begin{aligned} f(x) &= f\left(x_1 e_1+x_2 e_2+\ldots+x_n e_n\right) \\ &= x_1 f\left(e_1\right)+x_2 f\left(e_2\right)+\ldots+x_n f\left(e_n\right) \end{aligned}$$

■

Definition 2.1.3 — 仿射函数 (affine function). 其一般形式为 $f(x)=a^T x+\mathrm{b}$, 其中 $a\in\mathbb{R}^n$, $b\in\mathbb{R}$ 为标量.

Theorem 2.1.4 函数 $f:\mathbb{R}^n\to\mathbb{R}$ 为仿射函数需要满足
$f(\alpha x+\beta y)=\alpha f(x)+\beta f(y), \alpha+\beta=1, \alpha, \beta\in\mathbb{R}, x, y\in\mathbb{R}^n$

2.2 泰勒展开

Definition 2.2.1 — 函数 f 第 i 个分量的一阶偏导数. 假设 $f:\mathbb{R}^n\to\mathbb{R}$, 函数 f 在 z 点可微

$$\begin{aligned}\frac{\partial f}{\partial z_i}(z) &= \lim_{t\to 0}\frac{f\left(z_1,\cdots,z_{i-1},z_i+t,z_{i+1},\cdots,z_n\right)-f(z)}{t}\\ &= \lim_{t\to 0}\frac{f\left(z+te_i\right)-f(z)}{t}\end{aligned}$$

Definition 2.2.2 — f 在点 z 的梯度.

$$\nabla f(z)=\begin{bmatrix}\frac{\partial f}{\partial z_1}(z)\\ \vdots\\ \frac{\partial f}{\partial z_n}(z)\end{bmatrix}$$

Definition 2.2.3 — Taylor's Approximation. 假设 $f:\mathbb{R}^n\to\mathbb{R}$, 函数 f 在 z 点充分光滑, 即处处可导.

$$\begin{aligned}f(x) &= f(z)+\frac{\partial f}{\partial x_1}(z)\left(x_1-z_1\right)+\frac{\partial f}{\partial x_2}(z)\left(x_2-z_2\right)+\cdots+\frac{\partial f}{\partial x_n}(z)\left(x_n-z_n\right)\\ &+\frac{1}{2!}\sum_{j=1}^{n}\sum_{i=1}^{n}\frac{\partial f^2}{\partial x_i\partial x_j}(z)\left(x_i-z_i\right)\left(x_{\mathrm{j}}-z_j\right)+\cdots\end{aligned}$$

■ **Example 2.3** 泰勒公式利用多项式在一点附近逼近函数

$$\sin x=x-\frac{x^3}{3!}+\frac{x^5}{5!}-\cdots+(-1)^{k-1}\frac{x^{2k-1}}{(2k-1)!}+\frac{\sin\left[\xi+(2k+1)\frac{\pi}{2}\right]}{(2k+1)!}x^{2k+1}$$

一次逼近: $\sin x\approx x$
三次逼近: $\sin x\approx x-\frac{x^3}{3!}$ ■

Proof.

$$f(x)=P_n(x)+R_n(x)$$

$P_n(x)=a_0+a_1\left(x-x_0\right)+a_2\left(x-x_0\right)^2+\cdots+a_n\left(x-x_0\right)^n$
$R_n(x)=o\left(x-x_0\right)^n$
$f(x)\approx P_n(x)$
$\therefore P_n\left(x_0\right)=f\left(x_0\right), P_n'\left(x_0\right)=f'\left(x_0\right), P_n''\left(x_0\right)=f''\left(x_0\right),\cdots,P_n^{(n)}\left(x_0\right)=f^{(n)}\left(x_0\right)$
要求 $P_n\left(x_0\right)=f\left(x_0\right)\Rightarrow a_0=f\left(x_0\right)$

$$P_n'(x)=a_1+2a_2\left(x-x_0\right)+\cdots+na_n\left(x-x_0\right)^{n-1}\Rightarrow a_1=f'\left(x_0\right)$$

依此类推. $a_n=\frac{f^{(n)}(x_0)}{n!}$ ■

Corollary 2.2.1 — n 阶泰勒多项式.

$$\begin{aligned}P_n(x) &= f\left(x_0\right)+f'\left(x_0\right)\left(x-x_0\right)+\frac{f''\left(x_0\right)}{2!}\left(x-x_0\right)^2\\ &+\cdots+\frac{f^{(n)}\left(x_0\right)}{n!}\left(x-x_0\right)^n\end{aligned}$$

where $a_n = \frac{f^{(n)}(x_0)}{n!}$

Corollary 2.2.2 — 对于高阶余项的公式. 带拉格朗日余项的泰勒公式

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n + \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)^{n+1}$$

$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)^{n+1}$ (ξ 在 x_0 与 x 之间)

Corollary 2.2.3 — 麦克劳林 (Maclaurin) 公式. 在零点展开麦克劳林 (Maclaurin) 公式

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \cdots + \frac{f^{(n)}(0)}{n!}x^n + \frac{f^{(n+1)}(\theta x)}{(n+1)!}x^{n+1} \qquad (0 < \theta < 1)$$

Definition 2.2.4 — 一阶泰勒公式. 假设 $f : \mathbb{R}^n \to \mathbb{R}$, 函数 f 在 z 点可导

$$\hat{f}(x) = f(z) + \frac{\partial f}{\partial x_1}(z)(x_1 - z_1) + \ldots + \frac{\partial f}{\partial x_1}(z)(x_n - z_n)$$

当 x 非常接近 z 时, $\hat{f}(x)$ 也非常接近 $f(z)$. $\hat{f}(x)$ 是关于 x 的一个仿射函数.

Corollary 2.2.4 — 一阶泰勒公式的内积形式.

$$\hat{f}(x) = f(z) + \nabla f(z)^T(x - z) \quad \left(\nabla f(z) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(z) \\ \vdots \\ \frac{\partial f}{\partial x_n}(z) \end{bmatrix}\right)$$

一维时, $\hat{f}(x) = f(z) + f'(z)(x - z)$.

■ **Example 2.4**

$$f(x) = x_1 - 3x_2 + e^{2x_1 + x_2 - 1}$$

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \frac{\partial f}{\partial x_2}(x) \end{bmatrix} = \begin{bmatrix} 1 + 2e^{2x_1 + x_2 - 1} \\ -3 + e^{2x_1 + x_2 - 1} \end{bmatrix}$$

函数 f 在 o 点的一阶泰勒公式为:

$$\hat{f}(x) = f(0) + \nabla f(0)^T(x - 0) = e^{-1} + \left(1 + 2e^{-1}\right)x_1 + \left(-3 + e^{-1}\right)x_2$$

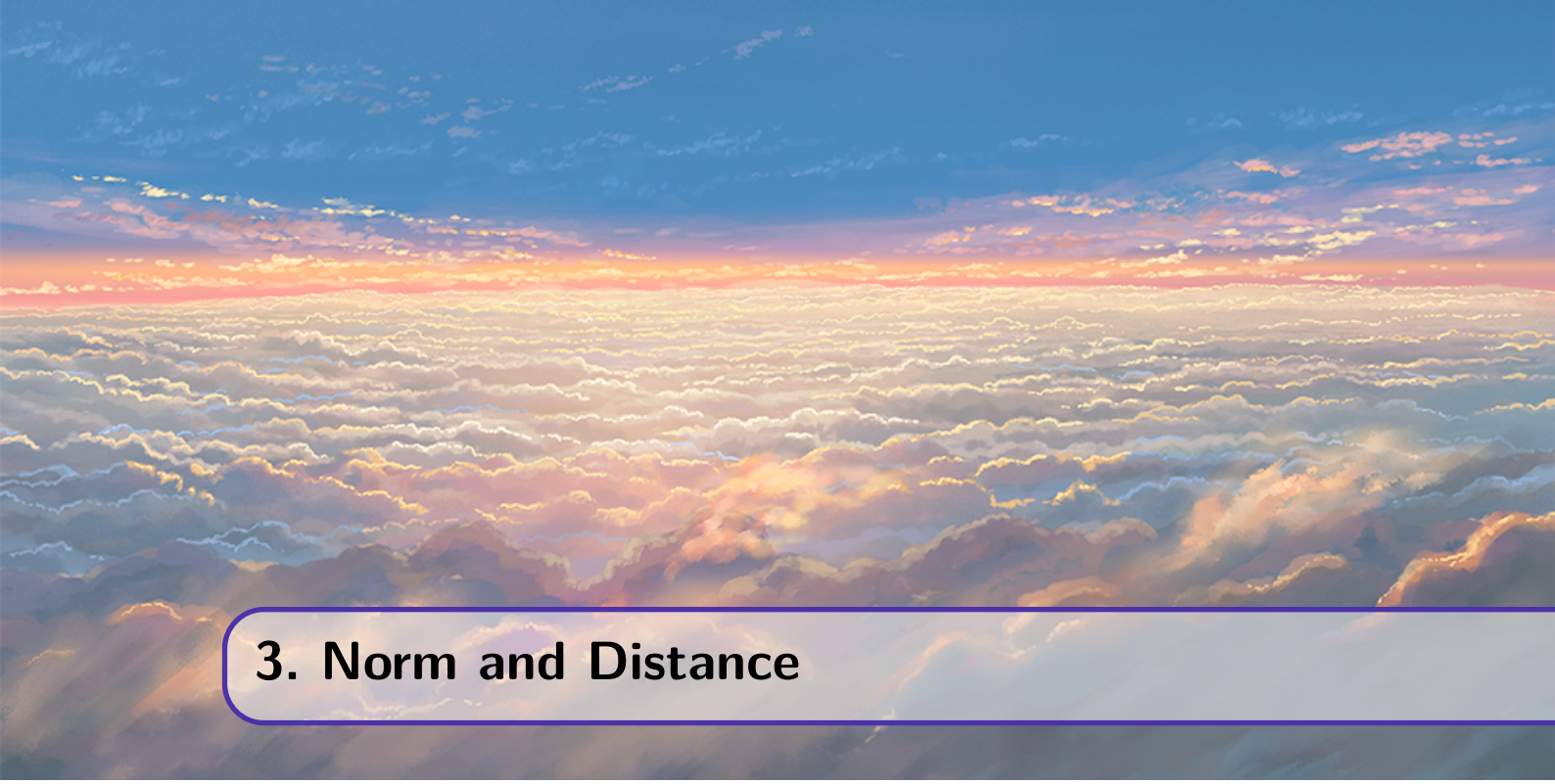
■

2.3 Regression Model

Definition 2.3.1 — Regression Model. 回归模型 (regression model) 为关于 x 的仿射函数

$$\hat{y} = x^T \beta + v$$

x 是特征向量 *(feature vector)*, 它的元素 x_i 称为回归元 *(regressors)*. n 维向量 β 是权重向量 *(weight vector)*. 标量 v 是偏移量 *(offset)*. 标量 $\hat{y}$ 是预测值 *(prediction)*. 表示某个实际结果或因变量, 用 y 表示.



3. Norm and Distance

3.1 Vector Norm

Definition 3.1.1 — Vector Norm. 在向量空间中存在一个函数 $\|\cdot\|:\mathbb{R}^n\to\mathbb{R}$, 且满足以下条件

- 齐次性: $\|\alpha x\|=|\alpha|\|x\|,\quad \alpha\in\mathbb{R}$ 且 $x\in\mathbb{R}^n$;
- 三角不等式: $\|x+y\|\le\|x\|+\|y\|,\quad x,y\in\mathbb{R}^n$;
- 非负性: $\|x\|\ge 0, \boldsymbol{x}\in\mathbb{R}^n$ 且 $\|\boldsymbol{x}\|=0\Leftrightarrow\boldsymbol{x}=0$;

则称 $\|\cdot\|$ 为向量范数.

▪ **Example 3.1 — ℓ_1-范数（曼哈顿范数, Manhattan norm）.**

$$\|x\|_1=|x_1|+|x_2|+\ldots+|x_n|\quad x,y\in\mathbb{R}^n,\alpha\in\mathbb{R}$$

▪

Proof.

$$\|\alpha x\|_1=|\alpha x_1|+|\alpha x_2|+\cdots+|\alpha x_n|=|\alpha|\|x\|_1\ge 0$$

$$\|x+y\|_1=|x_1+y_1|+\cdots+|x_n+y_n|\le|x_1|+|y_1|+\cdots+|x_n|+|y_n|=\|x\|_1+\|y\|_1$$

■

▪ **Example 3.2 — ℓ_2-范数（欧几里得范数, Euclidean norm）.**

$$\|x\|_2=\sqrt{\left(x_1^2+x_2^2+\cdots+x_n^2\right)}=\sqrt{x^Tx}=(\langle x,x\rangle)^{\frac{1}{2}}$$

▪

Proof.

$$\|\alpha x\|_2=(\langle\alpha x,\alpha x\rangle)^{\frac{1}{2}}=|\alpha|(\langle x,x\rangle)^{\frac{1}{2}}=|\alpha|\|x\|_2$$

$$\begin{aligned}\|x+y\|_2^2 &= \langle x+y, x+y\rangle = \langle x,x\rangle + \langle x,y\rangle + \langle y,x\rangle + \langle y,y\rangle \\ &= \|x\|_2^2 + 2\langle x,y\rangle + \|y\|_2^2 \le \|x\|_2^2 + 2\|x\|_2\|y\|_2 + \|y\|_2^2 \\ &= (\|x\|_2 + \|y\|_2)^2\end{aligned}$$

$$\|x+y\|_2 \le \|x\|_2 + \|y\|_2$$

■

Corollary 3.1.1 — 柯西—施瓦茨不等式.

$$|\langle x,y\rangle|^2 \le \langle x,x\rangle\langle y,y\rangle = \|x\|_2^2\|y\|_2^2$$

Definition 3.1.2 — ℓ_∞-范数.

$$\|x\|_\infty = \max_{1\le i\le n} |x_i|\,, x \in \mathbb{R}^n$$

Proof.

$$\begin{aligned}\max_{1\le i\le n} |x_i| &\le (|x_1|^p + \cdots + |x_i|^p + \cdots + |x_n|^p)^{1/p} \\ &\le \left(n \max_{1\le i\le n} |x_i|^p\right)^{1/p} \\ &= n^{1/p} \max_{1\le i\le n} |x_i| \\ &\to \max_{1\le i\le n} |x_i| \quad (p\to\infty)\end{aligned}$$

■

Definition 3.1.3 — ℓ_p-范数.

$$\|x\|_p = \left(x_1^p + x_2^p + \cdots + x_n^p\right)^{\frac{1}{p}}, \quad x \in \mathbb{R}^n, p \ge 1$$

ℓ_1 范数 $\|x\|_1$,ℓ_2-范数 $\|x\|_2$, ℓ_∞-范数是 ℓ_p-范数的特例.

证明可以使用以下两条不等式

Theorem 3.1.2 — Minkowshi Inequality.

$$\left(\sum_{i=1}^n |x_i+y_i|^p\right)^{\frac{1}{p}} \le \left(\sum_{i=1}^n |x_i|^p\right)^{\frac{1}{p}} + \left(\sum_{i=1}^n |y_i|^p\right)^{\frac{1}{p}}, p \ge 1, x, y \in \mathbb{R}^n$$

Theorem 3.1.3 — Hölder Inequality.

$$\sum_{i=1}^n |x_i y_i| \le \left(\sum_{i=1}^n |x_i|^p\right)^{1/p} \left(\sum_{i=1}^n |y_i|^q\right)^{1/q}, \frac{1}{p} + \frac{1}{q} = 1, 1 < p, q < \infty$$

3.2 Root Mean Square Value (RMS)

Definition 3.2.1 — 向量 x 的均方值 (mean-square value). 向量 $x \in \mathbb{R}^n$ 的均方值 (mean-square value)

$$\frac{x_1^2 + x_2^2 + \cdots + x_n^2}{n} = \frac{\|x\|_2^2}{n}$$

Definition 3.2.2 — n 维向量 x 的均方根 (root-mean-square value, RMS).

$$\mathrm{rms}(x) = \sqrt{\frac{x_1^2 + x_2^2 + \cdots + x_n^2}{n}} = \frac{\|x\|_2}{\sqrt{n}}$$

$\mathrm{rms}(x)$ 给出了 $|x_i|$ 的 “典型" (typical) 值. 例如, $\mathrm{rms}(\mathbf{1}) = 1$ (与 n 无关). 均方根 (RMS) 值对于比较不同长度的向量大小是比较有用的.

3.3 Chebyshev's Inequality

Theorem 3.3.1 — Chebyshev's Inequality.

$$P(|X - \mu| \geq \varepsilon) \leq \frac{\sigma^2}{\varepsilon^2}$$

$$P(|X - \mu| < \varepsilon) \geq 1 - \frac{\sigma^2}{\varepsilon^2}$$

Theorem 3.3.2 — Chebyshev's Inequality. 假设 k 为向量 x 分量满足条件 $|x_i| \geq a$ 的个数, 即 $x_i^2 \geq a^2$ 的个数.

因此: $\|x\|_2^2 = x_1^2 + x_2^2 + \cdots + x_n^2 \geq ka^2$

将 a^2 移项, 可得到 $k \leq \frac{\|x\|_2^2}{a^2}$

满足 $|x_i| \geq a$ 的 x_i 数量不会超过 $\frac{\|\|\|_2^2}{a^2}$

Corollary 3.3.3 — Chebyshev's Inequality Using RMS.

$$\mathrm{rms}(x) = \sqrt{\frac{x_1^2 + x_2^2 + \cdots + x_n^2}{n}} = \frac{\|x\|_2}{\sqrt{n}}$$

$|x_i| \geq a$ 的项数占整体的比例不会超过 $\left(\frac{\mathrm{rms}(x)}{a}\right)^2$, 即 $\frac{k}{n} \leq \left(\frac{\mathrm{rms}(x)}{a}\right)^2$

3.4 Distance

Definition 3.4.1 — Euclidean distance. n 维向量 a 和 b 之间的欧氏距离

$$\mathrm{dist}(a, b) = \|a - b\|_2$$

Definition 3.4.2 — RMS deviation. $\mathrm{rms}(a - b)$ 是 a 和 b 之间的均方根偏差.

Theorem 3.4.1 — Trianglar Inequality.

$$\|a-c\|_2 = \|(a-b)+(b-c)\|_2 \le \|a-b\|_2 + \|b-c\|_2$$

3.4.1 Feature Distance and Nearest Neighbor

Definition 3.4.3 — Feature Distance. 如果 x 和 y 分别为两个实体的特征向量, 那么它们的特征距离 (feature distance) 为 $\|x-y\|_2$

Definition 3.4.4 — Nearest Neighbor. 给定向量 x, 一个组向量 $Z_1, \dots, Z_m$, 当 $\hat{q}_j$ 满足:

$$\|x - z_j\|_2 \le \|x - z_i\|_2, \quad i = 1, \dots, m$$

则称 z_j 是 x 的最近邻 (nearest neighbor)

3.5 Standard Derivation

Definition 3.5.1 — 算术平均值. 对于 n 维向量 x

$$\text{avg}(x) = \frac{\mathbf{1}^T x}{n}$$

Definition 3.5.2 — De-meaned Vector.

$$\tilde{x} = x - \text{avg}(x)\mathbf{1}$$

因此 $\text{avg}\, \boldsymbol{g}(\tilde{x}) = 0$

Definition 3.5.3 — x 的标准差.

$$\text{std}(x) = \text{rms}(\tilde{x}) = \frac{\left\|x - \left(1^T x/n\right) 1\right\|_2}{\sqrt{n}}$$

$\text{std}(x)$ 表示数据元素的变化程度. 对于常数 α, 当且仅当 $x = \alpha\mathbf{1}$ 时, $\text{std}(x) = 0$.

Theorem 3.5.1

$$\text{rms}(x)^2 = \text{avg}(x)^2 + \text{std}(x)^2$$

3.6 Angle

Definition 3.6.1 — 两个非零向量 a 和 b 之间的角 (angle).

$$\angle(a,b) = \arccos\left(\frac{a^T b}{\|a\|_2 \|b\|_2}\right)$$

$\angle(a,b)$ 的取值范围为 $[0,\pi]$, 且满足

$$a^T b = \|a\|_2 \|b\|_2 \cos(\angle(a,b))$$

在二维和三维向量之中, 这里的角与普通角度 (ordinary angle) 是一致的.

- $\theta = \frac{\pi}{2} = 90$: a 和 b 为正交, 写作 $a \perp b (a^T b = 0)$.
- $\theta = 0$: a 和 b 为同向的 $(a^T b = \|a\|\|b\|)$.
- $\theta = \pi = 180$: a 和 b 为反向的 $(a^T b = -\|a\|\|b\|)$.

- $\theta < \frac{\pi}{2} = 90$：a 和 b 成锐角 ($a^T b > 0$).
- $\theta > \frac{\pi}{2} = 90$：a 和 b 成钝角 ($a^T b < 0$).

Definition 3.6.2 — 球面的距离.

$$\mathrm{R}\angle(a, b)$$

3.6.1 相关系数

给定向量 a 和 b, 其去均值向量为：

$$\tilde{a} = a - \mathrm{avg}(a)1, \tilde{b} = b - \mathrm{avg}(b)1$$

Definition 3.6.3 — a 和 b 的相关系数.

$$\rho = \frac{\tilde{a}^T \tilde{b}}{\|\tilde{a}\|_2 \|\tilde{b}\|_2} = \cos \angle(\tilde{a}, \tilde{b})$$

where $\tilde{a} \neq 0, \tilde{b} \neq 0$.

■ **Example 3.3** 高度相关的向量：
- 邻近地区的降雨时间序列.
- 类型密切相关文档的单词计数向量.
- 同行业中类似公司的日收益.

比较不相关的向量：
- 无关的向量.
- 音频信号 (比如, 在多轨录音中的不同轨).

负相关的向量：
- 深圳与墨尔本的每天气温变化

■

4. 优化问题初步

4.1 优化问题引入

Problem 4.1 假设 N 个样本向量 $x_1, \ldots, x_N \in \mathbb{R}^n$, 需要找到中心向量 z 满足

$$\min_{z \in \mathbf{R}^n} \sum_{i=1}^{N} \|x_i - z\|_2^2$$

■

Definition 4.1.1 — 高阶无穷小记号 o. 设 x, y 是同一变化过程中的无穷小, 即 $x \to 0, y \to 0$, 如果它们极限

$$\lim \frac{y}{x} = 0$$

则称 y 是 x 的高阶无穷小, 记作 $y = o(x)$.

Corollary 4.1.1

$$\lim \frac{y}{Cx} = \frac{1}{C} \lim \frac{y}{x} = 0$$

也即则称 y 是 Cx 的高阶无穷小, 记作 $y = o(Cx)$.

Proposition 4.1.2 — 优化求解的必要条件. 假设函数 f 在 $\hat{x}$ 可微, 则有

$$\hat{x} = \arg\min_{x \in \mathbb{R}^n} f(x) \Rightarrow \nabla f(\hat{x}) = 0$$

Proof. 假设函数 f 在 $\hat{x}$ 一阶泰勒展开, 有

$$f(x) = f(\hat{x}) + \langle \nabla f(\hat{x}), x - \hat{x} \rangle + o\left(\|x - \hat{x}\|_2\right)$$

假设 $\delta f(\hat{x}) \neq 0$, 则令 $\tilde{x} = \hat{x} - t \nabla f(\hat{x}), t > 0$, 可得

$$f(\tilde{x}) = f(\hat{x}) - t\|\nabla f(\hat{x})\|_2^2 + o\left(t\|\nabla f(\hat{x})\|_2\right)$$

当 $t \to 0$ 则 $t\|\nabla f(\hat{x})\|_2 \to 0$, 高阶无穷小 $o'\left(t\|\nabla f(\hat{x})\|_2\right) \to 0$
当 t 足够小时, 存在 $t\|\nabla f(\hat{x})\|_2 \geq o\left(t\|\nabla f(\hat{x})\|_2\right)$, 即

$$-t\|\nabla f(\hat{x})\|_2^2 + o\left(t\|\nabla f(\hat{x})\|_2\right) \leq 0$$

$$f(\tilde{x}) = f(\hat{x}) - t\|\nabla f(\hat{x})\|_2^2 + o\left(t\|\nabla f(\hat{x})\|_2\right) \leq f(\hat{x})$$

与 $\hat{x} = \arg\min_{\mathbf{R}^n} f(x)$ 矛盾.
$\nabla f(\widehat{x}) = 0$, 是最优问题解的必要条件. 通常 $\nabla f(\hat{x}) = 0 \Leftrightarrow \hat{x} = \arg\min_{\mathbf{R}^n} f(x)$. ■

■ **Example 4.1**

$$f(x) = -x^2, \quad x \in \mathbb{R}, \hat{x} = \operatorname{argmin}_{\mathbb{R}} f(x)$$

$\nabla f(\hat{x}) = 0$, 则有 $-2\hat{x} = 0$, 即 $\hat{x} = 0$

$$f(\hat{x}) = 0 \geq f(x), \quad x \in \mathbf{R}$$

(最大值!) ■

4.2 Convex Set

Definition 4.2.1 — 凸集. $\forall x, y \in \Omega, \alpha \in \mathbb{R}, 0 \leq \alpha \leq 1$ 有

$$\alpha x + (1-\alpha)y \in \Omega$$

则定义域 $\Omega \in \mathbb{R}^n$ 称为凸的 (Convex) 集合
(域内两点连线之间都属于这个域)

Definition 4.2.2 — 凸函数. 设函数 $f(x)$ 定义于称为凸的定义域 $\Omega \in \mathbb{R}^n$ 满足

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y), \forall x, y \in \Omega, \alpha \in \mathbf{R}, 0 \leq \alpha \leq 1$$

称其为凸函数.

■ **Example 4.2**

$$f(x) = x^2, x \in \mathbf{R}$$

$$\begin{aligned} f(\alpha x + (1-\alpha)y) &= (\alpha x + (1-\alpha)y)^2 \\ &= \alpha^2 x^2 + 2\alpha(1-\alpha)xy + (1-\alpha)^2 y^2 \\ &= \alpha x^2 + (1-\alpha)y^2 + \left(\alpha^2 - \alpha\right)x^2 + \left(\alpha^2 - \alpha\right)y^2 + 2\alpha(1-\alpha)xy \\ &= \alpha x^2 + (1-\alpha)y^2 - \alpha(1-\alpha)(x-y)^2 \\ &\leq \alpha x^2 + (1-\alpha)y^2 = \alpha f(x) + (1-\alpha)f(y) \end{aligned}$$

■

■ **Example 4.3** $f(x) = \|x\|$, 其中 $\|\ \|$ 表示 $\mathbb{R}^n$ 上的向量范数, $x \in \mathbb{R}^n$. ■

Proof.

$$\|\alpha x + (1-\alpha)y\| \leq \|\alpha x\| + \|(1-\alpha)y\| = |\alpha|\|x\| + |1-\alpha|\|y\|$$

■

■ **Example 4.4**

$$f(x) = \|x\|_2^2, x \in \mathbb{R}^n$$

■

Theorem 4.2.1 — 可微函数 f 是凸函数的充要条件.

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle, \quad \forall x, y$$

Proof. 首先, 证明一维情况 $f : \mathbb{R} \to \mathbb{R}, \alpha \in [0, 1]$.
⇒ 充分条件: $f(\alpha x + (1-\alpha)y) = f(x + (1-\alpha)(y-x)) \leq \alpha f(x) + (1-\alpha)f(y)$, 有

$$f(y) \geq f(x) + \frac{f(x + (1-\alpha)(y-x)) - f(x)}{(1-\alpha)(y-x)}(y-x)$$

令 $\alpha \to 1^-$, 则有 $f(y) \geq f(x) + f'(x)(y-x)$.
⇐ 必要条件: 令 $y \neq x, z = \alpha x + (1-\alpha)y$ 则有

$$f(x) \geq f(z) + f'(z)(x-z), f(y) \geq f(z) + f'(z)(y-z)$$

可得

$$\begin{aligned} \alpha f(x) + (1-\alpha)f(y) &\geq f(z) + \alpha f'(z)(x-z) + (1-\alpha)f'(z)(y-z) \\ &= f(z) + f'(z)(\alpha x + (1-\alpha)y - z) \\ &= f(z) \end{aligned}$$

证明 n 维情况 $f : \mathbb{R}^n \to \mathbb{R}$.
⇒ 充分条件: 令 $g(t) = f(tx + (1-t)y), t \in \mathbb{R}$, 则 $g'(t) = \langle \nabla f(tx + (1-t)y), x - y \rangle$
由于 f 是凸函数, 证明 $g(t)$ 也是凸函数; 并可得 $g(0) \geq g(1) + g'(1)(-1)$, 得证.
⇐ 必要条件: 与一维类似. (将 f' 改为 $\nabla f(z)^T$) ■

Theorem 4.2.2 如果可微函数 f 是凸函数, 则有

$$\hat{x} = \arg\min_{x \in \mathbb{R}^n} f(x) \Leftrightarrow \nabla f(\hat{x}) = 0$$

Proof. 已证 $\hat{x} = \arg\min_{x \in \mathbb{R}^n} f(x) \Rightarrow$ 可得 $\nabla f(\hat{x}) = 0$
只需证 $\nabla f(\hat{x}) = 0 \Rightarrow \hat{x} = \arg\min_{x \in \mathbb{R}^n} f(x)$.
由于函数 f 是可微凸的, 则有 $\forall x \in \mathbb{R}^n$,

$$\begin{aligned} f(x) &\geq f(\hat{x}) + \langle \nabla f(\hat{x}), x - \hat{x} \rangle \\ &\geq f(\hat{x}) + \langle 0, x - \hat{x} \rangle \geq f(\hat{x}) \end{aligned}$$

可得 $f(x) \geq f(\hat{x}), \hat{x} = \arg\min_{x \in \mathbb{R}^n} f(x)$. ■

4.3 向量偏导

Definition 4.3.1 — 向量对向量的导数.

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, z = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}$$

$$\nabla f(z) = \begin{bmatrix} \frac{\partial f(z)}{\partial z_1} \\ \vdots \\ \frac{\partial f(z)}{\partial z_n} \end{bmatrix}$$

■ **Example 4.5**

$$f(z) = x^T z + z^T z = \sum_{i=1}^{n} \left\{ x_i z_i + z_i^2 \right\}$$

$$\nabla f(z) = \begin{bmatrix} \frac{\partial f(z)}{\partial z_1} \\ \vdots \\ \frac{\partial f(z)}{\partial z_n} \end{bmatrix} = \begin{bmatrix} x_1 + 2z_1 \\ \vdots \\ x_n + 2z_n \end{bmatrix} = x + 2z$$

■

问题4.1中已知目标函数是凸函数.（见4.2, 4.3, 4.2）
则可以求解

Proof.

$$f(z) = \sum_{i=1}^{N} \|x_i - z\|_2^2 = \sum_{i=1}^{N} \langle x_i - z, x_i - z \rangle = \sum_{i=1}^{N} \left\{ x_i^T x_i - 2x_i^T z + z_i^T z \right\}$$

利用等价条件4.2.1

$$\nabla f(z) = \sum_{i=1}^{N} \{-2x_i + 2z\} = 0$$

(求导 4.3.1)

$$z = \frac{1}{N} \sum_{i=1}^{N} x_i$$

■

另解:

Proof. $J(x_0) = \sum_{i=1}^{n} \|x_0 - x_i\|^2$ ， 其中 $m = \frac{1}{n} \sum_{i=1}^{n} x_i$

$$\begin{aligned} J(x_0) &= \sum_{i=1}^{n} \|(x_0 - m) - (x_i - m)\|^2 \\ &= \sum_{i=1}^{n} \|x_0 - m\|^2 - 2 \sum_{i=1}^{n} (x_0 - m)^T (x_i - m) + \sum_{i=1}^{n} \|x_i - m\|^2 \\ &= \sum_{i=1}^{n} \|x_0 - m\|^2 - 2 (x_0 - m)^T \sum_{i=1}^{n} (x_i - m) + \sum_{i=1}^{n} \|x_i - m\|^2 \end{aligned}$$

因为

$$\sum_{i=1}^{n} (x_i - m) = \sum_{i=1}^{n} x_i - nm = \sum_{i=1}^{n} x_i - n \cdot \frac{1}{n} \sum_{i=1}^{n} x_i = 0$$

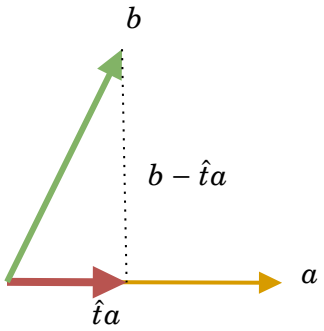
所以有

$$J(x_0) = \sum_{i=1}^{n} \|x_0 - m\|^2 + \sum_{i=1}^{n} \|x_i - m\|^2$$

即当 x_0 等于均值时，有最小均方. ■

4.4 投影问题: 标量优化问题

Figure 4.1: Projection onto a line



Problem 4.2 假设 $a, b \in \mathbb{R}^n, a \neq 0, t \in \mathbb{R}$, 当 t 多大时, ta 到 b 之间的距离最小

$$\hat{t} = \min_t \|ta - b\|_2^2$$

定义

$$f(t) = \|ta - b\|_2^2 = \langle ta - b, ta - b \rangle = t^2 a^T a - 2ta^T b + b^T b$$

可以验证 $f(t)$ 满足凸函数的定义。

$$\nabla f(t) = 2ta^T a - 2a^T b = 0$$

$$\hat{t} = \frac{a^T b}{a^T a} = \frac{a^T b}{\|a\|_2^2}$$

■

4.5 Clustering

将物理或抽象对象的集合分成由类似特征组成的多个类的过程称为聚类 *(clustering)*.

目标：分成 k 个集合, 尽量使得同一个集合中的向量彼此接近.

Notation 4.1. 给定 N 个 n 维向量 $x_1, \ldots, x_N \in \mathbb{R}^n$

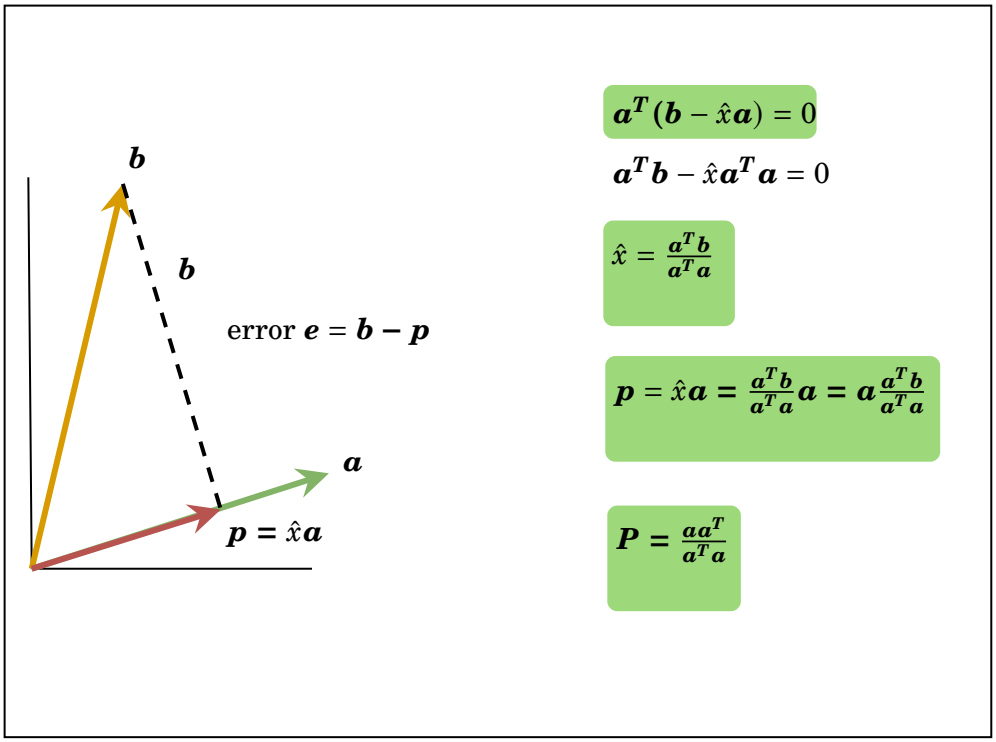
- 标签 $c_i \in \{1, 2, \cdots, k\}$ 表示向量 x_i 所属类别, 例如 $c_i = 2$ 表示 x_i 属于第 2 类.
- 对于 $j = 1, \ldots, k$, $G_j = \{i : c_i = j\}$ 表示属于第 j 类的向量 x_i 的下标集合.
- 向量 $z_j, j = 1, \ldots, k$, 表示同属于 j 类的向量 $x_i, i \in G_j$ 的聚类中心.

聚类目标是找到向量 x_i 的“标签 c_i ”和“聚类中心 z_j ”

Problem 4.3

$$\min_{z_j} \sum_{i \in G_j} \|x_i - z_j\|_2^2, j = 1, \cdots, k$$

Figure 4.2: Projection onto a line from the prospective of linear algebra



$$c_i = \underset{j=\{1,\cdots,k\}}{\operatorname{argmin}} \left\|x_i - z_j\right\|_2^2, i = 1, 2, \ldots, N$$

■

k-means 算法是将 N 向量 $x_i \in \mathbb{R}^n$ 划分成 k 类的迭代聚类算法.

Algorithm 1: k-means Algorithm

1. 在 N 个点中随机选取 k 个点, 分别作为聚类中心 z_j
2. 更新聚类标签 c_i : 计算每个点 x_i 到 k 个聚类中心 z_j 的距离, 并将其分配到最近的聚类中心 z_j 所在的聚类中 $c_i = j$
3. 更新聚类中心 z_j : 重新计算每个聚类现在的质心, 并以其作作为新的聚类中心, 根据更新标签 c_i, 更新属于第 j 类下标集合 $G_j = \{i : c_i = j\}$, 重新计算 c_i 类的聚类中心 z_j
4. 重复步骤 2、3, 直到所有聚类中心不再变化

Proof. 更新聚类标签 c_i:

$$\left\|x_i - z_j\right\|_2^2 = \operatorname{argmin}\left\{\|x_i - z_1\|_2^2, \|x_i - z_2\|_2^2, \cdots, \|x_i - z_k\|_2^2\right\}$$

更新聚类中心 z_j:

$$\nabla f_j\left(z_j\right) = \sum_{i \in G_j} 2\left(x_i - z_j\right) = 0$$

$$z_j = \frac{1}{\left|G_j\right|} \sum_{i \in G_j} x_i$$

$\left|G_j\right|$ 表示集合 G_j 中元素的数目. ■

在每一次迭代中目标函数 J 都会下降, 直到聚类中心 $z_1, \cdots, z_k$ 和划分聚类标签集合 $G_1, \cdots, G_k$ 不再变化.

但是 k-means 算法依赖于初始随机生成的聚类中心, 只可得到目标函数 J 的局部局部最优.

解决方案：使用不同的 (随机的) 初始聚类中心运行 k-means 算法若干次, 取目标函数 J 值最小的一次作为最终的聚类结果.

5. Linear Independence

5.1 线性相关、线性无关

Definition 5.1.1 — 线性相关 (linearly dependent). 定义：对于向量 $a_1, \dots a_m \in \mathbb{R}^n$, 如果存在不全为零的数 $\beta_1, \dots \beta_m \in \mathbb{R}$, 使得

$$\beta_1 a_1 + \cdots + \beta_m a_m = 0$$

则称向量 $a_1, \dots a_m$ 是线性相关 (linearly dependent).

线性相关等价于至少有一个向量 a_i 是其它向量的线性组合.

Corollary 5.1.1 向量集 $\{a_1\}$ 是线性相关的, 当且仅当 $a_1 = 0$.

Corollary 5.1.2 向量集 $\{a_1, a_2\}$ 是线性相关的, 当且仅当其中一个 $a_1 = \beta a_2, \beta \neq 0$.

Definition 5.1.2 — 线性独立 (linearly independent). 如果 n 维向量集 $\{a_1, \dots, a_m\}$ 不是线性相关的, 即线性独立 (linearly dependent), 也称线性无关, 即:

$$\beta_1 a_1 + \cdots + \beta_m a_m = 0$$

当且仅当 $\beta_1 = \cdots = \beta_m = 0$, 上述等式成立.

线性无关等价于不存在一个向量 a_i 是其它向量的线性组合.

Corollary 5.1.3 注：一个 n 维向量集最多有 n 个线性无关的向量, 也就是说如果 n 维向量集有 n + 1 个向量, 那它们必线性相关

■ **Example 5.1** n 维单位向量 $e_1, \dots, e_n$ 是线性独立的. ■

▪ **Example 5.2**

$$a_1 = \begin{bmatrix} 1 \\ -2 \\ 0 \end{bmatrix}, \quad a_2 = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}, \quad a_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$\beta_1 a_1 + \beta_2 a_2 + \beta_3 a_3 = \begin{bmatrix} \beta_1 - \beta_2 \\ -2\beta_1 + \beta_3 \\ \beta_2 + \beta_3 \end{bmatrix} = 0$$

$$\beta_1 = \beta_2 = \beta_3 = 0$$

▪

Theorem 5.1.4 假设 x 是线性无关向量 $a_1, \ldots, a_k$ 的线性组合:

$$x = \beta_1 a_1 + \cdots \beta_k a_k$$

则其系数 $\beta_1, \ldots \beta_k$ 是唯一的, 即如果有:

$$x = \gamma_1 a_1 + \cdots \gamma_k a_k$$

则对于 $i = 1, \ldots k$, 有 $\beta_i = \gamma_i$.

Proof. 系数是唯一的原因:

$$(\beta_1 - \gamma_1)\, a_1 + \cdots (\beta_k - \gamma_k)\, a_k = x - x = 0$$

由于向量 $a_1, \ldots, a_k$ 线性无关, 有 $\beta_1 - \gamma_1 = \beta_k - \gamma_k = 0$. ▪

5.2 Basis

Definition 5.2.1 — 基 (Basis). n 个线性独立的 n 维向量 $a_1, \ldots, a_n$ 的集合

Definition 5.2.2 — 向量 b 在基底 $a_1, \ldots, a_n$ 下的分解. 任何一个 n 维向量 b 都可以用它们的线性组合来表示

$$b = \beta_1 a_1 + \cdots + \beta_n a_n$$

Proof. 同一向量的系数是唯一的. ▪

▪ **Example 5.3** $e_1, \ldots, e_n$ 是一组基, 那么 b 在此基底下的分解为

$$b = b_1 e_1 + \cdots + b_n e_n, b = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \in \mathbb{R}^n$$

▪

5.3 标准正交向量

Definition 5.3.1 — Orthogonal Vectors. 在 n 维向量集 $a_1, \ldots, a_k$ 中, 如果对于 $i \neq j$, 都有 $a_i \perp a_j$, 则称它们相互正交 (orthogonal).

Definition 5.3.2 — Orthonormal Vectors. 如果 n 维向量集 $a_1, \ldots, a_k$ 相互正交, 且每个向量的模长都为单位长度 1 , 即对于 $i = 1, \ldots k$, 有 $\|a_i\|_2^2 = 1$, 则称它们是标准正交 (orthonormal) 的.

$$a_i^T a_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Corollary 5.3.1 标准正交的向量集是线性无关的.

Corollary 5.3.2 根据线性无关的性质, 必有向量集向量个数 $k \leq n$.

Proof. The proof is by induction on the dimension n.

First consider a linearly independent collection $a_1, \ldots, a_k$ of 1-vectors. We must have $a_1 \neq 0$. This means that every element a_i of the collection can be expressed as a multiple $a_i = (a_i/a_1)\, a_1$ of the first element a_1. This contradicts linear independence unless $k = 1$.

Next suppose $n \geq 2$ and the independence-dimension inequality holds for dimension $n - 1$.

Let $a_1, \ldots, a_k$ be a linearly independent list of n-vectors. We need to show that $k \leq n$. We partition the vectors as

$$a_i = \begin{bmatrix} b_i \\ \alpha_i \end{bmatrix}, \quad i = 1, \ldots, k$$

where b_i is an $(n-1)$-vector and α_i is a scalar.

First suppose that $\alpha_1 = \cdots = \alpha_k = 0$. Then the vectors $b_1, \ldots, b_k$ are linearly independent: $\sum_{i=1}^k \beta_i b_i = 0$ holds if and only if $\sum_{i=1}^k \beta_i a_i = 0$, which is only possible for $\beta_1 = \cdots = \beta_k = 0$ because the vectors a_i are linearly independent. The vectors $b_1, \ldots, b_k$ therefore form a linearly independent collection of $(n-1)$ -vectors. By the induction hypothesis we have $k \leq n - 1$, so certainly $k \leq n$.

Next suppose that the scalars α_i are not all zero. Assume $\alpha_j \neq 0$. We define a collection of $k - 1$ vectors c_i of length $n - 1$ as follows:

$$c_i = \begin{cases} b_i - \frac{\alpha_i}{\alpha_j} b_j, & i = 1, \ldots, j-1 \\ b_{i+1} - \frac{\alpha_{i+1}}{\alpha_j} b_j, & i = j, \ldots, k-1 \end{cases}$$

These $k - 1$ vectors are linearly independent: If $\sum_{i=1}^{k-1} \beta_i c_i = 0$ then

$$\sum_{i=1}^{j-1} \beta_i \begin{bmatrix} b_i \\ \alpha_i \end{bmatrix} + \gamma \begin{bmatrix} b_j \\ \alpha_j \end{bmatrix} + \sum_{i=j+1}^{k} \beta_{i-1} \begin{bmatrix} b_i \\ \alpha_i \end{bmatrix} = 0 \tag{5.1}$$

with

$$\gamma = -\frac{1}{\alpha_j} \left(\sum_{i=1}^{j-1} \beta_i \alpha_i + \sum_{i=j+1}^{k} \beta_{i-1} \alpha_i \right)$$

Since the vectors $a_i = (b_i, \alpha_i)$ are linearly independent, the eq. (5.1) only holds when all the coefficients β_i and γ are all zero. This in turns implies that the vectors $c_1, \ldots, c_{k-1}$ are linearly independent. By the induction hypothesis $k - 1 \leq n - 1$ so we have established that $k \leq n$ ∎

Definition 5.3.3 — **n 维向量的一个标准正交基.** 当 $k = n$ 时, $a_1, \ldots, a_n$ 是 n 维向量的一个标准正交基.

Definition 5.3.4 — **x 在标准正交基下的标准正交分解.** 如果 $a_1, \ldots, a_n$ 是一个标准正交基, 对于任意维向量 x;

$$x = \left(a_1^T x\right) a_1 + \cdots + \left(a_n^T x\right) a_n$$

则称其为 x 在标准正交基下的标准正交分解.

这个分解可以用于计算不同标准正交基下的系数.

Proof. 由于正交向量的性质

$$a_i^T a_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

所以

$$a_i^T x = \left(a_1^T x\right) a_i^T a_1 + \cdots + \left(a_i^T x\right) a_i^T a_i + \cdots + \left(a_n^T x\right) a_i^T a_n = a_i^T x$$

■

5.4 Gram-Schmidt Algorithm

Algorithm 2: Gram-Schmidt Algorithm

Input: n 维向量 $a_1, \ldots, a_k$
Output: 若这些向量线性无关，返回标准正交基 $q_1, \ldots, q_k$；若线性相关时判断 a_j 是 $a_1, \ldots, a_{j-1}$ 的线性组合

1 $q_1 = a_1 / \|a_1\|_2$
2 **while** $i = 2, \cdots, k$ **do**
3 正交化: $\widetilde{q}_i = a_i - \left(q_1^T a_i\right) q_1 - \cdots - \left(q_{i-1}^T a_i\right) q_{i-1}$
4 检验线性相关: 如果 $\widetilde{q}_i = 0$, 提前退出迭代
5 单位化: $q_i = \widetilde{q}_i / \|\widetilde{q}_i\|_2$
6 **end**

如果步骤 2 中未提前结束迭代, 那么 $a_1, \ldots, a_k$ 是线性独立的, 而且 $q_1, \ldots, q_k$ 是标准正交基.

如果在第 j 次迭代中提前结束, 说明 a_j 是 $a_1, \ldots, a_{j-1}$ 的线性组合, 因此 $a_1, \ldots, a_k$ 是线性相关的.

Theorem 5.4.1 $q_1, \ldots, q_{i-1}, q_i$ 是标准正交的.

Proof. 假设第 $i-1$ 次迭代成立, 即: $q_r \perp q_s, \forall r, s < i$.

正交化步骤保证有以下关系成立

$$\widetilde{q}_i = a_i - \left(q_1^T a_i\right) q_1 - \cdots - \left(q_{i-1}^T a_i\right) q_{i-1}$$

等式两边同时乘以 $q_j^T, j = 1, \ldots, i-1$

$$\begin{aligned} q_j^T \tilde{q}_i &= q_j^T a_i - \left(q_1^T a_i\right)\left(q_j^T q_1\right) - \cdots - \left(q_{i-1}^T a_i\right)\left(q_j^T q_{i-1}\right) \\ &= q_j^T a_i - q_j^T a_i \\ &= 0 \end{aligned}$$

$\because q_j^T q_r = 0, j \neq r, q_j^T q_j = 1$
$\therefore \widetilde{q}_i \perp q_1, \ldots, \widetilde{q}_i \perp q_{i-1}.$
单位化步骤保证了 $q_i = \widetilde{q}_i / \|\widetilde{q}_i\|_2$, 即 $q_1, \ldots, q_i$ 是标准正交. ■

Algorithm 3: Gram-Schmidt Algorithm for Three Vectors

Input: Three independent vectors $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$
Output: Three orthonormal vectors $\boldsymbol{q}_1 = \boldsymbol{A}/\|\boldsymbol{A}\|, \boldsymbol{q}_2 = \boldsymbol{B}/\|\boldsymbol{B}\|, \boldsymbol{q}_3 = \boldsymbol{C}/\|\boldsymbol{C}\|$.

1 Choose $\boldsymbol{A} = \boldsymbol{a}$

2
$$\boldsymbol{B} = \boldsymbol{b} - \frac{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}}{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}}\boldsymbol{A}$$

3
$$\boldsymbol{C} = \boldsymbol{c} - \frac{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{c}}{\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}}\boldsymbol{A} - \frac{\boldsymbol{B}^{\mathrm{T}}\boldsymbol{c}}{\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}}\boldsymbol{B}$$

4 单位化

5.4.1 The Analysis of Gram-Schmidt Algorithm

假设 Gram-Schmidt 正交法未在第 i 次迭代提前终止:

Corollary 5.4.2 a_i 是 $q_1, \ldots, q_i$ 的一个线性组合.

$$a_i = \|\tilde{q}_i\|_2 q_i + \left(q_1^T a_i\right) q_1 + \cdots + \left(q_{i-1}^T a_i\right) q_{i-1}$$

Proof.

$$\widetilde{q}_i = a_i - \left(q_1^T a_i\right) q_1 - \cdots - \left(q_{i-1}^T a_i\right) q_{i-1}$$

$$a_i = \widetilde{q}_i + \left(q_1^T a_i\right) q_1 + \cdots + \left(q_{i-1}^T a_i\right) q_{i-1}$$

注意有性质: $q_i = \widetilde{q}_i / \|\widetilde{q}_i\|_2$.

$$a_i = \|\tilde{q}_i\|_2 q_i + \left(q_1^T a_i\right) q_1 + \cdots + \left(q_{i-1}^T a_i\right) q_{i-1}$$

■

则有

Corollary 5.4.3

$$q_i = \frac{a_i - \left(q_1^T a_i\right) q_1 - \cdots - \left(q_{i-1}^T a_i\right) q_{i-1}}{\|\tilde{q}_i\|_2}$$

Corollary 5.4.4 q_i 是 $a_1, \ldots, a_i$ 的一个线性组合.

Proof. 归纳假设, 每个 q_{i-1} 都是 $a_1, \ldots, a_{i-1}$ 的线性组合:

$$\begin{aligned} q_2 &= \frac{a_2 - \left(q_1^T a_2\right) q_1}{\|\tilde{q}_2\|_2} \\ &= \frac{a_2 - \left(q_1^T a_2\right) \frac{a_1}{\|a_1\|_2}}{\|\tilde{q}_2\|_2} \end{aligned}$$

$$q_3 = \frac{a_3 - \left(q_1^T a_3\right) q_1 - \left(q_2^T a_3\right) q_2}{\|\tilde{q}_3\|_2}$$

通过对 i 的归纳证明, 可得 q_i 是 $a_1, \ldots, a_i$ 的线性组合. ■

假设 Schmidt 正交法在第 j 次迭代提前终止:

Corollary 5.4.5 a_j 是 $q_1, \ldots, q_{j-1}$ 的一个线性组合.

$$a_j = \left(q_1^T a_j\right) q_1 + \cdots + \left(q_{j-1}^T a_j\right) q_{j-1}$$

Proof.

$$\widetilde{q_i} = a_i - \left(q_1^T a_i\right) q_1 - \cdots - \left(q_{i-1}^T a_i\right) q_{i-1}$$

$$0 = a_i - \left(q_1^T a_i\right) q_1 - \cdots - \left(q_{i-1}^T a_i\right) q_{i-1}$$

$$a_i = \|\tilde{q}_i\|_2 \, q_i + \left(q_1^T a_i\right) q_1 + \cdots + \left(q_{i-1}^T a_i\right) q_{i-1}$$

■

Corollary 5.4.6 a_j 是 $a_1, \ldots, a_{j-1}$ 的线性组合.

Proof. 每一个 $q_1, \ldots, q_{j-1}$ 都是 $a_1, \ldots, a_{j-1}$ 的线性组合.
因此 a_j 是 $a_1, \ldots, a_{j-1}$ 的线性组合. ■

Matrices

6. Matrices

6.1 Matrices

Definition 6.1.1 — 矩阵. 矩阵是一个由数字构成的矩阵数组.

$$\begin{bmatrix} 0 & 1 & -2.3 & 0.1 \\ 1.3 & 4 & -0.1 & 0 \\ 4.1 & -1 & 0 & 1.7 \end{bmatrix} \quad \begin{pmatrix} 0 & 1 & -2.3 & 0.1 \\ 1.3 & 4 & -0.1 & 0 \\ 4.1 & -1 & 0 & 1.7 \end{pmatrix}$$

上述矩阵大小 *(size)* 为 3×4, 矩阵的每一个元素 *(element)* 又称为系数 *(coefficient)*;

Notation 6.1. 设 B_{ij} 表示矩阵 B 中第 i 行第 j 的元素
实数域中大小为 $m \times n$ 的矩阵集合写为 $\mathbb{R}^{m \times n}$
复数域中大小为 $m \times n$ 的矩阵集合写为 $\mathbb{C}^{m \times n}$

Definition 6.1.2 — 标量. 不区分一个 1×1 矩阵和一个标量.

Definition 6.1.3 — 向量. 不区分一个 $n \times 1$ 矩阵和一个向量.

Definition 6.1.4 — 行向量, 列向量. 一个 $1 \times n$ 矩阵被称为一个行向量.
一个 $n \times 1$ 矩阵杯称为一个列向量.

Definition 6.1.5 — 高形, 宽形和方形矩阵. 一个大小为 $m \times n$ 的矩阵为:

- 高的, 如果 $m > n$
- 宽的, 如果 $m < n$
- 方的, 如果 $m = n$

Definition 6.1.6 — 分块矩阵. 分块矩阵的每一个元都是一个矩阵.

$$A = \begin{bmatrix} B & C \\ D & E \end{bmatrix}$$

其中 B, C, D, E 都是矩阵 (被称为矩阵 A 的子矩阵).

分块矩阵位于同一行的子矩阵行维度必须相等, 位于同一列的子矩阵列维度必须相等.

Definition 6.1.7 — 矩阵的列向量表示. 矩阵 $A \in \mathbb{R}^{m\times n}$, 可通过其列向量 (m-vector) 进行表示, 假设其列向量为 $a_1, \ldots, a_n \in \mathbb{R}^m$, 则有

$$A = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}$$

Definition 6.1.8 — 矩阵的行向量表示. 矩阵 $A \in \mathbb{R}^{m\times n}$ 通过其行向量 $b_1, \ldots, b_m$ 进行表示

$$A = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}, b_i^T \in \mathbb{R}^n, i = 1, \cdots, m$$

6.2 矩阵运算

Definition 6.2.1 — 矩阵数乘. 设矩阵 $A \in \mathbb{R}^{m\times n}$

$$\beta A = \begin{bmatrix} \beta A_{11} & \beta A_{12} & \cdots & \beta A_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ \beta A_{m1} & \beta A_{m2} & \cdots & \beta A_{mn} \end{bmatrix}, \beta \in \mathbb{R}$$

Definition 6.2.2 — 矩阵加法. 矩阵 $A, B \in \mathbb{R}^{m\times n}$ 的和为

$$A + B = \begin{bmatrix} A_{11} + B_{11} & A_{12} + B_{12} & \cdots & A_{1n} + B_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} + B_{m1} & A_{m2} + B_{m2} & \cdots & A_{mn} + B_{mn} \end{bmatrix}$$

Definition 6.2.3 — Transpose. 矩阵 A 的转置表示为 A^T
若 $A \in \mathbb{R}^{m\times n}$, 则 $A^T \in \mathbb{R}^{n\times m}$, 其被定义为: $\left(A^T\right)_{ij} = A_{ji}, i = 1, \ldots, n; j = 1, \ldots, m$

转置将原矩阵的行向量转化为列向量.

Corollary 6.2.1 — 转置的性质. 有如下性质:

- $\left(A^T\right)^T = A$
- 对称矩阵满足 $A^T = A$
- $(\beta A)^T = \beta A^T, (A + B)^T = A^T + B^T$

Definition 6.2.4 — 共轭转置. 矩阵 A 的共轭转置表示为 A^{H}
若 $\mathrm{A} \in \mathbb{C}^{m\times n}$, 则 $A^H \in \mathbb{C}^{n\times m}$, 其被定义为: $\left(A^H\right)_{ij} = \bar{A}_{ji}, i = 1, \ldots, n; j = 1, \ldots, m;$
设矩阵 $A \in \mathbb{C}^{m\times n}$, 则其共轭转置为一个 $n \times m$ 矩阵

$$A^H = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{21} & \cdots & \bar{A}_{m1} \\ \bar{A}_{21} & \bar{A}_{22} & \cdots & \bar{A}_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{A}_{1n} & \bar{A}_{2n} & \cdots & \bar{A}_{mn} \end{bmatrix}$$

Corollary 6.2.2 — 共轭转置的性质. 有如下性质:

- $\left(A^H\right)^H = A$
- Hermitian 矩阵满足 $A = A^H$
- $(\beta A)^H = \beta A^H, (A+B)^H = A^H + B^H$

Definition 6.2.5 — 矩阵乘法. 设矩阵 $A \in \mathbb{R}^{m\times p}, B \in \mathbb{R}^{p\times n}$, 那么矩阵 A 与 B 的乘积, 记作 C = AB , 矩阵 $C \in \mathbb{R}^{m\times n}$ 的第 i 行第 j 列元素 C_{ij}

$$\mathrm{C}_{ij} = \sum_{k=1}^{p} A_{ik}B_{kj}$$

注意: 矩阵 A 的列大小必须等于 B 的行大小.

Corollary 6.2.3 — 矩阵乘法性质. 有如下性质:

- 结合律: $(AB)C = A(BC)$
- 分配律: $A(B+C) = AB + AC$
- $$(AB)^T = B^T A^T, \quad (AB)^{\mathrm{H}} = B^H A^H$$
- 一般情况下 $AB \neq BA$
- 对于方阵 A 有, $IA = AI = A$

Definition 6.2.6 — 分块矩阵乘法.

$$\begin{bmatrix} A & B \\ \mathrm{C} & D \end{bmatrix}\begin{bmatrix} W & Y \\ X & Z \end{bmatrix} = \begin{bmatrix} AW+BX & AY+BZ \\ CW+DX & CY+DZ \end{bmatrix}$$

Definition 6.2.7 — 矩阵-向量乘积 Ax. 矩阵 $A \in \mathbb{R}^{m\times n}$ 和一个向量 $x \in \mathbb{R}^n$ 的积为

$$Ax = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n \\ A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n \\ \vdots \\ A_{m1}x_1 + A_{m2}x_2 + \cdots + A_{mn}x_n \end{bmatrix}$$

Corollary 6.2.4 Ax 是矩阵 A 列向量的线性组合.

$$Ax = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1a_1 + \cdots + x_na_n$$

可以引出 A 的 Row Picture 和 Column Picture 的概念。

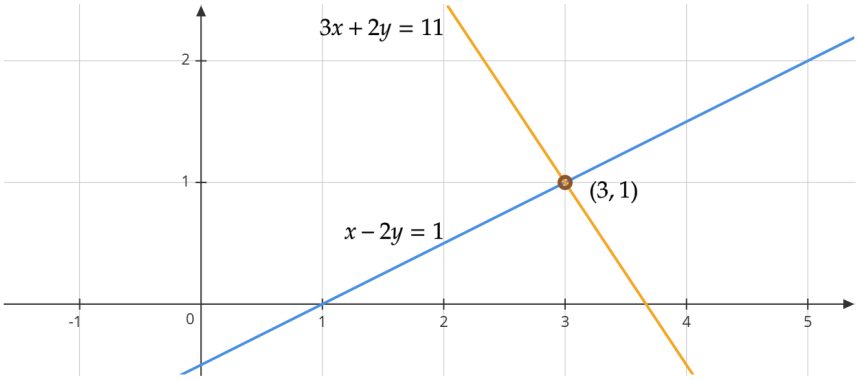
■ **Example 6.1**

$$\begin{cases} x - 2y = 1 \\ 3x + 2y = 11 \end{cases}$$

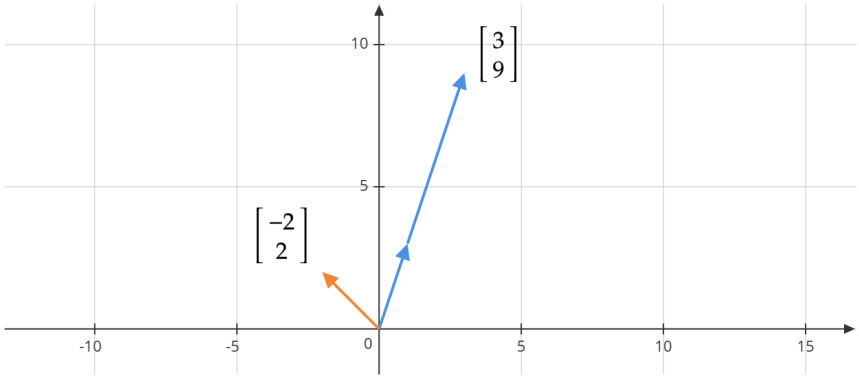
■

Figure 6.1: Row Picture and Column Picture for 6.1

(a) Row Picture



(b) Column Picture



Definition 6.2.8 — 矩阵-向量乘积函数 $f(x) = Ax$. 给定矩阵 $A \in \mathbb{R}^{m\times n}$，定义函数 $f : R^n \to R^m, f(x) = Ax$, 其中 $A = [f(e_1) \dots f(e_n)]$.

Proof. 该函数为一个线性函数：

$$A(\alpha x + \beta y) = \alpha(Ax) + \beta(Ay)$$

任意一个线性函数都可以写成矩阵-向量乘积函数的形式

$$\begin{aligned} f(x) &= f(x_1e_1 + x_2e_2 + \cdots + x_ne_n) \\ &= x_1f(e_1) + x_2f(e_2) + \cdots + x_nf(e_n) \\ &= \begin{bmatrix} f(e_1) & \cdots & f(e_n) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \end{aligned}$$

因此 $f(x) = Ax$, 其中 $A = [f(e_1) \dots f(e_n)]$ ∎

6.2.1 Matrix Power

Definition 6.2.9 — Matrix Power. It makes sense to multiply a square matrix A by itself to form AA. We refer to this matrix as A^2. Similarly, if k is a positive integer, then k copies of A multiplied together is denoted A^k. If k and l are positive integers, and A is square, then $A^kA^l = A^{k+l}$ and $\left(A^k\right)^l = A^{kl}$.

$$\left(A^{\ell+1}\right)_{ij} = \sum_{k=1}^{n} A_{ik}\left(A^{\ell}\right)_{kj}$$

By convention we take $A^0 = I$, which makes the formulas above hold for all nonnegative integer values of k and l.

▪ **Example 6.2 — Paths in a directed graph.**

$$A_{ij} = \begin{cases} 1 & \text{there is a edge from vertex } j \text{ to vertex } i \\ 0 & \text{otherwise} \end{cases}$$

▪

▪ **Example 6.3 — Linear dynamical system.**

$$x_{t+\ell} = A^{\ell}x_t$$

▪

6.2.2 矩阵乘法的算法复杂度

一般矩阵的乘法

$C = AB$ with A of size $m \times p$ and B of size $p \times n$

The product matrix C has size $m \times n$, so there are mn elements to compute.

The i, j element of C is the inner product of row i of A with column j of B. This is an inner product of vectors of length p and requires $2p - 1$ flops. Therefore the total is $mn(2p - 1)$ flops, which we approximate as $2mnp$ flops. $O(mnp)$

稀疏矩阵的乘法

Suppose that A is $m \times p$ and sparse, and B is $p \times n$, but not necessarily sparse.

The inner product of the i th row a_i^T of A with the j th column of B requires no more than $2\,\mathrm{nnz}\left(a_i^T\right)$ flops.

Summing over $i = 1, \ldots, m$ and $j = 1, \ldots, n$ we get $2\mathbf{nnz}(A)n$ flops.

If B is sparse, the total number of flops is no more that $2\mathbf{nnz}(B)m$ flops.

Note that these formulas agree with the one given above, $2mnp$, when the sparse matrices have all entries nonzero.

三重矩阵相乘

$$D = ABC$$

with A of size $m \times n$, B of size $n \times p$, and C of size $p \times q$.

The matrix D can be computed in two ways, as $(AB)C$ and as $A(BC)$.

In the first method we start with AB($2mnp$ flops) and then form $D = (AB)C$($2mpq$ flops), for a total of $2mp(n+q)$ flops.

In the second method we compute the product BC (2npq flops) and then form $D = A(BC)$($2mnq$ flops), for a total of $2nq(m+p)$ flops.

You might guess that the total number of flops required is the same with the two methods, but it turns out it is not. The first method is less expensive when $2mp(n+q) < 2nq(m+p)$, i.e., when

$$\frac{1}{n} + \frac{1}{q} < \frac{1}{m} + \frac{1}{p}$$

■ **Example 6.4** As a more specific example, consider the product

$$ab^T c$$

where a, b, c are n vectors.

If we first evaluate the outer product ab^T, the cost is n^2 flops, and we need to store n^2 values. We then multiply the vector c by this $n \times n$ matrix, which costs $2n^2$ flops. The total cost is $3n^2$ flops.

If we first evaluate the inner product $b^T c$, the cost is $2n$ flops, and we only need to store one number (the result). Multiplying the vector a by this number costs n flops, so the total cost is $3n$ flops. For n large, there is a dramatic difference between $3n$ and $3n^2$ flops.

(The storage requirements are also dramatically different for the two methods of evaluating $ab^T c$: 1 number versus n^2 numbers.) ■

6.2.3 矩阵向量乘积复杂度

矩阵 $A \in \mathbb{R}^{m \times n}$ 和向量 $x \in \mathbb{R}^n$ 的乘积 y = Ax, 需要 (2n − 1)m flops;

乘积 $y \in \mathbb{R}^m$, 每个元素需要做向量内积, 需要 $2n - 1$ flops;

当 n 足够大时, 复杂度近似于 2mn;

特殊情况:

- A 为对角矩阵：n flops
- A 为下三角矩阵：n^2 flops
- A 为稀疏矩阵时：flops << $2mn$

6.3 Special Matrices and Matrices in Different Applications

Definition 6.3.1 — Zero Matrix. 所有元素都为 0 的矩阵.
记作

$$0, 0_{m\times n}$$

Definition 6.3.2 — 单位矩阵. 为方形矩阵, 其中对角线元素为 1, 其它元素为 0.
记作 I 或者 I_n.

Corollary 6.3.1 I_n 的每一列是一个单位向量, 例如

$$\mathrm{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}$$

Definition 6.3.3 — Symmetric Matrices.

$$A_{ij} = A_{ji}, A^T = A$$

Definition 6.3.4 — Hermitian Matrices. $A_{ij} = \bar{A}_{ji}, A^H = A$ (共轭复数).

Definition 6.3.5 — Diagonal Matrices. 对角线上元素不全为 0, 其余元素全为 0.

对角矩阵用于膨胀 (dilation).

Definition 6.3.6 — 下三角矩阵. 方形矩阵且当 $i < j$ 时 $A_{ij} = 0$.

Definition 6.3.7 — 上三角矩阵. 方形矩阵且当 $i > j$ 时 $A_{ij} = 0$

通过 Gram-Schmidt 正交化算法可以化为上三角或者下三角矩阵。

6.3.1 $f(x) = Ax$ 中的 A

引入上节 $f(x) = Ax$(矩阵-向量乘积函数) 的概念,

▪ **Example 6.5 — Permutation Matrices.** f 颠倒向量 x 中的元素的顺序, 一个线性函数 $f(x) = Ax$

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

是一个置换矩阵. ▪

Proof.

$$Ax = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_3 \\ x_2 \\ x_1 \end{bmatrix}$$

■

▪ **Example 6.6** f 对向量 x 中的元素进行升序排序, 非线性; ▪

▪ **Example 6.7** f 将向量 x 中的元素替换成相应的绝对值, 非线性; ▪

▪ **Example 6.8 — 反转矩阵.**

$$A = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

▪

Proof.

$$Ax = \begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_2 \\ x_1 \end{bmatrix}$$

■

▪ **Example 6.9 — 循环移位矩阵.**

$$A = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

▪

Proof.

$$Ax = \begin{bmatrix} x_n \\ x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

■

▪ **Example 6.10 — 旋转矩阵.**

$$A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Ax 将向量 x 进行旋转, 角度为 θ. ▪

▪ **Example 6.11 — Reflection Matrices.** Suppose that y is the vector obtained by reflecting x through the line that passes through the origin, inclined θ radians with respect to horizontal. Then

$$y = \begin{bmatrix} \cos(2\theta) & \sin(2\theta) \\ \sin(2\theta) & -\cos(2\theta) \end{bmatrix} x$$

▪

6.3.2 Selectors

Definition 6.3.8 — Selector matrices.

Definition 6.3.9 — Downsampling.

6.3.3 图论：节点弧关联矩阵

Definition 6.3.10 — 关联矩阵. 假设有向图 G 有 m 个顶点, n 条弧, 则关联矩阵 A 大小为 $m \times n$, 其中

$$A_{ij} = \begin{cases} 1 & \text{如果点}i\text{是弧}j\text{的终点} \\ -1 & \text{如果点}i\text{是弧}j\text{的起点} \\ 0 & \text{其它} \end{cases}$$

6.3.4 Convolution

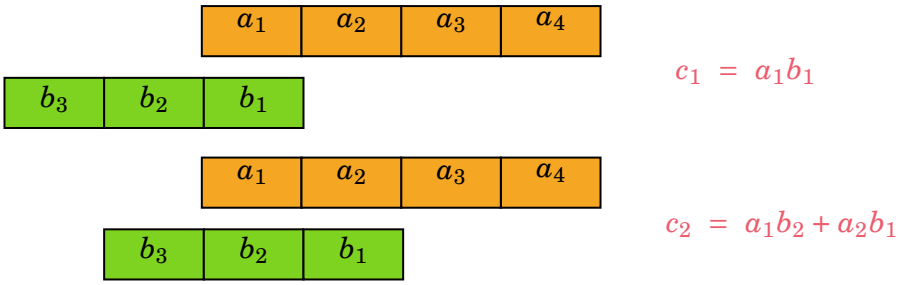
Definition 6.3.11 — 一维卷积. 向量 $a \in \mathbb{R}^n$ 和向量 $b \in \mathbb{R}^m$ 的卷积是一个 $(\mathrm{n}+\mathrm{m}-1)$ 维向量 $c \in \mathbb{R}^{m+\mathrm{n}-1}$

$$c_k = \sum_{i+j=k+1} a_i b_j, \quad k = 1, \ldots n+m-1$$

记为 $c = a * b$

Example 6.12 设 $n = 4, m = 3$

Figure 6.2: An example of convolution



$$\begin{aligned} c_1 &= a_1 b_1 \\ c_2 &= a_1 b_2 + a_2 b_1 \\ c_3 &= a_1 b_3 + a_2 b_2 + a_3 b_1 \\ c_4 &= a_2 b_3 + a_3 b_2 + a_4 b_1 \\ c_5 &= a_3 b_3 + a_4 b_2 \\ c_6 &= a_4 b_3 \end{aligned}$$

■

Corollary 6.3.2 假设向量 a 和 b 分别是以下多项式的系数

$$p(x) = a_1 + a_2 x + \cdots + a_n x^{n-1}, q(x) = b_1 + b_2 x + \cdots + b_m x^{m-1}$$

则 c = a* b 是多项式 $p(x)q(x)$ 的系数.

$$p(x)q(x) = c_1 + c_2 x + \cdots + c_{m+n-1} x^{m+n-2}$$

Corollary 6.3.3 — 卷积性质. 有如下性质:

- 对称性: $a * b = b * a$
- 结合律: $(a * b) * c = a * (b * c)$
- 如果 $a * b = 0$, 则 $a = 0$, 或者 $b = 0$

Corollary 6.3.4 如果固定 a 或 b, 则 $c = a * b$ 是一个线性函数

▪ **Example 6.13 — Toeplitz Matrix.** 4 维向量 a 和 3 维向量 b , 则 $c = a * b$

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{bmatrix} = \begin{bmatrix} a_1 & 0 & 0 \\ a_2 & a_1 & 0 \\ a_3 & a_2 & a_1 \\ a_4 & a_3 & a_2 \\ 0 & a_4 & a_3 \\ 0 & 0 & a_4 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} b_1 & 0 & 0 & 0 \\ b_2 & b_1 & 0 & 0 \\ b_3 & b_2 & b_1 & 0 \\ 0 & b_3 & b_2 & b_1 \\ 0 & 0 & b_3 & b_2 \\ 0 & 0 & 0 & b_3 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}$$

▪

6.3.5 多项式

Definition 6.3.12 — 多项式. 多项式 $p(t)$, 度为 $n-1$, 系数为 $x_1, x_2, \ldots, x_n$

$$p(t) = x_1 + x_2 t + x_3 t^2 + \cdots + x_n t^{n-1}$$

Definition 6.3.13 — Vandermonde Matrices. p(t) 在 m 个点中 $t_1, t_2, \ldots, t_m$ 的值为

$$\begin{bmatrix} p(t_1) \\ p(t_2) \\ \vdots \\ p(t_m) \end{bmatrix} = \begin{bmatrix} 1 & t_1 & \cdots & {t_1}^{n-1} \\ 1 & t_2 & \cdots & {t_2}^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_m & \cdots & {t_m}^{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = Ax$$

矩阵 A 被称为 *Vandermonde* 矩阵.

6.3.6 Fourier Transform

Definition 6.3.14 — Discrete Fourier Transform (DFT). DFT 将 n 维复向量 x 映射为 n 维复向量 y ($\mathbb{C}^n \to \mathbb{C}^n$)

$$y_k = \sum_{\ell=1}^{n} x_\ell e^{-i\frac{2\pi}{n}(k-1)(\ell-1)}, k = 1, \cdots, n$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ 1 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(n-1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \ldots & \omega^{-(n-1)(n-1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

其中 $\omega = e^{2\pi i/n}$.

DFT 矩阵 W 的第 k 行第 l 列的元素为 $W_{kl} = \omega^{-(k-1)(l-1)}$.

Definition 6.3.15 — Discrete Inverse Fourier Transform.

$$x_\ell = \frac{1}{n}\sum_{k=1}^{n} y_k e^{i\frac{2\pi}{n}(k-1)(\ell-1)}, \ell = 1, \cdots, n$$

6.3.7 Semi-Definite Matrices

Definition 6.3.16 — 半正定矩阵. 对称矩阵 $A \in \mathbb{R}^{n\times n}$ 称为半正定矩阵, 满足以下条件

$$x^T Ax \geq 0 \quad \forall x \in \mathbb{R}^n$$

Definition 6.3.17 — 正定矩阵. 对称矩阵 $A \in \mathbb{R}^{n\times n}$ 称为正定矩阵, 满足以下条件

$$x^T Ax > 0 \quad \forall x \neq 0$$

Definition 6.3.18 — 二次型. 如果对称矩阵 $A \in \mathbb{R}^{n\times n}$, 则 $x^T Ax$ 是二次型

Proof.

$$x^T Ax = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i A_{ij} x_j = \sum_{i=1}^{n} A_{ii}x_i^2 + 2\sum_{i>j} A_{ij}x_i x_j$$

■

▪ **Example 6.14**

$$A = \begin{bmatrix} 9 & 6 \\ 6 & a \end{bmatrix}$$

$$x^T Ax = 9x_1^2 + 12x_1x_2 + ax_2^2 = (3x_1 + 2x_2)^2 + (a-4)x_2^2$$

如果 $a > 4$, 矩阵 A 为正定矩阵:

$$x^T Ax > 0 \quad \forall x \neq 0$$

如果 $a = 4$, 矩阵 A 为半正定矩阵, 但不是正定矩阵:

$$x^T Ax \geq 0 \quad \forall x, \quad x^T Ax = 0 \quad \exists x = \begin{bmatrix} 2 \\ -3 \end{bmatrix}$$

如果 $a < 4$, 矩阵 A 不是半正定矩阵:

$$x^T Ax < 0 \quad \exists x = \begin{bmatrix} 2 \\ -3 \end{bmatrix}$$

▪

Corollary 6.3.5 正定矩阵 A 都是非奇异的.

Proof.

$$Ax = 0 \quad \Rightarrow \quad x^T Ax = 0 \quad \Rightarrow \quad x = 0$$

最后一步由正定性得到的.($x^T Ax > 0 \quad \forall x \neq 0$)

■

Theorem 6.3.6 — 正定矩阵对角元素性质. 正定矩阵 A 有正的对角元素.

$$A_{ii} = e_i^T A e_i > 0$$

Theorem 6.3.7 — 半正定矩阵对角元素性质. 每个半正定矩阵 A 都有非负的对角元素.

$$A_{ii} = e_i^T A e_i \geq 0$$

6.4 Gram 矩阵

Definition 6.4.1 — 实矩阵 A 的 Gram 矩阵.

$$G = A^T A = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_n^T \end{bmatrix} [a_1, a_2, \cdots, a_n] = \begin{bmatrix} a_1^T a_1 & a_1^T a_2 & \cdots & a_1^T a_n \\ a_2^T a_1 & a_2^T a_2 & \cdots & a_2^T a_n \\ \vdots & \vdots & \ddots & \vdots \\ a_n^T a_1 & a_n^T a_2 & \cdots & a_n^T a_n \end{bmatrix}$$

Definition 6.4.2 — 复矩阵的 A 的 Gram 矩阵.

$$G = A^H A = \begin{bmatrix} a_1^H a_1 & a_1^H a_2 & \cdots & a_1^H a_n \\ a_2^H a_1 & a_2^H a_2 & \cdots & a_2^H a_n \\ \vdots & \vdots & \ddots & \vdots \\ a_n^H a_1 & a_n^H a_2 & \cdots & a_n^H a_n \end{bmatrix}$$

Theorem 6.4.1 每个 Gram 矩阵都是半正定的.

Proof.

$$x^T A x = x^T B^T B x = \|Bx\|_2^2 \geq 0, \forall x$$

■

Theorem 6.4.2 如果 Gram 矩阵是正定的, 则要满足

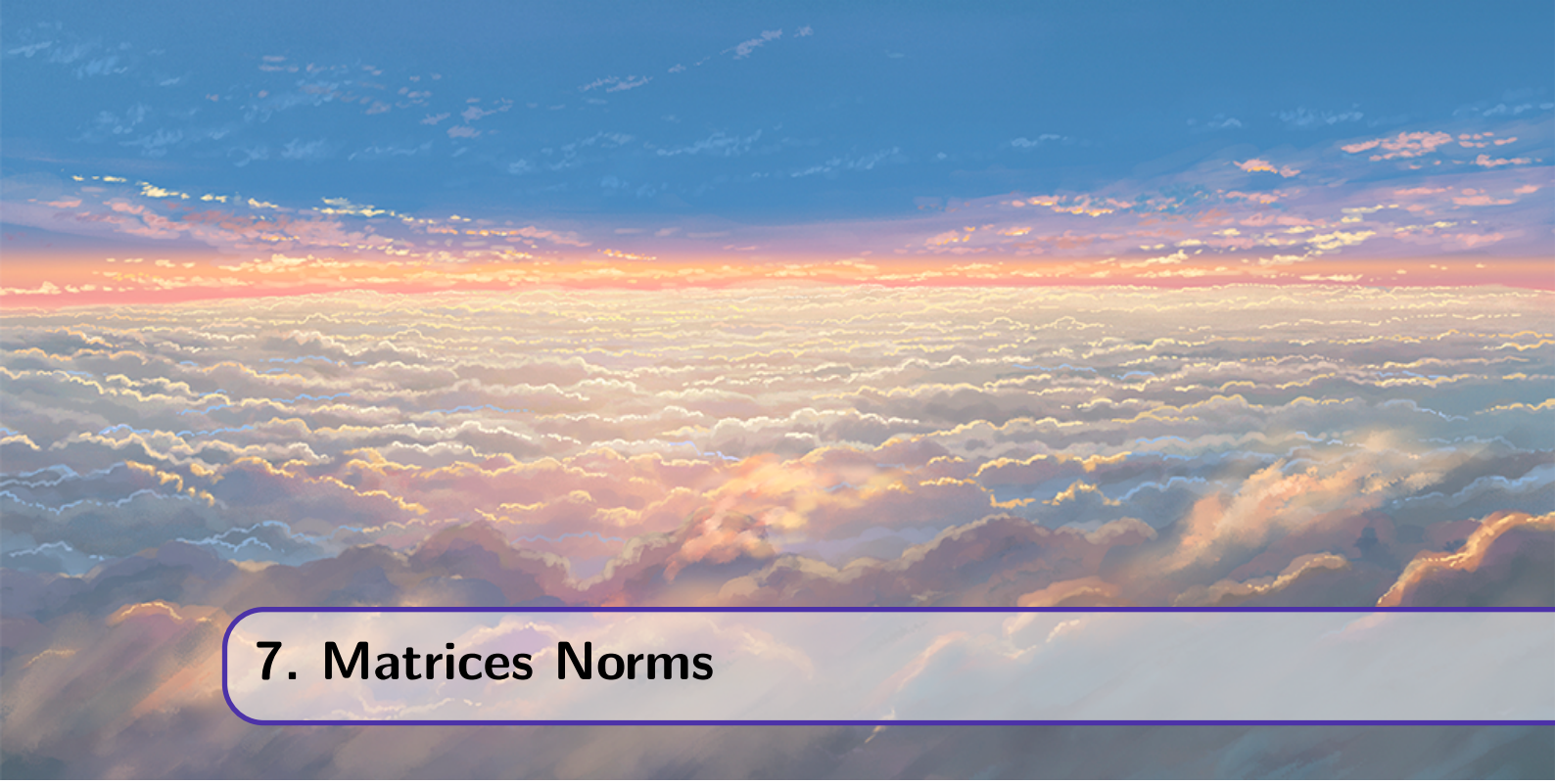
$$x^T A x = x^T B^T B x = \|Bx\|_2^2 > 0 (\forall x \neq 0)$$

Corollary 6.4.3 如果 Gram 矩阵是正定的, 则 B 的列向量是线性无关的.

Proof.

$$\|Bx\|_2^2 > 0 (\forall x \neq 0)$$

所以 $\forall x \neq 0, Bx \neq 0$.
注意和线性无关 5.1.2 的定义进行参照. ■



7. Matrices Norms

7.1 矩阵范数

Definition 7.1.1 — Matrix Norm. 向量空间中存在一个函数 $\|\cdot\|:\mathbb{R}^{m\times n}\to\mathbb{R}$ 且满足以下条件:

- 齐次性: $\|\alpha A\|=|\alpha|\|A\|, \alpha\in\mathbb{R}$ 且 $A\in\mathbb{R}^{m\times n}$;
- 三角不等式: $\|A+B\|\le\|A\|+\|B\|, A,B\in\mathbb{R}^{m\times n}$;
- 非负性: $\|A\|\ge 0, A\in\mathbb{R}^{m\times n}$ 且 $\|A\|=0\Leftrightarrow A=0$;

则称 $\|\cdot\|$ 为矩阵范数.

向量空间 $\mathbb{R}^{m\times n}$ 矩阵范数:

■ **Example 7.1 — F-范数 (Frobenius norm).**

$$\|A\|_F=\left(\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}^2\right)^{\frac{1}{2}}$$

■

Proof.

$$\|A\|_F\ge 0$$

$$\|\alpha A\|_F=|\alpha|\|A\|_F, \alpha\in\mathbb{R}$$

$$\begin{aligned}\|A+B\|_F&=\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\left(a_{ij}+b_{ij}\right)^2\right)^{\frac{1}{2}}\le\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\left(a_{ij}\right)^2\right)^{\frac{1}{2}}+\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\left(b_{ij}\right)^2\right)^{\frac{1}{2}}\\&=\|A\|_F+\|B\|_F\end{aligned}$$

■

Definition 7.1.2 — 从属于给定向量范数 $\|x\|_v$ 的矩阵范数. 设 $x \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n}, \|\cdot\|_v$ 为一种向量范数. 则 $\frac{\|Ax\|_v}{\|x\|_v}$ 对所有 $x \neq 0$ 有最大值, 令

$$\|A\|_v = \max_{x\neq 0}\left\{\frac{\|Ax\|_v}{\|x\|_v}\right\} = \max_{x\neq 0}\left\{\left\|A\frac{x}{\|x\|_v}\right\|_v\right\} = \max_{\|y\|_v=1}\{\|Ay\|_v\}$$

即

$$\|A\|_v = \max_{x\neq 0}\left\{\frac{\|Ax\|_v}{\|x\|_v}\right\}$$

$\|A\|_v$ 称为从属于给定向量范数 $\|x\|_v$ 的矩阵范数, 简称为从属范数或算子范数.

Proof. 可以验证 $\|A\|_v$ 满足矩阵范数定义.

$$\|A\|_v \geq 0$$

$$\|\alpha A\|_v = |\alpha|\|A\|_v, \alpha \in \mathbb{R}$$

$$\begin{aligned}\|A+B\|_v &= \max_{\|y\|_v=1}\|(A+B)y\|_v \\ &\leq \max_{\|y\|_v=1}\{\|Ay\|_v + \|By\|_v\} \\ &\leq \max_{\|y\|_v=1}\|Ay\|_v + \max_{\|y\|_v=1}\|By\|_v \\ &= \|A\|_v + \|B\|_v\end{aligned}$$

■

! 在本书中若未明确说明, $\|A\|$ 表示的是算子范数.

由定义 $\|A\|_v = \max_{x\neq 0}\left\{\frac{\|Ax\|_v}{\|x\|_v}\right\}$ 可得

Definition 7.1.3 — 向量范数和算子范数相容.

$$\frac{\|Ax\|_v}{\|x\|_v} \leq \|A\|_v \Rightarrow \|Ax\|_v \leq \|A\|_v\|x\|_v$$

称向量范数和算子范数相容.

Theorem 7.1.1 — 算子范数服从乘法范数相容性. 对于 $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{n\times p}$

$$\begin{aligned}\|AB\|_v &= \max_{x\neq 0}\left\{\frac{\|ABx\|_v}{\|x\|_v}\right\} \\ &\leq \max_{x\neq 0}\left\{\frac{\|A\|_v\|Bx\|_v}{\|x\|_v}\right\} \\ &\leq \|A\|_v \max_{x\neq 0}\left\{\frac{\|B\|_v\|x\|_v}{\|x\|_v}\right\} \\ &= \|A\|_v\|B\|_v\end{aligned}$$

算子范数服从乘法范数相容性.

根据向量的常用范数可以导出矩阵 $A \in \mathbb{R}^{m\times n}$ 的算子范数

Definition 7.1.4 — A 的列范数.

$$\|A\|_1 = \max_{x\neq 0}\left(\frac{\|Ax\|_1}{\|x\|_1}\right) = \max_{1\leq j\leq n}\sum_{i=1}^{m}\left|a_{ij}\right|$$

Definition 7.1.5 — A 的行范数.

$$\|A\|_\infty = \max_{x\neq 0}\left(\frac{\|Ax\|_\infty}{\|x\|_\infty}\right) = \max_{1\leq i\leq m}\sum_{j=1}^{n}\left|a_{ij}\right|$$

Definition 7.1.6 — A 的 2-范数.

$$\|A\|_2 = \max_{x\neq 0}\left(\frac{\|Ax\|_2}{\|x\|_2}\right) = \sqrt{\lambda_{\max}\left(A^TA\right)} \tag{7.1}$$

Proof. For any A Choose x to be the eigenvector of $A^{\mathrm{T}}A$ with largest eigenvalue $\lambda_{\max}$. The ratio in equation 7.1 is $\boldsymbol{x}^{\mathrm{T}}A^{\mathrm{T}}A\boldsymbol{x} = \boldsymbol{x}^{\mathrm{T}}\left(\lambda_{\max}\right)\boldsymbol{x}$ divided by $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$. This is $\lambda_{\max}$.

No $\boldsymbol{x}$ can give a larger ratio. The symmetric matrix $A^{\mathrm{T}}A$ has eigenvalues $\lambda_1,\ldots,\lambda_n$ and orthonormal eigenvectors $q_1,\boldsymbol{q}_2,\ldots,\boldsymbol{q}_n$. Every $\boldsymbol{x}$ is a combination of those vectors. Try this combination in the ratio and remember that $\boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{q}_j = 0$:

$$\frac{\boldsymbol{x}^{\mathrm{T}}A^{\mathrm{T}}A\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}} = \frac{\left(c_1\boldsymbol{q}_1+\cdots+c_n\boldsymbol{q}_n\right)^{\mathrm{T}}\left(c_1\lambda_1\boldsymbol{q}_1+\cdots+c_n\lambda_n\boldsymbol{q}_n\right)}{\left(c_1\boldsymbol{q}_1+\cdots+c_n\boldsymbol{q}_n\right)^{\mathrm{T}}\left(c_1\boldsymbol{q}_1+\cdots+c_n\boldsymbol{q}_n\right)} = \frac{c_1^2\lambda_1+\cdots+c_n^2\lambda_n}{c_1^2+\cdots+c_n^2}$$

The maximum ratio $\lambda_{\max}$ is when all c 's are zero, except the one that multiplies $\lambda_{\max}$. ■

The ratio in equation 7.1 is the Rayleigh quotient for the symmetric matrix $A^{\mathrm{T}}A$. Its maximum is the largest eigenvalue $\lambda_{\max}\left(A^{\mathrm{T}}A\right)$. The minimum ratio is $\lambda_{\min}\left(A^{\mathrm{T}}A\right)$. If you substitute any vector $\boldsymbol{x}$ into the Rayleigh quotient $\boldsymbol{x}^{\mathrm{T}}A^{\mathrm{T}}A\boldsymbol{x}/\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}$, you are guaranteed to get a number between $\lambda_{\min}\left(A^{\mathrm{T}}A\right)$ and $\lambda_{\max}\left(A^{\mathrm{T}}A\right)$.

The norm $\|A\|$ equals the largest singular value $\sigma_{\max}$ of A. The singular values $\sigma_1,\ldots,\sigma_r$ are the square roots of the positive eigenvalues of $A^{\mathrm{T}}A$. So certainly $\sigma_{\max} = (\lambda_{\max})^{1/2}$. Since U and V are orthogonal in $A = U\Sigma V^{\mathrm{T}}$, the norm is $\|\boldsymbol{A}\| = \sigma_{\max}$.

▪ **Example 7.2** 求矩阵 A 的各种常用范数

$$A = \begin{pmatrix} 1 & 2 & 0 \\ -1 & 2 & -1 \\ 0 & 1 & 1 \end{pmatrix}$$

$$\|A\|_1 = \max_{1\leq j\leq n}\sum_{i=1}^{n}\left|a_{ij}\right| = \max_{1\leq j\leq n}\{2,5,2\} = 5$$

$$\|A\|_\infty = \max_{1\le i\le n}\sum_{j=1}^{n}\left|a_{ij}\right| = \max_{1\le i\le n}\{3,4,2\} = 4$$

由于 $\|A\|_2 = \sqrt{\lambda_{\max}\left(A^TA\right)}$, 因此先求 A^TA 的特征值

$$A^TA = \begin{pmatrix} 1 & -1 & 0 \\ 2 & 2 & 1 \\ 0 & -1 & 1 \end{pmatrix}\cdot\begin{pmatrix} 1 & 2 & 0 \\ -1 & 2 & -1 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 1 \\ 0 & 9 & -1 \\ 1 & -1 & 2 \end{pmatrix}$$

特征方程为

$$\det\left(\lambda I - A^TA\right) = \begin{vmatrix} \lambda-2 & 0 & -1 \\ 0 & \lambda-9 & 1 \\ -1 & 1 & \lambda-2 \end{vmatrix} = 0$$

可得 A^TA 的特征值

$$\lambda_1 = 9.1428, \lambda_2 = 2.9211, \lambda_3 = 0.9361$$

■

对于 $\|A\|_2$ 需要计算 $\lambda_{\max}\left(A^TA\right)$, 直接根据特征方程计算特征值的算法复杂度太高.

8. 适定问题

8.1 The Definition of Well-posed Problem

In 1923, the French mathematician Hadamard introduced the notion of well-posed (适定) problem:

- A solution for the problem exists;
- The solution is unique;
- Perturbations in the data should cause small perturbations in the solution.

One of these conditions is not satisfied, the problem is said to be ill-posed (病态) and demands a special consideration.

▪ **Example 8.1** 假设 A 是非奇异矩阵

$$Ax = b$$

如果将 b 为 $b + \Delta b$, 方程新的解 $x + \Delta x$, 则有:

$$A(x + \Delta x) = b + \Delta b$$

即

$$\Delta x = A^{-1}\Delta b$$

如果小的变化 Δb 导致小变化 Δx, 则称解是稳定的. 如果小的变化 Δb 导致大变化 Δx, 则称解不稳定的.

设

$$A = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 + 10^{-10} & 1 - 10^{-10} \end{bmatrix}, \quad A^{-1} = \begin{bmatrix} 1 - 10^{10} & 10^{10} \\ 1 + 10^{10} & -10^{10} \end{bmatrix}$$

若 $b = (1, 1)$, 方程 Ax 的解 $x = (1, 1)$. 如果将 b 改为 $b + \Delta b$, 那么 x 的变化量为

$$\Delta x = A^{-1}\Delta b = \begin{bmatrix} \Delta b_1 - 10^{10}\,(\Delta b_1 - \Delta b_2) \\ \Delta b_1 + 10^{10}\,(\Delta b_1 - \Delta b_2) \end{bmatrix}$$

很小变化 Δb 会导致非常大变化 Δx! 由矩阵 A 定义的问题, 称为适定问题或病态问题.

▪

8.2 绝对误差的界限

假设 A 是非奇异的, 并给出定义:

Notation 8.1.

$$x = A^{-1}b$$

$$\Delta x = A^{-1}\Delta b$$

$\|\Delta x\|$ 的上界为:

$$\|\Delta x\|_2 \leq \left\|A^{-1}\right\|_2 \|\Delta b\|_2$$

矩阵范数 $\left\|A^{-1}\right\|_2$ 小时, 当 $\|\Delta b\|_2$ 变化很小, $\|\Delta x\|_2$ 也很小; $\left\|A^{-1}\right\|_2$ 大时, $\|\Delta x\|_2$ 可能很大, 即使 $\|\Delta b\|_2$ 很小.

8.3 相对误差的界限

假设 $b \neq 0$; 因此 $x \neq 0$

$\|\Delta x\|_2/\|x\|_2$ 的上界为:

$$\|\Delta x\|_2 = \left\|A^{-1}\Delta b\right\|_2 \leq \left\|A^{-1}\right\|_2 \|\Delta b\|_2 (\text{向量范数和算子范数相容})$$

$$\Rightarrow \frac{\|\Delta x\|_2}{\|x\|_2} \leq \frac{\left\|A^{-1}\right\|_2 \|\Delta b\|_2}{\|x\|_2} = \frac{\|A\|_2 \left\|A^{-1}\right\|_2 \|\Delta b\|_2}{\|x\|_2\|A\|_2} \leq \frac{\|A\|_2 \left\|A^{-1}\right\|_2 \|\Delta b\|_2}{\|b\|_2}$$

由 $\|b\|_2 = \|Ax\|_2 \leq \|A\|_2\|x\|_2$, 可得

$$\frac{\|\Delta x\|_2}{\|x\|_2} \leq \|A\|_2 \left\|A^{-1}\right\|_2 \frac{\|\Delta b\|_2}{\|b\|_2}$$

$\|A\|_2 \left\|A^{-1}\right\|_2$ 小，当 $\frac{\|\Delta b\|_2}{\|b\|_2}$ 相对变化很小时，$\frac{\|\Delta x\|_2}{\|x\|_2}$ 也变化很小;
$\|A\|_2 \left\|A^{-1}\right\|_2$ 大, $\frac{\|\Delta x\|_2}{\|x\|_2}$ 可远远大于 $\frac{\|\Delta b\|_2}{\|b\|_2}$.

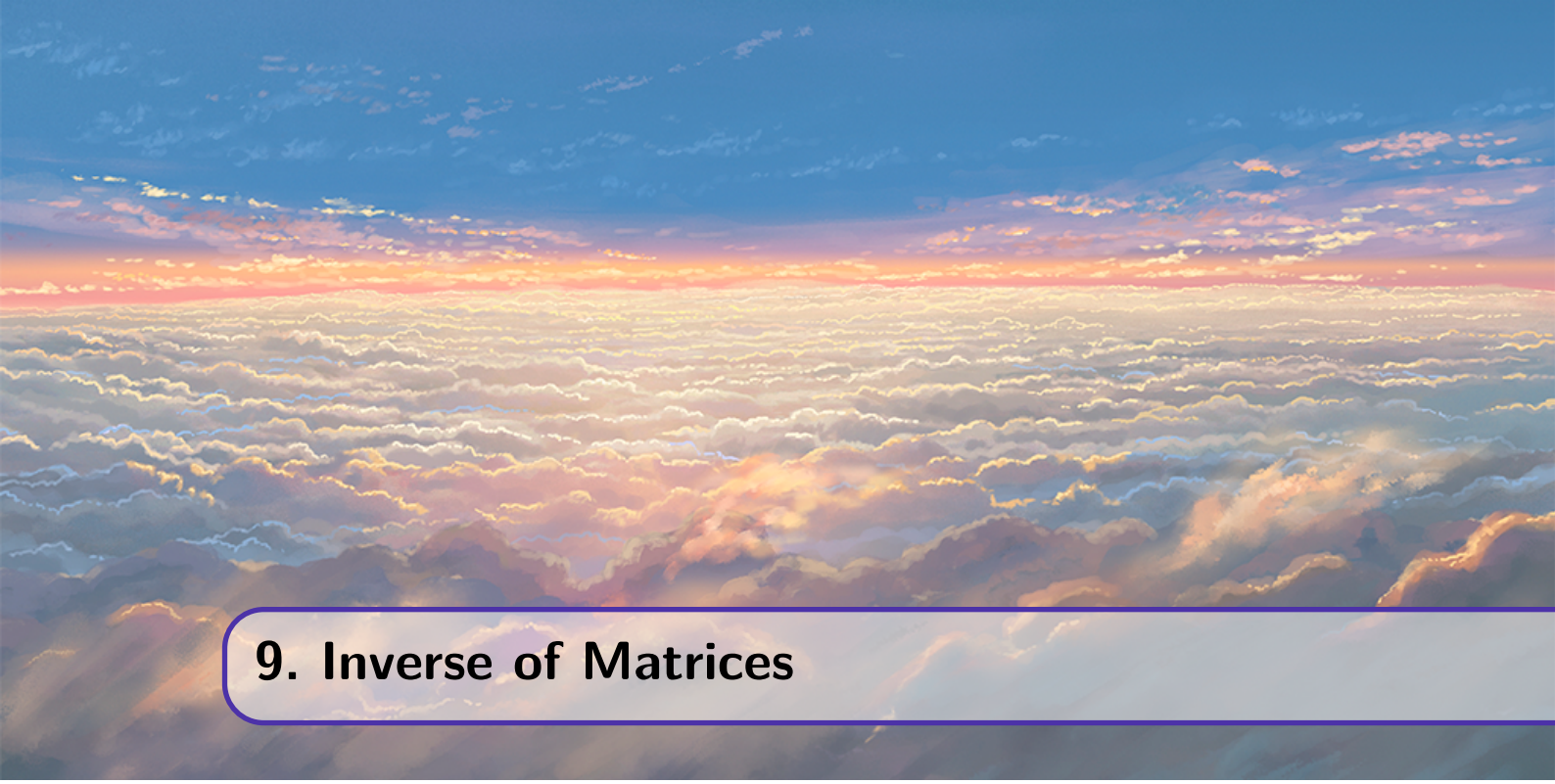
Definition 8.3.1 — 非奇异矩阵 A 的条件数 (condition number) . 条件数[a]定义

$$\kappa(A) = \|A\|_2 \left\|A^{-1}\right\|_2$$

[a] MathWorks Blog

Corollary 8.3.1 — 非奇异矩阵 A 的条件数 (condition number) 性质. 有如下性质:

- 对于所有 A, 有 $\kappa(A) \geq 1$;
- 如果 $\kappa(A)$ 比较小 (接近 1), x 的相对误差接近 b 的相对误差;
- 如果 $\kappa(A)$ 比较大（超过 100）, x 的相对误差比 b 的相对误差大得多.



9. Inverse of Matrices

9.1 Left Inverse, Right Inverse, Inverse

Definition 9.1.1 — A 的左逆. 当一个矩阵 X 满足

$$XA = I$$

X 被称为 A 的*左逆*; 当左逆存在时，则称 A 是*可左逆*的;

如果左逆矩阵存在, 则左逆矩阵有**无穷多**个.

▪ **Example 9.1**

$$A = \begin{bmatrix} -3 & -4 \\ 4 & 6 \\ 1 & 1 \end{bmatrix}$$

矩阵 A 是可左逆的，其左逆矩阵有两个

$$B = \frac{1}{9}\begin{bmatrix} -11 & -10 & 16 \\ 7 & 8 & -11 \end{bmatrix} \quad C = \frac{1}{2}\begin{bmatrix} 0 & -1 & 6 \\ 0 & 1 & -4 \end{bmatrix}$$

▪

Definition 9.1.2 — A 的右逆. 当左逆存在时，则称 A 是可左逆的;

如果右逆矩阵存在, 则右逆矩阵有**无穷多**个.

▪ **Example 9.2**

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

矩阵 B 可右逆，以下矩阵都是 B 的右逆

$$D = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \\ 1 & 1 \end{bmatrix}, E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, G = \begin{bmatrix} 1 & -1 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

■

一个大小为 $m \times n$ 的矩阵, 其左逆或右逆的维度为 $n \times m$.

Theorem 9.1.1 A 的左逆为 X 当且仅当 X^T 是 A^T 的右逆.

Proof.

$$A^T X^T = (XA)^T = I$$

■

Theorem 9.1.2 A 的右逆为 X 当且仅当 X^T 是 A^T 的左逆.

Proof.

$$X^T A^T = (AX)^{\mathrm{T}} = I$$

■

Theorem 9.1.3 如果矩阵 A 存在左逆和右逆，则左逆和右逆一定相等

Proof.

$$\begin{aligned} & XA = I, AY = I \\ \Rightarrow & X = XI = X(AY) = (XA)Y = Y \\ \Rightarrow & X = Y \end{aligned}$$

■

Definition 9.1.3 — 逆 A^{-1}. 如果矩阵 A 存在左逆和右逆, 此时 X 称为矩阵的*逆*，记作 A^{-1} 当矩阵的逆存在时，则称矩阵 A *可逆*.

9.2 Linear Equation Systems

Definition 9.2.1 — Linear Equation Systems. 有 n 个变量的 m 个方程为

$$\begin{cases} A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n = b_1 \\ A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n = b_2 \\ \vdots \\ A_{m1}x_1 + A_{m2}x_2 + \cdots + A_{mn}x_n = b_m \end{cases}$$

写成矩阵形式为：$\mathrm{A}x = \mathrm{b}$. 其中 A 为系数矩阵, x 为 n 维列向量.

该方程组可能**无解**，**有唯一解**和**无穷解**.

9.2.1 线性方程组求解

Theorem 9.2.1 如果矩阵 A 可左逆，假设 X 是矩阵 A 的左逆，则**至多**一个解, 如有解则 $x = Xb$.

Proof.

$$Ax = b \Rightarrow x = XAx = Xb$$

列满秩时 (下面证明), 列向量线性无关, 所以其零空间中只有零解，方程 $Ax = b$ 可能有一个唯一解 (b 在 A 的列空间中, 此特解就是全部解, 因为通常的特解可以通过零空间中

的向量扩展出一组解集，而此时零空间只有 0 向量），也可能无解（b 不在 A 的列空间中）
. ■

Theorem 9.2.2 如果矩阵 A 可右逆，假设 Y 是矩阵 A 的右逆，则**至少**一个解, 即 $x = Yb$
.

Proof. 设 $x = Yb$

$$x = Yb \Rightarrow Ax = AYb = b$$

右逆就是研究 $m \times n$ 矩阵 A 行满秩的情况, 此时 n > m = rank(A) . 对称的, 其左零空间中仅有零向量，即没有行向量的线性组合能够得到零向量.（$N(A^T) = \{0\}$）■

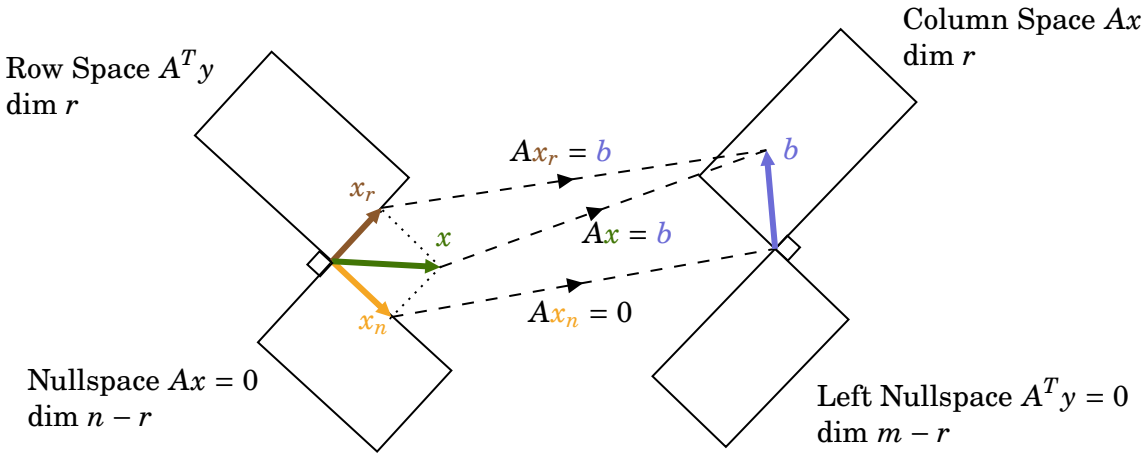
Theorem 9.2.3 如果矩阵 A 可逆的，假设 X 是矩阵 A 的逆，则

$$Ax = b \Rightarrow x = A^{-1}b$$

唯一解.

9.3 Fundamental Theorem of Linear Algebra

Figure 9.1: Four Subspace of Matrix A



$\boldsymbol{r = m}$	$\boldsymbol{r = n}$	Square and invertible	$\boldsymbol{Ax = b}$ has 1 solution
$\boldsymbol{r = m}$	$r < n$	Short and wide	$\boldsymbol{Ax = b}$ has ∞ solutions
$r < m$	$\boldsymbol{r = n}$	Tall and thin	$\boldsymbol{Ax = b}$ has 0 or 1 solution
$r < m$	$r < n$	Not full rank	$\boldsymbol{Ax = b}$ has 0 or ∞ solutions

The set of linear equations is called *over-determined* if $m > n$, *under-determined* if $m \leq n$, and *square* if $m = n$.

A set of equations with zero right-hand side, $Ax = 0$, is called a *homogeneous* set of equations. Any homogeneous set of equations has $x = 0$ as a solution.

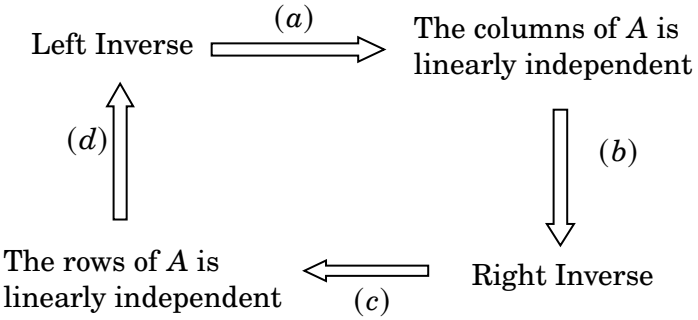
9.4 Invertible Matrices

Theorem 9.4.1 对于方阵 $A \in \mathbb{R}^{n\times n}$ ，以下条件都是等价的:

1. A 可左逆
2. A 的列向量线性无关
3. A 可右逆
4. A 的行向量线性无关
5. A 可逆

此时矩阵 A 为非奇异矩阵，由条件 1 与 3，可得 A 为可逆矩阵.

Proof. 可以通过以下方式证明:



- 性质 (a) 对任意矩阵 $A \in \mathbb{R}^{m\times n}$ 都成立
- 性质 (b) 对方阵矩阵 $A \in \mathbb{R}^{n\times n}$ 都成立
- 对于性质 (c) 与 (d), 可利用 A^T 证明

■

Theorem 9.4.2 (a): A 可左逆，则 A 列向量线性无关.

Proof. 假设 A 的左逆是 B ，则

$$\begin{aligned} Ax = 0 \Rightarrow \quad BAx = 0 \\ \Rightarrow Ix = 0 \end{aligned}$$

假设 A 的列向量 $A = [a_1, a_2, \cdots, a_n]$

$$Ax = x_1a_1 + x_2a_2 + \cdots + x_na_n = 0$$

则当该等式 $Ax = 0$ 成立时，其解 $x = 0$, 则 A 的列向量线性无关.

Corollary 9.4.3 如果 $A \in \mathbb{R}^{m\times n}$ 有左逆，则有 $m \geq n = r$.

即 A 是高或方的矩阵, 如 $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$.

此时 A 的行向量可能线性相关，而 A 的列向量线性无关. $N(A) = \{0\}$.

假设 A 的列向量 $A = [a_1, a_2, \cdots, a_n]$

$$\begin{aligned} Ax = x_1a_1 + x_2a_2 + \cdots + x_na_n = b \\ Ay = y_1a_1 + y_2a_2 + \cdots + y_na_n = b \end{aligned}$$

$$Ax - Ay = A(x - y) = 0 \Rightarrow x = y$$

当 $b \in \mathbb{R}^m, b \notin \{y \mid y = Ax, x \in \mathbb{R}^n\}$ 时（即 b 不在 A 的列空间，$m \geq n$ 时），线性方程组无解. $Ax = b$ 至多一个解，如有解则 $x = Xb$. ■

Theorem 9.4.4 矩阵的行秩等于列秩.

Proof. 令 A 是一个 $m \times n$ 的矩阵，其列秩为 r. 因此矩阵 A 的列空间的维度是 r .

令 $c_1, c_2, \ldots, c_r$ 是 A 的列空间的一组基,构成 $m \times r$ 矩阵 C 的列向量 $C = [c_1, c_2, \ldots, c_r]$, 并使得 A 的每个列向量是 C 的 r 个列向量的线性组合.

由矩阵乘法的定义，存在一个 $r \times n$ 矩阵 R, 使得 $A = CR$. (A 的 (i, j) 元素是 c_i 与 R 的第 j 个行向量的点积.)

现在，由于 $A = CR$, A 的每个行向量是 R 的行向量的线性组合，这意味着 A 的行向量空间被包含于 R 的行向量空间之中. 因此 A的行秩 $\leq R$的行秩. 但 R 仅有 r 行, 所以 R的行秩 $\leq r = A$的列秩. 这就证明了 A的行秩 $\leq A$的列秩.

把上述证明过程中的“行”与“列”交换，利用对偶性质同样可证 A的列秩 $\leq A$的行秩. 更简单的方法是考虑 A 的转置矩阵 A^{T}, 则 A的列秩 $= A^{\mathrm{T}}$的行秩 $\leq A^{\mathrm{T}}$的列秩 $= A$的行秩. 这证明了 A 的列秩等于 A 的行秩. 证毕. ■

Theorem 9.4.5 (c): 矩阵 $A \in \mathbb{R}^{m \times n}$ 有右逆 X, 则 A 行向量线性无关.

Proof.

$$\mathrm{X}^T A^T = (AX)^T = I$$

则有 X^T 是 A^T 的左逆, A^T 的列向量线性无关.

即 $A^T \in \mathbb{R}^{n \times m}$.

Corollary 9.4.6 如果 $A \in \mathbb{R}^{m \times n}$ 有左逆，则有 $r = m \leq n$.

即 A 是宽或方的矩阵.

此时 A 的列向量可能线性相关，而 A 的行向量线性无关.

$N(A^T) = \{0\}, \dim N(A) = n - r, r = m$. ($Ax = b$ 有无穷解)

根据定理“矩阵的行秩等于列秩”，A^T 的列向量线性无关，则矩阵 A 有 m 个线性无关列向量（行向量），即通过 Gram-Schmidt 正交化可得 m 个正交基.

$\forall b \in \mathbb{R}^m$, 有 $b \in \{y \mid y = Ax, x \in \mathbb{R}^n\}$ $(m \leq n)$, 方程 $Ax = b$ 有解，其解为 $x = Xb$. ■

Theorem 9.4.7 (b): 若方阵 A 列向量线性无关，则 A 可右逆.

Proof. 假设 $A \in \mathbb{R}^{n \times n}$ 为方阵且列向量线性无关

$$A = [a_1, a_2, \cdots, a_n]$$

则对于任意向量 b $\in \mathbb{R}^n$, 则向量组 $[a_1, a_2, \ldots, a_n,$ b] 线性相关，存在不全为 0 的系数，使得以下等式成立

$$x_1 a_1 + x_2 a_2 + \cdots + x_n a_n + x_{n+1} b = 0$$

因为 A 列向量线性无关，则 $x_{n+1} \neq 0$（假设 $x_{n+1} = 0$ 会推出违反线性无关假设的结论），即 b 是 A 列向量的线性组合;

$$b = -\frac{x_1}{x_{n+1}} a_1 - \frac{x_2}{x_{n+1}} a_2 - \cdots - \frac{x_n}{x_{n+1}} a_n$$

存在向量 $c_1, \ldots, c_n \in \mathbb{R}^n$, 使得

$$\begin{aligned} Ac_1 &= e_1 \\ Ac_2 &= e_2 \\ &\cdots \\ Ac_n &= e_n \end{aligned}$$

则矩阵 $C = [c_1 c_2 \ldots c_n]$ 是矩阵 A 的右逆, $AC = I$.

■

9.5 转置和共轭转置的逆

Theorem 9.5.1 — 转置 A^T 和共轭转置 A^{H}. 如果矩阵 A 为非奇异矩阵，则其转置 A^T 和共轭转置 A^{H} 都为非奇异矩阵，则有

$$\left(A^T\right)^{-1} = \left(A^{-1}\right)^T, \quad \left(A^H\right)^{-1} = \left(A^{-1}\right)^H$$

Proof.

$$\left(AA^{-1}\right)^T = I \Rightarrow \underbrace{\left(A^{-1}\right)^T}_{\text{the inverse of } A^T} A^T = I$$

■

Corollary 9.5.2 如果矩阵 A 和矩阵 B 都为非奇异矩阵，则乘积 AB 也为非奇异矩阵

$$(AB)^{-1} = B^{-1}A^{-1}$$

Proof.

$$(AB) \underbrace{B^{-1}A^{-1}}_{\text{the inverse of AB}} = I$$

■

9.6 Gram Matrix 非奇异的性质

Gram 矩阵的定义见 6.4.1.

Corollary 9.6.1 — Gram Matrix 可逆等价于 A 列线性无关. 矩阵 $A \in \mathbb{R}^{m\times n}, \mathrm{G} = A^T A$

矩阵 A 列向量线性无关 $\Leftrightarrow$ Gram 矩阵 G 非奇异.

Proof. " $\Rightarrow$ ":

假设矩阵 A 列向量线性无关, $A^T A$ 奇异. 则存在 $A^T Ax = 0, x \neq 0$, 可得 $x^T A^T Ax = \|Ax\|_2^2 = 0$, 即 $Ax = 0$ 与列向量线性无矛盾.

" $\Leftarrow$ ":

假设 $A^T A$ 非奇异, 矩阵 A 列向量线性相关. 则有 $Ax = 0, x \neq 0$, 可得 $A^T Ax = 0$, 即 $A^T A$ 是奇异矩阵. ■

9.7 伪逆

Definition 9.7.1 — Pseudo-inverse.

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

$$A^{\dagger} = V\Sigma^{+}U^T = [v_1 \cdots v_r \cdots v_n] \begin{bmatrix} \sigma_1^{-1} & & \\ & \ddots & \\ & & \sigma_r^{-1} \end{bmatrix} [u_1 \cdots u_r \cdots u_m]^{\mathrm{T}}$$

Theorem 9.7.1 伪逆 $A^{\dagger}$ 为 A 的右逆

Proof.

$$AA^{\dagger} = AA^T \left(AA^T\right)^{-1} = \left(AA^T\right)^{-1} \left(AA^T\right) = I$$

■

Theorem 9.7.2 当 A 为方阵时，右逆等于矩阵的逆

Proof.

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

■

Theorem 9.7.3

Corollary 9.7.4 以下三个结论为等价的，对于实矩阵 A

- A 是可左逆的
- A 的列向量线性无关
- $A^T A$ 为非奇异矩阵

Corollary 9.7.5 以下三个结论为等价的，对于实矩阵 A

- A 是可右逆的
- A 的行向量线性无关
- AA^T 为非奇异矩阵

By choosing good bases, A multiplies $\boldsymbol{v}_i$ in the row space to give $\sigma_i \boldsymbol{u}_i$ in the column space. A^{-1} must do the opposite!

If $A\boldsymbol{v} = \sigma \boldsymbol{u}$ then $A^{-1}\boldsymbol{u} = \boldsymbol{v}/\sigma$. The singular values of A^{-1} are $1/\sigma$, just as the eigenvalues of A^{-1} are $1/\lambda$. The bases are reversed. The u 's are in the row space of A^{-1}, the v 's are in the column space.

The pseudoinverse $A^{+}$is an n by m matrix. **If A^{-1} exists, then $A^{+}$is the same as** A^{-1}. In that case $m = n = r$ and we are inverting $U\Sigma V^{\mathrm{T}}$ to get $V\Sigma^{-1}U^{\mathrm{T}}$.

The new symbol $A^{+}$is needed when $r < m$ or $r < n$. Then A has no two-sided inverse, but it has a *pseudo*inverse $A^{+}$with that same rank r :

$$A^{+}\boldsymbol{u}_i = \frac{1}{\sigma_i}\boldsymbol{v}_i \quad \text{for } i \le r \quad \text{and} \quad A^{+}\boldsymbol{u}_i = \boldsymbol{0} \quad \text{for } i > r$$

The vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r$ in the column space of A go back to $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r$ in the row space.

The other vectors $\boldsymbol{u}_{r+1}, \ldots, \boldsymbol{u}_m$ are in the left nullspace, and A^+sends them to zero. When we know what happens to all those basis vectors, we know A^+.

Notice the pseudoinverse of the diagonal matrix Σ. Each σ in Σ is replaced by σ^{-1} in Σ^+.The product $\Sigma^+\Sigma$ is as near to the identity as we can get. It is a projection matrix, $\Sigma^+\Sigma$ is partly I and otherwise zero. We can invert the σ 's, but we can't do anything about the zero rows and columns.

Figure 9.2: $A\boldsymbol{x}^\dagger$ in the column space goes back to $A^\dagger A\boldsymbol{x}^\dagger = \boldsymbol{x}^\dagger$ in the row space

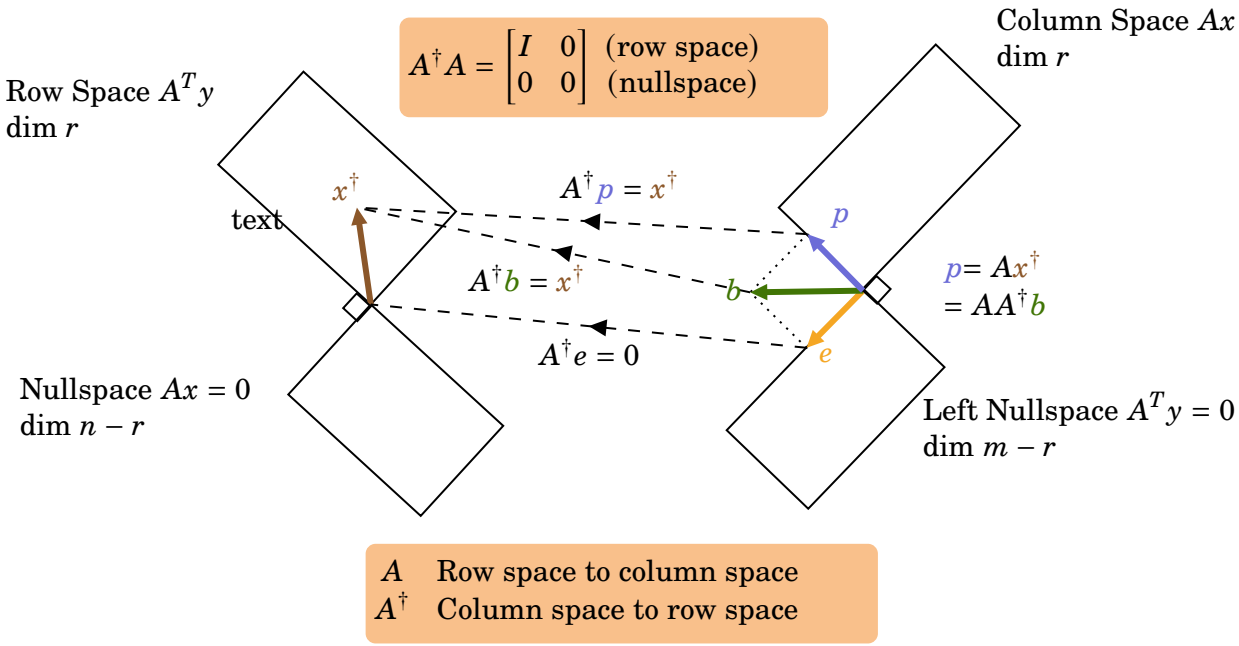
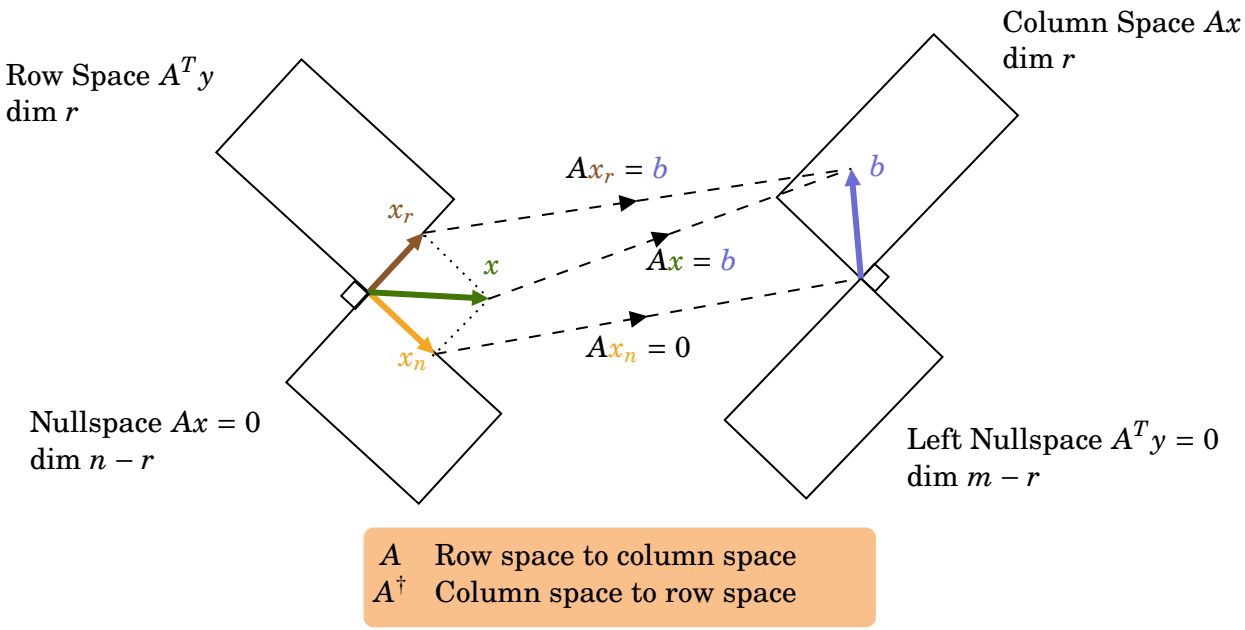


Figure 9.3: Projection from row space to column space



10. Orthogonal Matrices

10.1 预备知识

10.1.1 标准正交向量

参见 5.3.2。

10.1.2 Gram 矩阵与标准正交的关系

For the definition of Gram matrices, refer to 6.4.1. 关于它与非奇异的性质，参见 9.6.

Theorem 10.1.1 如果 A 的 Gram 矩阵为单位矩阵，则 $A \in \mathbb{R}^{m\times n}$ 具有标准正交列.

Proof.

$$
\begin{aligned}
A^TA &= \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix}^T \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \\
&= \begin{bmatrix} a_1^Ta_1 & a_1^Ta_2 & \cdots & {a_1}^Ta_n \\ a_2^Ta_1 & {a_2}^Ta_2 & \cdots & a_2^Ta_n \\ \vdots & \vdots & \ddots & \vdots \\ a_n^Ta_1 & a_n^Ta_2 & \cdots & a_n^Ta_n \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \\
&= I
\end{aligned}
$$

■

10.1.3 矩阵-向量乘积与标准正交的关系

如果 $A \in \mathbb{R}^{m\times n}$ 具有标准正交列，则线性函数 $f(x) = Ax$

Theorem 10.1.2 保持原内积.

$$\langle Ax, Ay\rangle = x^T y$$

Proof.

$$\langle Ax, Ay\rangle = (Ax)^T(Ay) = x^T A^T A y = x^T y$$

■

Theorem 10.1.3 保持原范数.

$$\|Ax\|_2 = \|x\|_2$$

Proof.

$$\|Ax\|_2 = \left((Ax)^T(Ax)\right)^{1/2} = \left(x^T x\right)^{1/2} = \|x\|_2$$

■

Theorem 10.1.4 保持原距离.

$$\|Ax - Ay\|_2 = \|x - y\|_2$$

Proof.

$$\|Ax - Ay\|_2 = \left((Ax - Ay)^T(Ax - Ay)\right)^{1/2} = \left((x - y)^T(x - y)\right)^{1/2} = \|x - y\|_2$$

■

Theorem 10.1.5 保持原角度.

$$\angle(Ax, Ay) = \angle(x, y)$$

Proof.

$$\angle(Ax, Ay) = \arccos\left(\frac{(Ax)^T(Ay)}{\|Ax\|_2\|Ay\|_2}\right) = \arccos\left(\frac{x^T y}{\|x\|_2\|y\|_2}\right) = \angle(x, y)$$

■

10.1.4 左可逆性与正交的关系

如果矩阵 $A \in \mathbb{R}^{m\times n}$ 有标准正交列，则：

Theorem 10.1.6 A 是左可逆的，其左逆为 A^T.

Proof. 根据定义：

$$A^T A = I$$

■

Theorem 10.1.7 A 有线性无关的列向量.

Proof.

$$Ax = 0 \quad \Rightarrow \quad A^T Ax = x = 0$$

■

Theorem 10.1.8 A 是高的或者方的，即 $m \geq n$.

Proof. 列向量 $a_1, a_2, \ldots, a_n \in \mathbb{R}^m$, 由维度定理可得 $n \leq m$。 ■

10.2 正交矩阵

Definition 10.2.1 — 正交矩阵. 所有列两两相互正交的**方形**实矩阵。

Theorem 10.2.1 — 正交矩阵满足非奇异性. 即如果矩阵 A 是正交的，则 A 是可逆的，左逆等于右逆，且它的逆为 A^T.

$$\left.\begin{array}{l} A^T A = I \\ A \text{ 是方的} \end{array}\right\} \quad \Rightarrow \quad AA^T = I$$

Corollary 10.2.2 A^T 也是一个正交矩阵。

Corollary 10.2.3 A 的行是标准正交的，即 a_i 范数为 1 且相互正交。

如果 $A \in \mathbb{R}^{m \times n}$ 有标准正交列以及 $m > n$, 则 $AA^T \neq I$。

10.3 Permutation Matrices

Notation 10.1. $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ 为 $(1, 2, \ldots, n)$ 的一个重新排序的排列。
将 π 与一个置换矩阵 $A \in \mathbb{R}^{n \times n}$ 联系起来:

$$A_{i\pi_i} = 1, \quad A_{ij} = 0 \text{ 如果} j \neq \pi_i$$

Definition 10.3.1 — 置换. Ax 是 x 的一个置换:

$$Ax = \left(x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n}\right)$$

A 在每一行和每一列中都有一个等于 1 的元素。

Theorem 10.3.1 置换矩阵满足正交性，即所有置换矩阵都是正交的.

Corollary 10.3.2

$$A^T A = I$$

Proof. 因为 A 的每一行有一个元素等于 1

$$\left(A^T A\right)_{ij} = \sum_{k=1}^{n} A_{ki} A_{kj} = \begin{cases} 1 & i = j \\ 0 & \text{其它} \end{cases}$$

■

Corollary 10.3.3 $A^T = A^{-1}$ 是逆置换矩阵.

▪ **Example 10.1** 若 $\{1, 2, 3, 4\}$ 的置换为:

$$(\pi_1, \pi_2, \pi_3, \pi_4) = (2, 4, 1, 3)$$

相应的置换矩阵及其逆矩阵为

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, A^{-1} = A^T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

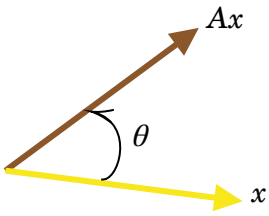
A^T 是与置换相关的置换矩阵

$$(\tilde{\pi}_1, \tilde{\pi}_2, \tilde{\pi}_3, \tilde{\pi}_4) = (3, 1, 4, 2)$$

■

10.4 平面旋转

Figure 10.1: An example of rotation



▪ **Example 10.2 — Rotation Matrices in $\mathbb{R}^2$.** 在一个 2 维平面的旋转可以用矩阵表示为

$$A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

■

▪ **Example 10.3 — Rotation Matrices in $\mathbb{R}^3$.**

$$A = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

描述了在 $\mathbb{R}^3$ 中 (x_1, x_3) 平面的旋转。 ■

10.5 Householder Matrix

Definition 10.5.1 — Householder matrix, reflator matrix.

$$A = I - 2aa^T$$

其中，向量 a 满足 $\|a\|_2 = 1$ 。

Theorem 10.5.1 反射矩阵 (reflector matrix) 是对称的.

$$A^T = A$$

Theorem 10.5.2 反射矩阵 (reflector matrix) 是正交的。

Proof.

$$A^T A = \left(I - 2aa^T\right)\left(I - 2aa^T\right) = I - 4aa^T + 4aa^T aa^T = I$$

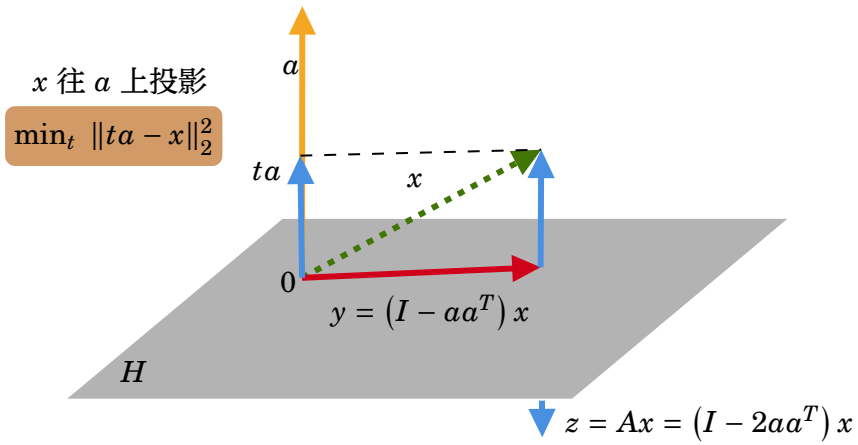
■

Theorem 10.5.3 反射矩阵是对合矩阵 (Involutory matrix)，即它的逆是它本身。

$$H = H^{-1}$$

10.5.1 The Geometry of Householder Transformation

Figure 10.2: Elementary Reflection



$H = \{u \mid a^T u = 0\}$ 是与 a 正交的向量的 (超) 平面。

Corollary 10.5.4 如果 $\|a\|_2 = 1$， x 在 H 上的投影为

$$y = x - \left(a^T x\right) a = x - a\left(a^T x\right) = \left(I - aa^T\right) x$$

Proof. 1. $y \in H$.

$$a^T y = a^T\left(x - a\left(a^T x\right)\right) = a^T x - \left(a^T a\right)\left(a^T x\right) = a^T x - a^T x = 0$$

2. 考虑任意 $z \in H (z \neq y)$, 证明 $\|x - z\| > \|x - y\|$

$$\begin{aligned}\|x-z\|_2^2 &= \|x-y+y-z\|_2^2 \\ &= \|x-y\|_2^2 + 2(x-y)^T(y-z) + \|y-z\|_2^2 \\ &= \|x-y\|_2^2 + 2\left(a^T x\right)a^T(y-z) + \|y-z\|_2^2 \\ &= \|x-y\|_2^2 + \|y-z\|_2^2 \quad (\text{因为} a^T y = a^T z = 0) \\ &\geq \|x-y\|_2^2\end{aligned}$$

■

Corollary 10.5.5 x 通过超平面的反射由反射算子的乘积给出

$$z = y + (y-x) = \left(I - 2aa^T\right)x$$

10.6 正交矩阵乘积

若 $A_1, \ldots, A_k \in \mathbb{R}^{n\times n}$ 是正交矩阵，那么它们的乘积为:

$$A = A_1 A_2 \cdots A_k$$

Corollary 10.6.1 — 正交矩阵乘积的正交性.

$$\begin{aligned}A^T A &= (A_1 A_2 \cdots A_k)^T (A_1 A_2 \cdots A_k) \\ &= A_k^T \cdots A_2^T A_1^T A_1 A_2 \cdots A_k \\ &= I\end{aligned}$$

10.7 具有正交矩阵的线性方程

系数正交矩阵 $A \in \mathbb{R}^{n\times n}$ 的线性方程;

$$Ax = b$$

解为:

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

10.7.1 The Complexity of the Multiplication Ax of Orthogonal Matrix A

可以在 $2n^2$ 个 flop 内计算矩阵向量乘法。

如果 A 有特殊性质，代价将会小于 n^2。例如，

- 置换矩阵：0 flop。
- 反射算子 (给定 a)：$4n$ flops。
- 平面旋转：$O(1)$ flop。

10.8 列标准正交的高矩阵

Theorem 10.8.1 假设矩阵 $A \in \mathbb{R}^{m\times n}$ 是高的 $(m > n)$, 具有标准正交列, 则有 A^T 具有标准正交行.

Theorem 10.8.2 A^T 是 A 的一个左逆。

$$A^T A = I$$

Theorem 10.8.3 A 没有右逆。

10.9 值域范围、列空间

Definition 10.9.1 — 向量集合张成的空间. 一个向量集合张成的空间是其所有线性组合的集合.

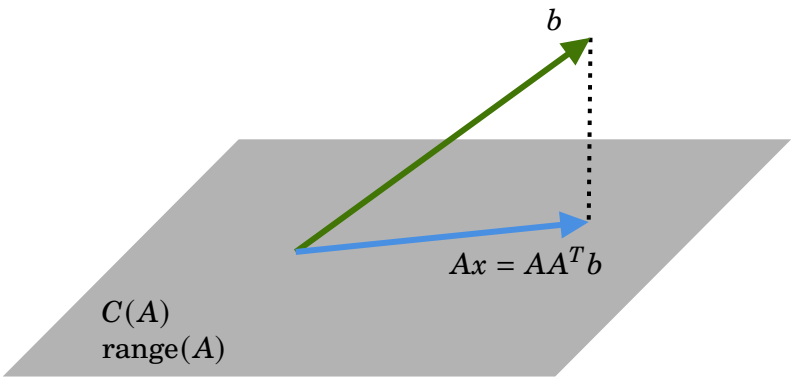
$$\text{span}\,(a_1, a_2, \cdots, a_n) = \{x_1 a_1 + x_2 a_2 + \cdots + x_n a_n \mid x \in \mathbb{R}^n\}$$

Definition 10.9.2 — A 的范围（列空间）.

$$\text{range}(A) = \{Ax \mid x \in \mathbb{R}^n\}$$

10.9.1 投影到列标准正交的矩阵 A 的列空间

Figure 10.3: Projection onto the column space of A, A has orthonormal columns



Problem 10.1 假设矩阵 $A \in \mathbb{R}^{m\times n}$ **具有标准正交列**, 求投影 b 在 $C(A)$ 上的投影。 ∎

即向量 Ax 与 b 有最短距离

$$\min_x \|Ax - b\|_2^2$$

$$\begin{aligned} f(x) &= \|Ax - b\|_2^2 \\ &= (Ax - b)^T (Ax - b) = x^T A^T Ax - 2x^T A^T b + b^T b \\ &= x^T x - 2x^T A^T b + b^T b \left(\because A^T A = I\right) \end{aligned}$$

$$\nabla f(x) = 2x - 2A^T b = 0 \Rightarrow x = A^T b$$

$AA^T b$ 称为向量 $\mathrm{b} \in \mathbb{R}^m$ 在 $\text{range}(A)$ 上的正交投影。

Theorem 10.9.1

$$Ax = AA^T b \in \text{range}(A)$$

且是 b 在 $C(A)$ 上的投影。

Proof. 1.

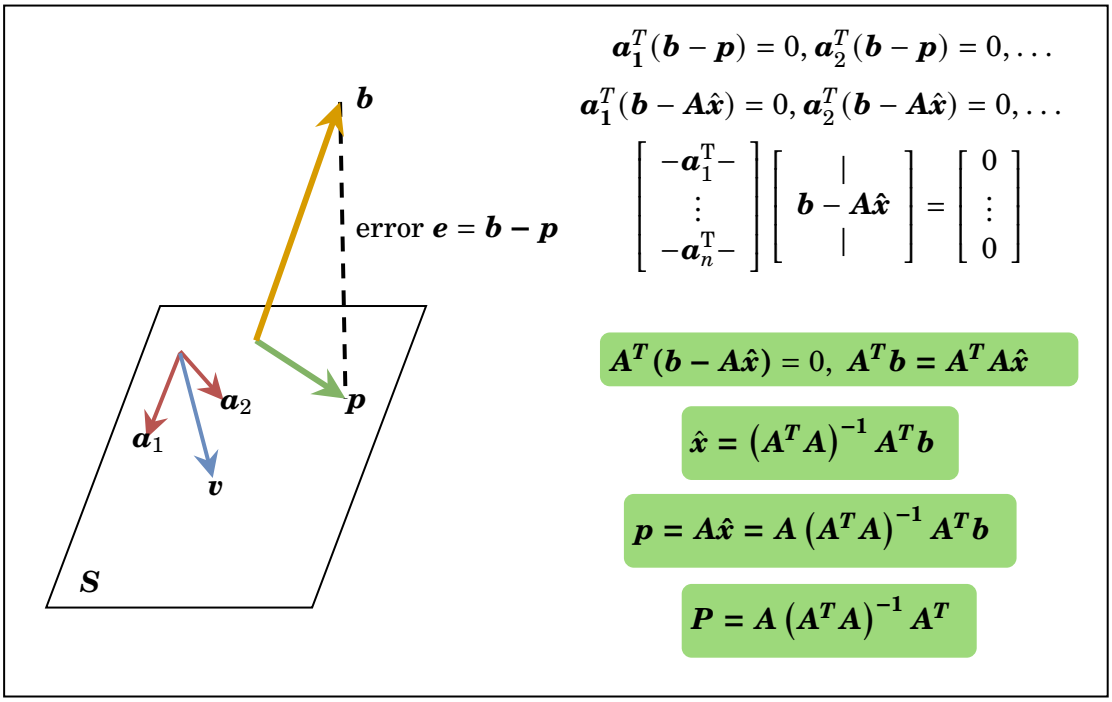
$$Ax = AA^T b \in \text{range}(A)$$

2. 可以证明 $\hat{x} = A^T b$ 满足 $\|A\hat{x} - b\| < \|Ax - b\|$, 对于所有 $x \neq \hat{x}$。
b 到 range(A) 内任意点 Ax 的距离的平方和为:

$$\begin{aligned}
\|Ax - b\|_2^2 &= \|A(x - \hat{x}) + A\hat{x} - b\|_2^2 \quad \left(\text{其中}\hat{x} = A^T b\right) \\
&= \|A(x - \hat{x})\|_2^2 + \|A\hat{x} - b\|_2^2 + 2(x - \hat{x})^T A^T (A\hat{x} - b) \\
&= \|A(x - \hat{x})\|_2^2 + \|A\hat{x} - b\|_2^2 \\
&= \|(x - \hat{x})\|_2^2 + \|A\hat{x} - b\|_2^2 \\
&\geq \|A\hat{x} - b\|_2^2
\end{aligned}$$

当且仅当 $x = \hat{x}$, 等号成立。
第 3 行成立是因为 $A^T(A\hat{x} - b) = \hat{x} - A^T b = 0$ 。 ■

Figure 10.4: Projection of $\boldsymbol{b}$ into the column space of A, A is any matrice. Sourced from [**Strang1993IntroductionTL**]



11. QR 分解与 Householder 变换

11.1 Triangular Matrices

Definition 11.1.1 — Lower Triangular Matrices. 矩阵 $A \in \mathbb{R}^{n\times n}$ 为下三角 *(Lower Triangular)* 矩阵, $A_{ij} = 0, j > i$ 。

$$A = \begin{bmatrix} A_{11} & 0 & \cdots & 0 & 0 \\ A_{21} & A_{22} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_{n-1,1} & A_{n-1,2} & \cdots & A_{n-1,n-1} & 0 \\ A_{n1} & A_{n2} & \cdots & A_{n,n-1} & A_{nn} \end{bmatrix}$$

Definition 11.1.2 — Upper Triangular Matrices. A^T 为上三角 *(Upper Triangular)* 矩阵。

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1,n-1} & A_{1,n} \\ 0 & A_{22} & \cdots & A_{2,n-1} & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & A_{n-1,n-1} & A_{n-1,n} \\ 0 & 0 & \cdots & 0 & A_{nn} \end{bmatrix}$$

Definition 11.1.3 — 单位上三角矩阵，单位下三角矩阵. 对角元素 a_{ii} 都等于 1 的上（下）三角矩阵。

11.1.1 高斯消元法

Problem 11.1 当 A 是具有非零对角元素的下三角矩阵时，解 $Ax = b$ 。 ∎

使用前向回代 (Forward Substitution) 算法求解。

时间复杂度：$1 + 3 + 5 + \ldots + (2n - 1) = n^2$ flops

Problem 11.2 当 A 是具有非零对角元素的上三角矩阵, 解 $Ax = b$ 。 ∎

使用后向回代 (Back Substitution) 算法来求解.

Algorithm 4: Forward Substitution

Input: $A \in R^{n\times n}$(A是下三角矩阵), $b \in R^{n\times 1}$
Output: $x \in R^{n\times 1}$

$$x_1 = \frac{b_1}{A_{11}}$$
$$x_2 = \frac{b_2 - A_{21}x_1}{A_{22}}$$
$$x_3 = \frac{b_3 - A_{31}x_1 - A_{32}x_2}{A_{33}}$$
$$\cdots$$
$$x_n = \frac{b_n - A_{n1}x_1 - A_{n2}x_2 - \cdots - A_{n,n-1}x_{n-1}}{A_{nn}}$$

Algorithm 5: Backward Substitution

Input: $A \in R^{n\times n}$(A是上三角矩阵), $b \in R^{n\times 1}$
Output: $x \in R^{n\times 1}$

$$x_n = \frac{b_n}{A_{nn}}$$
$$x_{n-1} = \frac{b_{n-1} - A_{n-1,n}x_n}{A_{n-1,n-1}}$$
$$x_{n-2} = \frac{b_{n-2} - A_{n-2,n-1}x_{n-1} - A_{n-2,n}x_n}{A_{n-2,n-2}}$$
$$\cdots$$
$$x_1 = \frac{b_1 - A_{12}x_2 - A_{13}x_3 - \cdots - A_{1n}x_n}{A_{11}}$$

时间复杂度：$1+3+\ldots+2\mathrm{n}-1=\mathrm{n}^2$ flops

11.1.2 The Inverses of Triangular Matrices

Theorem 11.1.1 对角元素非零的三角矩阵 A 是非奇异的，即:

$$Ax=0 \quad \Rightarrow \quad x=0$$

Theorem 11.1.2 — **高斯消元法**. A 的逆可以通过逐列解方程 $AX=I$ 来计算得到

$$A\begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}=\begin{bmatrix} e_1 & e_2 & \cdots & e_n \end{bmatrix}$$

Theorem 11.1.3 下三角矩阵的逆是下三角矩阵，上三角矩阵的逆是上三角矩阵。

上/下三角矩阵 $\mathrm{A}\in\mathbb{R}^{n\times n}$ 逆的复杂度

$$n^2+(n-1)^2+\cdots+1\approx\frac{1}{3}n^3 \text{ flops}$$

11.2 QR Factorization

如果矩阵 $A\in\mathbb{R}^{m\times n}$ 的列向量线性无关，则可以将其分解为

Theorem 11.2.1 — **QR Factorization.**

$$\begin{aligned} A_{n\times n} &= \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \\ &= \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1n} \\ 0 & R_{22} & \cdots & R_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{nn} \end{bmatrix} \\ &= Q_{n\times n}R_{n\times n} \end{aligned}$$

向量 $q_1,\ldots,q_n\in\mathbb{R}^m$ 是标准正交向量:

$$\|q_i\|_2=1, \quad q_i^Tq_j=0, \text{ if } \quad i\neq j$$

对角元素 R_{ii} 是非零的。若 $R_{ii}<0$, 改变 $R_{ii},\cdots,R_{in}$ 和向量 q_i 的符号。大多数定义要求 $R_{ii}>0$, 使得 Q 和 R 是唯一的。

Corollary 11.2.2 $Q\in\mathbb{R}^{m\times n}$ 具有标准正交列 $\left(Q^TQ=I\right)$.

Corollary 11.2.3 如果 A 是方阵 $(\mathrm{m}=\mathrm{n})$, 则 Q 是正交的 $\left(Q^TQ=QQ^T=I\right)$.

Corollary 11.2.4 $R\in\mathbb{R}^{n\times n}$ 的上三角矩阵.

Corollary 11.2.5 R 是非奇异的 (对角元素是非零的).

Corollary 11.2.6

$$R = Q^{-1}A \Rightarrow R = Q^{T}A$$

QR 分解可通过 Gram-Schmidit 正交化法（参见5.4）、Householder 变换等进行。

Algorithm 6: QR Decomposition Using Gram-Schmidt Algorithm

1. 设矩阵 A 的列向量依次为 $a_1, a_2, \cdots, a_n$ ，由于 A 为非奇异矩阵, 则列向量线性无关
2. 对列向量 $a_1, a_2, \ldots, a_n$ 按照 Gram-Schmidt 方法进行正交化，然后单位化
3. 单位化得到的标准正交向量 $q_1, q_2, \ldots, q_n$, 即得到标准正交矩阵 Q
4. 根据 $R = Q^{-1}A \Rightarrow R = Q^{T}A$, 得到上三角矩阵 R
5. QR 分解 $A = QR$

■ **Example 11.1** 矩阵 A 的 QR 分解过程

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 1 \\ 0 & 0 & 2 \end{bmatrix}$$

令 $a_1 = (1,1,0)^T, a_2 = (1,-1,0)^T, a_3 = (0,1,2)^T$, 由 Schmidt 方法正交单位化后, 得到 $q_1 = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0\right)^T, q_2 = \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0\right)^T, q_3 = (0,0,1)^T$ 。

所以 $a_1 = \sqrt{2}q_1, \quad a_2 = \sqrt{2}q_2, \quad a_3 = \frac{1}{\sqrt{2}}q_1 - \frac{1}{\sqrt{2}}q_2 + 2q_3$ 。

$$A = QR = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 & \frac{1}{\sqrt{2}} \\ 0 & \sqrt{2} & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 2 \end{bmatrix}$$

■

■ **Example 11.2**

$$\begin{aligned} \begin{bmatrix} -1 & -1 & 1 \\ 1 & 3 & 3 \\ -1 & -1 & 5 \\ 1 & 3 & 7 \end{bmatrix} &= \begin{bmatrix} -1/2 & 1/2 & -1/2 \\ 1/2 & 1/2 & -1/2 \\ -1/2 & 1/2 & 1/2 \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} 2 & 4 & 2 \\ 0 & 2 & 8 \\ 0 & 0 & 4 \end{bmatrix} \\ &= \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \\ 0 & 0 & R_{33} \end{bmatrix} \\ &= QR \end{aligned}$$

■

11.2.1 QR 分解的存在唯一性

可以证明，QR 分解具有唯一性[1] [2]。

[1] SciComp
[2] Kyle Kloster's Website

Theorem 11.2.7 — $m = n$ 矩阵 A 进行 QR 分解的唯一性. If $\boldsymbol{A} = \boldsymbol{Q}_1\boldsymbol{R}_1 = \boldsymbol{Q}_2\boldsymbol{R}_2$ are two QR decompositions of full rank, square A, then

$$\begin{aligned}\boldsymbol{Q}_2 &= \boldsymbol{Q}_1\boldsymbol{S}\\ \boldsymbol{R}_2 &= \boldsymbol{S}\boldsymbol{R}_1\end{aligned}$$

for some square diagonal $\boldsymbol{S}$ with entries ± 1.

If we require the diagonal entries of $\boldsymbol{R}$ to be positive, then the decomposition is unique.

Theorem 11.2.8 — $m < n$ 矩阵 A 进行 QR 分解的唯一性. If

$$\boldsymbol{A} = \boldsymbol{Q}_1 \begin{bmatrix} \boldsymbol{R}_1 & \boldsymbol{N}_1 \end{bmatrix} = \boldsymbol{Q}_2 \begin{bmatrix} \boldsymbol{R}_2 & \boldsymbol{N}_2 \end{bmatrix}$$

are two QR decompositions of a full rank, $m \times n$ matrix $\boldsymbol{A}$ with $m < n$, then

$$\boldsymbol{Q}_2 = \boldsymbol{Q}_1\boldsymbol{S}, \quad \boldsymbol{R}_2 = \boldsymbol{S}\boldsymbol{R}_1, \quad \text{and} \quad \boldsymbol{N}_2 = \boldsymbol{S}\boldsymbol{N}_1$$

for square diagonal S with entries ± 1.

If we require the diagonal entries of $\boldsymbol{R}$ to be positive, then the decomposition is unique.

Proof. Let $\boldsymbol{Q}_1 \begin{bmatrix} \boldsymbol{R}_1 & \boldsymbol{N}_1 \end{bmatrix} = \boldsymbol{Q}_2 \begin{bmatrix} \boldsymbol{R}_2 & \boldsymbol{N}_2 \end{bmatrix}$ with $\boldsymbol{Q}_i$ being $m \times m$ and orthogonal, $\boldsymbol{R}_i$ being $m \times m$ and upper triangular, and $\boldsymbol{N}_i$ being an arbitrary $m \times (n-m)$ matrix.

Then multiplying through yields $\boldsymbol{Q}_1\boldsymbol{R}_1 = \boldsymbol{Q}_2\boldsymbol{R}_2$, two QR decompositions of a full rank, $m \times m$ matrix.

Using the theorem above, we get that $\boldsymbol{Q}_2 = \boldsymbol{Q}_1\boldsymbol{S}$ and $\boldsymbol{R}_2 = \boldsymbol{S}\boldsymbol{R}_1$ for a diagonal matrix $\boldsymbol{S}$ with entries ± 1.

Looking at the right-most partition of the original product yields $\boldsymbol{Q}_1\boldsymbol{N}_1 = \boldsymbol{Q}_2\boldsymbol{N}_2$. But we've shown $\boldsymbol{Q}_2 = \boldsymbol{Q}_1\boldsymbol{S}$, so now we have $\boldsymbol{Q}_1\boldsymbol{N}_1 = \boldsymbol{Q}_1\boldsymbol{S}\boldsymbol{N}_2$.

Left-multiplying by $\boldsymbol{Q}_1^T$ and then by $\boldsymbol{S}$ then proves $\boldsymbol{N}_2 = \boldsymbol{S}\boldsymbol{N}_1$, completing the theorem. ∎

Theorem 11.2.9 — $m > n$ 矩阵 A 进行 QR 分解的唯一性. If $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{Q}_1 & \boldsymbol{U}_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q}_2 & \boldsymbol{U}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_2 \\ 0 \end{bmatrix}$ are two QR decompositions of a full rank, $m \times n$ matrix $\boldsymbol{A}$ with $m > n$, then

$$\boldsymbol{Q}_2 = \boldsymbol{Q}_1\boldsymbol{S}, \quad \boldsymbol{R}_2 = \boldsymbol{S}\boldsymbol{R}_1, \quad \text{and} \quad \boldsymbol{U}_2 = \boldsymbol{U}_1\boldsymbol{T}$$

for square diagonal $\boldsymbol{S}$ with entries ± 1, and square orthogonal $\boldsymbol{T}$.

If we require the diagonal entries of $\boldsymbol{R}$ to be positive, then $\boldsymbol{Q}$ and $\boldsymbol{R}$ are unique.

Proof. Let $\boldsymbol{A}$ be full rank and $m \times n$ with $m > n$. Suppose it has decompositions

$$\boldsymbol{A} = \tilde{\boldsymbol{Q}}_1\tilde{\boldsymbol{R}}_1 = \tilde{\boldsymbol{Q}}_2\tilde{\boldsymbol{R}}_2$$

for $m \times m$ orthogonal matrices $\tilde{\boldsymbol{Q}}_i$, $m \times n$ and upper-triangular matrices $\tilde{\boldsymbol{R}}_i$. (We know we can do this because the QR decomposition always exists).

Since $m > n$, we can write $\tilde{\boldsymbol{Q}}_i = \begin{bmatrix} \boldsymbol{Q}_i & \boldsymbol{U}_i \end{bmatrix}$ and $\tilde{\boldsymbol{R}}_i = \begin{bmatrix} \boldsymbol{R}_i \\ 0 \end{bmatrix}$ where $\boldsymbol{Q}_i$ is $m \times n$ and $\boldsymbol{U}_i$ is $m \times (m-n)$. Then

$$\boldsymbol{A} = \tilde{\boldsymbol{Q}}_i \tilde{\boldsymbol{R}}_i = \begin{bmatrix} \boldsymbol{Q}_i & \boldsymbol{U}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_i \\ 0 \end{bmatrix} = \boldsymbol{Q}_i \boldsymbol{R}_i$$

where $\boldsymbol{R}_i$ is square, upper-triangular, invertible (because $\boldsymbol{A}$ is full rank), and the columns of $\boldsymbol{Q}_i$ are orthonormal so $\boldsymbol{Q}_i$ satisfies $\boldsymbol{Q}_i^T \boldsymbol{Q}_i = \boldsymbol{I}$. Then we have

$$\boldsymbol{Q}_1 \boldsymbol{R}_1 = \boldsymbol{Q}_2 \boldsymbol{R}_2$$

and left-multiplying by $\boldsymbol{Q}_2^T$ and right-multiplying by $\boldsymbol{R}_1^{-1}$ yields

$$\boldsymbol{Q}_2^T \boldsymbol{Q}_1 = \boldsymbol{R}_2 \boldsymbol{R}_1^{-1}$$

Note that the right-hand side of Eqn (2) is upper-triangular (since $\boldsymbol{R}_i$ is). On the other hand, left-multiplying Eqn (1) by $\boldsymbol{Q}_1^T$ and right-multiplying by $\boldsymbol{R}_2^{-1}$ gives $Q_1^T Q_2 = R_1 R_2^{-1}$, and taking the transpose yields a lower-triangular expression for $\boldsymbol{Q}_2^T \boldsymbol{Q}_1$. Therefore $\boldsymbol{Q}_1^T \boldsymbol{Q}_2 = \boldsymbol{R}_1 \boldsymbol{R}_2^{-1}$ is both lower- and upper-triangular, and so it is diagonal. Call it $\boldsymbol{D}$. Then right-multiplying Eqn (1) by $\boldsymbol{R}_2^{-1}$ yields

$$\boldsymbol{Q}_2 \boldsymbol{R}_2 \boldsymbol{R}_2^{-1} = \boldsymbol{Q}_2 = \boldsymbol{Q}_1 \boldsymbol{R}_1 \boldsymbol{R}_2^{-1} = \boldsymbol{Q}_1 \boldsymbol{D}$$

and so $\boldsymbol{Q}_2 = Q_1 \boldsymbol{D}$. Multiplying this by its transpose and using orthogonality of $\boldsymbol{Q}_i$ we get $\boldsymbol{I} = \boldsymbol{Q}_2^T \boldsymbol{Q}_2 = (\boldsymbol{Q}_1 \boldsymbol{D})^T (\boldsymbol{Q}_1 \boldsymbol{D}) = \boldsymbol{D}^T \boldsymbol{Q}_1^T \boldsymbol{Q}_1 \boldsymbol{D} = \boldsymbol{D}^T \boldsymbol{D} = \boldsymbol{D}^2$. This proves $\boldsymbol{D}^2 = \boldsymbol{I}$, so $\boldsymbol{D} = \boldsymbol{S}$, a diagonal matrix with entries ± 1. So $\boldsymbol{Q}_2 = \boldsymbol{Q}_1 \boldsymbol{S}$. Left multiplying Eqn (1) by $\boldsymbol{Q}_2^T = \boldsymbol{S} \boldsymbol{Q}_1^T$ then yields

$$\boldsymbol{S} \boldsymbol{Q}_1^T \boldsymbol{Q}_1 \boldsymbol{R}_1 = \boldsymbol{S} \boldsymbol{R}_1 = \boldsymbol{Q}_2^T \boldsymbol{Q}_2 \boldsymbol{R}_2 = \boldsymbol{R}_2$$

proving that $\boldsymbol{R}_2 = \boldsymbol{S} \boldsymbol{R}_1$.

■

11.2.2 复矩阵的 QR 分解

Theorem 11.2.10 如果 $A \in \mathbb{C}^{m \times n}$ 的列向量是线性无关的，则可以将其分解为

$$A = QR$$

$Q \in \boldsymbol{C}^{m \times n}$ 具有正交列。$\left(Q^H Q = I\right)$
$R \in \boldsymbol{C}^{n \times n}$ 具有实非零对角元素的上三角矩阵。

大多数情况下，会优先选择对角线元素 R_{ii} 为正数。
如果没有特别说明，之后默认矩阵 A 都是实数的。

11.3 QR 分解的应用

可用 QR 分解求解以下问题：

- 线性方程
- 最小二乘问题
- 带约束的最小二乘问题

11.3.1 QR 分解和求解线性方程组 $Ax = b$

QR 分解的思想可以用于加快求逆矩阵速度。

Corollary 11.3.1

$$Ax = b \Rightarrow x = A^{-1}b$$
$$QRx = b \Rightarrow x = R^{-1}\left(Q^T b\right)$$

Algorithm 8: Solving linear equations via QR factorization

Input: $n \times n$ invertible matrix A

1. QR factorization. Compute the QR factorization $A = QR$
2. Compute $Q^T b$
3. Back substitution. Solve the triangular equation $Rx = Q^T b$ using back substitution

对于普通矩阵使用 QR 分解求解线性方程组，正交分解需要 $2n^3$ flops, 计算 $Q^T b$ 需要 $2n^2$ flops, 第三步回代求解 $Rx = Q^T b$ 需要 n^2 flops.(总时间复杂度是 $O(n^3)$).

对于稀疏矩阵求解线性方程组，时间复杂度接近 $\text{nnz}(A)$。内存使用和复杂度在很大程度上取决于系数矩阵的稀疏模式。内存使用量通常是 $\text{nnz}(A)+n$ 的适度倍数，$\text{nnz}(A)+n$ 是指定问题数据 A 和 b 所需的标量数量，通常远小于 n^2+n(如果 A 和 b 不是稀疏时存储它们所需的标量数)。求解稀疏线性方程的 flop 数通常也更接近 $\text{nnz}(A)$，而不是 n^3（矩阵 A 不稀疏时的阶数）。

11.3.2 QR 分解和求解伪逆 $A^\dagger$、逆 A^{-1}

Definition 11.3.1 — 线性无关列向量的矩阵 A 的伪逆.

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

Theorem 11.3.2

$$\begin{aligned} A^\dagger &= \left((QR)^T (QR)\right)^{-1} (QR)^T \\ &= \left(R^T Q^T QR\right)^{-1} R^T Q^T \\ &= \left(R^T R\right)^{-1} R^T Q^T \quad \left(Q^T Q = I\right) \\ &= R^{-1} R^{-T} R^T Q^T \quad (R\text{是非奇异的}) \\ &= R^{-1} Q^T \end{aligned}$$

Corollary 11.3.3 对于方阵非奇异矩阵 A, 其逆为

$$A^{-1} = (QR)^{-1} = R^{-1} Q^T$$

Corollary 11.3.4 对于方阵非奇异矩阵 $A = QR$

$$RA^{-1} = Q^T$$

Proof.

$$\begin{aligned} A = QR \\ \Rightarrow AA^{-1} &= QRA^{-1} \\ \Rightarrow I &= Q\underbrace{RA^{-1}}_{Q^{-1}} \end{aligned}$$

■

Algorithm 9: Computing the inverse via QR factorization

Input: $n \times n$ invertible matrix A

1 QR factorization. Compute the QR factorization $A = QR$

2 **for** $i = 1, \ldots, n$ **do**

3 Solve the triangular equation $Rb_i = \tilde{q}_i$ using back substitution.

4 **end**

对于通过 QR 分解求 A^{-1}，QR 分解需要 $2n^3$ flops，n 次回代需要 n^3 flops，时间复杂度是 $O(3n^3)$.

11.3.3 A 的列空间和 Q 的列空间相同

矩阵 $A \in \mathbb{R}^{m\times n}$ 的值域范围定义为:

$$\text{range}(A) = \{Ax \mid x \in \mathbf{R}^n\}$$

Theorem 11.3.5 假设 A 有线性无关的列向量，且其 QR 因子为 Q, R，则 Q 和 A 的值域范围相同（有相同的列空间）。

即 Q 的列向量是标准正交的，并且和 A 的列向量张成相同的空间。

Proof.

$$\begin{aligned} y \in \text{range}(A) &\Leftrightarrow y = Ax, x \in \mathbb{R}^n \\ &\Leftrightarrow y = QRx, z = Rx \\ &\Leftrightarrow y = Qz, z \in \mathbb{R}^n \\ &\Leftrightarrow y \in \text{range}(Q) \end{aligned}$$

■

11.3.4 往 A 列空间上的投影也是往 Q 列空间上的投影

结合 $A = QR$ 和 $A^\dagger = R^{-1}Q^T$, 可得:

Theorem 11.3.6

$$AA^\dagger = QRR^{-1}Q^T = QQ^T$$

Theorem 11.3.7

$$R^{\mathrm{T}}R\widehat{\boldsymbol{x}} = R^{\mathrm{T}}Q^{\mathrm{T}}\boldsymbol{b} \text{ or } R\widehat{\boldsymbol{x}} = Q^{\mathrm{T}}\boldsymbol{b} \text{ or } \quad \widehat{\boldsymbol{x}} = R^{-1}Q^{\mathrm{T}}\boldsymbol{b}$$

可以用于求解最小二乘法，而不用直接求解 $Ax = b$（当 A 不可逆时）

注意在 $AA^\dagger$ 中乘积的顺序与 $A^\dagger A = I$ 的差异。

$$\begin{aligned}&\min_y \|Qy - x\|_2^2\\ \Rightarrow &Q^T(Qy - x) = 0\\ \Rightarrow &Q^TQy = Q^Tx\\ \Rightarrow &y = Q^Tx\end{aligned}$$

QQ^Tx 是 x 在 Q 值域（列空间）上的投影，也是往 A 的列空间上的投影（见10.9.1）.

Figure 11.1: Projecting onto the column space of A is also projecting onto the column space of Q

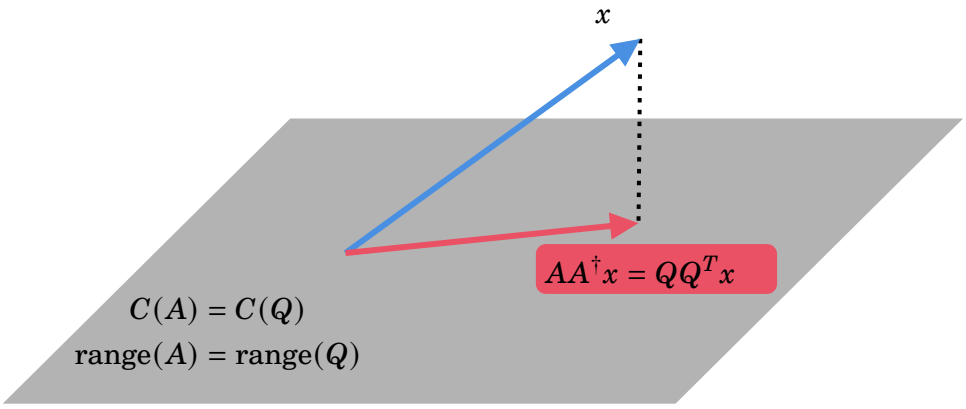
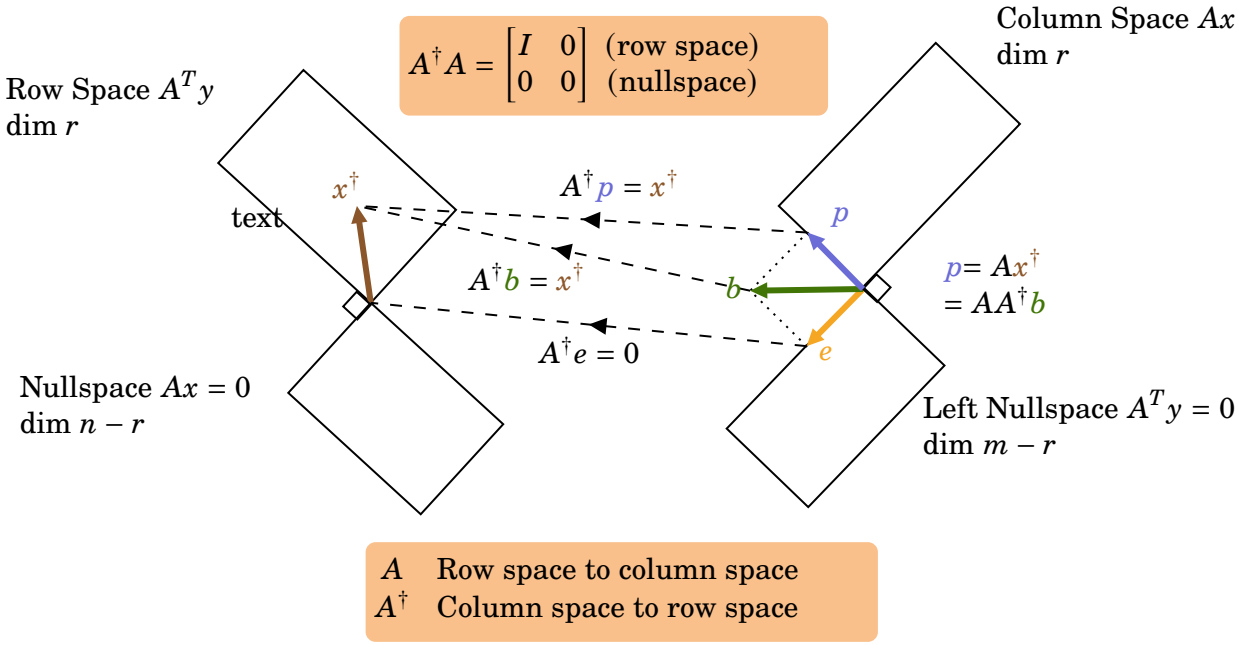


Figure 11.2: $A\boldsymbol{x}^\dagger$ in the column space goes back to $A^\dagger A\boldsymbol{x}^\dagger = \boldsymbol{x}^\dagger$ in the row space



The pseudoinverse $A^\dagger$ is the n by m matrix that makes $AA^\dagger$ and $A^\dagger A$ into projections.

Trying for $AA^{-1} = A^{-1}A = I$, $AA^{\dagger}$ = projection matrix onto the column space of A (refer to projection onto the column space of A)
$A^{+}A$ = projection matrix onto the row space of A

11.4 QR Algorithm Using Gram-Schmidt Algorithm

Gram-Schmidt QR 算法将逐列计算 Q 和 R.

k 步后我们得到了 QR 的部分分解:

$$A = \begin{bmatrix} a_1 & a_2 & \cdots & a_k \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & \cdots & q_k \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1k} \\ 0 & R_{22} & \cdots & R_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{kk} \end{bmatrix}$$

Corollary 11.4.1 QR 的部分分解列向量 $q_1, \ldots, q_k$ 是标准正交的。

Corollary 11.4.2 对角线元素 $R_{11}, R_{22}, \ldots, R_{kk}$ 是正的。

Corollary 11.4.3 列向量 $q_1, \ldots, q_k$ 和 $a_1, \ldots, a_k$ 张成的空间相同。

Theorem 11.4.4 $A = QR$ 矩阵中的 R 矩阵为

$$R_{1k} = q_1^T a_k, R_{2k} = q_2^T a_k, \ldots, R_{k-1,k} = q_{k-1}^T a_k$$

Gram-Schmidt 正交化的两条基本公式

$$\widetilde{q}_i = a_i - \left(q_1^T a_i\right) q_1 - \cdots - \left(q_{i-1}^T a_i\right) q_{i-1}$$

$$a_i = \left(q_1^T a_i\right) q_1 + \cdots + \left(q_{i-1}^T a_i\right) q_{i-1} + \underbrace{\|\tilde{q}_i\|_2 \, q_i}_{\widetilde{q}_i}$$

$$= R_{1i} q_1 + \cdots + R_{ii} q_i$$

Proof. 假设已经实现 $k-1$ 列的 QR 分解，方程 $A = QR$ 的第 k 列可以计算为:

$$a_k = R_{1k} q_1 + R_{2k} q_2 + \cdots + R_{k-1,k} q_{k-1} + R_{kk} q_k$$

无论如何选择 $R_{1k}, \ldots, R_{k-1,k}$, 向量

$$\tilde{q}_k = a_k - R_{1k} q_1 - R_{2k} q_2 - \cdots - R_{k-1,k} q_{k-1} \neq 0$$

都将是非零的.（由于 $a_1, \ldots, a_k$ 是线性无关的，即 $q_1, \ldots, q_{i-1}, q_i$ 是线性无关的，假设 Gram-Schmidt 算法过程中没有出现中途退出的状况）

因此

$$a_k \notin \operatorname{span}\{q_1, \ldots, q_{k-1}\} = \operatorname{span}\{a_1, \ldots, a_{k-1}\}$$

q_k 是 $\tilde{q}_k$ 的单位化：选择 $R_{kk} = \|\tilde{q}_k\|_2$, 以及 $q_k = \left(\frac{1}{R_{kk}}\right)\tilde{q}_k$ 。
$\tilde{q}_k$ 和 q_k 正交于 $q_1, \ldots, q_{k-1}$, 则 $R_{1k}, \ldots, R_{k-1,k}$ 为:

$$R_{1k} = q_1^T a_k, R_{2k} = q_2^T a_k, \ldots, R_{k-1,k} = q_{k-1}^T a_k$$

■

11.4.1 Gram-Schmidt Algorithm

Algorithm 10: QR Decomposition Using Gram-Schmidt Algorithm

Input: 矩阵 $A \in \mathbb{R}^{m\times n}$ ，列向量 $a_1, \ldots, a_n$ 线性无关
Output: 分解得到的 Q、R 矩阵

$R_{11} = \|a_1\|_2$
$q_1 = \frac{1}{R_{11}} a_1$
for $k = 2$ *to* n **do**
  for $l = 1$ *to* $k-1$ **do**
    $R_{l,k} = q_l^T a_k$
  end
  $\tilde{q}_k = a_k - \left(R_{1k}q_1 + R_{2k}q_2 + \cdots + R_{k-1,k}q_{k-1}\right)$
  $R_{kk} = \|\tilde{q}_k\|_2$
  $q_k = \dfrac{1}{R_{kk}}\tilde{q}_k$
end

■ **Example 11.3**

$$\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} = \begin{bmatrix} -1 & -1 & 1 \\ 1 & 3 & 3 \\ -1 & -1 & 5 \\ 1 & 3 & 7 \end{bmatrix}$$
$$= \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \\ 0 & 0 & R_{33} \end{bmatrix}$$

Q 和 R 的第一列:

$$\tilde{q}_1 = a_1 = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix}, R_{11} = \|\tilde{q}_1\| = 2, q_1 = \frac{1}{R_{11}}\tilde{q}_1 = \begin{bmatrix} -1/2 \\ 1/2 \\ -1/2 \\ 1/2 \end{bmatrix}$$

Q 和 R 的第二列：计算得到 $R_{12} = q_1^T a_2 = 4$ 。
正交化计算:

$$\tilde{q}_2 = a_2 - R_{12}q_1 = \begin{bmatrix} -1 \\ 3 \\ -1 \\ 3 \end{bmatrix} - 4\begin{bmatrix} -1/2 \\ 1/2 \\ -1/2 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

将其单位化得到：

$$R_{22} = \|\tilde{q}_2\| = 2, \quad q_2 = \frac{1}{R_{22}}\tilde{q}_2 = \begin{bmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{bmatrix}$$

Q 和 R 的第三列: 计算得到 $R_{13} = q_1^T a_3 = 2$ 以及 $R_{23} = q_2^T a_3 = 8$ 。

正交化计算: $\tilde{q}_3 = a_3 - R_{13}q_1 - R_{23}q_2 = \begin{bmatrix} 1 \\ 3 \\ 5 \\ 7 \end{bmatrix} - 2\begin{bmatrix} -1/2 \\ 1/2 \\ -1/2 \\ 1/2 \end{bmatrix} - 8\begin{bmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{bmatrix} = \begin{bmatrix} -2 \\ -2 \\ 2 \\ 2 \end{bmatrix}$

将其单位化得到：

$$R_{33} = \|\tilde{q}_3\| = 4, \quad q_3 = \frac{1}{R_{33}}\tilde{q}_3 = \begin{bmatrix} -1/2 \\ -1/2 \\ 1/2 \\ 1/2 \end{bmatrix}$$

最终结果：

$$\begin{bmatrix} -1 & -1 & 1 \\ 1 & 3 & 3 \\ -1 & -1 & 5 \\ 1 & 3 & 7 \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & q_3 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \\ 0 & 0 & R_{33} \end{bmatrix}$$

$$= \begin{bmatrix} -1/2 & 1/2 & -1/2 \\ 1/2 & 1/2 & -1/2 \\ -1/2 & 1/2 & 1/2 \\ 1/2 & 1/2 & 1/2 \end{bmatrix} \begin{bmatrix} 2 & 4 & 2 \\ 0 & 2 & 8 \\ 0 & 0 & 4 \end{bmatrix}$$

■

11.4.2 基于 Gram-Schmidt 方法进行 QR 分解的时间复杂度

Gram-Schmidt 方法第 k 次循环的复杂度：

- a_k 有 k-1 个 $q_i^T a_k$ 内积操作: (k − 1)(2 m − 1) flops
- 计算 $\tilde{q}_k$: 2(k − 1)m flops.

$$\widetilde{q_i} = a_i - \left(q_1^T a_i\right) q_1 - \cdots - \left(q_{i-1}^T a_i\right) q_{i-1}$$

- 计算 R_{kk} 和 q_k : 3 m flops。$R_{kk} = \|\tilde{q}_k\|_2, q_k = \tilde{q}_k / R_{kk}$

第 k 次循环的总和：$(4m-1)(k-1) + 3m$ flops

$A \in \mathbb{R}^{m\times n}$ 分解的复杂度:

$$\sum_{k=1}^{n}((4m-1)(k-1)+3m) = (4m-1)\frac{n(n-1)}{2} + 3mn$$

$$\approx 2mn^2 \text{ flops}$$

对于稀疏矩阵（空间存储小于 $m \times n$ ）的 QR 分解，时间复杂度可以低于 $O(2mn^2)$.

11.5 The Numerical Instability of QR Decomposition based on Gram-Schmidt Algorithm

Gram-Schmidt 算法复杂度为 $2mn^2$ flops. 在实际情况中**不推荐使用** (容易被舍入误差影响)。

修正 Gram-Schmidt 算法复杂度为 $2mn^2$ flops，有更好的数值计算性能。修正之处在于求解投影时每次只减去一个投影。

Householder 算法复杂度为 $2mn^2 - (2/3)n^3$ flops 。将 Q 表示为初等正交矩阵的乘积。是最广泛使用的算法 (在 MATLAB 和 Julia 中的 qr 函数使用该算法)。

本书中认为 QR 分解的复杂度为 $Q(2mn^2)$.

■ **Example 11.4**

```
[m,n]=size(A);
Q = zeros(m,n);
R = zeros(n,n);
for k = 1:n
    R(1:k-1,k)=Q(:,1:k-1)'*A(:,k);
    v= A(:,k)-Q(:,1:k-1)*R(1:k-1,k);
    R(k,k) = norm(v);
    Q(:,k)=v/R(k,k);
end

```

Listing 11.1: Gram Schmidt

Algorithm 11: Gram-Schmidt 的 MATLAB 算法

Input: 矩阵 A

Output: QR 分解得到的矩阵 Q、R

for $k = 1$ *to* n **do**

  $R_{\mathrm{jk}} = q_j^T a_k, j < k, j = 1, \ldots, k-1$

  $v = \tilde{q}_k = a_k - Q_{:,1:k-1} R_{1:k-1,k}$

  $R_{\mathrm{kk}} = \|\tilde{q}_k\|_2$

  $q_k = \left(\frac{1}{R_{kk}}\right) \tilde{q}_k$

end

构造一个矩阵 $A = USV$, 其中 U 和 V 是正交矩阵，S 是对角矩阵

$$S_{ii} = 10^{-10(i-1)/(n-1)}, \quad i = 1, \cdots, n$$

把 Gram-Schmidt 算法应用到一个大小为 m = n = 50 的方形矩阵 A 上。

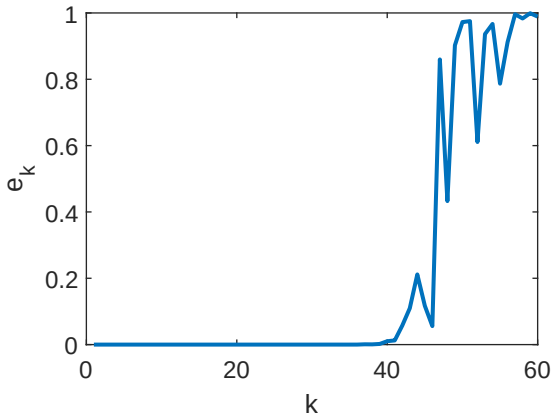
图中显示了 q_k 与前面列之间的正交性的偏差:

$$e_k = \max_{1 \le i < k} \left| q_i^T q_k \right|, \quad k = 2, \ldots, n$$

失去正交性是由于**浮点数存储的舍入误差**。

```
for j = 1:n
    v=A(:,j);
    for i=1:j-1
        R(i,j)=Q(:,i)'*v;
        v=v-R(i,j)*Q(:,i);
    end
    R(j,j)= norm(v);
```

Figure 11.3: $\max_{1\le i<k}\left|q_i^T q_k\right|$ (Gram-Schmidt Algorithm)




```
    Q(:,j)=v/R(j,j);
end

```

Listing 11.2: modified Gram-Schmidt

Algorithm 12: Modified Gram-Schmidt Algorithm

Input: 矩阵 A
Output: QR 分解得到的矩阵 Q、R

for $j = 1$ *to* n **do**
  $v = \tilde{q}_k = a_k$
  for $i = 1$ *to* $j-1$ **do**
    $R_{ij} = q_i^T v$
    $v = v - R_{ij} q_i$
  end
  $R_{\mathrm{kk}} = \|\tilde{q}_k\|_2$
  $q_k = \left(\frac{1}{R_{kk}}\right)\tilde{q}_k$
end

修正 Gram-Schmidt 算法的误差更小。 ∎

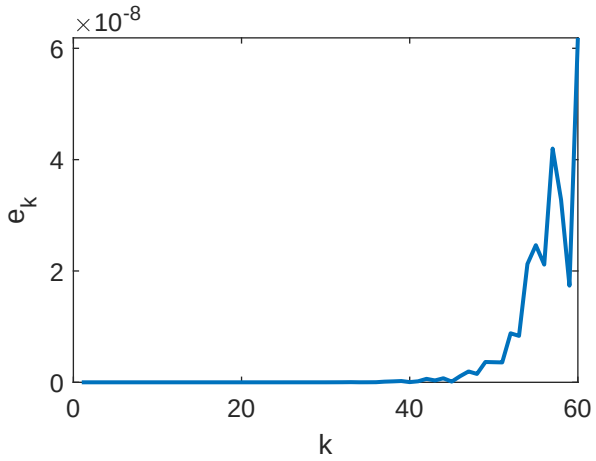
11.6 QR Decomposition Using Householder Transformation

Householder 算法是 QR 分解常用的算法 (MATLAB 和 Julia 中的 qr 函数)。与 Gram-Schmidt 相比，对舍入误差更有鲁棒性。

Householder 算法计算一个“完整的”QR 分解：

$$A_{m\times n} = \begin{bmatrix} Q_{m\times n} & \tilde{Q}_{m\times(m-n)} \end{bmatrix} \begin{bmatrix} R_{n\times n} \\ 0_{(m-n)\times n} \end{bmatrix}, \quad \begin{bmatrix} Q & \tilde{Q} \end{bmatrix} \text{是列正交的矩阵}$$

Figure 11.4: 在同一组数据中使用修正 Gram-Schmidt 算法求得的误差，$e_k = \max_{1 \le i < k} \left| q_i^T q_k \right|, k = 2, \ldots, n$



Proof.

$$\begin{aligned} A &= \begin{bmatrix} Q & \tilde{Q} \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} \\ &= QR + \tilde{Q}0 \\ &= QR \end{aligned}$$

■

where R is an $n \times n$ upper triangular matrix, 0 is an $(m-n) \times n$ zero matrix, Q is $m \times n$, $\tilde{Q}$ is $m \times (m-n)$, and Q and $\tilde{Q}$ both have orthogonal columns.

! $A \in R^{m \times n}, m \ge n$，当 $m < n$ 时列线性相关，无法进行 QR 分解。

完整的 Q 因子被构造成正交矩阵的乘积：

$$\begin{bmatrix} Q & \tilde{Q} \end{bmatrix} = H_1 H_2 \cdots H_n$$

每个 $H_i \in \mathbb{R}^{m \times m}$ 是对称正交的 Householder 矩阵。

11.6.1 Householder Matrix

Theorem 11.6.1 $H = I - 2vv^T$, 其中 $\|v\|_2 = 1$, Hx 是 x 关于超平面 $\left\{u \mid v^T u = 0\right\}$ 反对称.

H 是对称的

$$H^T = H$$

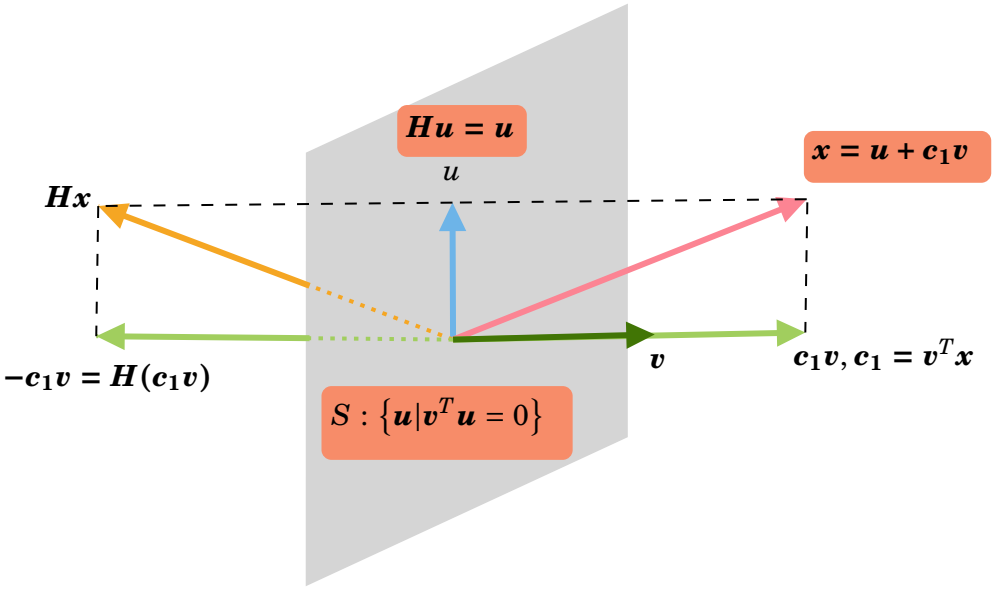
H 是正交的

$$H^T H = I$$

Proof.

$$\begin{aligned} Hv &= (I - 2vv^T)(c_1 v) \\ &= c_1 v - 2vv^T c_1 v \\ &= c_1 v - 2c_1 vv^T v \\ &= -c_1 v \end{aligned}$$

Figure 11.5: Reflection of x



$$
\begin{aligned}
Hu &= (I - 2vv^T)u \\
&= u - 2vv^T u \\
&= u \quad (v^T u = 0)
\end{aligned}
$$

$$
\begin{aligned}
Hx &= H(u + c_1 v) \\
&= u - c_1 v \\
&= x - 2(v^T x)v
\end{aligned}
$$

■

Theorem 11.6.2 矩阵向量积 Hx 能化简为

$$Hx = x - 2\left(v^T x\right)v$$

Hx 的算法复杂度

如果 v 和 x 的长度是 p, 复杂度是 $4p$ flops 。

11.6.2 构造反射算子

给定非零 p 维向量 $y = (y_1, y_2, \ldots, y_p)$, 定义

Definition 11.6.1

$$w = \begin{bmatrix} y_1 + \operatorname{sign}(y_1)\|y\|_2 \\ y_2 \\ \vdots \\ y_p \end{bmatrix}$$

$$v = \frac{1}{\|w\|_2} w$$

$\operatorname{sign}(x)$ 是符号函数，$\operatorname{sign}(0) = 0$。

Theorem 11.6.3 向量 w 满足

$$\|w\|_2^2 = 2y^T w$$

$$\begin{aligned}\|w\|_2^2 &= w^T w \\ &= 2\left(\|y\|_2^2 + |y_1|\,\|y\|_2\right) \\ &= 2y^T\left(y + \operatorname{sign}(y_1)\,\|y\|_2 e_1\right) \\ &= 2y^T w\end{aligned}$$

Definition 11.6.2 — Constructed Householder Matrix. 将 y 变换为单位基向量 $e = [1, 0, 0, \cdots, 0]^T$ 乘以一个常数的 Householder 矩阵为

$$H = I - 2\frac{ww^T}{\|w\|_2^2}$$

Proof.

$$H = I - 2vv^T = I - 2\frac{ww^T}{\|w\|_2^2} \quad \left(v = \frac{1}{\|w\|_2}w\right)$$

■

Theorem 11.6.4 Householder 矩阵 $H = I - 2vv^T = I - 2\frac{ww^T}{\|w\|_2^2}$ 将 y 映射为

$$Hy = \begin{bmatrix} -\operatorname{sign}(y_1)\,\|y\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = r_1$$

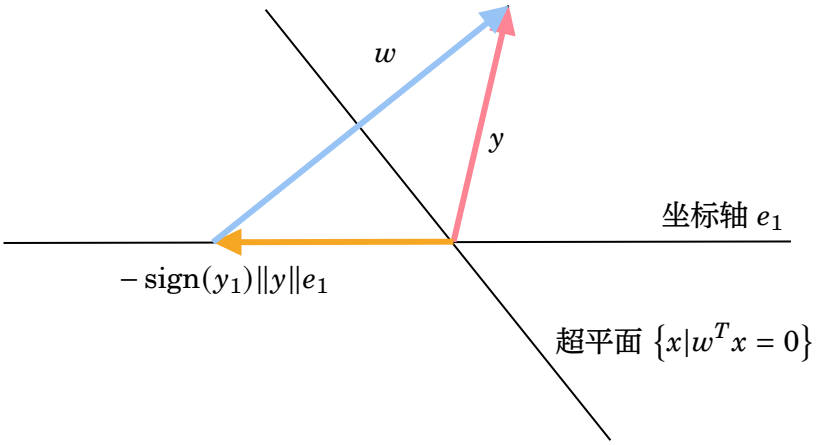
Proof.

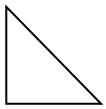
$$\begin{aligned}Hy &= y - \frac{2\left(w^T y\right)}{\|w\|_2^2}w \\ &= y - w \\ &= -\operatorname{sign}(y_1)\,\|y\|_2 e_1 \\ &= \begin{bmatrix} -\operatorname{sign}(y_1)\,\|y\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}\end{aligned}$$

■

即 Householder 变换的结果 Hy 只保留第一个元素的值，其余元素的值变为 0. 又因为 H 是正交、对称的，它与 QR 分解求解 R 有一定联系。

Figure 11.6: 构造的 Householder 矩阵的几何意义





构造的 Householder 矩阵几何意义
关于超平面 $\{x \mid w^Tx = 0\}$, 其法向量 w, v:

$$w = y + \text{sign}(y_1)\|y\|_2 e_1, v = \frac{w}{\|w\|_2}$$

反射算子 H 将 y 映射到向量 $-\text{sign}(y_1)\|y\|_2 e_1$ 。

11.6.3 Householder 三角化

计算反射算子 $H_1, \ldots, H_n$ 将 A 简化为上三角矩阵形式:

$$H_n H_{n-1} \cdots H_1 A = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

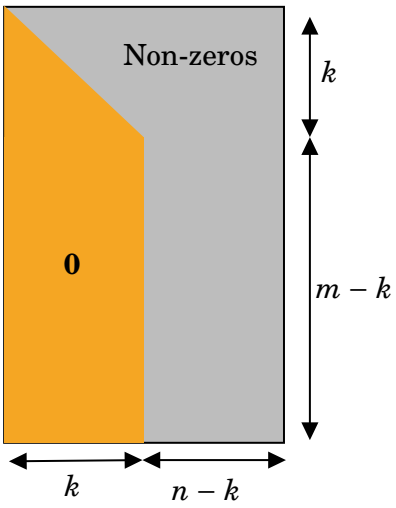
第 k 个步骤之后，矩阵 $H_k H_{k-1} \ldots H_1 A$ 具有以下结构:
对于 $i > j$ 和 $j \leq k$, 第 i, j 个位置的元素为零。
其过程如下:

$$A = \begin{bmatrix} X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X \\ X & X & X & X & X & X & X & X \end{bmatrix}$$

在第一次处理之后，A_1 第一列只剩下第一个元素不为 0.

Figure 11.7: The structure of $H_k H_{k-1} \dots H_1 A$

$$H_n H_{n-1} \cdots H_1 A =$$



$$H_1 A = A_1 = \begin{bmatrix} X & X & X & X & X & X & X & X \\ 0 & X & X & X & X & X & X & X \\ 0 & X & X & X & X & X & X & X \\ 0 & X & X & X & X & X & X & X \\ 0 & X & X & X & X & X & X & X \\ 0 & X & X & X & X & X & X & X \\ 0 & X & X & X & X & X & X & X \\ 0 & X & X & X & X & X & X & X \end{bmatrix}$$

第二次处理对于 $H_1 A_{2:m,1:n}$（$A_{1_{2:m,1:n}}$）进行处理。

$$H_2 A_1 = A_2 = \begin{bmatrix} X & X & X & X & X & X & X & X \\ 0 & X & X & X & X & X & X & X \\ 0 & 0 & X & X & X & X & X & X \\ 0 & 0 & X & X & X & X & X & X \\ 0 & 0 & X & X & X & X & X & X \\ 0 & 0 & X & X & X & X & X & X \\ 0 & 0 & X & X & X & X & X & X \\ 0 & 0 & X & X & X & X & X & X \end{bmatrix}$$

11.6.4 Household-QR Algorithm

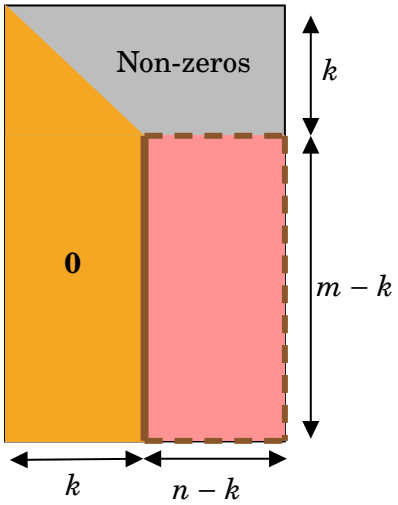
Theorem 11.6.5 在算法步骤 2 中，将 $A_{k:m,k:n}$ 与反射算子 $I - 2v_k v_k^T$ 相乘

$$\left(I - 2v_k v_k^T\right) A_{k:m,k:n} = A_{k:m,k:n} - 2v_k \left(v_k^T A_{k:m,k:n}\right)$$

等价于用 $H_k \in \mathbb{R}^{m \times m}$ 乘以 $A \in \mathbb{R}^{m \times n}$

$$H_k = \begin{bmatrix} I & 0 \\ 0 & I - 2v_k v_k^T \end{bmatrix} = I - 2 \begin{bmatrix} 0 \\ v_k \end{bmatrix} \begin{bmatrix} 0 \\ v_k \end{bmatrix}^T$$

Figure 11.8: 迭代计算过程中 A 的结构



Algorithm 13: QR Decomposition Using Householder Transformation

Input: Matrix A

Output: $v_1, \cdots, v_n, v_k \in \mathbb{R}^{m-k+1}$ (Q), $A = \begin{bmatrix} R \\ 0 \end{bmatrix}$ (R)

1 **for** k *in* $1 : n$ **do**

2 令 $y = A_{k:m,k} \in \mathbb{R}^{m-k+1}$, 计算向量 v_k

$$w = y + \text{sign}\,(y_1)\,\|y\| e_1$$

$$v_k = \frac{1}{\|w\|} w$$

3 将 $A_{k:m,k:n} \in \mathbb{R}^{(m-k+1)\times(n-k+1)}$ 与反射矩阵 $I - 2v_k v_k^T$ 相乘

$$A_{k:m,k:n} := A_{k:m,k:n} - 2v_k \left(v_k^T A_{k:m,k:n}\right)$$

4 **end**

算法的最终结果将下列矩阵来代替 $A \in \mathbb{R}^{m\times n}$

$$\begin{bmatrix} R \\ 0 \end{bmatrix}$$

返回向量 $v_1, \ldots, v_n$, 其中 v_k 的长度为 $m-k+1$ 。

11.6.5 An Example for Householder Algorithm

Problem 11.3

$$A = \begin{bmatrix} -1 & -1 & 1 \\ 1 & 3 & 3 \\ -1 & -1 & 5 \\ 1 & 3 & 7 \end{bmatrix} = H_1 H_2 H_3 \begin{bmatrix} R \\ 0 \end{bmatrix}$$

计算反射算子 H_1, H_2, H_3 来将矩阵 A 三角化

$$H_3 H_2 H_1 A = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ 0 & R_{22} & R_{23} \\ 0 & 0 & R_{33} \\ 0 & 0 & 0 \end{bmatrix}$$

R 的第一列：计算将 A 的第一列映射到 e_1 乘积的反射算子

$$y = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \end{bmatrix}, \quad w = y - \|y\|_2 e_1 = \begin{bmatrix} -3 \\ 1 \\ -1 \\ 1 \end{bmatrix}, \quad v_1 = \frac{1}{\|w\|_2} w = \frac{1}{2\sqrt{3}} \begin{bmatrix} -3 \\ 1 \\ -1 \\ 1 \end{bmatrix}$$

用 $I - 2v_1 v_1^T$ 和 A 的乘积代替 A:

$$A := \left(I - 2v_1 v_1^T\right) A = \begin{bmatrix} 2 & 4 & 2 \\ 0 & 4/3 & 8/3 \\ 0 & 2/3 & 16/3 \\ 0 & 4/3 & 20/3 \end{bmatrix}$$

R 的第二列：计算将 $A_{2:4,2}$ 映射到 e_1 乘积的反射算子

$$y = \begin{bmatrix} 4/3 \\ 2/3 \\ 4/3 \end{bmatrix}, \quad w = y + \|y\|_2 e_1 = \begin{bmatrix} 10/3 \\ 2/3 \\ 4/3 \end{bmatrix}, \quad v_2 = \frac{1}{\|w\|_2} w = \frac{1}{\sqrt{30}} \begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix}$$

用 $I - 2v_2 v_2^T$ 和 $A_{2:4,2:3}$ 的乘积代替 $A_{2:4,2:3}$:

$$A := \begin{bmatrix} 1 & 0 \\ 0 & I - 2v_2 v_2^T \end{bmatrix} A = \begin{bmatrix} 2 & 4 & 2 \\ 0 & -2 & -8 \\ 0 & 0 & 16/5 \\ 0 & 0 & 12/5 \end{bmatrix}$$

R 的第三列：计算将 $A_{3:4,3}$ 映射到 e_1 乘积的反射算子

$$y = \begin{bmatrix} 16/5 \\ 12/5 \end{bmatrix}, \quad w = y + \|y\|_2 e_1 = \begin{bmatrix} 36/5 \\ 12/5 \end{bmatrix}, \quad v_3 = \frac{1}{\|w\|_2} w = \frac{1}{\sqrt{10}} \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

用 $I-2v_3v_3^T$ 和 $A_{3:4,3}$ 的乘积代替 $A_{3:4,3}$：

$$A := \begin{bmatrix} I & 0 \\ 0 & I-2v_3v_3^T \end{bmatrix} A = \begin{bmatrix} 2 & 4 & 2 \\ 0 & -2 & -8 \\ 0 & 0 & -4 \\ 0 & 0 & 0 \end{bmatrix}$$

总的求解式为

$$\begin{aligned} H_3H_2H_1A &= \begin{bmatrix} I_2 & 0 \\ 0 & I_2-2v_3v_3^T \end{bmatrix} \begin{bmatrix} I_1 & 0 \\ 0 & I_3-2v_2v_2^T \end{bmatrix} \left(I_4-2v_1v_1^T\right) A \\ &= \begin{bmatrix} I_2 & 0 \\ 0 & I_2-2v_3v_3^T \end{bmatrix} \begin{bmatrix} I_1 & 0 \\ 0 & I_3-2v_2v_2^T \end{bmatrix} \begin{bmatrix} 2 & 4 & 2 \\ 0 & 4/3 & 8/3 \\ 0 & 2/3 & 16/3 \\ 0 & 4/3 & 20/3 \end{bmatrix} \\ &= \begin{bmatrix} I_2 & 0 \\ 0 & I_2-2v_3v_3^T \end{bmatrix} \begin{bmatrix} 2 & 4 & 2 \\ 0 & -2 & -8 \\ 0 & 0 & 16/5 \\ 0 & 0 & 12/5 \end{bmatrix} \\ &= \begin{bmatrix} 2 & 4 & 2 \\ 0 & -2 & -8 \\ 0 & 0 & -4 \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

■

11.6.6 Complexity of Householder Algorithm

$$H_k = \begin{bmatrix} I & 0 \\ 0 & I-2v_kv_k^T \end{bmatrix} = I - 2\begin{bmatrix} 0 \\ v_k \end{bmatrix}\begin{bmatrix} 0 \\ v_k \end{bmatrix}^T$$

Householder 方法第 k 次循环的复杂度:
- $v_k^TA_{k:m,k:n}$ 的乘积：$(2(m-k+1)-1)(n-k+1)$ flops
- v_k 的外积：$(m-k+1)(n-k+1)$ flops
- $A_{k:m,k:n}$ 的减法：$(m-k+1)(n-k+1)$ flops

第 k 次循环的总和：$4(m-k+1)(n-k+1)$ flops
计算 R 和 $v_1,\dots,v_n$ 的总复杂度

$$\begin{aligned} \sum_{k=1}^{n} 4(m-k+1)(n-k+\mathbf{2}) &\approx \int_0^n 4(m-t)(n-t+1)dt \\ &\approx 2mn^2-\frac{2}{3}n^3 \text{ flops} \end{aligned}$$

! $\sum_{k=1}^{n} 4(m-k+1)(n-k+\mathbf{2})$ 是因为

11.7 Householder 变换进行 QR 分解的 Q 因子

Householder 算法返回向量 $v_1,\dots,v_n$, 其定义为:

Definition 11.7.1 — $v_1, \ldots, v_n$ 的完整表示.

$$\begin{bmatrix} Q & \tilde{Q} \end{bmatrix} = H_1 H_2 \cdots H_n$$

通常不需计算矩阵 $\begin{bmatrix} Q & \tilde{Q} \end{bmatrix}$。向量 $v_1, \ldots, v_n$ 是 $\begin{bmatrix} Q & \tilde{Q} \end{bmatrix}$ 简单表示 (economical representation)。

Theorem 11.7.1 $\begin{bmatrix} Q & \tilde{Q} \end{bmatrix}$ 或其转置的乘积可以计算为:

$$\begin{bmatrix} Q & \tilde{Q} \end{bmatrix} x = H_1 H_2 \cdots H_n x$$
$$\begin{bmatrix} Q & \tilde{Q} \end{bmatrix}^T y = H_n H_{n-1} \cdots H_1 y$$

11.7.1 Multiplication with Q factor

Definition 11.7.2 — 矩阵-向量积 $H_k x$. 矩阵-向量积 $H_k x$ 定义为:

$$H_k x = \begin{bmatrix} I_{k-1} & 0 \\ 0 & I - 2 v_k v_k^T \end{bmatrix} \begin{bmatrix} x_{1:k-1} \\ x_{k:m} \end{bmatrix} = \begin{bmatrix} x_{1:k-1} \\ x_{k:m} - 2 \left(v_k^T x_{k:m} \right) v_k \end{bmatrix}$$

11.7.2 矩阵-向量积 $H_k x$ 算法复杂度

$H_k x$ 乘积的复杂度为: $4(\mathrm{m} - \mathrm{k} + 1)$ flops。

$H_1 H_2, \ldots H_n$ 或其转置的乘积的复杂度为:

$$\sum_{k=1}^{n} 4(m - k + 1) \approx 4mn - 2n^2 \text{ flops}$$

其复杂度约等于 $m \times n$ 矩阵的矩阵-向量乘积 ($2mn$ flops)。

11.8 Fast Orthogonalization (Givens and Householder)

This section is quoted from [**Strang1993IntroductionTL**].

There are three ways to reach the important factorization $A = QR$.

Gram-Schmidt works to find the orthonormal vectors in Q. Then R is upper triangular because of the order of Gram-Schmidt steps.

Now we look at better methods (Householder and Givens), which use a product of specially simple Q 's that we know are orthogonal.

Elimination gives $A = LU$, orthogonalization gives $A = QR$. We don't want a triangular L, we want an orthogonal Q.L is a product of E' s from elimination, with 1's on the diagonal and the multiplier ℓ_{ij} below. Q **will be a product of orthogonal matrices.**

There are two simple orthogonal matrices to take the place of the E's. The *reflection matrices* $I - 2uu^{\mathrm{T}}$ are named after *Householder*. The *plane rotation matrices* are named after *Givens*. The simple matrix that rotates the xy plane by θ is Q_{21}:

Definition 11.8.1 — Givens Rotation in the 1-2 plane.

$$Q_{21} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Use Q_{21} the way you used E_{21}, to produce a zero in the $(2,1)$ position. That determines the angle θ. Bill Hager gives this example in Applied Numerical Linear Algebra:

■ **Example 11.5 — 使用 Givens 进行 QR 分解.**

$$Q_{21}A = \begin{bmatrix} .6 & .8 & 0 \\ -.8 & .6 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 90 & -153 & 114 \\ 120 & -79 & -223 \\ 200 & -40 & 395 \end{bmatrix} = \begin{bmatrix} 150 & -155 & -110 \\ \mathbf{0} & 75 & -225 \\ 200 & -40 & 395 \end{bmatrix}$$

The zero came from $-8(90)+.6(120)$. No need to find θ, what we needed was $\cos\theta$

$$\cos\theta = \frac{90}{\sqrt{90^2+120^2}} \quad \text{and} \quad \sin\theta = \frac{-120}{\sqrt{90^2+120^2}}$$

Now we attack the $(3,1)$ entry. The rotation will be in rows and columns 3 and 1 . The numbers $\cos\theta$ and $\sin\theta$ are determined from 150 and 200 , instead of 90 and 120 .

$$Q_{31}Q_{21}A = \begin{bmatrix} .6 & 0 & .8 \\ 0 & 1 & 0 \\ -.8 & 0 & .6 \end{bmatrix} \begin{bmatrix} 150 & \cdot & \cdot \\ 0 & \cdot & \cdot \\ 200 & \cdot & \cdot \end{bmatrix} = \begin{bmatrix} 250 & -125 & 250 \\ \mathbf{0} & 75 & -225 \\ \mathbf{0} & 100 & 325 \end{bmatrix}$$

One more step to R. The $(3,2)$ entry has to go. The numbers $\cos\theta$ and $\sin\theta$ now come from 75 and 100 . The rotation is now in rows and columns 2 and 3:

$$Q_{32}Q_{31}Q_{21}A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & .6 & .8 \\ 0 & -.8 & .6 \end{bmatrix} \begin{bmatrix} 250 & -125 & \cdot \\ 0 & 75 & \cdot \\ 0 & 100 & \cdot \end{bmatrix} = \begin{bmatrix} 250 & -125 & 250 \\ \mathbf{0} & 125 & 125 \\ \mathbf{0} & \mathbf{0} & 375 \end{bmatrix}$$

We have reached the upper triangular R. What is Q? Move the plane rotations Q_{ij} to the other side to find $A = QR$-just as you moved the elimination matrices E_{ij} to the other side to find $A = LU$:

Theorem 11.8.1

$$Q_{32}Q_{31}Q_{21}A = R \quad \text{means} \quad A = \left(Q_{21}^{-1}Q_{31}^{-1}Q_{32}^{-1}\right)R = QR$$

The inverse of each Q_{ij} is Q_{ij}^{T} (rotation through $-\theta$). The inverse of E_{ij} was not an orthogonal matrix! LU **and** QR **are similar but** L **and** Q **are not the same.** ■

Householder reflections are faster than rotations because each one clears out a whole column below the diagonal. Watch how the first column a_1 of A becomes column r_1 of R :

■ **Example 11.6 — Reflection by H_1.**

$$H_1 = I - 2u_1u_1^T$$

$$H_1\boldsymbol{a}_1 = \begin{bmatrix} \|\boldsymbol{a}_1\| \\ 0 \\ \vdots \\ 0 \end{bmatrix} \text{ or } \begin{bmatrix} -\|\boldsymbol{a}_1\| \\ 0 \\ \vdots \\ 0 \end{bmatrix} = r_1$$

The length was not changed, and u_1 is in the direction of $a_1 - r_1$. We have $n-1$ entries in the unit vector $\boldsymbol{u}_1$ to get $n-1$ zeros in $\boldsymbol{r}_1$. (Rotations had one angle θ to get one zero.) When we reach column k, we have $n-k$ available choices in the unit vector u_k. This leads to $n-k$ zeros in $\boldsymbol{r}_k$. We just store the u's and $\boldsymbol{r}$'s to know the final Q and R :

Theorem 11.8.2 — H 的逆是它本身.

$$(H_{n-1}\dots H_1)\,A = R \quad \text{means} \quad A = (H_1\dots H_{n-1})\,R = QR$$

■

11.9 Recap：QR Decomposition

11.9.1 分治策略

求解线性方程组 $Ax = b$, 矩阵 $A \in \mathbb{R}^{n\times n}$ 分解成“结构简单”的矩阵相乘:

$$A = A_1A_2\cdots A_k$$

■ **Example 11.7** — 求解 k 个线性方程组 $A_1A_2\cdots A_kx = b$.

$$\begin{aligned} A_1\underbrace{(A_2\cdots A_kx)}_{z_1} &= b \\ A_2\underbrace{(A_3\cdots A_kx)}_{z_2} &= z_1 \\ \cdots & \\ A_{k-1}\underbrace{(A_kx)}_{z_{k-1}} &= z_{k-2} \\ \boldsymbol{A_kx} &= z_{k-1} \end{aligned}$$

■

■ **Example 11.8** — QR 分解 $Ax = b$.

$$\begin{aligned} Qy &= b \\ Rx &= y \end{aligned}$$

■

通常**分解复杂度远大于求解复杂度**。

11.9.2 非奇异矩阵的 QR 分解

Theorem 11.9.1 任意非奇异矩阵 $A \in \mathbb{R}^{n\times n}$, 都可以进行 QR 分解。

Corollary 11.9.2 $Q \in \mathbb{R}^{n\times n}$ 是一个正交矩阵

Corollary 11.9.3 $R \in \mathbb{R}^{n\times n}$ 是一个上三角矩阵并且对角元素都为正数

11.9.3 使用 QR 分解求 A^{-1} 可以转换成 $R^{-1}Q^T$

Theorem 11.9.4 $A^{-1} = (QR)^{-1} = R^{-1}Q^{-1} = R^{-1}Q^T$

11.9.4 QR 分解求解 A^{-1}

计算非奇异矩阵 $\mathrm{A} \in \mathbb{R}^{n\times n}$ 的逆 A^{-1}, $X = [x_1, x_2, \cdots, x_n], x_i \in \mathbb{R}^n, i = 1, \cdots, n$, $I = [e_1, e_2, \cdots, e_n], e_i \in \mathbb{R}^n, i = 1, \cdots, n$：

Theorem 11.9.5 — QR 分解求解 A^{-1}.

$$Rx_1 = Q^T e_1, Rx_2 = Q^T e_2, \cdots, Rx_n = Q^T e_n$$

$$\begin{aligned} & AX = I \\ \Rightarrow & QRX = I \\ \Rightarrow & RX = Q^T I \\ \Rightarrow & Rx_1 = Q^T e_1, Rx_2 = Q^T e_2, \cdots, Rx_n = Q^T e_n \end{aligned}$$

The Complexity of $QRX = I$

复杂度: $2n^3 + n^3 \approx 3n^3$ flops

- QR 分解复杂度: $2n^3$
- 回代法：一次回代 n^2, 则 n 次回代 n^3

11.9.5 QR 分解求解线性方程组

使用 QR 分解求解线性方程组 $Ax = b$, 矩阵 $A \in \mathbb{R}^{n\times n}$ 为非奇异矩阵

Algorithm 14: QR 分解求解线性方程组

1. 首先对 A 进行 QR 分解，得到 $A = QR$
2. 计算 $y = Q^T b$
3. 通过回代法求解 $Rx = y$

The Complexity of Solving Linear Equation Systems Using QR Decomposition

复杂度: $2n^3 + 3n^2 \approx 2n^3$ flops

- QR 分解复杂度: $2n^3$
- 矩阵向量乘法: $2n^2$
- 回代法：n^2

12. LU 分解

12.1 Solving Linear Equation Systems

12.1.1 Linear Equation Systems

- **Example 12.1**

$$A\boldsymbol{x}=\boldsymbol{b} \Leftrightarrow \begin{array}{rrrrrrr} & x+ & 2 & y+ & 3 & z= & 6 \\ 2 & x+ & 5 & y+ & 2 & z= & 4 \\ 6 & x- & 3 & y+ & & z= & 2 \end{array}$$

■

见 Row Picture，Column Picture 的概念。

12.1.2 Elimination

- **Example 12.2 — Elimination.** Before

$$\begin{array}{rrrrrr} x- & 2 & y & = & 1 \\ 3\ x+ & 2 & y & = & 11 \end{array}$$

After

$$\begin{array}{rrrrr} x- & 2 & y & = & 1 \\ & 8 & y & = & 8 \end{array}$$

■

Definition 12.1.1 — Pivot. The first nonzero in the row that does elimination. **Zero is not allowed as a pivot.**

Definition 12.1.2 — Multiplier. (Entry to eliminate) divided by (pivot)

消元法通过 elimination matrices E 进行消元操作，使得主对角线以下的元素为 0. Multiply the j^{th} equation by ℓ_{ij} and subtract from the i^{th} equation. (This eliminates x_j from equation i.) We need a lot of these simple matrices E_{ij}, one for every nonzero to be eliminated below the main diagonal.

Definition 12.1.3 — Elementary matrix, Elimination matrix. The elementary matrix or elimination matrix E_{ij} has the extra nonzero entry $-\ell$ in the i, j position. Then E_{ij} subtracts a multiple ℓ of row j from row i.

Theorem 12.1.1 消元法的本质是

$$Ax = b \Rightarrow EAx = Eb$$

Definition 12.1.4 — Permutation matrices P. P_{ij} is the identity matrix with rows i and j reversed. When this "permutation matrix" P_{ij} multiplies a matrix, it exchanges rows i and j.

Definition 12.1.5 — Augmented matrix.

$$\begin{bmatrix} A & b \end{bmatrix}$$

Computing A^{-1} by Gauss-Jordan Elimination

$$\text{Multiply } \begin{bmatrix} A & I \end{bmatrix} \text{ by } A^{-1} \text{ to get } \begin{bmatrix} I & A^{-1} \end{bmatrix}$$

■ **Example 12.3**

$$\begin{bmatrix} K & e_1 & e_2 & e_3 \end{bmatrix} = \begin{bmatrix} \mathbf{2} & \mathbf{-1} & \mathbf{0} & 1 & 0 & 0 \\ \mathbf{-1} & \mathbf{2} & \mathbf{-1} & 0 & 1 & 0 \\ \mathbf{0} & \mathbf{-1} & \mathbf{2} & 0 & 0 & 1 \end{bmatrix} \quad \text{Start Gauss-Jordan on } K$$

$$\rightarrow \begin{bmatrix} 2 & -1 & 0 & 1 & 0 & 0 \\ \mathbf{0} & \frac{3}{2} & \mathbf{-1} & \frac{1}{2} & \mathbf{1} & \mathbf{0} \\ 0 & -1 & 2 & 0 & 0 & 1 \end{bmatrix} \quad \left(\frac{\mathbf{1}}{\mathbf{2}} \text{ row } 1 + \text{ row } \mathbf{2}\right)$$

$$\rightarrow \begin{bmatrix} 2 & -1 & 0 & 1 & 0 & 0 \\ 0 & \frac{3}{2} & -1 & \frac{1}{2} & 1 & 0 \\ \mathbf{0} & \mathbf{0} & \frac{4}{3} & \frac{1}{3} & \frac{2}{3} & \mathbf{1} \end{bmatrix} \quad \left(\frac{\mathbf{2}}{3} \text{ row } \mathbf{2} + \text{row } 3\right)$$

$$\begin{pmatrix} \text{Zero above} \\ \text{third pivot} \end{pmatrix} \rightarrow \begin{bmatrix} 2 & -1 & 0 & 1 & 0 & 0 \\ 0 & \frac{3}{2} & 0 & \frac{3}{4} & \frac{3}{2} & \frac{3}{4} \\ 0 & 0 & \frac{4}{3} & \frac{1}{3} & \frac{2}{3} & 1 \end{bmatrix} \quad \left(\frac{3}{4} \text{ row } \mathbf{3} + \text{ row } \mathbf{2}\right)$$

$$\begin{pmatrix} \text{Zero above} \\ \text{second pivot} \end{pmatrix} \rightarrow \begin{bmatrix} 2 & 0 & 0 & \frac{3}{2} & 1 & \frac{1}{2} \\ 0 & \frac{3}{2} & 0 & \frac{3}{4} & \frac{3}{2} & \frac{3}{4} \\ 0 & 0 & \frac{4}{3} & \frac{1}{3} & \frac{2}{3} & 1 \end{bmatrix} \quad \left(\frac{2}{3} \text{ row } 2 + \text{ row } 1\right)$$

它变成 Reduced echelon form R.

$$\begin{matrix} (\textit{divided by } 2) \\ (\textit{divided by } \frac{3}{2}) \\ (\textit{divided by } \frac{4}{3}) \end{matrix} \begin{bmatrix} 1 & 0 & 0 & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 1 & 0 & \frac{1}{2} & 1 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} \end{bmatrix} = \begin{bmatrix} I & x_1 & x_2 & x_3 \end{bmatrix} = \begin{bmatrix} I & K^{-1} \end{bmatrix}$$

1. K is symmetric across its main diagonal. Then K^{-1} is also symmetric.
2. K is tridiagonal (only three nonzero diagonals). But K^{-1} is a dense matrix with no zeros. That is another reason we don't often compute inverse matrices. The inverse of a band matrix is generally a dense matrix.
3. The product of pivots is $2\left(\frac{3}{2}\right)\left(\frac{4}{3}\right) = 4$. This number 4 is the determinant of K.

K^{-1} involves division by the determinant of K $\quad K^{-1} = \frac{1}{4}\begin{bmatrix} 3 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 3 \end{bmatrix}$.

This is why an invertible matrix cannot have a zero determinant: we need to divide.

■

12.2 LU 分解

$(E_{32}E_{31}E_{21})\,A = U$ becomes $A = \left(E_{21}^{-1}E_{31}^{-1}E_{32}^{-1}\right)U$ which is $A = LU$

Theorem 12.2.1 When a row of A starts with zeros, so does that row of L.
When a column of A starts with zeros, so does that column of U.

■ **Example 12.4 — The key reason why A equals LU.** Ask yourself about the pivot rows that are subtracted from lower rows. Are they the original rows of A? No, elimination probably changed them.

Are they rows of U? Yes, the pivot rows never change again.

When computing the third row of U, we subtract multiples of earlier rows of U (not rows of A!):

$$\text{Row 3 of } U = (\text{Row 3 of } A) - \ell_{31}(\text{Row 1 of } U) - \ell_{32}(\text{Row 2 of } of\, U)$$

Rewrite this equation to see that the row $\begin{bmatrix} \ell_{31} & \ell_{32} & 1 \end{bmatrix}$ is multiplying the matrix U :

$$(\text{Row 3 of } A) = \ell_{31}(\text{Row 1 of } U) + \ell_{32}(\text{Row 2 of } U) + 1(\text{Row 3 of } U)$$

This is exactly row 3 of $A = LU$.

That row of L holds $\ell_{31}, \ell_{32}, 1$. All rows look like this, whatever the size of A. With no row exchanges, we have $A = LU$. ■

Definition 12.2.1 — A 的 LU 分解.

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1k} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & & \vdots \\ a_{k1} & \cdots & a_{kk} & \cdots & a_{kn} \\ \vdots & & \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nk} & \cdots & a_{nn} \end{pmatrix} = LU$$

$$\text{where } L = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ l_{21} & 1 & \ddots & 0 \\ \vdots & \vdots & \ddots & 0 \\ l_{n1} & l_{n2} & \cdots & 1 \end{pmatrix}, U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & 0 & u_{nn} \end{pmatrix}$$

所以

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1r} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & & \vdots \\ a_{r1} & \cdots & a_{rr} & \cdots & a_{rn} \\ \vdots & & \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nr} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & 0 & \ddots & 0 \\ l_{r1} & \cdots & 1 & \ddots & \vdots \\ \vdots & & \vdots & \ddots & 0 \\ l_{n1} & \cdots & l_{nr} & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & 0 & u_{nn} \end{pmatrix}$$

12.2.1 $A = LDU$

$A = LU$ 是不对称的。但是可以改写为对称形式。

$$\text{Split } U \text{ into } \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \begin{bmatrix} 1 & u_{12}/d_1 & u_{13}/d_1 & \cdot \\ & 1 & u_{23}/d_2 & \cdot \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}.$$

■ **Example 12.5** $(A = LU) \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 8 \\ 0 & 5 \end{bmatrix}$ splits further into

$$(A = LDU) \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 2 & \\ & 5 \end{bmatrix} \begin{bmatrix} 1 & 4 \\ 0 & 1 \end{bmatrix}$$

■

Theorem 12.2.2 当 A 是对称矩阵的时候，且消元的时候不需要行交换：
$S = LDL^T$

■ **Example 12.6**

$$\begin{bmatrix} 1 & 2 \\ 2 & 7 \end{bmatrix} = \begin{bmatrix} \mathbf{1} & 0 \\ \mathbf{2} & \mathbf{1} \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \quad \begin{bmatrix} \mathbf{1} & \mathbf{2} \\ 0 & \mathbf{1} \end{bmatrix}$$

■

12.2.2 L、U 矩阵的性质

根据矩阵的乘法原理, 有

Theorem 12.2.3 A 的第一行元素 a_{1j} 为

$$a_{1j} = u_{1j}, j = 1, \cdots, n$$

Corollary 12.2.4 U 的第一行元素 u_{1j} 为

$$u_{1j} = a_{1j}, j = 1, \cdots, n$$

Proof.

$$A = \begin{pmatrix} \boldsymbol{a_{11}} & \cdots & \boldsymbol{a_{1r}} & \cdots & \boldsymbol{a_{1n}} \\ \vdots & \ddots & \vdots & & \vdots \\ a_{r1} & \cdots & a_{rr} & \cdots & a_{rn} \\ \vdots & & \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nr} & \cdots & a_{nn} \end{pmatrix}$$

$$= \begin{pmatrix} \boldsymbol{1} & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & 0 & \ddots & 0 \\ l_{r1} & \cdots & 1 & \cdots & \vdots \\ \vdots & \ddots & \vdots & \ddots & 0 \\ l_{n1} & \cdots & l_{nr} & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{u_{11}} & \cdots & \boldsymbol{u_{1r}} & \cdots & \boldsymbol{u_{1n}} \\ 0 & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & u_{rr} & \cdots & u_{rn} \\ \vdots & \ddots & 0 & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & u_{nn} \end{pmatrix}$$

■

Corollary 12.2.5 L 的第一列元素 l_{i1} 为

$$l_{i1} = \frac{a_{i1}}{u_{11}}, i = 2, 3, \cdots, n$$

Proof.

$$A = \begin{pmatrix} \boldsymbol{a_{11}} & \cdots & a_{1r} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & & \vdots \\ \boldsymbol{a_{r1}} & \cdots & a_{rr} & \cdots & a_{rn} \\ \vdots & & \vdots & \ddots & \vdots \\ \boldsymbol{a_{n1}} & \cdots & a_{nr} & \cdots & a_{nn} \end{pmatrix}$$

$$= \begin{pmatrix} \boldsymbol{1} & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & 0 & \ddots & 0 \\ \boldsymbol{l_{r1}} & \cdots & 1 & \ddots & \vdots \\ \vdots & & \vdots & \ddots & 0 \\ \boldsymbol{l_{n1}} & \cdots & l_{nr} & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{u_{11}} & \cdots & u_{1r} & \cdots & u_{1n} \\ 0 & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & u_{rr} & \cdots & u_{rn} \\ \vdots & \ddots & 0 & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & u_{nn} \end{pmatrix}$$

■

Theorem 12.2.6 A 的第 r 行主对角线以右元素元素 $a_{1j}(j = 1, \cdots, n)$ 为

$$a_{rj} = \sum_{k=1}^{r} l_{rk} u_{kj}, r = 1, 2, \cdots, n, j = r, \cdots, n$$

Proof.

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1r} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & & \vdots \\ a_{r1} & \cdots & \boldsymbol{a_{rr}} & \cdots & \boldsymbol{a_{rn}} \\ \vdots & & \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nr} & \cdots & a_{nn} \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & 0 & \ddots & 0 \\ l_{r1} & \cdots & \boldsymbol{1} & \cdots & 0 \\ \vdots & \cdots & \vdots & \ddots & 0 \\ l_{n1} & \cdots & l_{nr} & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} u_{11} & \cdots & u_{1r} & \cdots & u_{1n} \\ 0 & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \boldsymbol{u_{rr}} & \cdots & \boldsymbol{u_{rn}} \\ \vdots & \ddots & 0 & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & u_{nn} \end{pmatrix}$$

■

Corollary 12.2.7 U 第 r 行主对角线以右元素 u_{rj}

$$u_{rj} = a_{rj} - \sum_{k=1}^{r-1} l_{rk} u_{kj}, j = r, \cdots, n$$

Proof.

$$a_{rj} = \sum_{k=1}^{r} l_{rk} u_{kj}, r = 1, 2, \cdots, n, j = r, \cdots, n$$

$$\Rightarrow a_{rj} = \sum_{k=1}^{r-1} l_{rk} u_{kj} + l_{rr} u_{rj}, r = 1, 2, \cdots, n, j = r, \cdots, n$$

$$\Rightarrow u_{rj} = a_{rj} - \sum_{k=1}^{r-1} l_{rk} u_{kj}, j = r, \cdots, n$$

■

Corollary 12.2.8 U 的对角线元素 u_{rr}

$$u_{rr} = a_{rr} - \sum_{k=1}^{r-1} l_{rk} u_{kr}$$

Theorem 12.2.9 A 的第 r 列元素主对角线以下元素 $a_{ir} (i = r+1, \cdots, n)$ 为

$$a_{ir} = \sum_{k=1}^{r} l_{ik} u_{kr}, i = r+1, \cdots, n, r = 1, 2, \cdots, n-1$$

Proof.

$$A=\begin{pmatrix} a_{11} & \cdots & a_{1r} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & & \vdots \\ a_{r1} & \cdots & \boldsymbol{a_{rr}} & \cdots & a_{rn} \\ \vdots & & \vdots & \ddots & \vdots \\ a_{n1} & \cdots & \boldsymbol{a_{nr}} & \cdots & a_{nn} \end{pmatrix}$$

$$=\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & 0 & \ddots & 0 \\ l_{r1} & \cdots & \boldsymbol{1} & \cdots & 0 \\ \vdots & \cdots & \vdots & \ddots & 0 \\ l_{n1} & \cdots & \boldsymbol{l_{nr}} & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} u_{11} & \cdots & u_{1r} & \cdots & u_{1n} \\ 0 & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \boldsymbol{u_{rr}} & \cdots & u_{rn} \\ \vdots & \ddots & 0 & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & u_{nn} \end{pmatrix}$$

■

Corollary 12.2.10 显然, $r=1$ 时

$$a_{i1}=l_{i1}u_{11}, i=2,3,\cdots,n$$

Corollary 12.2.11 L 第 r 列主对角线以下元素 l_{ir}

$$l_{ir}=\frac{a_{ir}-\sum_{k=1}^{r-1}l_{ik}u_{kr}}{u_{rr}}, i=r+1,\cdots,n$$

12.2.3 Complexity of LU Decomposition

求解 $Ax=b, A$ 为非奇异矩阵，LU 算法为求解方程组 $Ax=b$ 的标准解法，复杂度: $\frac{2}{3}n^3+2n^2\approx\frac{2}{3}n^3$ flops

1. 对矩阵 A 进行 LU 分解 $\left(\frac{2}{3}n^3 \text{ flops}\right)$
2. 回代法: 求解 $Ly=b$ $\left(n^2 \text{ flops}\right)$
3. 回代法: 求解 $Ux=y$ $\left(n^2 \text{ flops}\right)$

12.2.4 Example of LU Decomposition

▪ **Example 12.7** 对矩阵 A 进行 LU 分解

$$A=\begin{bmatrix} 8 & 2 & 9 \\ 4 & 9 & 4 \\ 6 & 7 & 9 \end{bmatrix}$$

$$A=\begin{bmatrix} 8 & 2 & 9 \\ 4 & 9 & 4 \\ 6 & 7 & 9 \end{bmatrix}=\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & L_{32} & 1 \end{bmatrix}\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

计算 U 的第一行和 L 的第一列

$$(u_{11},u_{12},u_{13})=(8,2,9),(l_{21},l_{31})=\left(\frac{1}{2},\frac{3}{4}\right)$$

然后计算 U 的第二行和 L 的第二列

$$u_{22} = a_{22} - l_{21}u_{12} = 8, u_{23} = a_{23} - l_{21}u_{13} = -\frac{1}{2}, l_{32} = \frac{a_{32} - l_{31}u_{12}}{u_{22}} = \frac{11}{16}$$

最后计算 U 的第三行

$$u_{33} = a_{33} - l_{31}u_{13} - l_{32}u_{23} = -\frac{83}{32}$$

■

12.3 Problem of LU Decomposition

■ **Example 12.8**

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & L_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

计算 U 的第一行和 L 的第一列

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & l_{32} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

然后计算 U 的第二行和 L 的第二列

$$\begin{matrix} u_{22} = a_{22} - l_{21}u_{12} = 0 \\ u_{23} = a_{23} - l_{21}u_{13} = 2 \end{matrix} \quad l_{32} = \frac{a_{32} - l_{31}u_{12}}{u_{22}} = \frac{1}{\mathbf{0}}$$

即该矩阵无法 LU 分解！ ■

12.4 $PA = LU$

Theorem 12.4.1 非奇异矩阵 $\mathrm{A} \in \mathbb{R}^{n \times n}$，则可分解为 $A = P^T LU$
P 是一个置换矩阵，L 为下三角矩阵并且对角线元素全为 1，U 为上三角矩阵

$PA = LU$ 分解方法不唯一，随着 P 的选择不同，L、U 也不同。

■ **Example 12.9** — $PA = LU$. $A = \begin{bmatrix} 0 & 5 & 5 \\ 2 & 9 & 0 \\ 6 & 8 & 8 \end{bmatrix}, P_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, P_2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

易知

$$P_1^T = P_1^{-1} = P_1, P_2^T = P_2^{-1} = P_2$$

计算可得

$$P_1 A = \begin{bmatrix} 6 & 8 & 8 \\ 2 & 9 & 0 \\ 0 & 5 & 5 \end{bmatrix}, P_2 A = \begin{bmatrix} 2 & 9 & 0 \\ 0 & 5 & 5 \\ 6 & 8 & 8 \end{bmatrix}$$

LU 分解不唯一:

$$P_1 A = \begin{bmatrix} 6 & 8 & 8 \\ 2 & 9 & 0 \\ 0 & 5 & 5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ 0 & \frac{15}{19} & 1 \end{bmatrix} \begin{bmatrix} 6 & 8 & 8 \\ 0 & \frac{19}{3} & \frac{-8}{3} \\ 0 & 0 & \frac{135}{19} \end{bmatrix} = L_1 U_1 \Rightarrow A = P_1 L_1 U_1$$

$$P_2A = \begin{bmatrix} 2 & 9 & 0 \\ 0 & 5 & 5 \\ 6 & 8 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & \frac{-19}{5} & 1 \end{bmatrix} \begin{bmatrix} 2 & 9 & 2 \\ 0 & 5 & 5 \\ 0 & 0 & 27 \end{bmatrix} = L_2U_2 \Rightarrow A = P_2L_2U_2$$

■

Theorem 12.4.2 这个方法等价于对 A 进行行初等变换然后对 PA 进行分解 $PA = LU$

The Complexity of $PA = LU$
复杂度: $\frac{2}{3}n^3$ flops

12.5 舍入误差的影响

■ **Example 12.10**

$$\begin{bmatrix} 10^{-5} & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

解得:

$$x_1 = -\frac{1}{1-10^{-5}}, x_2 = \frac{1}{1-10^{-5}}$$

使用 LU 分解求解上述上述方程，并且使用以下两个置换矩阵:

$$P_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{or} \quad P_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

计算过程中，中间结果四舍五入到小数点后四位。
选择 1: $P_1 = I$。

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

L 和 U 四舍五入到小数点后四位

$$L = \begin{bmatrix} 1 & 0 \\ 10^5 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 10^{-5} & 1 \\ 0 & -10^5 \end{bmatrix}$$

向前回代

$$\begin{bmatrix} 1 & 0 \\ 10^5 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \Rightarrow z_1 = 1, z_2 = -10^5$$

向后回代

$$\begin{bmatrix} 10^{-5} & 1 \\ 0 & -10^5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -10^{-5} \end{bmatrix} \Rightarrow x_1 = 0, x_2 = 1$$

! x_1 的误差为 100%。

选择 2: 行进行交换。

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

L 和 U 四舍五入到小数点后四位

$$L = \begin{bmatrix} 1 & 0 \\ 10^{-5} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

向前回代

$$\begin{bmatrix} 1 & 0 \\ 10^{-5} & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow z_1 = 0, z_2 = 1$$

向后回代

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow x_1 = -1, x_2 = 1$$

x_1, x_2 的误差约为 10^{-5}。

■

不同置换矩阵 P ，算法可能导致产生不同的误差的结果; 由于数值存储存在误差: 第一种 P_1 行交换，算法不稳定; 第二种 P_2 行交换, 算法是稳定得到“准确”近似解;

在数值分析中，一些比较简单的规则去挑选置换矩阵 P，使得算法结果比较稳定。

$$\begin{bmatrix} 10^{-5} & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ 10^{-5} & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

12.6 稀疏线性方程组

Theorem 12.6.1 如果矩阵 A 是系稀疏矩阵, 则它一般可以被分解为

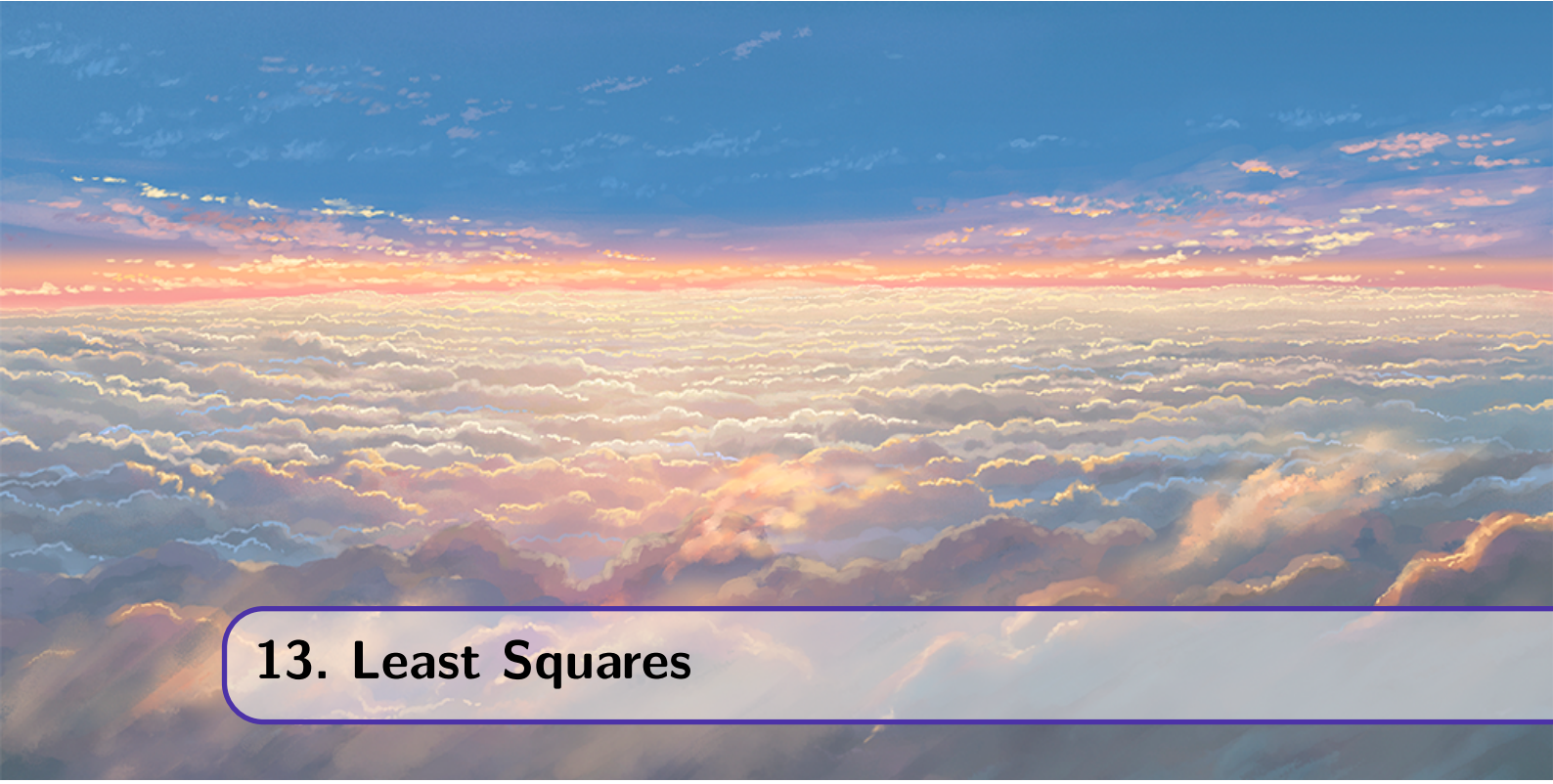
$$A = P_1 L U P_2$$

矩阵 P_1, P_2 都为置换矩阵

Corollary 12.6.2 对矩阵 A 进行行变换和列变换得到: $\tilde{A} = P_1^T A P_2^T$
然后进行分解: $\widetilde{\mathrm{A}} = LU$

P_1 和 P_2 的选择会影响 L 和 U 的稀疏度.

III Least Squares

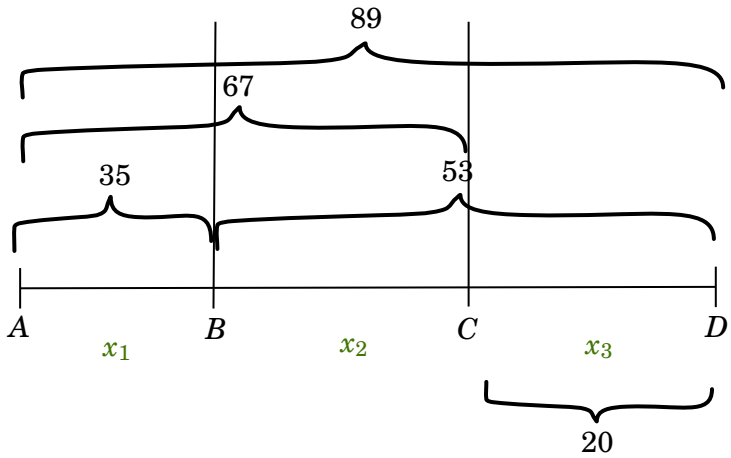


13. Least Squares

13.1 An Example: Measurement Problem

Problem 13.1 已知测量量路段长度：$AD = 89, AC = 67, BD = 53, AB = 35, CD = 20$，$x_1, x_2$ 和 x_3 的长度是多少? ▪

Figure 13.1: Measurement Problem



由 x_1, x_2 和 x_3 的关系可得方程组:

$$\begin{cases} x_1 + x_2 + x_3 = 89 \\ x_1 + x_2 = 67 \\ x_2 + x_3 = 53 \\ x_1 = 35 \\ x_3 = 20 \end{cases} \Leftrightarrow Ax = b, A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, b = \begin{bmatrix} 89 \\ 67 \\ 53 \\ 35 \\ 20 \end{bmatrix}$$

取后三个式子求解方程组，回代前两个式子

$$\begin{cases} x_2 + x_3 = 53 \\ x_1 = 35 \\ x_3 = 20 \end{cases} \Rightarrow x_1 = 35, x_2 = 33, x_3 = 20.$$
$$\begin{cases} x_1 + x_2 + x_3 = 88 \neq 89 \\ x_1 + x_2 = 68 \neq 67 \end{cases}$$

由于测量存在误差，方程组之间相互矛盾，该超定方程组无解。

Problem 13.2 — **最小二乘问题**. 寻找该方程组的近似解，并尽可能逼近方程组的目标 b, 即残差向量 $r = Ax - b$ 某种度量下尽可能小

$$\min_x \|Ax - b\|_2^2 = \|r\|_2^2 \quad (\ell_2\text{范数度量残差})$$

■

使用 ℓ_2、ℓ_∞ 等也可以度量误差，但是函数在零点处不光滑，不能求导。

Problem 13.3 — **求解最小二乘解**. 给定 $\mathrm{A} \in \mathbb{R}^{m\times n}$, $\mathrm{b} \in \mathbb{R}^m$, 求解 $x \in \mathbb{R}^n$ 让目标函数最小

$$\min_x \|Ax - b\|_2^2 = \min_x \sum_{i=1}^{m} \left(\sum_{j=1}^{n} A_{ij}x_j - b_i \right)^2$$

■

Notation 13.1 (最小二乘法的解). 最小二乘法的解为 $\hat{x}$

$$\hat{x} = \arg\min_x \|Ax - b\|_2^2 = \arg\min_x \sum_{i=1}^{m} \left(\sum_{j=1}^{n} A_{ij}x_j - b_i \right)^2$$

■ **Example 13.1** $f(x) = \|Ax - b\|_2^2, \mathrm{A} = \begin{bmatrix} 2 & 0 \\ -1 & 1 \\ 0 & 2 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$

求解

$$\hat{x} = \arg\min_x \|Ax - b\|_2^2$$

解:

$$f(x) = \|Ax - b\|_2^2 = (2x_1 - 1)^2 + (-x_1 + x_2)^2 + (2x_2 + 1)^2$$

$$\frac{\partial f}{\partial x_1} = 10x_1 - 2x_2 - 4, \frac{\partial f}{\partial x_2} = -2x_1 + 10x_2 + 4$$

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = 0 \Rightarrow \hat{x} = \left(\frac{1}{3}, -\frac{1}{3} \right)^T$$

■

Theorem 13.1.1 设最小二乘法的解为 $\hat{x}$ ，满足:

$$\|A\hat{x} - b\|_2^2 \leq \|Ax - b\|_2^2, \forall x \in \mathbf{R}^n$$

当残差 $\hat{r} = A\hat{x} - b = 0$ 时，则 $\hat{x}$ 是线性方程组 $Ax = b$ 的解; 否则其为误差最小平方

和意义下方程组的近似解。

13.2 求解最小二乘法

给定 $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, x \in \mathbb{R}^n$ 目标函数：

$$f(x) = \|Ax - b\|_2^2 = \sum_{i=1}^{m}\left(\sum_{j=1}^{n} A_{ij}x_j - b_i\right)^2$$

为使目标函数最小，求最优解 $\hat{x}$：$\hat{x} = \arg\min_x f(x)$

Theorem 13.2.1 可微函数 $f(x)$ 的最优解 $\hat{x}$ 满足条件：梯度 $\nabla f(\hat{x}) = \mathbf{0}$，即：

$$\nabla f(\hat{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\hat{x}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\hat{x}) \end{bmatrix} = 2A^T(A\hat{x} - b) = 0$$

Theorem 13.2.2 — 正规方程与最小二乘解.

$$A^TAx = A^Tb$$

A 的列向量**线性无关**时，则 $\hat{x} = \left(A^TA\right)^{-1}A^Tb$。

Proof. 设函数 $g_i(x) = \sum_{j=1}^{n} A_{ij}x_j - b_i$，则有

$$g_i(x) = \sum_{j=1}^{n} A_{ij}x_j - b_i \Rightarrow \begin{pmatrix} A_{1,1} & \cdots & A_{1,k} & \cdots & A_{1,n} \\ \vdots & & \vdots & & \vdots \\ \boldsymbol{A_{j,1}} & \cdots & \boldsymbol{A_{j,k}} & \cdots & \boldsymbol{A_{j,n}} \\ \vdots & & \vdots & & \vdots \\ A_{m,1} & \cdots & A_{m,k} & \cdots & A_{m,n} \end{pmatrix}\begin{pmatrix} \boldsymbol{x_1} \\ \vdots \\ \boldsymbol{x_j} \\ \vdots \\ \boldsymbol{x_n} \end{pmatrix} - \begin{pmatrix} b_1 \\ \vdots \\ \boldsymbol{b_j} \\ \vdots \\ b_n \end{pmatrix}$$

$$f(x) = \|Ax - b\|_2^2 = \sum_{i=1}^{m}\left(\sum_{j=1}^{n} A_{ij}x_j - b_i\right)^2 = \sum_{i=1}^{m}(g_i(x))^2$$

函数 $f(x)$ 对变量 x_k 偏导为

$$\frac{\partial f(x)}{\partial x_k} = \sum_{i=1}^{m}\left((2g_i(x))\left(\frac{\partial g_i(x)}{\partial x_k}\right)\right)$$

又因为

$$\frac{\partial g_i(x)}{\partial x_k} = A_{ik}$$

所以

$$\begin{aligned}\frac{\partial f}{\partial x_k}(x) &= \sum_{i=1}^{m} 2\left(g_i(x)\right)\left(A_{ik}\right) \\ &= 2\sum_{i=1}^{m}\left(\left(\sum_{j=1}^{n} A_{ij}x_j - b_i\right)\left(A_{ik}\right)\right) \\ &= 2\sum_{i=1}^{m}\left(\left(\sum_{j=1}^{n} A_{ij}x_j\right)\left(A_{ik}\right)\right) - 2\sum_{i=1}^{m}\left(\left(b_i\right)\left(A_{ik}\right)\right)\end{aligned}$$

注意有

$$\sum_{j=1}^{n} A_{ij}x_j = \begin{pmatrix} A_{1,1} & \cdots & A_{1,k} & \cdots & A_{1,n} \\ \vdots & & \vdots & & \vdots \\ \boldsymbol{A_{j,1}} & \cdots & \boldsymbol{A_{j,k}} & \cdots & \boldsymbol{A_{j,n}} \\ \vdots & & \vdots & & \vdots \\ A_{m,1} & \cdots & A_{m,k} & \cdots & A_{m,n} \end{pmatrix}\begin{pmatrix} \boldsymbol{x_1} \\ \vdots \\ \boldsymbol{x_j} \\ \vdots \\ \boldsymbol{x_n} \end{pmatrix} = \begin{pmatrix} Result_1 \\ \vdots \\ \boldsymbol{Result_k} \\ \vdots \\ Result_m \end{pmatrix} = Ax$$

$$\sum_{i=1}^{m}\left(\underbrace{\left(\sum_{j=1}^{n} A_{ij}x_j\right)}_{Result}\left(A_{ik}\right)\right) = \begin{array}{c} A_{1,\boldsymbol{k}} \times Result_1 \\ + \\ \vdots \\ + \\ \boldsymbol{A_{i,k}} \times \boldsymbol{Result_i} \\ + \\ \vdots \\ + \\ \boldsymbol{A_{m,k}} \times Result_m \end{array} = \begin{pmatrix} A_{1,\boldsymbol{k}} \\ \vdots \\ A_{i,\boldsymbol{k}} \\ \vdots \\ A_{m,\boldsymbol{k}} \end{pmatrix}^T Ax = a_k^T Ax$$

类似地，有

$$\sum_{i=1}^{m}\left(\left(b_i\right)\left(A_{ik}\right)\right) = \begin{array}{c} A_{1,\boldsymbol{k}} \times b_1 \\ + \\ \vdots \\ + \\ \boldsymbol{A_{i,k}} \times \boldsymbol{b_i} \\ + \\ \vdots \\ + \\ A_{\boldsymbol{m},\boldsymbol{k}} \times b_m \end{array} = a_k^T b$$

所以

$$\begin{aligned}\frac{\partial f}{\partial x_k}(x) &= 2a_k^T Ax - 2a_k^T b \\ &= 2a_k^T(Ax - b)\end{aligned}$$

所以函数 $f(x)$ 的梯度

$$\begin{aligned}\nabla f(x) &= \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix} \\ &= 2 \begin{bmatrix} a_1^T(Ax-b) \\ a_2^T(Ax-b) \\ \vdots \\ a_n^T(Ax-b) \end{bmatrix} \\ &= 2\left[a_1, a_2, \cdots, a_n\right]^T (Ax-b) \\ &= 2A^T(Ax-b)\end{aligned}$$

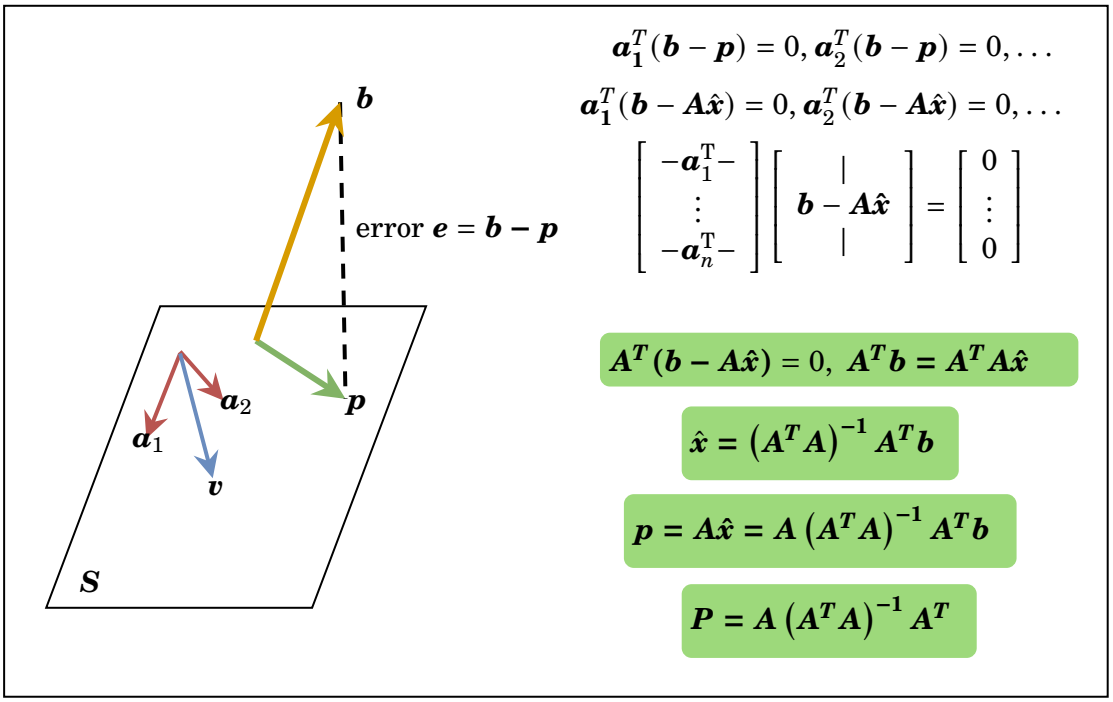
$$\nabla f(x) = 2\left(A^TAx - A^Tb\right) = 0 \Rightarrow A^TAx = A^Tb$$

A 的列向量无关时，则 $\hat{x} = \left(A^TA\right)^{-1}A^Tb$。

■

13.3 The Geometry of Least Squares: 投影与 A 列空间的关系

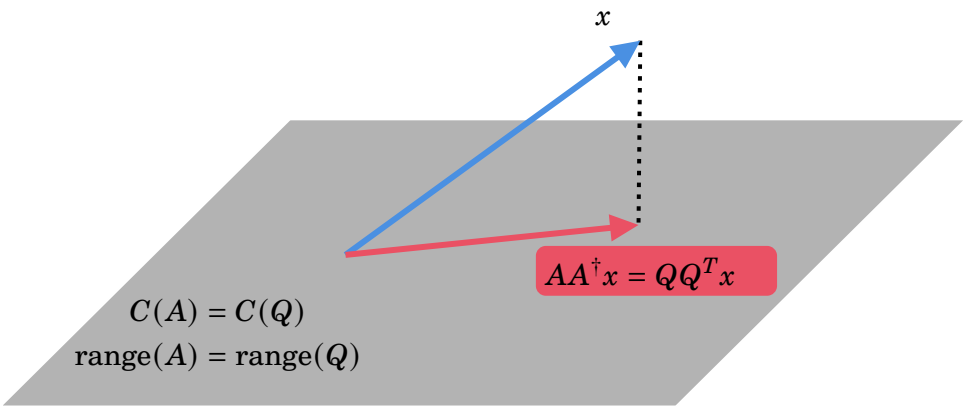
Figure 13.2: Projection of $\boldsymbol{b}$ into the column space of A, A is any matrice. Sourced from [**Strang1993IntroductionTL**]



矩阵 A $\in \mathbb{R}^{m\times n}$ 的列 $a_1, a_2, \ldots, a_n \in \mathbb{R}^m$ 的最小二乘法问题

$$\hat{x} = \arg\min_x \|Ax-b\|_2^2, \|Ax-b\|_2^2 = \left\|\sum_{j=1}^n a_j x_j - b\right\|_2^2$$

Figure 13.3: Projecting onto the column space of A is also projecting onto the column space of Q



向量 b 在 range(A) 上的投影是 $A\left(A^TA\right)^{-1}A^Tb$。

残余向量 $\hat{r} = A\hat{x} - b$ 满足 $A^T\hat{r} = A^T(A\hat{x} - b) = 0$。残余向量 $\hat{r}$ 正交于 A 的每一列, 因此正交于 range(A)。

Theorem 13.3.1 — 投影与 A 列空间的关系. $A\hat{x} \in$ range(A) 是 A 的列空间中最接近 b 的向量。

$\hat{r} = A\hat{x} - b$ 正交于 A 的列空间（值域空间）range(A)。

13.4 正规方程

Theorem 13.4.1 — 最小二乘法问题的正规方程.

$$A^TAx = A^Tb$$

等价于

$$\nabla f(x) = 0, f(x) = \|Ax - b\|_2^{2\prime\prime}$$

系数矩阵 A^TA 是 A 的 Gram 矩阵，最小二乘法问题所有的解都满足正规方程。

Theorem 13.4.2 如果 A 的列线性无关，则

A^TA 为非奇异矩阵，正规方程此时有唯一解。

13.5 QR 分解求解最小二乘法

Theorem 13.5.1 — QR 分解求解最小二乘法. 若 A $\in \mathbb{R}^{m\times n}$ 的列向量线性无关，则存在 A = QR 分解，$Q \in \mathbb{R}^{m\times n}$ ，$R \in \mathbb{R}^{n\times n}$

最小二乘法问题的解

$$\begin{aligned}\hat{x} = \left(A^T A\right)^{-1} A^T b &= \left((QR)^T (QR)\right)^{-1} (QR)^T b \\ &= \left(R^T Q^T QR\right)^{-1} R^T Q^T b \\ &= \left(R^T R\right)^{-1} R^T Q^T b \\ &= R^{-1} Q^T b\end{aligned}$$

■ **Example 13.2**

$$A = \begin{bmatrix} 3 & -6 \\ 4 & -8 \\ 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} -1 \\ 7 \\ 0 \end{bmatrix}$$

首先对 A 进行 QR 分解

$$Q = \begin{bmatrix} 3/5 & 0 \\ 4/5 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 5 & -10 \\ 0 & 1 \end{bmatrix}$$

计算 $d = Q^T b = (5, 2)$
求解 $Rx = d$

$$\begin{bmatrix} 5 & -10 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

解得 $x_1 = 5, x_2 = 2$

■

13.5.1 The Complexity of Solving Least Square Problem via QR Decomposition

算法复杂度:

- 首先对 A 进行 QR 分解 $A = QR$ $\left(2mn^2 \text{ flops }\right)$
- 计算矩阵向量乘积 $d = Q^T b$(2mn flops)
- 通过回代求解 $Rx = d$ $\left(n^2 \text{ flops }\right)$
- 复杂度: $2mn^2$ flops

13.6 求解正规方程可能带来的严重误差

直接求解正规方程组求解:

$$A^T Ax = A^T b$$

可能会造成严重的舍入误差。

■ **Example 13.3** 一个列向量“几乎”线性相关的矩阵

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

将中间结果四舍五入到小数点后 8 位.
方法 1： 通过 Gram 矩阵求解

$$A^T A = \begin{bmatrix} 1 & -1 \\ -1 & 1 + 10^{-10} \end{bmatrix} \approx \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, A^T b = \begin{bmatrix} 0 \\ 10^{-10} \end{bmatrix} \Rightarrow x = \begin{bmatrix} 10^{-10} \\ 10^{-10} \end{bmatrix}$$

经过四舍五入之后，Gram 矩阵为奇异矩阵。

方法 2 : 通过对 A 进行 QR 分解

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, R = \begin{bmatrix} 1 & -1 \\ 0 & 10^{-5} \end{bmatrix}$$

$$\begin{aligned} \hat{x} &= \left(A^T A\right)^{-1} A^T b = R^{-1} Q^T b \\ \Rightarrow & Rx = Q^T b \\ \Rightarrow & \begin{bmatrix} 1 & -1 \\ 0 & 10^{-5} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 10^{-5} \end{bmatrix} \\ \Rightarrow & x = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \end{aligned}$$

■

方法 2 比方法 1 更稳定，因为它避免构造 Gram 矩阵。

13.7 梯度下降法

给定 $A \in \mathbb{R}^{m \times n}$, b $\in \mathbb{R}^m, x \in \mathbb{R}^n$ 目标函数:

$$f(x) = \|Ax - b\|_2^2 = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} A_{ij} x_j - b_i \right)^2$$

为使目标函数最小, 可求最优解 $\hat{x}$: $\hat{x} = \arg\min_x f(x)$。

Problem 13.4 $A \in \mathbb{R}^{m \times n}$ 列向量线性相关或 n 非常大 $A^T A \in \mathbb{R}^{n \times n}$ 不可逆，无法直接代入求得最小二乘解。 ■

通过迭代求解目标的最优解过程:

$$x^{(1)}, x^{(2)}, \cdots, x^{(k)} \to \hat{x}$$

设 $x^{(k)}$ 是第 k 步迭代，期望更新 $x^{(k+1)}$ ，满足 $f\left(x^{(k+1)}\right) < f\left(x^{(k)}\right)$.

设函数 $f(x)$ 可微，根据泰勒公式，在 $x^{(k)}$ 的一阶公式为

$$f\left(x^{(k+1)}\right) = f\left(x^{(k)}\right) + \left\langle \nabla f\left(x^{(k)}\right), x^{(k+1)} - x^{(k)} \right\rangle + o\left(\left\| x^{(k+1)} - x^{(k)} \right\|\right)$$

如果 $\left\| x^{(k+1)} - x^{(k)} \right\|_2$ 足够小, 则有

$$f\left(x^{(k+1)}\right) - f\left(x^{(k)}\right) \approx \left\langle \nabla f\left(x^{(k)}\right), x^{(k+1)} - x^{(k)} \right\rangle$$

Corollary 13.7.1 根据柯西不等式 1.5.1

$\left|\left\langle \nabla f\left(x^{(k)}\right), x^{(k+1)} - x^{(k)} \right\rangle\right| \le \left\| \nabla f\left(x^{(k)}\right) \right\|_2 \left\| x^{(k+1)} - x^{(k)} \right\|_2$

所以有

$$\left\langle \nabla f\left(x^{(k)}\right), x^{(k+1)} - x^{(k)} \right\rangle \ge -\left\| \nabla f\left(x^{(k)}\right) \right\|_2 \left\| x^{(k+1)} - x^{(k)} \right\|_2$$

当 $x^{(k+1)} - x^{(k)} = -\alpha_k \nabla f\left(x^{(k)}\right), \alpha_k > 0$ 时，等式成立。

由于 $-\left\|\nabla f\left(x^{(k)}\right)\right\|_2\left\|x^{(k+1)}-x^{(k)}\right\|_2$ 是非负的，此时 $f\left(x^{(k+1)}\right)-f\left(x^{(k)}\right)\leq 0$。

迭代公式为

$$x^{(k+1)}=x^{(k)}-\alpha_k\nabla f\left(x^{(k)}\right), f\left(x^{(k+1)}\right)<f\left(x^{(k)}\right)$$

Definition 13.7.1 — 梯度下降法求解最小二乘法.

$$\min_{x\in\mathbb{R}^n}\frac{1}{2}\|Ax-b\|_2^2,\quad A\in\mathbb{R}^{m\times n}, b\in\mathbb{R}^m$$

令

$$f(x)=\frac{1}{2}\|Ax-b\|_2^2$$

则 f 为凸函数, 并有 $\nabla f(x)=A^T(Ax-b)$。

则 $A^TA\in\mathbb{R}^{n\times n}$ 。如果列向量**线性相关**会导致其**不可逆**或 n 非常大。可以通过梯度下降法迭代

$$x^{(k+1)}=x^{(k)}-\alpha_k\nabla f\left(x^{(k)}\right), f\left(x^{(k+1)}\right)<f\left(x^{(k)}\right)$$

求解

$$x^{(k+1)}=x^{(k)}-\alpha^{(k)}A^T\left(Ax^{(k)}-b\right)$$

13.8 估计学习率（步长）α

Problem 13.5

$$\min_{x\in\mathbb{R}^n}\frac{1}{2}\|Ax-b\|_2^2,\quad A\in\mathbb{R}^{m\times n}, b\in\mathbb{R}^m$$

令 $f(x)=\frac{1}{2}\|Ax-b\|_2^2$

$$x^{(k+1)}=x^{(k)}-\alpha^{(k)}A^T\left(Ax^{(k)}-b\right)$$

需要估计 $\alpha^{(k)}$。

为了估计 $\alpha^{(k)}$, 通过线性搜索估计:

$$\alpha^{(k)}=\arg\min_{\alpha\in\Re}f\left(x^{(k)}-\alpha A^T\left(Ax^{(k)}-b\right)\right)$$

即 $\alpha^{(k)}$ 是最优步长。在上面的优化式中 $x^{(k)}$、A、b 均视为定值。

Theorem 13.8.1 — 线性搜索估计的最优步长.

$$\alpha^{(k)}=\frac{\left\|A^T\left(Ax^{(k)}-b\right)\right\|_2^2}{\left\|AA^T\left(Ax^{(k)}-b\right)\right\|_2^2}$$

Proof. 令 $g(\alpha)=f\left(x^{(k)}-\alpha A^T\left(Ax^{(k)}-b\right)\right)$ 是关于 α 的凸函数, 则有

$$\min_\alpha g(\alpha)\Rightarrow g'(\alpha)=0\Rightarrow\alpha^{(k)}=\frac{\left\|A^T\left(Ax^{(k)}-b\right)\right\|_2^2}{\left\|AA^T\left(Ax^{(k)}-b\right)\right\|_2^2}$$

$$
\begin{aligned}
&f(x)=\frac{1}{2}\|Ax-b\|_2^2, g\left(\alpha^{(k)}\right)=f\left(x^{(k)}-\alpha^{(k)}A^T\left(Ax^{(k)}-b\right)\right)\\
\Rightarrow &g\left(\alpha^{(k)}\right)=\frac{1}{2}\left\|A\left(x^{(k)}-\alpha^{(k)}A^T\left(Ax^{(k)}-b\right)\right)-b\right\|_2^2\\
&=\frac{1}{2}\left\|\left(Ax^{(k)}-b\right)-\left(\alpha^{(k)}A^T\left(Ax^{(k)}-b\right)\right)\right\|_2^2\\
&=\frac{1}{2}\left(\left(Ax^{(k)}-b\right)^T\left(Ax^{(k)}-b\right)+\left(\alpha^{(k)}A^T\left(Ax^{(k)}-b\right)\right)^T\left(\alpha^{(k)}A^T\left(Ax^{(k)}-b\right)\right)\right.\\
&\left.-\left(Ax^{(k)}-b\right)^T\left(\alpha^{(k)}A^T\left(Ax^{(k)}-b\right)\right)\right)\\
\Rightarrow &g'\left(\alpha^{(k)}\right)=\alpha^{(k)}\left(A^T\left(Ax^{(k)}-b\right)\right)^T\left(A^T\left(Ax^{(k)}-b\right)\right)-\left(Ax^{(k)}-b\right)^T\left(A^T\left(Ax^{(k)}-b\right)\right)=0\\
\Rightarrow &\alpha^{(k)}=\frac{\left\|A^T\left(Ax^{(k)}-b\right)\right\|_2^2}{\left\|AA^T\left(Ax^{(k)}-b\right)\right\|_2^2}
\end{aligned}
$$

■

Algorithm 15: 使用线性搜索估计步长的梯度下降法

初始 $x^{(0)}, k=0$
while *Not Convergent* **do**
  $p^{(k)}=A^T\left(Ax^{(k)}-b\right)$
  $\alpha^{(k)}=\dfrac{\left\|p^{(k)}\right\|_2^2}{\left\|Ap^{(k)}\right\|_2^2}$
  $x^{(k+1)}=x^{(k)}-\alpha^{(k)}p^{(k)}$
end

14. Multi-objective Least Squares

14.1 Definition of Multi-objective Least Squares

Problem 14.1 假设有以下多个目标

$$J_1(x) = \|A_1x - b_1\|_2^2, \cdots, J_k(x) = \|A_kx - b_k\|_2^2$$

矩阵 $A_i \in \mathbb{R}^{m_i \times n}$, 向量 $b_i \in \mathbb{R}^{m_i}$;
寻找一个向量 $x \in \mathbb{R}^n$ 使得这 k 个目标 $_i(x), i = 1, \cdots, k$ 最小。
可以将上述多目标规划问题转换为加权最小二乘法问题。

Problem 14.2 — 加权最小二乘法问题.

$$\min_x \left\{J(x) = \lambda_1 J_1(x) + \cdots \lambda_k J_k(x) = \lambda_1 \|A_1x - b_1\|_2^2 + \cdots + \lambda_k \|A_kx - b_k\|_2^2\right\}$$

$\lambda_i > 0, i = 1, \cdots, k$, 表示不同目标的相对重要程度。

$$\begin{aligned} J(x) &= \lambda_1 \|A_1x - b_1\|_2^2 + \cdots + \lambda_k \|A_kx - b_k\|_2^2 \\ &= \left\|\sqrt{\lambda_1}\,(A_1x - b_1)\right\|_2^2 + \cdots + \left\|\sqrt{\lambda_k}\,(A_kx - b_k)\right\|_2^2 \end{aligned}$$

利用 ℓ_2 范数平方的可加性, 目标函数 $J(x)$ 可以写成紧密形式:

$$J(x) = \left\| \begin{bmatrix} \sqrt{\lambda_1}\,(A_1x - b_1) \\ \vdots \\ \sqrt{\lambda_k}\,(A_kx - b_k) \end{bmatrix} \right\|_2^2$$

进一步可简化为 $J(x) = \|\tilde{A}x - \tilde{b}\|_2^2$, 其中

$$\tilde{A} = \begin{bmatrix} \sqrt{\lambda_1}A_1 \\ \vdots \\ \sqrt{\lambda_k}A_k \end{bmatrix}, \quad \tilde{b} = \begin{bmatrix} \sqrt{\lambda_1}b_1 \\ \vdots \\ \sqrt{\lambda_k}b_k \end{bmatrix}$$

因此将多目标问题转化为单目标问题，使用最小二乘法进行求解。

■ **Example 14.1 — 双目标规划问题.**

$$\min_x \|A_1x - b_1\|_2^2 + \lambda \|A_2x - b_2\|_2^2, A_1, A_2 \in \mathbb{R}^{10\times5}$$

■

14.2 求解多目标最小二乘问题

$$\tilde{A} = \begin{bmatrix} \sqrt{\lambda_1}A_1 \\ \vdots \\ \sqrt{\lambda_k}A_k \end{bmatrix}, \quad \tilde{b} = \begin{bmatrix} \sqrt{\lambda_1}b_1 \\ \vdots \\ \sqrt{\lambda_k}b_k \end{bmatrix}$$

如果 $\tilde{A}$ 的列向量线性无关时，则该问题的解唯一；
每一个矩阵 A_i 的行向量可以线性相关；

$$\begin{aligned} \hat{x} &= \left(\tilde{A}^T\tilde{A}\right)^{-1}\tilde{A}^T\tilde{b} \\ &= \left(\lambda_1{A_1}^TA_1 + \cdots + \lambda_k{A_k}^TA_k\right)^{-1}\left(\lambda_1{A_1}^Tb_1 + \cdots + \lambda_k{A_k}^Tb_k\right) \end{aligned}$$

可对 $\tilde{A}$ 进行 QR 分解计算 $\hat{x}$ 。

14.3 正则化数据拟合

线性模型拟合数据 $\left(x^{(1)}, y^{(1)}\right), \cdots, \left(x^{(N)}, y^{(N)}\right)$

$$\hat{f}(x) = \theta_1 f_1(x) + \cdots \theta_p f_p(x), \theta = \left[\theta_1, \cdots, \theta_p\right]^T$$

$f_1(x)$ 为常数函数，且恒等于 1 。

较大的参数 θ_i 会让模型对 $f_i(x)$ 的变化更加敏感。让参数 $\theta_2, \ldots, \theta_p$ 更小，可以避免模型过拟合。

即可引出两个目标函数:

$$J_1(\theta) = \sum_{k=1}^{N}\left(\hat{f}\left(x^{(k)}\right) - y^{(k)}\right)^2, \quad J_2(\theta) = \sum_{j=2}^{p}\theta_j^2$$

首要目标 $J_1(\theta)$ 是误差的平方和。

$$\min_\theta J_1(\theta) + \lambda J_2(\theta) = \sum_{k=1}^{N}\left(\hat{f}\left(x^{(k)}\right) - y^{(k)}\right)^2 + \lambda\sum_{j=2}^{p}\theta_j^2$$

正则化参数 $\lambda > 0$ ； 该问题等价于最小二乘法问题:

$$\min_\theta \left\| \begin{bmatrix} A_1 \\ \sqrt{\lambda}A_2 \end{bmatrix}\theta - \begin{bmatrix} y_{d\times1} \\ 0 \end{bmatrix} \right\|_2^2$$

$$A_1 = \begin{bmatrix} 1 & f_2\left(x^{(1)}\right) & \ldots & f_p\left(x^{(1)}\right) \\ 1 & f_2\left(x^{(2)}\right) & \ldots & f_p\left(x^{(2)}\right) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & f_2\left(x^{(N)}\right) & \cdots & f_p\left(x^{(N)}\right) \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}, y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix}$$

14.4 图像逆问题

Problem 14.3

$$y = Ax_{ex} + v$$

向量 $x_{ex} \in \mathbb{R}^n$ 表示未知原始信息 (需要估计) 向量 $v \in \mathbb{R}^m$ 表示未知的误差或者噪声向量 $y \in \mathbb{R}^m$ 为观测的已知数据矩阵 $A \in \mathbb{R}^{m\times n}$ 将测量值 y 和原始信息 x_{ex} 之间的关系

最小二乘估计法:

$$\min_x \|Ax - y\|_2^2$$

■

利用未知 x_{ex} 先验信息，对目标进行约束，构成多目标优化问题。

例如：Tikhonov 正则化

$$\min_x \|Ax - y\|_2^2 + \lambda\|x\|_2^2, \lambda > 0$$

目标在于使 $\|Ax - y\|_2^2$ 足够小，同时 x 的能量也要小; 该优化模型等价于求解:

$$\left(A^TA + \lambda I\right)x = A^Ty$$

即使矩阵 A 的列线性相关时，也有唯一解!

14.5 信号去噪

观察信号向量 $y \in \mathbb{R}^n$ ，

$$y = x_{ex} + v$$

$x_{ex} \in \mathbb{R}^n$ 是未知信号, $v \in \mathbb{R}^n$ 是噪声。

目标是找一个信号变换缓慢，信号既有光滑性，同时逼近 y ，其优化模型:

$$\min_x \|x - y\|_2^2 + \lambda\sum_{i=1}^{n-1}(x_{i+1} - x_i)^2, \lambda > 0$$

Definition 14.5.1 — 差分矩阵. 令矩阵 $D \in \mathbb{R}^{(n-1)\times n}$ 为差分矩阵:

$$D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}$$

则有 $\sum_{i=1}^{n-1}(x_{i+1} - x_i)^2 = \|Dx\|_2^2$

$$\min_x \left\| \begin{bmatrix} I \\ \sqrt{\lambda}D \end{bmatrix} x - \begin{bmatrix} y \\ 0 \end{bmatrix} \right\|_2^2$$

优化模型等价于求解线性方程:

$$\left(I + \lambda D^TD\right)x = y$$

二维图像 $X \in \mathbb{R}^{M\times N}$ ，可按列存储成向量 $x \in \mathbb{R}^{MN}$:

$$x = \begin{bmatrix} X_{1:M,1} \\ X_{1:M,2} \\ \vdots \\ X_{1:M,N} \end{bmatrix}$$

■ **Example 14.2 — reshape.**

$$X = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix} \Rightarrow x = \begin{bmatrix} 1 \\ 4 \\ 2 \\ 5 \end{bmatrix}$$

■

$$y = Ax_{ex} + v$$

未知图像 $x_{ex} \in \mathbb{R}^{MN}$ ，观察图像 $y \in \mathbb{R}^{MN}$; 模糊矩阵 $A \in \mathbb{R}^{MN\times MN}$ 是已知观测图像与未知图像关系噪声 $v \in \mathbb{R}^{MN}$ (未知)

图像具有光滑性：图像相邻两个像素值之间变化不大。

- 水平方向：$X[n_1, n_2+1] \approx X[n_1, n_2]$
- 垂直方向：$X[n_1+1, n_2] \approx X[n_1, n_2]$

Definition 14.5.2 — 垂直差分矩阵. 垂直差分矩阵是大小为 $N \times N$ 的块矩阵，每块大小 $(M-1)\times M$。

$$D_v = \begin{bmatrix} D & 0 & \cdots & 0 \\ 0 & D & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \vdots & D \end{bmatrix}, D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}$$

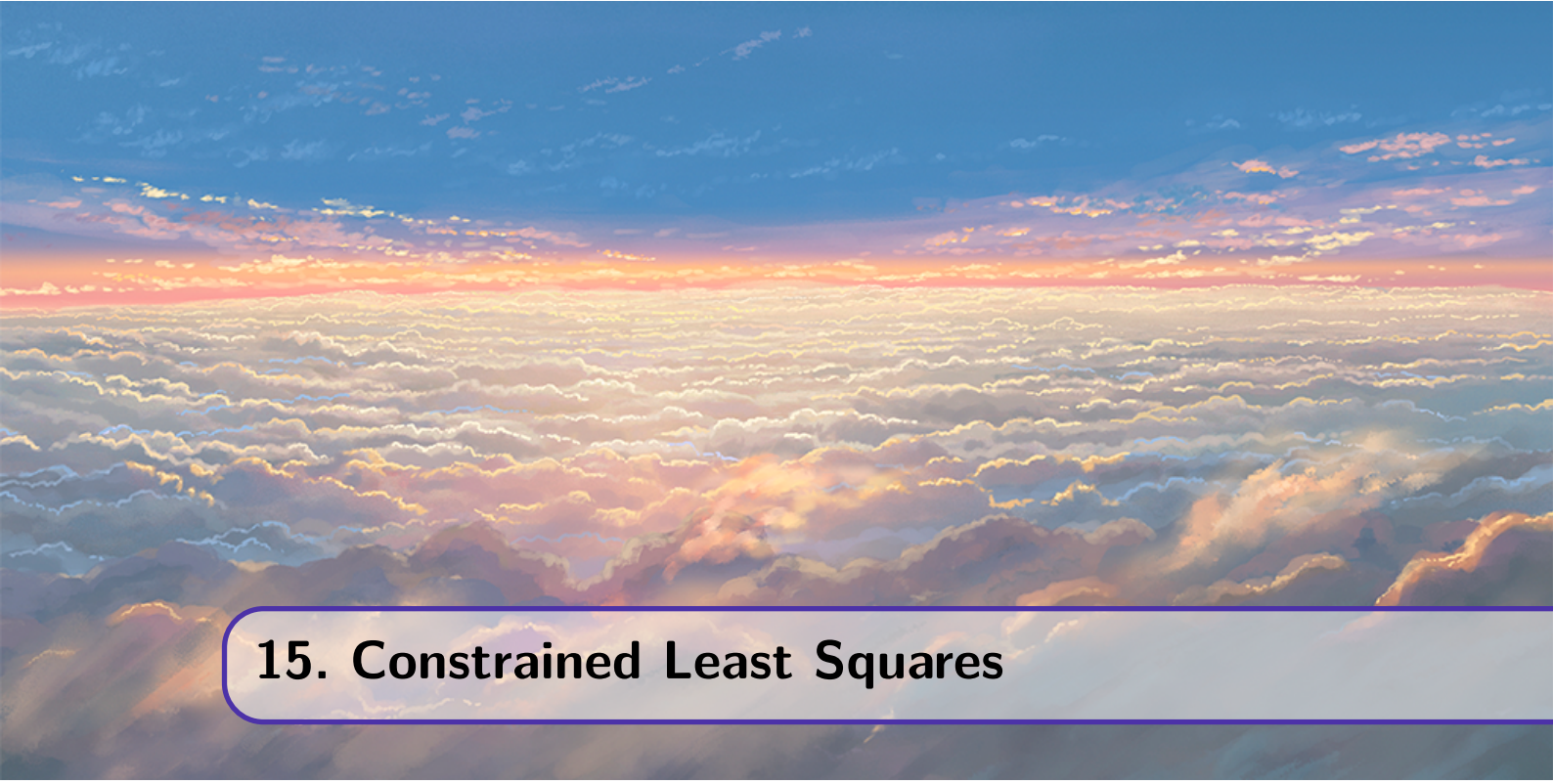
$$X = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix} \Rightarrow x = \begin{bmatrix} 1 \\ 4 \\ 2 \\ 5 \end{bmatrix}, D_v x \Rightarrow \begin{bmatrix} 4-1 \\ 5-2 \end{bmatrix}$$

$$\hat{x} = \arg\min_x \|Ax - y\|_2^2 + \lambda \|D_v x\|_2^2 + \lambda \|D_h x\|_2^2, \lambda > 0$$

$\|Ax - y\|_2^2$ 称为保证项: 保证 $A\hat{x} \approx y$.

$\lambda \|D_v x\|_2^2 + \lambda \|D_h x\|_2^2$ 为惩罚项，惩罚相邻像素值的差异变化

$$\|D_h x\|_2^2 + \|D_v x\|_2^2 = \sum_{i=1}^{M}\sum_{j=1}^{N-1}\left(X_{i,j+1} - X_{ij}\right)^2 + \sum_{i=1}^{M-1}\sum_{j=1}^{N}\left(X_{i+1,j} - X_{ij}\right)^2$$



15. Constrained Least Squares

Problem 15.1

$$\begin{aligned} &\min_x \left\{ f(x) = 2x_1^2 + x_2^2 \right\} \\ &\text{s.t.} \quad h(x) = x_1 + x_2 - 1 = 0 \end{aligned}$$

直接利用无约束优化问题求解：

$$\nabla f(x) = \begin{bmatrix} 4x_1 \\ 2x_2 \end{bmatrix} = 0 \Rightarrow x = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

显然不满足约束条件 $x_1 + x_2 - 1 = 0 + 0 - 1 \neq 0$, 不是优化问题的解。 ∎

由约束条件可得 $x_1 = 1 - x_2$ ，代入目标函数则有 $f(x) = 3x_2^2 - 4x_2 + 2$ ，即当 $\hat{x}_2 = \frac{2}{3}$ 时，目标函数值最小，并有 $\hat{x}_1 = 1 - \hat{x}_2 = \frac{1}{3}$ 。

惩罚未能满足约束条件, 引入拉格朗日函数 (Lagrange Function)

$$L(x, \lambda) = f(x) - \lambda h(x) = 2x_1^2 + x_2^2 + \lambda (1 - x_1 - x_2)$$

$$\left.\begin{aligned} \frac{\partial L}{\partial x_1} &= 4x_1 - \lambda = 0 \\ \frac{\partial L}{\partial x_2} &= 2x_2 - \lambda = 0 \\ \frac{\partial L}{\partial \lambda} &= 1 - x_1 - x_2 = 0 \end{aligned}\right\} \Rightarrow \hat{x}_1 = \frac{1}{3}, \hat{x}_2 = \frac{2}{3}, \hat{\lambda} = \frac{4}{3}$$

Definition 15.0.1 — Lagrange Functions. $\min_x / \max f(x)$ s.t. $h_i(x) = 0, i \in I \triangleq \{1, \cdots, p\}$ $g_j(x) \leq 0, j \in J \triangleq \{1, \cdots, q\}$

假设 $\lambda_i \in \mathbb{R}, i \in \mathrm{I}, u_j \in \mathbb{R}^+, j \in J$, 拉格朗日乘子 (Lagrange Multipliers)

引入拉格朗日函数: $L(x, \lambda, u) = f(x) - \sum_{i \in I} \lambda_i h_i(x) - \sum_{j \in J} u_j g_j(x)$, $\lambda = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_p \end{bmatrix}, u =$

$$\begin{bmatrix} u_1 \\ \vdots \\ u_q \end{bmatrix}$$

$$\nabla_x L(x, \lambda, u) = \nabla_x f(x) - \sum_{i \in I} \lambda_i \nabla_x h_i(x) - \sum_{j \in J} u_j \nabla_x g_j(x) = 0$$

Theorem 15.0.1 — Karush-Kuhn-Tucker Conditions. $h_i(x) = 0, i \in I$ $\lambda_i h_i(x) = 0, i \in I$ $g_j(x) \leq 0, j \in J$ $u_j g_j(x) = 0, j \in J$ $u_j \geq 0, j \in J$

Example
Example 2

15.1 An Example for Karush-Kuhn-Tucker Conditions

■ **Example 15.1**

Problem 15.2 $\max_x \{f(x) = 20x_1 + 10x_2\}$ s.t. $g_1(x) = x_1^2 + x_2^2 \leq 1, g_2(x) = x_1 + 2x_2 \leq 2$

■

$g_3(x) = -x_1 \leq 0, g_4(x) = -x_2 \leq 0$

$\nabla_x L(x, u) = \nabla_x f(x) - u_1 \nabla_x g_1(x) - u_2 \nabla_x g_2(x) - u_3 \nabla_x g_3(x) - u_4 \nabla_x g_4(x) = 0, u_j \geq 0$

$\nabla_x f(x) = \begin{bmatrix} 20 \\ 10 \end{bmatrix}, \nabla_x g_1(x) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix}, \nabla_x g_2(x) = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \nabla_x g_3(x) = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \nabla_x g_4(x) = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$.

$$\nabla_x f(x) = u_1 \nabla_x g_1(x), \nabla_x f(x) \neq u_2 \nabla_x g_2(x), \nabla_x f(x) \neq u_3 \nabla_x g_3(x), \nabla_x f(x) \neq u_4 \nabla_x g_4(x)$$

$$u_2 = u_3 = u_4 = 0, \quad u_1 \neq 0$$

即$\nabla_x f(x) = u_1 \nabla_x g_1(x), \begin{bmatrix} 20 \\ 10 \end{bmatrix} = u_1 \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} \Rightarrow x_1 = 2x_2$, 代入$g_1(x) = x_1^2 + x_2^2 - 1 = 5x_2^2 - 1 = 0$.

由于 $x_2 \geq 0$, 可得 $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \frac{\sqrt{5}}{5} \begin{bmatrix} 2 \\ 1 \end{bmatrix}, u_1 = 5\sqrt{5}$.

■

15.2 Supplement Material: Karush-Kuhn-Tucker (KKT) 条件

Cited from `https://zhuanlan.zhihu.com/p/38163970`.

15.2.1 等式约束优化问题

给定一个目标函数 $f : \mathbb{R}^n \to \mathbb{R}$, 我们希望找到 $\mathbf{x} \in \mathbb{R}^n$, 在满足约束条件 $g(\mathbf{x}) = 0$ 的前提下, 使得 $f(\mathbf{x})$ 有最小值。这个约束优化问题记为

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g(\mathbf{x}) = 0 \end{aligned}$$

为方便分析, 假设 f 与 g 是连续可导函数。Lagrange 乘数法是等式约束优化问题的典型解法。定义 Lagrangian 函数

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x})$$

其中 λ 称为 Lagrange 乘数。Lagrange 乘数法将原本的约束优化问题转换成等价的无约束优化问题

$$\min_{\mathbf{x},\lambda} L(\mathbf{x}, \lambda)$$

计算 L 对 $\mathbf{x}$ 与 λ 的偏导数并设为零，可得最优解的必要条件：

$$\begin{aligned} \nabla_{\mathbf{x}} L &= \frac{\partial L}{\partial \mathbf{x}} = \nabla f + \lambda \nabla g = \mathbf{0} \\ \nabla_{\lambda} L &= \frac{\partial L}{\partial \lambda} = g(\mathbf{x}) = 0 \end{aligned}$$

其中第一式为定常方程式 (stationary equation), 第二式为约束条件。解开上面 $n+1$ 个方程式可得 $L(\mathbf{x}, \lambda)$ 的驻点 (stationary point) $\mathbf{x}^\star$ 以及 λ 的值 (正负数皆可能)。

15.2.2 不等式约束优化问题

接下来我们将约束等式 $g(\mathbf{x}) = 0$ 推广为不等式 $g(\mathbf{x}) \leq 0$ 。考虑这个问题

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g(\mathbf{x}) \leq 0 \end{aligned}$$

约束不等式 $g(\mathbf{x}) \leq 0$ 称为原始可行性 (primal feasibility), 据此我们定义可行域 (feasible region) $K = \mathbf{x} \in \mathbb{R}^n \mid g(\mathbf{x}) \leq 0$ 。假设 $\mathbf{x}^\star$ 为满足约束条件的最佳解, 分开两种情况讨论:

(1) $g(\mathbf{x}^*) < 0$, 最佳解位于 K 的内部, 称为内部解 (interior solution), 这时约束条 是无效的 (inactive); (2) $g(\mathbf{x}^*) = 0$, 最佳解落在 K 的边界, 称为边界解 (boundary solution), 此时约束条件是有效的 (active).

这两种情况的最佳解具有不同的必要条件。(1) 内部解：在约束条件无效的情形下, $g(\mathbf{x})$ 不起作用, 约束优化问题退化为无约束优化问题, 因此驻点 $\mathbf{x}^\star$ 满足 $\nabla f = \mathbf{0}$ 且 $\lambda = 0$ 。

(2) 边界解：在约束条件有效的情形下, 约束不等式变成等式 $g(\mathbf{x}) = 0$, 这与前述 Lagrange 乘数法的情况相同。我们可以证明驻点 $\mathbf{x}^\star$ 发生于 $\nabla f \in \operatorname{span} \nabla g$, 换句话说, 存在 λ 使得 $\nabla f = -\lambda \nabla g$, 但这里 λ 的正负号是有其意义的。因为我们希望最小化 f, 梯度 ∇f (函数 f 在点 $\mathbf{x}$ 的最陡上升方向) 应该指向可行域 K 的内部 (因为你的最优解最小值是在边界取得的), 但 ∇g 指向 K 的外部 (即 $g(\mathbf{x}) > 0$ 的区域, 因为你的约束是小于等于 0), 因此 $\lambda \geq 0$, 称为对偶可行性 (dual feasibility)。

因此, 不论是内部解或边界解, $\lambda g(\mathbf{x}) = 0$ 恒成立, 称为互补松弛性 (complementary slackness)。整合上述两种情况, 最佳解的必要条件包括 Lagrangian 函数 $L(\mathbf{x}, \lambda)$ 的定常方程式、原始可行性、对偶可行性，以及互补松弛性：

$$\begin{aligned} \nabla_{\mathbf{x}} L = \nabla f + \lambda \nabla g &= \mathbf{0} \\ g(\mathbf{x}) &\leq 0 \\ \lambda &\geq 0 \\ \lambda g(\mathbf{x}) &= 0 \end{aligned}$$

这些条件合称为 Karush-Kuhn-Tucker (KKT) 条件。如果我们要最大化 $f(\mathbf{x})$ 且受限于 $g(\mathbf{x}) \leq 0$, 那么对偶可行性要改成 $\lambda \leq 0$ 。上面结果可推广至多个约束等式与约束不等式的情况。考虑标准约束优化问题 (或称非线性规划):

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_j(\mathbf{x}) = 0, \quad j = 1, \ldots, m \\ & h_k(\mathbf{x}) \leq 0, \quad k = 1, \ldots, p \end{aligned}$$

定义 Lagrangian 函数

$$L\left(\mathbf{x},\left\{\lambda_j\right\},\left\{\mu_k\right\}\right)=f(\mathbf{x})+\sum_{j=1}^{m}\lambda_j g_j(\mathbf{x})+\sum_{k=1}^{p}\mu_k h_k(\mathbf{x})$$

其中 λ_j 是对应 $g_j(\mathbf{x})=0$ 的 Lagrange 乘数, μ_k 是对应 $h_k(\mathbf{x})\le 0$ 的 Lagrange 乘数 (或称 KKT 乘数)。KKT 条件包括

$$\begin{aligned}\nabla_{\mathbf{x}}L&=\mathbf{0}\\ g_j(\mathbf{x})&=0,\quad j=1,\ldots,m\\ h_k(\mathbf{x})&\le 0\\ \mu_k&\ge 0\\ \mu_k h_k(\mathbf{x})&=0,\quad k=1,\ldots,p\end{aligned}$$

15.2.3 An Example

Problem 15.3 考虑这个问题

$$\begin{aligned}\min\quad & x_1^2+x_2^2\\ \text{s.t.}\quad & x_1+x_2=1\\ & x_2\le\alpha\end{aligned}$$

其中 $(x_1,x_2)\in\mathbb{R}^2,\alpha$ 为实数。 ▪

写出 Lagrangigan 函数

$$L\left(x_1,x_2,\lambda,\mu\right)=x_1^2+x_2^2+\lambda\left(1-x_1-x_2\right)+\mu\left(x_2-\alpha\right)$$

KKT 方程组如下:

$$\begin{aligned}\frac{\partial L}{\partial x_i}&=0,\quad i=1,2\\ x_1+x_2&=1\\ x_2-\alpha&\le 0\\ \mu&\ge 0\\ \mu\left(x_2-\alpha\right)&=0\end{aligned}$$

求偏导可得 $\frac{\partial L}{\partial x_1}=2x_1-\lambda=0$ 且 $\frac{\partial L}{\partial x_2}=2x_2-\lambda+\mu=0$, 分别解出 $x_1=\frac{\lambda}{2}$ 且 $x_2=\frac{\lambda}{2}-\frac{\mu}{2}$ 。代入约束等式 $x_1+x_2=\lambda-\frac{\mu}{2}=1$ 或 $\lambda=\frac{\mu}{2}+1$ 。合并上面结果,

$$x_1=\frac{\mu}{4}+\frac{1}{2},\quad x_2=-\frac{\mu}{4}+\frac{1}{2}$$

最后再加入约束不等式 $-\frac{\mu}{4}+\frac{1}{2}\le\alpha$ 或 $\mu\ge 2-4\alpha$ 。底下分开三种情况讨论。(1) $\alpha>\frac{1}{2}$: 不难验证 $\mu=0>2-4\alpha$ 满足所有的 KKT 条件, 约束不等式是无效的, $x_1^\star=x_2^\star=\frac{1}{2}$ 是内部解，目标函数的极小值是 $\frac{1}{2}$。 (2) $\alpha=\frac{1}{2}$: 如同 1, $\mu=0=2-4\alpha$ 满足所有的 KKT 条件, $x_1^\star=x_2^\star=\frac{1}{2}$ 是边界解, 因为 $x_2^\star=\alpha$

(3) $\alpha<\frac{1}{2}$: 这时约束不等式是有效的, $\mu=2-4\alpha>0$, 则 $x_1^\star=1-\alpha$ 且 $x_2^\star=\alpha$, 目标函数的极小值是 $(1-\alpha)^2+\alpha^2$。

15.3 Supplement Material: 浅谈最优化问题的 KKT 条件

Cited from https://zhuanlan.zhihu.com/p/26514613.

对于具有等式和不等式约束的一般优化问题

$$\begin{aligned} &\min f(\mathbf{x}) \\ &\text{s.t. } g_j(\mathbf{x}) \le 0 (j = 1, 2, \cdots, m) \\ &h_k(\mathbf{x}) = 0 (k = 1, 2, \cdots, l) \end{aligned}$$

KKT 条件给出了判断 $\mathbf{x}^*$ 是否为最优解的必要条件, 即:

$$\begin{cases} \frac{\partial f}{\partial x_i} + \sum_{j=1}^{m} \mu_j \frac{\partial g_j}{\partial x_i} + \sum_{k=1}^{l} \lambda_k \frac{\partial h_k}{\partial x_i} = 0, (i = 1, 2, \ldots, n) \\ h_k(\mathbf{x}) = 0, (k = 1, 2, \cdots, l) \\ \mu_j g_j(\mathbf{x}) = 0, (j = 1, 2, \cdots, m) \\ \mu_j \ge 0 \end{cases}$$

15.3.1 等式约束优化问题

所谓的等式约束优化问题是指

$$\begin{aligned} &\min f\left(x_1, x_2, \ldots, x_n\right) \\ &\text{s.t. } h_k\left(x_1, x_2, \ldots, x_n\right) = 0 \end{aligned}$$

我们令 $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{k=1}^{l} \lambda_k h_k(\mathbf{x})$, 函数 $L(x, y)$ 称为 Lagrange 函数, 参数 λ 称为 Lagrange 乘子. 再联立方程组: $\begin{cases} \frac{\partial L}{\partial x_i} = 0 (i = 1, 2, \cdots, n) \\ \frac{\partial L}{\partial \lambda_k} = 0 (k = 1, 2, \cdots, l) \end{cases}$, 得到的解为可能极值点，由于我们用的是必要条件，具体是否为极值点需根据问题本身的具体情况检验. 这个方程组称为等式约束的极值必要条件.

上式我们对 n 个 x_i 和 l 个 λ_k 分别求偏导, 回想一下在无约束优化问题 $f\left(x_1, x_2, \ldots, x_n\right) = 0$ 中, 我们根据极值的必要条件, 分别令 $\frac{\partial f}{\partial x_i} = 0$, 求出可能的极值点. 因此可以联想到：等式约束下的 Lagrange 乘数法引入了 l 个 Lagrange 乘子，或许我们可以把 λ_k 也看作优化变量（x_i 就叫做优化变量）. 相当于将优化变量个数增加到 $(n + l)$ 个, x_i 与 λ_k 一视同仁, 均为优化变量, 均对它们求偏导.

15.3.2 不等式约束优化问题

以上我们讨论了等式约束的情形，接下来我们来介绍不等式约束的优化问题. 我们先给出其主要思想：转化的思想——将不等式约束条件变成等式约束条件. 具体做法：引入松弛变量. 松弛变量也是优化变量，也需要一视同仁求偏导.

具体而言, 我们先看一个一元函数的例子:

$$\min f(x)$$

$$\text{s.t. } g_1(x) = a - x \le 0$$

$$g_2(x) = x - b \le 0$$

(注：优化问题中，我们必须求得一个确定的值，因此不妨令所有的不等式均取到等号，即 $\le$ 的情况.)

对于约束 g_1 和 g_2, 我们分别引入两个松弛变量 a_1^2 和 b_1^2, 得到 $h_1\left(x, a_1\right) = g_1 + a_1^2 = 0$ 和 $h_2\left(x, b_1\right) = g_2 + b_1^2 = 0$. 注意, 这里直接加上平方项 a_1^2、b_1^2 而非 a_1、b_1, 是因为 g_1 和 g_2 这两个不等式的左边必须加上一个正数才能使不等式变为等式. 若只加上 a_1 和 b_1, 又会引入新的约束 $a_1 \ge 0$ 和 $b_1 \ge 0$, 这不符合我们的意愿.

由此我们将不等式约束转化为了等式约束, 并得到 Lagrange 函数

$$L\left(x, a_1, b_1, \mu_1, \mu_2\right)=f(x)+\mu_1\left(a-x+a_1^2\right)+\mu_2\left(x-b+b_1^2\right)$$

我们再按照等式约束优化问题（极值必要条件）对其求解, 联立方程

$$\begin{cases} \frac{\partial F}{\partial x}=\frac{\partial f}{\partial x}+\mu_1 \frac{d g_1}{d x}+\mu_2 \frac{d g_2}{d x}=\frac{d f}{d x}-\mu_1+\mu_2=0 \\ \frac{\partial F}{\partial \mu_1}=g_1+a_1^2=0, \quad \frac{\partial F}{\partial \mu_2}=g_2+b_1^2=0 \\ \frac{\partial F}{\partial a_1}=2 \mu_1 a_1=0, \quad \frac{\partial F}{\partial b_1}=2 \mu_2 b_1=0 \\ \mu_1 \geq 0, \quad \mu_2 \geq 0 \end{cases}$$

(注: 这里的 $\mu_1 \geq 0, \mu_2 \geq 0$ 先承认, 我们待会再解释! (先上车再买票, 手动斜眼). 实际上对于不等式约束前的乘子, 我们要求其大于等于 0）得出方程组后, 便开始动手解它. 看到第 3 行的两式 $\mu_1 a_1=0$ 和 $\mu_1 a_1=0$ 比较简单, 我们就从它们入手吧对于 $\mu_1 a_1=0$, 我们有两种情况: 情形 1： $\mu_1=0, a_1 \neq 0$ 此时由于乘子 $\mu_1=0$, 因此 g_1 与其相乘为零, 可以理解为约束 g_1 不起作用, 且有 $g_1(x)=a-x<0$. 情形 2: $\mu_1 \geq 0, a_1=0$ 此时 $g_1(x)=a-x=0$ 且 $\mu_1>0$, 可以理解为约束 g_1 起作用, 且有 $g_1(x)=0$. 合并情形 1 和情形 2 得: $\mu_1 g_1=0$, 且在约束起作用时 $\mu_1>0, g_1(x)=0$; 约束不起作用时 $\mu_1=0, g_1(x)<0$. 同样地, 分析 $\mu_2 b_1=0$, 可得出约束 g_2 起作用和不起作用的情形, 并分析得到 $\mu_2 g_2=0$.

由此, 方程组（极值必要条件）转化为

$$\begin{cases} \frac{d f}{d x}+\mu_1 \frac{d g_1}{d x}+\mu_2 \frac{d g_2}{d x}=0 \\ \mu_1 g_1(x)=0, \mu_2 g_2(x)=0 \\ \mu_1 \geq 0, \mu_2 \geq 0 \end{cases}$$

这是一元一次的情形. 类似地, 对于多元多次不等式约束问题

$$\begin{aligned} & \min f(\mathbf{x}) \\ & \text { s.t. } g_j(\mathbf{x}) \leq 0(j=1,2, \cdots, m) \end{aligned}$$

我们有

$$\begin{cases} \frac{\partial f\left(x^*\right)}{\partial x_i}+\sum_{j=1}^m \mu_j \frac{\partial g_j\left(x^*\right)}{\partial x_i}=0(i=1,2, \ldots, n) \\ \mu_j g_j\left(x^*\right)=0(j=1,2, \ldots, m) \\ \mu_j \geq 0(j=1,2, \ldots, m) \end{cases}$$

上式便称为不等式约束优化问题的 KKT（Karush-Kuhn-Tucker）条件. μ_j 称为 KKT 乘子, 且约束起作用时 $\mu_j \geq 0, g_j(x)=0$； 约束不起作用时 $\mu_j=0, g_j(x)<0$. 别急, 还木有完, 我们还剩最后一个问题没有解决：为什么 KKT 乘子必须大于等于零一我将用几何性质来解释. 由于

$$\frac{\partial f\left(x^*\right)}{\partial x_i}+\sum_{j=1}^m \mu_j \frac{\partial g_j\left(x^*\right)}{\partial x_i}=0(i=1,2, \ldots, n)$$

用梯度表示: $\nabla f\left(\mathbf{x}^*\right)+\sum_{j \in J} \mu_j \nabla g_j\left(\mathbf{x}^*\right)=0, J$ 为起作用约束的集合. 移项: $-\nabla f\left(\mathbf{x}^*\right)=\sum_{j \in J} \mu_j \nabla g_j\left(\mathbf{x}^*\right)$,

注意到梯度为向量. 上式表示在约束极小值点 $\mathbf{x}^*$ 处，函数 $f\left(\mathbf{x}^*\right)$ 的负梯度一定可以表示成：所有起作用约束在该点的梯度（等值线的法向量）的线性组合.（复习课本中梯度的性质：某点梯度的方向就是函数等值线 $f(\mathbf{x})=C$ 在这点的法线方向, 等值线就是地理的等高线)

为方便作图, 假设现在只有两个起作用约束, 我们作出图形如下图. 注意我们上面推导过, 约束起作用时 $g_j(\mathbf{x}) = 0$, 所以此时约束在几何上应该是一簇约束平面. 我们假设在 $\mathbf{x}^*$ 取得极小值点, 若同时满足 $g_1(\mathbf{x}) = 0$ 和 $g_2(\mathbf{x}) = 0$, 则 $\mathbf{x}^k$ 一定在这两个平面的交线上, 且 $-\nabla f(\mathbf{x}^*) = \sum_{j\in J} \mu_j \nabla g_j(\mathbf{x}^*)$, 即 $-\nabla f\left(\mathbf{x}^k\right)$ 、$\nabla g_1\left(\mathbf{x}^k\right)$ 和 $\nabla g_2\left(\mathbf{x}^k\right)$ 共面.

下图是在点 $\mathbf{x}^k$ 处沿 x_1Ox_2 面的截面, 过点 $\mathbf{x}^k$ 作目标函数的负梯度 $-\nabla f\left(\mathbf{x}^k\right)$, 它垂直于目标函数的等值线 $f(\mathbf{x}) = C$ （高数课本：一点的梯度与等值线相互垂直），且指向目标函数 $f(\mathbf{x})$ 的最速减小方向. 再作约束函数 $g_1(\mathbf{x}) = 0$ 和 $g_2(\mathbf{x}) = 0$ 的梯度 $\nabla g_1\left(\mathbf{x}^k\right)$ 和 $\nabla g_2\left(\mathbf{x}^k\right)$, 它们分别垂直 $g_1(\mathbf{x}) = 0$ 和 $g_2(\mathbf{x}) = 0$ 两曲面在 $\mathbf{x}^k$ 的切平面, 并形成一个雉形夹角区域. 此时, 可能有 a、 b 两种情形:

我们先来看情形 b： 若 3 个向量的位置关系如 b 所示, 即 $-\nabla f$ 落在 ∇g_1 和 ∇g_2 所形成的雉角区外的一侧. 此时, 作等值面 $f(\mathbf{x}) = C$ 在点 $\mathbf{x}^k$ 的切平面（它与 $-\nabla f\left(\mathbf{x}^k\right)$ 垂直）, 我们发现：沿着与负梯度 $-\nabla f$ 成锐角的方向移动（如下图红色箭头方向）, 只要在红色区域取值, 目标函数 $f(\mathbf{x})$ 总能减小. 而红色区域是可行域 ($f(\mathbf{x}) = C$, C 取不同的常数能得到不同的等值线, 因此能取到红色区域）, 因此既可减小目标函数值, 又不破坏约束条件. 这说明 $\mathbf{x}^k$ 仍可沿约束曲面移动而不破坏约束条件, 且目标函数值还能够减小. 所以 $\mathbf{x}^k$ 不是稳定的最优点, 即不是局部极值点.

反过头来看情形 a: $-\nabla f$ 落在 ∇g_1 和 ∇g_2 形成的雉角内. 此时, 同样作 $f(\mathbf{x}) = C$ 在点 $\mathbf{x}^k$ 与 $-\nabla f$ 垂直的切平面. 当从 $\mathbf{x}^k$ 出发沿着与负梯度 $-\nabla f$ 成锐角的方向移动时, 虽然能使目标函数值减小, 但此时任何一点都不在可行区域内. 显然, 此时 $\mathbf{x}^k$ 就是局部最优点 $\mathbf{x}^*$, 再做任何移动都将破坏约束条件, 故它是稳定点.

由于 $-\nabla f(\mathbf{x}^*)$ 和 $\nabla g_1(\mathbf{x}^*)$ 、$\nabla g_2(\mathbf{x}^*)$ 在一个平面内, 所以前者可看成是后两者的线性组合. 又由上面的几何分析知, $-\nabla f(\mathbf{x}^*)$ 在 $\nabla g_1(\mathbf{x}^*)$ 和 $\nabla g_2(\mathbf{x}^*)$ 的夹角之间, 所以线性组合的系数为正, 有

$$-\nabla f(\mathbf{x}^*) = \mu_1 \nabla g_1(\mathbf{x}^*) + \mu_2 \nabla g_2(\mathbf{x}^*), \text{ 且} \mu_1 > 0, \mu_2 > 0.$$

这就是 $\mu_j > 0$ 的原因. 类似地, 当有多个不等式约束同时起作用时, 要求 $-\nabla f(\mathbf{x}^*)$ 处于 $\nabla g_j(\mathbf{x}^*)$ 形成的超角锥（高维图形, 我姑且称之为“超”）之内.

15.3.3 总结：同时包含等式和不等式约束的一般优化问题

$$\begin{aligned} &\min f(\mathbf{x}) \\ &\text{s.t. } g_j(\mathbf{x}) \le 0 (j = 1, 2, \cdots, m) \\ &h_k(\mathbf{x}) = 0 (k = 1, 2, \cdots, l) \end{aligned}$$

KKT 条件 (x^* 是最优解的必要条件）为

$$\begin{cases} \frac{\partial f}{\partial x_i} + \sum_{j=1}^{m} \mu_j \frac{\partial g_j}{\partial x_i} + \sum_{k=1}^{l} \lambda_k \frac{\partial h_k}{\partial x_i} = 0, (i = 1, 2, \ldots, n) \\ h_k(\mathbf{x}) = 0, (k = 1, 2, \cdots, l) \\ \mu_j g_j(\mathbf{x}) = 0, (j = 1, 2, \cdots, m) \\ \mu_j \ge 0 \end{cases}$$

注意，对于等式约束的 Lagrange 乘子，并没有非负的要求！以后求其极值点，不必再引入松弛变量，直接使用 KKT 条件判断！

IV Extensive Reading

16. Fourier Series, Fourier Transform

16.1 基本概念

Definition 16.1.1 — 波. 波的基本属性：频率、振幅、相位。

Definition 16.1.2 — Complex Numbers.

$$C = R + jI$$

where R and I are real numbers and $j = \sqrt{-1}$. Here, R denotes the real part of the complex number and I its imaginary part.

Real numbers are a subset of complex numbers in which $I = 0$.

Definition 16.1.3 — Complex Number in Polar Coordinates.

$$C = |C|(\cos\theta + j\sin\theta) = |C|e^{j\theta}$$

where $|C| = \sqrt{R^2 + I^2}$ is the length of the vector extending from the origin of the complex plane to point (R, I), and θ is the angle between the vector and the real axis.

上式使用了欧拉公式

Theorem 16.1.1 — Euler's Formular.

$$e^{j\theta} = \cos\theta + j\sin\theta$$

Corollary 16.1.2 $e^{2\pi it}$ 可以表示每秒 1 圈的旋转.

Definition 16.1.4 — 正弦函数.

$$y = A\sin(\omega t + \varphi)$$

就是一个以 $\frac{2\pi}{\omega}$ 为周期的正弦函数, 其中 y 表示动点的位置, t 表示时间, A 为振幅, ω 为角频率, φ 为初相.

如何深入研究非正弦周期函数呢? 将周期函数展开成由简单的周期函数例如三角函数组成的级数. 具体地说, 将周期为 $T\left(=\frac{2\pi}{\omega}\right)$ 的周期函数用一系列以 T 为周期的正弦函数 $A_n \sin(n\omega t+\varphi_n)$ 组成的级数来表示, 记为

$$f(t)=A_0+\sum_{n=1}^{\infty} A_n \sin(n\omega t+\varphi_n)$$

其中 $A_0, A_n, \varphi_n (n=1,2,3,\cdots)$ 都是常数.

将周期函数按上述方式展开, 它的物理意义是很明确的, 这就是把一个比较复杂的周期运动看成是许多不同频率的简谐振动的叠加. 在电工学上, 这种展开称为*谐波分析*, 其中常数项 A_0 称为 $f(t)$ 的*直流分量*, $A_1 \sin(\omega t+\varphi_1)$ 称为*一次谐波* (又叫做*基波*) , $A_2 \sin(2\omega t+\varphi_2)$, $A_3 \sin(3\omega t+\varphi_3)$, $\cdots$ 依次称为*二次谐波*, *三次谐波*, 等等.

Definition 16.1.5 — 三角级数. 令 $\frac{a_0}{2}=A_0, a_n=A_n \sin\varphi_n, b_n=A_n\cos\varphi_n, \omega=\frac{\pi}{l}$(即 $T=2l$), 则 $f(t)=A_0+\sum_{n=1}^{\infty} A_n \sin(n\omega t+\varphi_n)$ 可以改写为

$$\frac{a_0}{2}+\sum_{n=1}^{\infty}\left(a_n\cos\frac{n\pi t}{l}+b_n\sin\frac{n\pi t}{l}\right)$$

其中 $a_0, a_n, b_n (n=1,2,3,\cdots)$ 都是常数.

Definition 16.1.6 — 三角函数系（基波）.

$$1, \cos x, \sin x, \cos 2x, \sin 2x, \cdots, \cos nx, \sin nx, \cdots$$

在区间 $[-\pi,\pi]$ 上正交, 就是指在三角函数系 $(7-4)$ 中任何不同的两个函数的乘积在区间 $[-\pi,\boldsymbol{\pi}]$ 上的积分等于零, 即

$$\begin{array}{l}\int_{-\pi}^{\pi}\cos nx\,\mathrm{d}x=0 \quad (n=1,2,3,\cdots),\\ \int_{-\pi}^{\pi}\sin nx\,\mathrm{d}x=0 \quad (n=1,2,3,\cdots),\\ \cdots\end{array}$$

Definition 16.1.7 — 复数域的 Gram 矩阵. 如果 $A\in\mathbb{C}^{m\times n}$ 的 Gram 矩阵为单位矩阵, 则 A 具有正交列:

$$\begin{aligned}A^HA&=\begin{bmatrix}a_1 & a_2 & \cdots & a_n\end{bmatrix}^H\begin{bmatrix}a_1 & a_2 & \cdots & a_n\end{bmatrix}\\&=\begin{bmatrix}a_1^Ha_1 & a_1^Ha_2 & \cdots & a_1^Ha_n\\ a_2^Ha_1 & a_2^Ha_2 & \cdots & a_2^Ha_n\\ \vdots & \vdots & & \vdots\\ a_n^Ha_1 & a_n^Ha_2 & \cdots & a_n^Ha_n\end{bmatrix}\\&=\begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & 1 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 1\end{bmatrix}\\&=I\end{aligned}$$

Gram 矩阵列有单位范数: $\|a_i\|_2^2=a_i^Ha_i=1$ 。

Gram 矩阵列是相互正交的：对于 $i \neq j, a_i^H a_j = 0$ 。

Definition 16.1.8 — 酉矩阵. 列正交的方形复数矩阵称为酉矩阵。

Definition 16.1.9 — 酉矩阵的逆.

$$\left.\begin{array}{l} A^H A = I \\ A \text{ 是方的} \end{array}\right\} \quad \Rightarrow \quad AA^H = I$$

酉矩阵是具有逆 A^H 的非奇异矩阵。如果 A 是酉矩阵, 那么 A^H 也是酉矩阵。

16.2 Fourier Series

Definition 16.2.1 — 傅里叶级数.

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx) .$$

where

$$\begin{cases} a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos nx \, \mathrm{d}x & (n = 0, 1, 2, 3, \cdots), \\ b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin nx \, \mathrm{d}x & (n = 1, 2, 3, \cdots). \end{cases}$$

Proof. 设 $f(x)$ 是周期为 2π 的周期函数, 且能展开成三角级数

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx) .$$

先求 a_0. 上式一从 $-\pi$ 到 π 积分, 由于假设式右端级数可逐项积分，因此有

$$\int_{-\pi}^{\pi} f(x)\mathrm{d}x = \int_{-\pi}^{\pi} \frac{a_0}{2} \, \mathrm{d}x + \sum_{k=1}^{\infty} \left[a_k \int_{-\pi}^{\pi} \cos kx \, \mathrm{d}x + b_k \int_{-\pi}^{\pi} \sin kx \, \mathrm{d}x \right]$$

根据三角函数系的正交性，等式右端除第一项外，其余各项均为零，所以

$$\int_{-\pi}^{\pi} f(x)\mathrm{d}x = \frac{a_0}{2} \cdot 2\pi$$

于是得

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x)\mathrm{d}x$$

其次求 a_n. 用 $\cos nx$ 乘式两端, 再从 $-\pi$ 到 π 积分, 得到

$$\begin{aligned} &\int_{-\pi}^{\pi} f(x) \cos nx \, \mathrm{d}x \\ =&\frac{a_0}{2} \int_{-\pi}^{\pi} \cos nx \, \mathrm{d}x + \sum_{k=1}^{\infty} \left[a_k \int_{-\pi}^{\pi} \cos kx \cos nx \, \mathrm{d}x + b_k \int_{-\pi}^{\pi} \sin kx \cos nx \, \mathrm{d}x \right] \end{aligned}$$

根据三角函数系的正交性, 等式右端除 $k = n$ 的一项外, 其余各项均为 0. 所以

$$\int_{-\pi}^{\pi} f(x)\cos nx\,\mathrm{d}x = a_n \int_{-\pi}^{\pi} \cos^2 nx\,\mathrm{d}x = a_n\pi$$

于是得

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\cos nx\,\mathrm{d}x \quad (n = 1, 2, 3, \cdots)$$

类似地, 用 $\sin nx$ 乘 $(7-5)$ 式的两端, 再从 $-\pi$ 到 π 积分, 可得

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\sin nx\,\mathrm{d}x \quad (n = 1, 2, 3, \cdots)$$

■

Theorem 16.2.1 — 收敛定理, 狄利克雷 (Dirichlet) 充分条件. 设 $f(x)$ 是周期为 2π 的周期函数, 如果它满足:

1. 在一个周期内连续或只有有限个第一类间断点,
2. 在一个周期内至多只有有限个极值点

那么 $f(x)$ 的傅里叶级数收敛, 并且

- 当 x 是 $f(x)$ 的连续点时, 级数收敛于 $f(x)$;
- 当 x 是 $f(x)$ 的间断点时, 级数收敛于 $\frac{1}{2}\left[f\left(x^-\right)+f\left(x^+\right)\right]$.

收敛定理告诉我们: 只要函数在 $[-\pi,\pi]$ 上至多有有限个第一类间断点，并且不作无限次振动，函数的傅里叶级数在连续点处就收敛于该点的函数值，在间断点处收敛于该点左极限与右极限的算术平均值. 可见，函数展开成傅里叶级数的条件比展开成幂级数的条件低得多.

Corollary 16.2.2 记 $C = \left\{x \mid f(x) = \frac{1}{2}\left[f\left(x^-\right)+f\left(x^+\right)\right]\right\}$
在 C 上就成立 $f(x)$ 的傅里叶级数展开式

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty}\left(a_n\cos nx + b_n\sin nx\right), x \in C$$

Definition 16.2.2 — 正弦级数. 当 $f(x)$ 为奇函数时, $f(x)\cos nx$ 是奇函数, $f(x)\sin nx$ 是偶函数, 故

$$\left.\begin{aligned} a_n &= 0 \quad (n = 0, 1, 2, \cdots), \\ b_n &= \tfrac{2}{\pi}\textstyle\int_0^{\pi} f(x)\sin nx\,\mathrm{d}x \quad (n = 1, 2, 3, \cdots). \end{aligned}\right\}$$

即知奇函数的傅里叶级数是只含有正弦项的正弦级数.

$$\sum_{n=1}^{\infty} b_n \sin nx$$

Definition 16.2.3 — 余弦级数. 当 $f(x)$ 为偶函数时, $f(x)\cos nx$ 是偶函数, $f(x)\sin nx$ 是奇函数, 故

$$\left.\begin{aligned} a_n &= \tfrac{2}{\pi}\textstyle\int_0^{\pi} f(x)\cos nx\,\mathrm{d}x \quad (n = 0, 1, 2, \cdots) \\ b_n &= 0 \quad (n = 1, 2, 3, \cdots) \end{aligned}\right\}$$

即知偶函数的傅里叶级数是只含常数项和余弦项的余弦级数

$$\frac{a_0}{2}+\sum_{n=1}^{\infty}a_n\cos nx$$

在实际应用 (如研究某种波动问题, 热的传导、扩散问题) 中, 有时还需要把定义在区间 $[0,\pi]$ 上的函数 $f(x)$ 展开成正弦级数或余弦级数.

根据前面讨论的结果, 这类展开问题可以按如下的方法解决: 设函数 $f(x)$ 定义在区间 $[0,\pi]$ 上并且满足收玫定理的条件, 我们在开区间 $(-\pi,0)$ 内补充函数 $f(x)$ 的定义, 得到定义在 $(-\pi,\pi]$ 上的函数 $F(x)$, 使它在 $(-\pi,\pi)$ 上成为奇函数 𝕀(偶函数). 按这种方式拓广函数定义域的过程称为奇延拓 (偶延拓). 然后将奇延拓 (偶延拓) 后的函数展开成傅里叶级数, 这个级数必定是正弦级数 (余弦级数). 再限制 x 在 $(0,\pi]$ 上, 此时 $F(x)\equiv f(x)$, 这样便得到 $f(x)$ 的正弦级数 (余弦级数) 展开式.

Theorem 16.2.3 设周期为 $2l$ 的周期函数 $f(x)$ 满足收敛定理的条件, 则它的傅里叶级数展开式为

$$f(x)=\frac{a_0}{2}+\sum_{n=1}^{\infty}\left(a_n\cos\frac{n\pi x}{l}+b_n\sin\frac{n\pi x}{l}\right)(x\in C),\quad(8-1)$$

其中

$$\left.\begin{array}{l}a_n=\frac{1}{l}\int_{-l}^{l}f(x)\cos\frac{n\pi x}{l}\,\mathrm{d}x\quad(n=0,1,2,\cdots)\\ b_n=\frac{1}{l}\int_{-l}^{l}f(x)\sin\frac{n\pi x}{l}\,\mathrm{d}x\quad(n=1,2,3,\cdots),\\ C=\left\{x\mid f(x)=\frac{1}{2}\left[f\left(x^{-}\right)+f\left(x^{+}\right)\right]\right\}\end{array}\right\}$$

当 $f(x)$ 为奇函数时,

$$f(x)=\sum_{n=1}^{\infty}b_n\sin\frac{n\pi x}{l}\quad(x\in C)$$

其中

$$b_n=\frac{2}{l}\int_0^l f(x)\sin\frac{n\pi x}{l}\,\mathrm{d}x\quad(n=1,2,3,\cdots)$$

当 $f(x)$ 为偶函数时,

$$f(x)=\frac{a_0}{2}+\sum_{n=1}^{\infty}a_n\cos\frac{n\pi x}{l}(x\in C)$$

其中

$$a_n=\frac{2}{l}\int_0^l f(x)\cos\frac{n\pi x}{l}\,\mathrm{d}x\quad(n=0,1,2,\cdots)$$

Definition 16.2.4 — 傅里叶级数的复数形式.

$$\sum_{n=-\infty}^{\infty}c_n\mathrm{e}^{\frac{n\pi x}{l}j}$$

where $c_n=\frac{1}{2l}\int_{-l}^{l}f(x)\mathrm{e}^{-\frac{n\pi x}{l}j}\mathrm{d}x\quad(n=0,\pm1,\pm2,\cdots)$.

Proof. 设周期为 $2l$ 的周期函数 $f(x)$ 的傅里叶级数为

$$\frac{a_0}{2}+\sum_{n=1}^{\infty}\left(a_n\cos\frac{n\pi x}{l}+b_n\sin\frac{n\pi x}{l}\right)$$

其中系数 a_n 与 b_n 为

$$\left.\begin{aligned}a_n&=\frac{1}{l}\int_{-l}^{l}f(x)\cos\tfrac{n\pi x}{l}\,\mathrm{d}x\quad(n=0,1,2,\cdots),\\ b_n&=\frac{1}{l}\int_{-l}^{l}f(x)\sin\tfrac{n\pi x}{l}\,\mathrm{d}x\quad(n=1,2,3,\cdots).\end{aligned}\right\}$$

利用欧拉公式

$$\cos t=\frac{\mathrm{e}^{t\mathrm{i}}+\mathrm{e}^{-t i}}{2},\sin t=\frac{\mathrm{e}^{t\mathrm{i}}-\mathrm{e}^{-t\mathrm{i}}}{2\mathrm{i}}$$

记

$$\frac{a_0}{2}=c_0,\quad\frac{a_n-b_n\mathrm{i}}{2}=c_n,\quad\frac{a_n+b_n\mathrm{i}}{2}=c_{-n}\quad(n=1,2,3,\cdots)$$

则表示为

$$c_0+\sum_{n=1}^{\infty}\left(c_n\mathrm{e}^{\frac{n\pi x}{l}i}+c_{-n}\mathrm{e}^{-\frac{n\pi x}{l}i}\right)=\left(c_n\mathrm{e}^{\frac{n\pi x}{l}i}\right)_{n=0}+\sum_{n=1}^{\infty}\left(c_n\mathrm{e}^{\frac{n\pi x}{l}i}+c_{-n}\mathrm{e}^{-\frac{n\pi x}{l}i}\right)$$

$$c_0=\frac{a_0}{2}=\frac{1}{2l}\int_{-l}^{l}f(x)\mathrm{d}x$$

$$\begin{aligned}c_n&=\frac{a_n-b_n i}{2}\\&=\frac{1}{2}\left[\frac{1}{l}\int_{-l}^{l}f(x)\cos\frac{n\pi x}{l}\,\mathrm{d}x-\frac{\mathrm{i}}{l}\int_{-l}^{l}f(x)\sin\frac{n\pi x}{l}\,\mathrm{d}x\right]\\&=\frac{1}{2l}\int_{-l}^{l}f(x)\left(\cos\frac{n\pi x}{l}-\mathrm{i}\sin\frac{n\pi x}{l}\right)\mathrm{d}x\\&=\frac{1}{2l}\int_{-l}^{l}f(x)\mathrm{e}^{-\frac{n\pi}{l}}\,\mathrm{d}x\quad(n=1,2,3,\cdots);\end{aligned}$$

$$c_{-n}=\frac{a_n+b_n\mathrm{i}}{2}=\frac{1}{2l}\int_{-l}^{l}f(x)\mathrm{e}^{\frac{n\pi x}{l}i}\,\mathrm{d}x\quad(n=1,2,3,\cdots)$$

将已得的结果合并写为

$$c_n=\frac{1}{2l}\int_{-l}^{l}f(x)\mathrm{e}^{-\frac{n\pi x}{l}}\,\mathrm{d}x\quad(n=0,\pm1,\pm2,\cdots)$$

■

16.3 Fourier Transform

Definition 16.3.1 — The Fourier transform of a continuous function $f(t)$ of a continuous variable t.

$$F(\mu)=\Im\{f(t)\}=\int_{-\infty}^{\infty}f(t)e^{-j2\pi\mu t}dt$$

Using Euler's formula, we can write as

$$F(\mu) = \int_{-\infty}^{\infty} f(t)[\cos(2\pi\mu t) - j\sin(2\pi\mu t)]dt$$

傅里叶变换将一个复杂的波变换成不同的正弦波（基波），时频图（时域、频域、某一刻相位）。对于任意一个时域的曲线，均可以分解成为不同的基波的

$$\text{时域} f(t) \Leftrightarrow F(u,v) \text{频域}$$

Definition 16.3.2 — inverse Fourier transform.

$$f(t) = \int_{-\infty}^{\infty} F(\mu)e^{j2\pi\mu t}d\mu$$

16.4 Discrete Fourier Transform

Definition 16.4.1 — Discrete Fourier Transform.

$$F_m = \sum_{n=0}^{M-1} f_n e^{-j2\pi mn/M} \quad m = 0, 1, 2, \ldots, M-1$$

Definition 16.4.2 — inverse discrete Fourier transform (IDFT).

$$f_n = \frac{1}{M}\sum_{m=0}^{M-1} F_m e^{j2\pi mn/M} \quad n = 0, 1, 2, \ldots, M-1$$

Definition 16.4.3 — 离散傅里叶变换矩阵 W.

$$W = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ 1 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \cdots & \omega^{-(n-1)(n-1)} \end{bmatrix}$$

where $\omega = e^{2\pi j/n}$, $\quad j = \sqrt{-1}$

Corollary 16.4.1 矩阵 $(1/\sqrt{n})W$ 是酉矩阵:

$$\frac{1}{n}W^H W = \frac{1}{n}WW^H = I$$

Corollary 16.4.2 W 的逆 $W^{-1} = (1/n)W^H$ 。

Corollary 16.4.3 W 的共轭转置为

$$W^H = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{bmatrix}$$

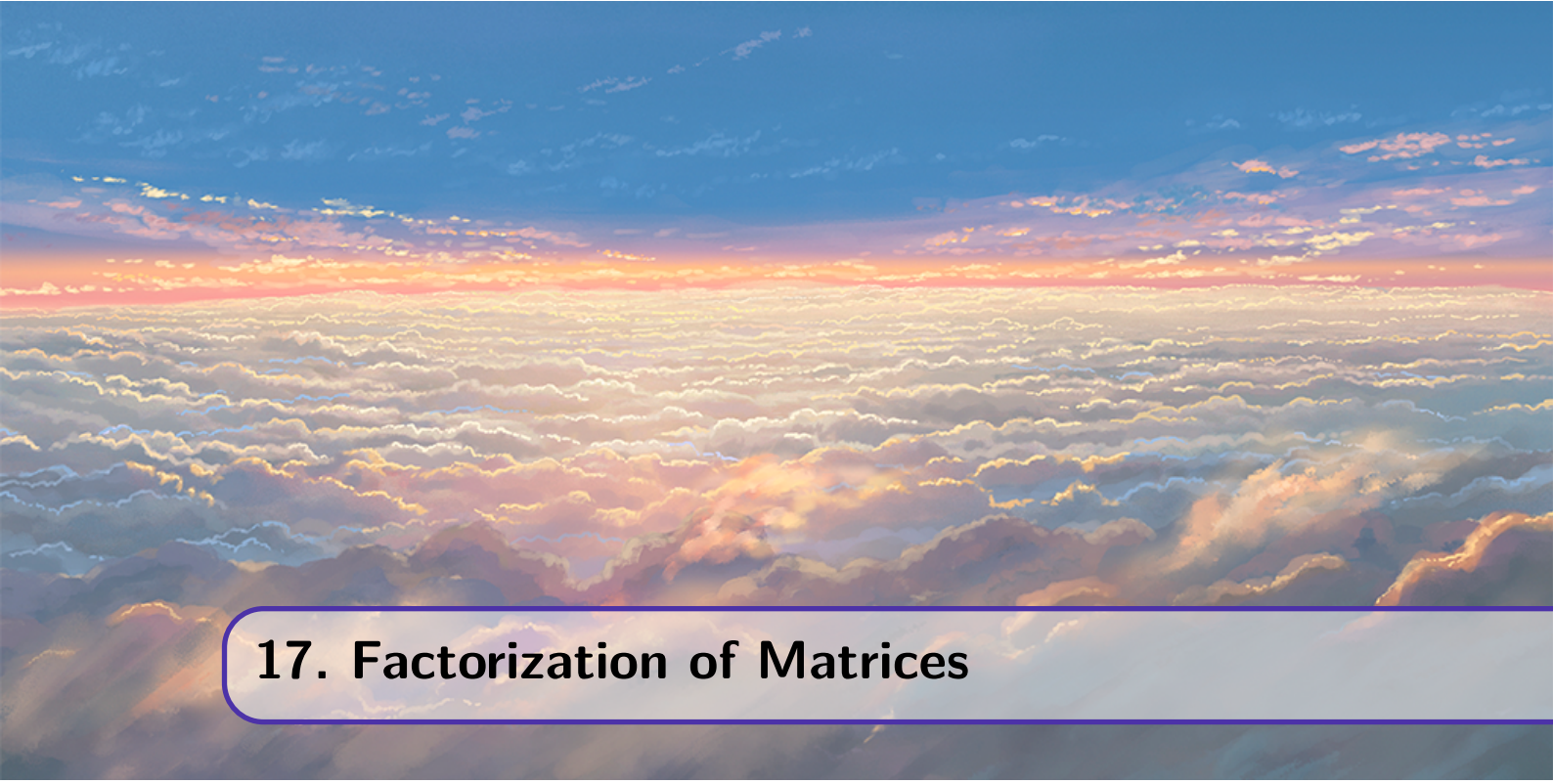
Corollary 16.4.4 Gram 矩阵的第 i, j 个元素为

$$\left(W^H W\right)_{ij} = 1 + \omega^{i-j} + \omega^{2(i-j)} + \cdots + \omega^{(n-1)(i-j)}$$

$\left(W^H W\right)_{ii} = n, \left(W^H W\right)_{ij} = \frac{\omega^{n(i-j)}-1}{\omega^{i-j}-1} = 0$ 如果 $i \neq j$
最后一步因为 $\omega^n = 1$ 。

Definition 16.4.4 — 离散傅里叶反变换. n 维向量 x 的离散傅里叶反变换是

$$W^{-1}x = (1/n)W^H x$$



17. Factorization of Matrices

17.1 主要的矩阵分解

$$A = LU$$

$$A = LPU$$

$$A = QR$$

$$A = X\Lambda X^{-1}$$

$$S = Q\Lambda Q^T$$

$$A = U\Sigma V^T$$

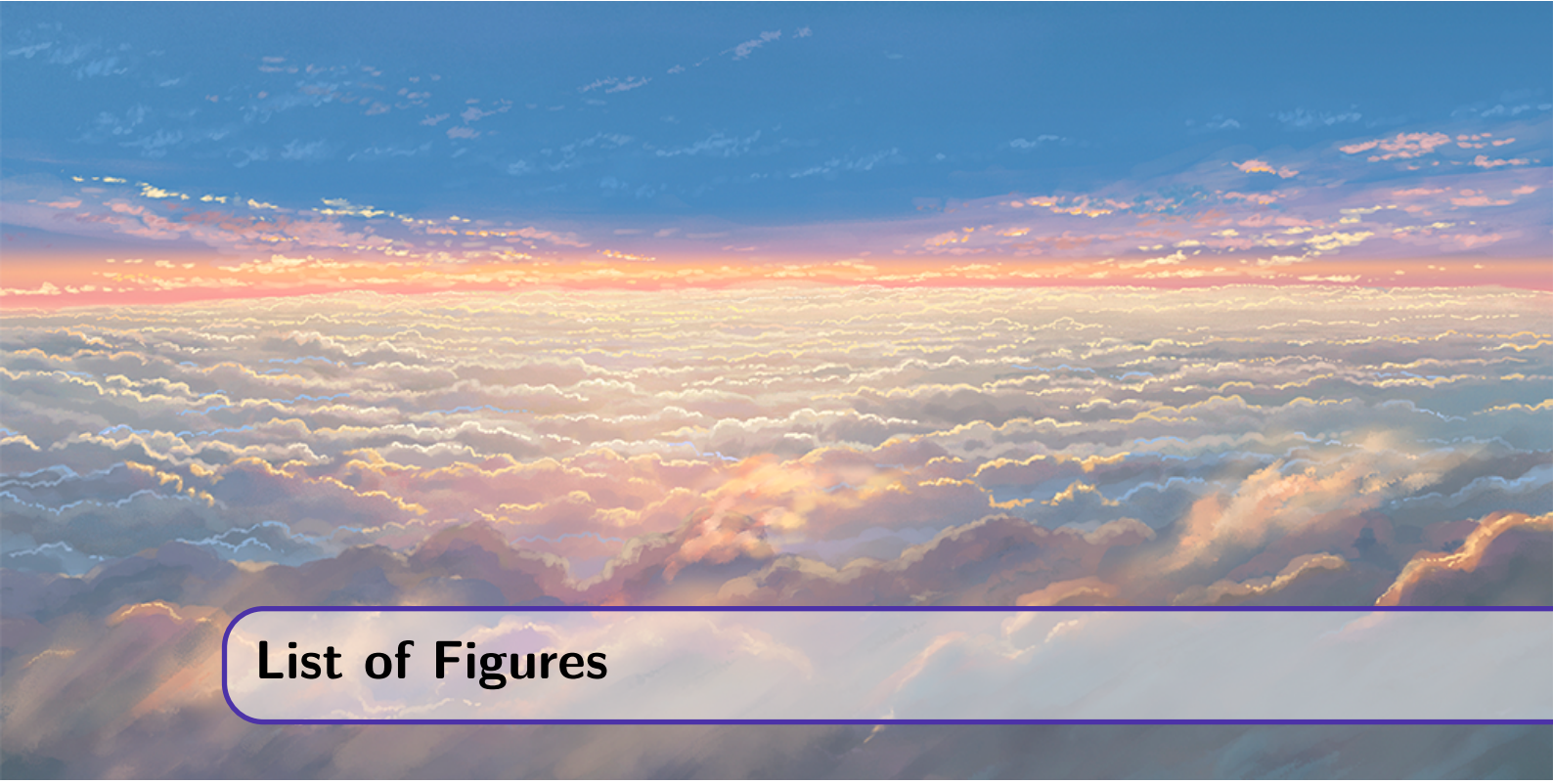
$$A = CMR$$

17.2 $A = LU$

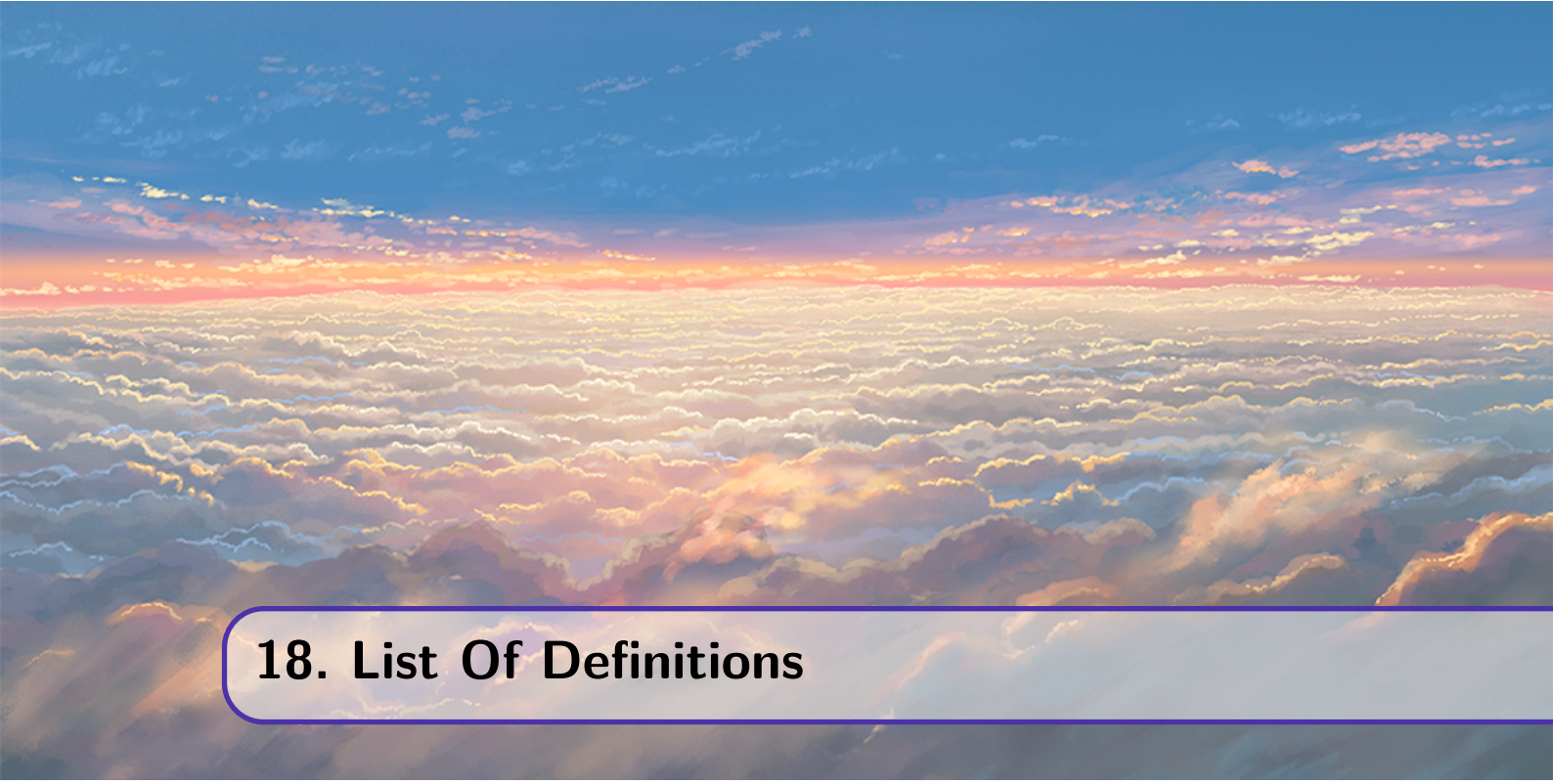
$$(E_{32}E_{31}E_{21})\,A = U \text{ becomes } A = \left(E_{21}^{-1}E_{31}^{-1}E_{32}^{-1}\right)U \quad \text{which is} \quad A = LU$$

高斯消元法将 A 变成上三角矩阵，等价于左乘以一系列的初等矩阵。

Problem 17.1 ∎



List of Figures



18. List Of Definitions