

恰定方程组指系数矩阵秩与其阶数相同，且与增广矩阵秩相同的方程组。它有且仅有一个解向量。

它的一般形式是

$$Ax = b$$

我们需要通过某种手段解出符合上述条件的 x 。

Cramer法则

我们将上述列向量 b 替换 A 中的第 i 个列向量，把产生的新矩阵记作 A_i ，那么解向量的第 i 项：

$$x_i = \frac{\det(A_i)}{\det(A)} \tag{2.1}$$

式(2.1)被称为Cramer法则。它要求系数矩阵 A 是满秩的。

证明

对于满秩的系数矩阵 A ，我们可以有如下推导

$$Ax = b \tag{2.2-1}$$

$$x = A^{-1}b \tag{2.2-2}$$

由伴随矩阵的定义：

$$x = \frac{A^*b}{\det(A)} \tag{2.2}$$

而由矩阵运算法则，解向量 x 的第 i 个分量可以描述为：

$$x_i = \sum_{j=0}^n A_{ij}^* b_j \tag{2.3-1}$$

而 A_{ij}^* 是 a_{ji} (请注意 i 和 j 的顺序)的代数余子式。

故由行列式的Laplace展开运算，不难得到

$$x_i = \sum_{j=0}^n A_{ij}^* b_j = A_i \tag{2.3}$$



复杂度

一个 n 阶行列式具有 $n!$ 个Laplace展开项，每一个展开项由 n 项相乘，故计算单个 n 阶行列式需要 $n!(n - 1)$ 次乘法运算。而对于 n 阶系数矩阵，Cramer法则要求计算 $(n + 1)$ 个矩阵。故最终的计算复杂度为 $(n + 1)!(n -$

1), 过大, 不可接受.

是什么深层次的原因导致了Cramer法则具有如此的计算复杂度呢?

Gauss消元法

Gauss消元法的核心任务是, 对于增广矩阵 $[A, b] \in \mathbb{R}^n$, 执行 n 次消元, 第 i 次消元通过基本矩阵变换将 a_{ii} 下方的所有元素归零.

简单Gauss消元法

数学语言描述:

算法2.1:

对于 $A = [A, b] \in \mathbb{R}^n$ 的第1至 $(n - 1)$ 行向量 $A_i (i < n)$, 重复执行下述步骤:

for $j = i + 1, j \leq n$, 令

$$[A_j, b_j] \leftarrow [A_j, b_j] - \frac{a_{ji}}{a_{ii}} [A_i, b_i] \quad (2.4)$$

我们不需要对最后一列元素进行消元, 因此使用了“第1至 $(n - 1)$ 行”的提法. 经过这样的消元, 系数矩阵化为了上三角矩阵, 可以立即解出 x_n , 从而逐个回代求解出解向量. 最终的解公式太过于啰嗦丑陋, 且推导难度不大, 故不在此重复.



复杂度分析

根据上述算法, 对于第 i 次执行来讲, 赋值表达式(2.4)中首先执行了1次 $\frac{a_{ji}}{a_{ii}}$, 而行向量中共有 $(n - i + 1)$ 个非零元素, 故每次执行赋值表达式(2.4)均要求 $(n - i + 2)$ 次乘法运算. 对于每一个行向量 i , 赋值表达式(2.4)均执行 $(n - i)$ 次, 故Gauss消元法的复杂度可以表述为:

$$\sum_{i=1}^n (n - 1)(n - 1 + 2) = \frac{n^3}{3} + n^2 - \frac{n}{3}$$

失效情况

我们在之前提到了, 实际计算中应尽量避免小数除大数, 而上述的算法实际上无法避免这种情况, 可能在某些矩阵的解决过程中会有某一极小的数作主元的情况发生. 故为了避免这样的问题, 我们提出了如下的改进算法:

(列)主元素法

(列)主元素法在算法2.1执行过程中，每面对一个新的 i 值，它都会遍历矩阵的第 i 列，将拥有绝对值最大的第 i 列元素的行向量替换至第 i 行。这样的算法附加一个 $O(n)$ 开销，但可以很大程度上保证算法的数值稳定。

全主元素法

继续改进，在在算法2.1执行过程中，每面对一个新的 i 值，它都会遍历整个矩阵，将拥有绝对值最大的元素的行向量替换至第 i 行，列向量替换至第 i 列，同时记录这个列变换，便于修正解向量的顺序。这样的算法附加一个 $O(n^2)$ 开销，但可以极大程度上保证算法的数值稳定，是求解中小型稠密线性方程组的最优方法之一。

在不少应用场景中，系数矩阵 A 是保持不变的，而方程右值 b 来源于输入值，是时刻变化的。因此在这种情境下，使用简单Gauss消元法及其衍生方法就会产生复用性低的问题：面对每一个新的 b ，我们都需要重新执行一遍完全一样的Gauss消元步骤，这是愚蠢的。故我们试图找到某一种方式来记录Gauss消元法的步骤，我们可以将步骤记录在某种数据结构中，但这种方式太过于随意，也太过于工程化，故难以与理论的误差分析等兼容，故我们提出了结构化描述Gauss消元步骤的方式，这种方式将实践的具体情况剔除，将上述的“记录”抽象为理论。

分解法

用基本初等矩阵刻画线性变换

在高等代数中我们知道，任何一个基本矩阵变换均可以被一个基本初等矩阵刻画。假使我们有基本初等矩阵 $L \in \mathbb{R}^n$ ，矩阵 $A \in \mathbb{R}^n$ ，那么 LA 代表对 A 执行一系列初等行变换，而 AL 代表对 A 执行一系列初等列变换。

在上述的情况下， L 对 A 执行的变换操作相当于将单位矩阵 E 变换为 L 所需的行或列操作。

例2.1 设矩阵

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \quad A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad \text{求} LA \text{和} AL.$$

解答：从单位矩阵 E 变换为 L ，若是行变换，需“将第二行的1倍加到第三行上”；若是列变换，需执行“将第三列的1倍加到第二列上”。故 LA 可以视为“将 A 的第二行的1倍加到第三行上”，故

$$LA = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 2 & 2 & 2 \end{pmatrix}$$

同理， AL 可以视为“将 A 的第三列的1倍加到第二列上”，故

$$AL = \begin{pmatrix} 1 & 2 & 1 \\ 1 & 2 & 1 \\ 1 & 2 & 1 \end{pmatrix}$$

Doolittle分解

我们在之后的讨论中，为了方便，将彻底不考虑增广矩阵，只考虑系数矩阵。

因此，我们可以通过一个基本初等矩阵 L ，将上述对系数矩阵 A 执行的一系列消元法结构化地描述。根据算法2.1，面对行向量 A_i ，我们执行的操作可以刻画为：

$$A \leftarrow L_i A$$

其中：

$$L = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & -l_{i+1,i} & 1 & \dots & 0 \\ 0 & 0 & \dots & -l_{i+2,i} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -l_{n,i} & 0 & \dots & 1 \end{pmatrix}$$

$$\text{其中, } l_{k,i} = \frac{a_{k,i}}{a_{i,i}}$$

因此，算法2.1的一系列操作可以记作：

$$A \leftarrow L_{n-1} L_{n-2} \dots L_1 A \quad (2.5')$$

用新的符号重写上式以避免在下述推导中出现歧义：

$$U = L_{n-1} L_{n-2} \dots L_1 A \quad (2.5)$$

左乘移项：

$$L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1} U = A \quad (2.6)$$

而上述的各个 L_i 有着很好的运算性质(实际上，与线性变换的性质息息相关)：

$$\text{在 } L = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & -l_{i+1,i} & 1 & \dots & 0 \\ 0 & 0 & \dots & -l_{i+2,i} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -l_{n,i} & 0 & \dots & 1 \end{pmatrix} \text{ 时, } L^{-1} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & l_{i+1,i} & 1 & \dots & 0 \\ 0 & 0 & \dots & l_{i+2,i} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & l_{n,i} & 0 & \dots & 1 \end{pmatrix}$$

$$\text{且在 } L_1 = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & -l_{i+1,i} & 1 & \dots & 0 \\ 0 & 0 & \dots & -l_{i+2,i} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -l_{n,i} & 0 & \dots & 1 \end{pmatrix}, L_2 = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 0 & -l_{i+2,i+1} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & -l_{n,i+1} & \dots & 1 \end{pmatrix}$$

时，有

$$L_1 L_2 = L = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & -l_{i+1,i} & 1 & \dots & 0 \\ 0 & 0 & \dots & -l_{i+2,i} & -l_{i+2,i+1} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -l_{n,i} & -l_{n,i+1} & \dots & 1 \end{pmatrix}$$

故我们可以根据上述的运算性质，将式(2.6)的所有 L_i 项合并为一个 L 项：

$$LU = A \quad (2.7)$$

其中：

$$L = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ l_{2,1} & 1 & \dots & 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ l_{3,1} & l_{3,2} & \dots & 1 & 0 & \dots & 0 \\ l_{4,1} & l_{4,2} & \dots & l_{i+1,i} & 1 & \dots & 0 \\ l_{5,1} & l_{5,2} & \dots & l_{i+2,i} & l_{i+2,i+1} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ l_{n,1} & l_{n,2} & \dots & l_{n,i} & l_{n,i+1} & \dots & 1 \end{pmatrix}$$

是一个单位下三角矩阵。而Gauss消元的结果 U 是一个上三角矩阵，故我们可以藉由Gauss消元的原理，将矩阵 A 分解为一个单位下三角矩阵和一个一般上三角矩阵的积。这种分解被称为Doolittle分解。

有效条件

正如我们在所有工程学、自然科学和形式科学中所作的工作一样，在提出一个新的工具之后，我们需要关注一个问题：在什么情况下，这种工具能够正常发挥作用？

我们之前提到，Doolittle分解是使用基本初等矩阵来记录Gauss消元的步骤，因此Gauss消元与Doolittle分解理应是等价的。因此，适用Gauss消元的矩阵也理应适用Doolittle分解。

定理2.1： 设 $A \in \mathbb{R}^n$ ，若 A 的1至 $(n-1)$ 阶顺序主子式 A_i 均非零，则矩阵 A 存在唯一的Doolittle分解。

这条定理是直观的：在执行消元时，我们需要一个非零的主元 a_{ii} 来执行消元操作。而易证明，经过先前的消元操作之后，主元 a_{ii} 非零当且仅当主子式矩阵 A_i 是满秩的。分析的证明难度不大，请见讲义。

我们应当注意到，定理2.1中的条件(1至 $(n-1)$ 阶顺序主子式 A_i 均非零)弱于方程可解条件(系数矩阵 A 满秩)。这一差距的来源在于，我们在执行消元的过程中，不需要对最后一列元素执行消元操作，因此，最后一个主元 a_{nn} 是否非零并不重要。这表明，可以执行Doolittle分解是矩阵可解的必要非充分条件。这也表示着，Doolittle分解已将消元法从解方程的实际情景中彻底地抽象了出来。

LC的计算方法

至此，我们提出了Doolittle分解的来源、分析定义和有效条件，是时候来讨论其具体的计算和应用方式了。

考察Doolittle分解的定义式：

$$LU = A \quad (2.7)$$

其中, L 是一个单位下三角矩阵, U 是一个上三角矩阵. 故将上式表示如下:

$$\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{2,1} & 1 & 0 & \dots & 0 \\ l_{3,1} & l_{3,2} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ l_{n,1} & l_{n,2} & l_{n,3} & \dots & 1 \end{pmatrix} \begin{pmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \dots & u_{1,n} \\ 0 & u_{2,2} & u_{2,3} & \dots & u_{2,n} \\ 0 & 0 & u_{3,3} & \dots & u_{3,n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & u_{n,n} \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ a_{3,1} & a_{3,2} & a_{3,3} & \dots & a_{3,n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & a_{n,3} & \dots & a_{n,n} \end{pmatrix} \quad (2.8)$$

由矩阵的乘法法则:

$$a_{ij} = \sum_{k=0}^n l_{ik} u_{kj} \quad (2.9)$$

根据 L 和 U 矩阵中1和0的分布, 我们将式(2.9)的计算简化(实际上, 简化之后的形式变得非常丑陋)

$$\begin{cases} a_{ij} = u_{ij}, i = 1 \\ a_{ij} = \sum_{k=1}^{i-1} l_{ik} u_{kj} + u_{ij}, j \leq i, i > 1 \\ a_{ij} = \sum_{k=0}^n l_{ik} u_{kj}, j < i, i > 1 \end{cases} \quad (2.10)$$

由此可以反解出

$$\begin{cases} u_{1j} = a_{1j} \\ u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}, j \leq i, i > 1 \\ l_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}}{u_{jj}}, j < i, i > 1 \end{cases} \quad (2.11)$$

实际的计算顺序可以简单描述为"逐层计算, 先行后列". 实际的计算步骤见讲义, 看上去吓人但难度偏低(张宇纸老虎.gif), 按照上述公式实际操作一遍即可学会.

在解方程中应用Doolittle分解

假设我们通过上述繁杂而无趣的计算方法求出了系数矩阵 A 的Doolittle分解:

$$LU = A \quad (2.7)$$

将其回代到方程组 $Ax = b$ 中:

$$LUx = b$$

将向量 Ux 表示为 y , 有

$$\begin{cases} Ly = b \\ Ux = y \end{cases}$$

相当于解两个已经过消元的恰定(如果系数矩阵 A 是满秩的)方程组.

至于课本中描述的紧凑格式解方程组, 可以认为是一种类似于小学数学"列竖式计算"的内容, 重要性偏低.

而所谓追赶法解三对角方程, 即是上述 L 和 U 矩阵中多出了很多个0, 在式(2.11)中多代入几个0即可. 不是很理解为什么还要将其单列一节.

开销分析

由于基本原理相同, Doolittle分解法与Gauss消元法具有相同的运算量, 但Doolittle分解法有着复用性的优势.

Cholesky分解

在实践中, 有相当一部分线性求解问题应用到了正定二次型的系数矩阵, 而正定二次型具有更优的性质, 可能可以简化求解运算. 因此我们有必要针对正定二次型来提出一种新的算法.

正定二次型

若一个矩阵满足 $A = A^T$, 则将其称为对称矩阵, 而术语“二次型”常被用于提及一个对称矩阵.

当一个二次型 A 满足以下四个等价条件之一:

1. $\forall x \neq 0, x^T A x > 0$
2. $\forall A$ 的特征值 $\lambda, \lambda > 0$
3. A 的所有顺序主子式均大于0
4. $C \in \mathbb{R}^n, A = C^T C$

则 A 被称为正定二次型. 上述四个条件均为命题" A 是正定二次型"的充分必要条件

平方根法

由上述性质4: $C \in \mathbb{R}^n, A = C^T C$, 我们知道, 一个正定二次型可以被分解为一个实矩阵和其转置的积.

定理2.2: 若 A 属 $n \times n$ 正定二次型, 则存在唯一的可逆下三角矩阵 C , 使得

$$A = C C^T \quad (2.12)$$

其中, C 的对角元素均为正实数.

证明:

由定理2.1, 存在 $LU = A$, 其中 L 是下三角单位矩阵, U 为上三角矩阵. 我们将 U 的对角线元素抄到矩阵 D 中, 即记矩阵 $D = \text{diag}(u_{11}, u_{22}, \dots, u_{nn})$, 再记矩阵 $P = D^{-1}U$, 则 $A = LDP$. 请注意, **此时的 P 之对角元素全部为1**. 又 A 属二次型, D 属对角矩阵, 故:

$$LDP = A = A^T = P^T D^T L^T = P^T D L^T$$

而, 定理2.1指出, 同一个矩阵的Doolittle分解是唯一的. 因此不难得到 $L = P^T$, 即

$$A = LDP = P^T DP \quad (2.13)$$

接下来，我们试图将对角矩阵 D 拆分为形如 $F^T F$ 的形式。但这要求着 F 也为对角矩阵，且 D 的对角线元素值为 F 相应对角线元素值的平方，这就要求 D 的对角线元素均为正。因此证明：

由正定性质：任取一非零列向量 $x \in \mathbb{R}^{1 \times n}$ ，有

$$x^T A x = x^T P^T D P x > 0$$

这种形式表示： D 也是一个正定二次型，而 D 是一个对角阵，对角阵的对角元素就是它的特征值。因此 D 的对角线元素为正。

故 $D = F^T F$, $F = \text{diag}(\sqrt{u_{ii}})$ 。即

$$A = P^T D P = P^T F^T F P = (F P)^T F P = C C^T$$

易知 FP 是上三角矩阵，故 $C = (FP)^T$ 是一个下三角矩阵。

■

唯一性的证明是容易的，请见讲义。

计算方法

Cholesky分解的计算原理与Doolittle分解一致，均是应用矩阵乘法法则。直接把结果贴在此处。

$$\begin{cases} c_{kk} = (a_{kk} - \sum_{j=1}^{k-1} c_{kj}^2)^{1/2} \\ c_{ik} = (a_{ik} - \sum_{j=1}^{k-1} c_{ij} c_{kj})^{1/2}, i > k \end{cases} \quad (2.11)$$

完成Cholesky分解后的解方程步骤与Doolittle分解一致。

容易得知，Cholesky分解应用了对称性，故计算复杂度为Gauss消元法的一半，为 $O(n^3/6)$ 。

但是，Cholesky分解法调用了平方运算，可能会导致运算时间的延长。

改进的平方根法

在Cholesky分解的证明过程中，我们提到：

$$A = L D P = L D L^T \quad (2.13)$$

其中， L 是单位下三角矩阵， D 为对角矩阵。

因此我们可以将矩阵 A 作 LDL^T 分解以避免平方根运算。

计算方法

改进的平方根法在求矩阵 L 和 D 的过程中采用了与Doolittle分解完全一致的方式：首先将 LD 组合为一个新的矩阵 U 处理，再通过与Doolittle分解完全一致的计算方式得到如下的计算式：

$$\begin{cases} d_k = a_{kk} - \sum_{j=1}^{k-1} l_{kj}^2 d_j \\ l_{ij} = (a_{ik} - \sum_{j=1}^{k-1} l_{ij} d_j l_{kj}) / d_k, \quad k \leq i \end{cases} \quad (2.14)$$

如果对式子的形式有着充分的观察力，会发现式(2.14)与(2.11)有着完全一致的形式(2.11的第二条由于对称性被简化掉了)。

这样的运算式尚未把正定二次型的性质充分发挥，故提出了一个使用辅助量 u_{ik} $SS^T = L D L^T$ (实际上就是将 LD 的积 U 中的元素显式地记录了下来)的计算式，见课本。

使用改进的平方根法解方程的步骤与上述两方法稍有出入：

$$Ax = b$$

$$LDL^T x = b$$

$$Ly = b, L^T x = D^{-1}b$$

针对矩阵的误差分析

直观地，误差分析首先要求我们能够确定且统一地描述误差(即测定值与实际值之间的"距离")，而单靠矩阵显然做不到这一点，因此我们借用了范数这一工具。

度量方法：范数

范数实际上是实分析中的一条重要概念，是长度概念的推广，我们在此将其简化描述。

假设有一线性空间 \mathbb{D} ，存在某一函数 $\|\cdot\|$ 将 C 映射至非负实数空间 \mathbb{R}^* ，且这个函数满足：

1. $\|x\| = 0$ 当且仅当 $x = 0$ (正定性)
2. $\forall x \in \mathbb{D}, \alpha \in \mathbb{R}, \|\alpha x\| = |\alpha| \|x\|$ (绝对齐次性)
3. $\forall x, y \in \mathbb{D}, \|x + y\| \leq \|x\| + \|y\|$ (三角不等式)

则这一函数被称为 \mathbb{D} 上的一个**范数**，二元体 $(\mathbb{D}, \|\cdot\|)$ 被称为一个赋范线性空间或Banach空间。

一句话描述：**范数是某种距离的刻画**。

我们称具有如下性质的范数是等价的：

在一线性空间 \mathbb{D} 上的两个范数 $\|\cdot\|_1$ 和 $\|\cdot\|_2$ ，如 $\forall x \in \mathbb{D}, \exists M, m \in \mathbb{R}, M > m > 0$ ，有：

$$m\|x\|_2 \leq \|x\|_1 \leq M\|x\|_2$$

则称范数 $\|\cdot\|_1$ 和 $\|\cdot\|_2$ 等价。

这种等价性给出了一种很好的性质：等价范数之间只差一个常数倍，因此，想要得到向量的某种性质，无论用哪种范数来估计，都可以获得相同的结果。

向量范数

针对于 $1 \times n$ 向量空间的常用范数有：

- 1. 1-范数： $\|x\|_1 = \sum_{i=1}^n x_i$
- 2. 2-范数： $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$
- 3. ∞ -范数： $\|x\|_\infty = \max x_i$

所有的向量范数均是等价的(证明请参见任一实分析教材，如陈建功《实函数论》)

矩阵范数

一般来讲，在矩阵空间上的某一函数 $\|\cdot\|$ 想要被称之为范数，还需要附加一个相容性条件：

$$\|AB\| \leq \|A\|\|B\|$$

诱导范数

若 $\|x\|$ 属 $1 \times n$ 向量空间上的向量范数， A 是一 $n \times n$ 矩阵，则

$$\|A\|_m = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}$$

则 $\|A\|_m$ 是一矩阵范数，称为诱导范数或算子范数。

证明属实分析内容，略。

由上述向量范数可以诱导出矩阵范数：

- 1. 1-范数：最大的列总和
- 2. 2-范数(谱范数)：矩阵的最大奇异值之平方根
- 3. ∞ -范数：最大的行总和

所有的矩阵范数均是等价的

矩阵的误差定义

提出了范数工具，我们即可定义矩阵的误差：

记测量值为 x^* ，真实值为 x ，则 $\|x - x^*\|$ 为向量的绝对误差， $\frac{\|x - x^*\|}{\|x^*\|}$ 为向量的相对误差。

记测量值为 A^* ，真实值为 A ，则 $\|A - A^*\|$ 为矩阵的绝对误差， $\frac{\|A - A^*\|}{\|A^*\|}$ 为矩阵的相对误差。

方程组的条件数

在某些情况下，方程组 $Ax = b$ 中，若在右端 b 施加一小扰动 δb ，其解 x 会获得一个相当大的扰动 δx ，这种方程组被称为病态的。

为了定量刻画方程组的病态程度，我们讨论上述情况：

$$A(x + \delta x) = b + \delta b \quad (2.15)$$

试图得到其解的相对变化 $\frac{\|\delta x\|}{\|x\|}$ 与右值的相对变化 $\frac{\|\delta b\|}{\|b\|}$ 的相关关系，从而找出其病态的根源：

由(2.15)得到 $A\delta x = \delta b$ ，而 A 是可逆矩阵，故：

$$\|\delta x\| = \|A^{-1}\delta b\| \leq \|A^{-1}\| \|\delta b\|$$

又 $\|Ax\| \leq \|A\|\|x\|$ ，得到

$$\|x\| \geq \frac{\|Ax\|}{\|A\|} = \frac{\|b\|}{\|A\|}$$

故：

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\|A^{-1}\| \|\delta b\|}{\frac{\|b\|}{\|A\|}} = \frac{\|A^{-1}\| \|A\| \|\delta b\|}{\|b\|}$$

故我们发现 $\|A^{-1}\| \|A\|$ 控制着这种扰动的大小，这个量被称为条件数，描述为 $\text{cond}(A)$ 。直观地，条件数的大小与所用范数有关。

性质

条件数有如下性质：

1. 任意方阵的条件数大于1
2. $\forall \alpha \neq 0 \in \mathbb{R}, \text{cond}(\alpha A) = \text{cond}(A)$
3. 正交矩阵的2-范数条件数为1

在如下情况下，方阵的条件数会变得很大：

1. 有两行/列向量非常相近时(这种问题通常被称为多重共线性问题)
2. 有某一主元有着较小的绝对值时
3. 各行/列向量元素的数量级差距较大时

2021.10.18

Hautbois