



矩阵的运算

调用函数

在上一章中，我们介绍了常见的数学运算函数，例如abs,sin,round,log等。事实上，这些函数可以直接应用到矩阵上面，所表示的含义是：对矩阵中的每个元素分别运用这些数学运算函数，因此返回的结果也是一个矩阵。

- 使用频率较高的函数
- sum 求和函数
  - prod 求乘积函数 (product)
  - cumsum 计算累积和 (cumulative sum)
  - diff 计算差分 (difference)
  - mean 计算平均值
  - median 计算中位数
  - mode 计算众数
  - var 计算方差 (variance)
  - std 计算标准差 (standard deviation)
  - min 求最小值 (minimum value)
  - max 求最大值 (maximum value)

算术运算

+ (加)、- (减) 在线性代数中，只有两个大小完全相同的矩阵才可以进行相加相减，而在MATLAB中，只要两个矩阵的大小兼容，就能够进行计算。

情形	示例：计算 A+B		计算结果	解释
两个大小完全相同的输入	A =	B =	ans =	将 A 和 B 对应位置的元素相加
	6 5 6 2 9 2	7 8 9 8 6 2	13 13 15 10 15 4	
有一个输入是标量（常数）	A =	B =	ans =	矩阵的每个元素都加上这个标量
	2 1 3 7 2 4	4	6 5 7 11 6 8	
一个输入是矩阵，另一个输入是具有相同行数的列向量	A =	B =	ans =	把 B 堆叠成完全相同的三列，然后再和 A 相加 相当于 repmat(B,1,3)
	3 6 5 2 6 8	6 5	9 12 11 7 11 13	
一个输入是矩阵，另一个输入是具有相同列数的行向量	A =	B =	ans =	把 B 堆叠成完全相同的两行，然后再和 A 相加 相当于 repmat(B,2,1)
	3 5 6 6 9 4	3 9 6	6 14 12 9 18 10	
一个输入是列向量，另一个输入是行向量。	A =	B =	ans =	把 A 堆叠成完全相同的三列，把 B 堆叠成完全相同的两行，然后相加 相当于 repmat(A,1,3) repmat(B,2,1)
	2 5	1 8 3	3 10 5 6 13 8	

五种兼容模式

- 矩阵的乘法
- 矩阵的乘法：使用的运算符是乘号“\*”，例如矩阵A\*B，矩阵的乘法必须要满足前面矩阵A的列数和后面矩阵B的行数相等
  - 对应位置的元素相乘：使用的运算符是点乘“.\*”，例如A.\*B，这时A和B的大小只需兼容就能运算。
- 矩阵的除法
- 命令“x = B/A”表示对线性方程组x\*A = B求解x
  - 命令“x = A\B”则表示对线性方程组A\*x = B求解x
  - 大家只需要掌握“A/B”的用法即可 用A的每个元素除以B的对应元素，A和B的大小必须兼容
- 矩阵的乘方
- “^”表示矩阵的幂运算，例如A是一个方阵，那么A ^ 3等价于A\*A\*A
  - A^(-1)可用来计算A的逆矩阵，等价于inv(A)
  - “.^”表示对矩阵中的每一个元素分别进行乘方计算，例如A.^ 0.5表示对矩阵A中的每一个元素开根号。
- 矩阵的转置
- 矩阵的转置符号为英文的单引号：'，它也可以在前面加上点变成.'，两者的区别在于对矩阵中复数的处理，使用'会在转置的同时将复数变为共轭复数（实部不变虚部反号），使用.'则会保持原来的复数。
  - transpose(A)等价于A.'

关系运算

- Matlab中的关系运算符有六个： ==等于 ~=不等于 >大于 >=大于等于 <小于 <=小于等于
- 关系运算符用来比较两个数组中的元素，并返回逻辑值1(true)或0(false)来指示关系是否成立。
- 算术运算中介绍的兼容模式在关系运算中也支持
  - NaN(不定值或缺失值)相互之间不相等。
- 易错点：连续使用关系运算符
- 0 == 0 == 0 % 返回 logical 0 (false)
  - 1 == 1 == 1 % 返回 logical 1 (true)
  - 1 < 0 < 1 % 返回 logical 0 (false)
  - 1 > 0 > -1 % 返回 logical 1 (true)
- 我们应该使用下一小节中介绍的逻辑运算函数（逻辑与&、逻辑或|、逻辑非~等）连接多个关系表达式，例如判断a<b<c对应的命令应为(a<b)&(b<c)
- 易错点：浮点数的比较
- 解决方法：使用容差 tol（而不是使用 ==）比较浮点数。
  - 例如：要比较A和B两个数是否相等，只需要满足：|A-B|<=tol，这里的tol是一个非常非常小的正数，例如tol可以取成1e-12(10的-12次方)。tol越小，要求越严格。

逻辑运算

- MATLAB中使用逻辑值表示布尔变量，逻辑值1代表真 (true)，逻辑值0代表假 (false)。而逻辑运算就是对逻辑值进行的运算。
- 四种逻辑运算方法
- | 运算方法 | 函数名 | 运算符 | 运算规则（针对逻辑值）              | 示例   |
|------|-----|-----|--------------------------|--|
| 逻辑与  | and | &   | 都为 1 时返回 1，只要有一个是 0 返回 0 | and(1,1) % 返回 1<br>and(1,0) % 返回 0<br>and(0,0) % 返回 0<br>1 & 1 % 返回 1 1 & 0 % 返回 0<br>0 & 0 % 返回 0 |
| 逻辑或  | or  |     | 只要有一个为 1 返回 1，都是 0 时返回 0 | or(1,1) % 返回 1<br>or(1,0) % 返回 1<br>or(0,0) % 返回 0<br>1   1 % 返回 1 1   0 % 返回 1<br>0   0 % 返回 0    |
| 逻辑非  | not | ~   | 原来为 1 时返回 0，为 0 时返回 1    | not(1) % 返回 0<br>not(0) % 返回 1<br>~1 % 返回 0<br>~0 % 返回 1   |
| 逻辑异或 | xor | 无   | 不相同时取 1，相同时取 0           | xor(1,0) %返回 1<br>xor(1,1) %返回 0   |
- 三个注意点
- MATLAB推荐大家直接使用运算符进行计算，因此&、|和~这三个符号的功能大家要牢记
  - “逻辑与&”是对数组A和B进行计算的，计算时会比较A和B对应位置的元素。数组A和B的大小不一定要完全相同，只需要符合算术运算中介绍的五种兼容模式即可。
  - “逻辑与&”不仅可以作用在逻辑值0和1上，还可以用于普通的数值上，这时候，MATLAB会将非零数值视为逻辑1，将数值零视为逻辑0进行运算。例如：3&5返回逻辑值1，-4&0返回逻辑值0。
- logical函数：
- L = logical(A) 将 A 转换为一个逻辑值数组。A 中的任意非零元素都将转换为逻辑值 1 (true)，零则转换为逻辑值 0 (false)。复数值和 NaN 不能转换为逻辑值。
- true和false函数
- 分别创建全为逻辑1和逻辑0的逻辑矩阵。
  - true(n)可以生成一个n行n列全为逻辑值1的方阵
  - true(m,n)可以生成一个m行n列全为逻辑值1的矩阵
- &&和||
- 这两个运算符和“逻辑与&”和“逻辑或|”非常相似，但它们有两个非常重要的区别：
  - (1) &&和||只能对标量（只有一个值）进行逻辑运算，不能对有多元素的向量或者矩阵进行运算，而&和|可以。
  - (2) &&和||进行逻辑运算时具有短路功能，可以提高运行效率：
  - 计算A && B时，如果A为逻辑0，则B不会被判断，因为最后的结果一定是逻辑0；
  - 计算A || B时，如果A为逻辑1，则B不会被判断，因为最后的结果一定是逻辑1。
- 利用逻辑值对矩阵进行索引
- 假设有一个m行n列的矩阵A，我们要提取其指定位置的元素，那么我们可以生成一个和A同样大小的逻辑矩阵L，L中的元素要么是逻辑值1，要么是逻辑值0，其中：等于逻辑值1的元素所处的位置是我们所需要的。接着我们只需要使用命令A(L)，就能够在A中提取出指定位置的元素。通常逻辑矩阵L是由一系列的逻辑运算或者条件运算得来的。
  - L必须是逻辑矩阵，即里面的0和1都必须是逻辑值，不能是由数字0和1构成的数值矩阵。如果L是数值矩阵，可以使用logical函数进行转换。
  - 例如找出A中所有小于等于5的元素 A(A<=5)
  - 随机生成200名同学的考试得分(0-100整数)，统计得分大于等于60分的同学人数，并计算得分位于区间[60,80]内的所有同学的平均分。
  - A = randi([0,100],200,1); % 随机生成200名同学的分  
sum(A>=60) % 大于等于60分的人数  
tmp = A(A<=80 & A >=60); % 区间[60,80]内的所有分数  
mean(tmp) % 计算平均值
- all、any和find函数
- all 沿着给定的维度判断矩阵元素是否全为非零值，全为非零值时返回逻辑1
  - any 沿着给定的维度判断矩阵元素是否存在至少一个非零值，是的话返回逻辑1
  - find 查找矩阵中的非零元素，并返回其索引

集合运算

- unique 返回数组中的唯一值
- ismember 判断一个数组的元素是否在另一个数组内
- intersect 返回两个数组的交集
- union 返回两个数组的并集
- setdiff 返回两个数组的差集
- setxor 返回两个数组的对称差集

