

我是8位的

I am 8 bits, what about you?

随笔 - 205, 文章 - 0, 评论 - 103, 阅读 - 101万

导航

博客园

首页

新随笔

联系

订阅

管理

<

2022年3月

>

日	一	二	三	四	五	六
27	28	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

公告

你的支持是我的动力

欢迎关注微信公众号“我是8位的”



昵称：我是8位的

园龄：4年7个月

粉丝：288

关注：5

+加关注

盖楼抽奖

#她的梦想在发光#

HWD科技女性故事有奖征集

分享最打动的科技女性故事

活动时间：2022年3月8日-3月18日

马上参与



搜索

找找看

谷歌搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

积分与排名

积分 - 457097

排名 - 1198

线性代数笔记20——行列式和代数余子式

行列式

如果有两个向量 $\langle a_1, a_2 \rangle$ 和 $\langle b_1, b_2 \rangle$ ，那么这两个向量组成的行列式是：

$$\begin{vmatrix} a_1 & a_2 \\ b_1 & b_2 \end{vmatrix} = a_1 b_2 - a_2 b_1$$

看起来只是表示一个简单的计算，仅仅计算了一个数值，但是别忘了，行列式是由向量组成的，它一定会表示向量间的某种关系。

在《线性代数笔记4——向量3（叉积）》中我们看到，二阶行列式表示了二维平面中以两个向量为临边的平行四边形的面积；三阶行列式表示在三维空间中以三个向量为临边的平行六面体的体积；推广到n维空间，n阶行列式表示在n维空间中图形的n维体积。实际上我们无法有效表示出三维以上的空间。对于物理世界中更多维的空间，绝大多数人都无法想象，但是数学却可以给出明确的定义。

对于n维空间的行列式，可以表示为：

$$D_n = |A_{n \times n}|$$

其中A是一个 $n \times n$ 的矩阵。

行列式是由向量引出的，解释的也是向量的性质，在看到行列式时一定要在头脑中映射出向量，实际上线性代数的本质就是对向量的研究。

行列式的性质

性质0，单位矩阵的行列式为1

这个不解释。

性质1，如果 $D_n = |A|$ 中某行的元素全为0，那么 $D_n = 0$

这个性质较为明显，在多维空间中，行列式表示的是体积，如果其中一个维度的模为0，那么体积也是0。

性质2，如果 $D_n = |A|$ 中某两行元素对应成比例，那么 $D_n = 0$

很多时候我们都喜欢用实例推导性质，像下边这样：

$$\begin{vmatrix} 1 & 2 \\ 2 & 4 \end{vmatrix} = 1 \times 4 - 2 \times 2 = 0$$

或者用代数形式：

$$\begin{vmatrix} a_1 & a_2 \\ ka_1 & ka_2 \end{vmatrix} = a_2 ka_1 - a_1 ka_2 = 0$$

随笔分类 (211)

- ★★资源下载★★(1)
- Java并发编程(1)
- 程序员的数学(24)
- 单变量微积分(31)
- 多变量微积分(24)
- 概率(24)
- 机器学习(27)
- 软件设计(1)
- 数据分析(6)
- 数据结构与算法(27)
- 随笔(5)
- 线性代数(34)
- 项目管理(2)
- 转载(4)

随笔档案 (205)

- 2021年2月(1)
- 2020年3月(2)
- 2020年2月(6)
- 2020年1月(4)
- 2019年12月(7)
- 2019年11月(15)
- 2019年9月(3)
- 2019年8月(6)
- 2019年7月(1)
- 2019年6月(8)
- 2019年5月(3)
- 2019年4月(5)
- 2019年3月(7)
- 2019年2月(3)
- 2019年1月(7)
- 更多

阅读排行榜

- 1. 使用Apriori进行关联分析（一）(29768)
- 2. 线性代数笔记12——列空间和零空间(28772)
- 3. FP-growth算法发现频繁项集（一）——构建FP树(24430)
- 4. 寻找“最好”（2）——欧拉-拉格朗日方程(23099)
- 5. 多变量微积分笔记3——二元函数的极值(22772)

评论排行榜

- 1. 隐马尔可夫模型（一）(8)
- 2. 线性代数笔记12——列空间和零空间(7)
- 3. 线性代数笔记3——向量2（点积）(7)
- 4. FP-growth算法发现频繁项集（一）——构建FP树(5)
- 5. 寻找“最好”（2）——欧拉-拉格朗日方程(4)

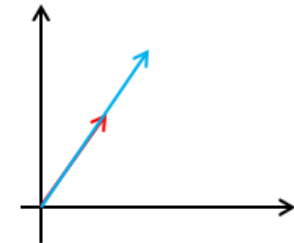
推荐排行榜

- 1. 寻找“最好”（2）——欧拉-拉格朗日方程(7)
- 2. FP-growth算法发现频繁项集（一）——构建FP树(7)
- 3. 线性代数笔记3——向量2（点积）(6)
- 4. FP-growth算法发现频繁项集（二）——发现频繁项集(5)
- 5. 隐马尔可夫模型（一）(5)

最新评论

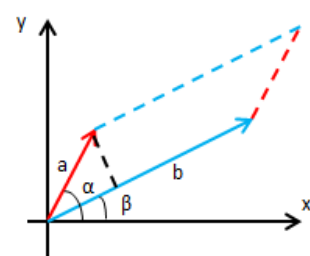
- 1. Re:线性代数笔记3——向量2（点积）
如果点积小于0，即夹角小于90°，这个写错了吧。应该是夹角大于90°
--猫猫猫猫猫大人

但是性质应当由定义推导，然后用计算去验证，而不是用计算去推导。现在我们尝试用行列式的定义去推导。行列式表示的是向量间的关系，以二维空间为例，如果某两行元素对应成比例，那么说明一个向量是另一个向量的延伸，它们的夹角是0°或180°，即二者平行，两个平行的向量围成的面积是0：



性质3，如果 $D_n = |A|$ 中某两行元素互换，那么互换后的行列式变号，即 $|A| = -|A|$

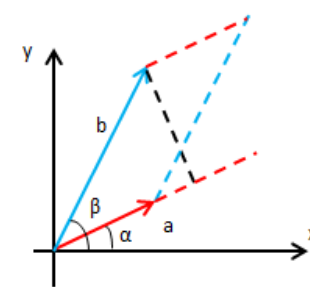
两个向量的模长是a和b，与x轴的夹角分别是 α 和 β ，如下图所示：



平行四边形的面积：

$$S = basin(\alpha - \beta)$$

如果两个向量互换：



$$S' = absin(\beta - \alpha) = basin(\alpha - \beta) = -S$$

在代数学中，角度、面积、体积可以是负的。用计算去验证：

$$\begin{vmatrix} a_1 & a_2 \\ b_1 & b_2 \end{vmatrix} = a_1b_2 - a_2b_1 = -(b_1a_2 - b_2a_1) = -\begin{vmatrix} b_1 & b_2 \\ a_1 & a_2 \end{vmatrix}$$

性质4，倍乘性质

$$D_n = |A| = \begin{vmatrix} \cdots & a_{i_1} & a_{i_2} & \cdots & a_{i_n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{vmatrix} \text{ then } k \begin{vmatrix} \cdots & a_{i_1} & a_{i_2} & \cdots & a_{i_n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{vmatrix} = \begin{vmatrix} \cdots & ka_{i_1} & ka_{i_2} & \cdots & ka_{i_n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{vmatrix}$$

2. Re:线性代数笔记10——矩阵的LU分解写的很好，不过LU分解的前提是错的，LU分解只需要第三个条件，如果允许行置换就是下面写到的PLU，可以分解所有矩阵

--wiki3D

3. Re:单变量微积分笔记20——三角替换1 (sin和cos) 很nice

--尹保棕

4. Re:线性代数笔记24——微分方程和exp(At) 有些图片挂了呢

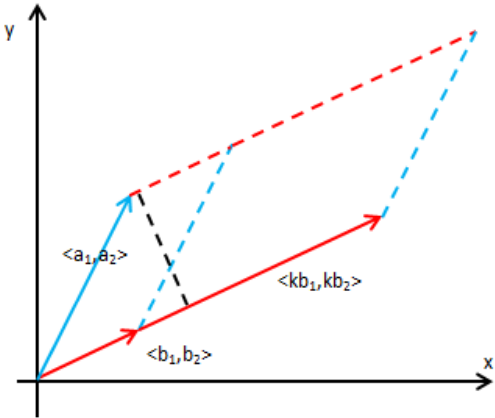
--ccchendada

5. Re:寻找“最好” (2) ——欧拉-拉格朗日方程

提个issue，最速降线中 $v = \{2gh\}^{1/2}$ 与配图不一致，建议以起点为原点，向右伸出x轴，向下伸出y轴建立坐标系

--trustInU

实际上是将外部的k乘到其中的一行，把平行四边形的一条边扩大k倍，则面积也扩大了k倍，如下图所示：



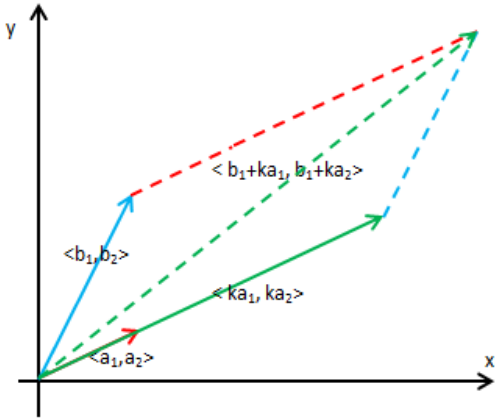
需要注意的是行列式与矩阵的区别，矩阵扩大k倍是将矩阵中的全部元素都乘以k（矩阵中的每个元素都对应了一个向量的分量，这在下文关于矩阵的介绍中会有所说明），这将有下面的关系：

$$|kA_{n \times n}| = k^n |A_{n \times n}|$$

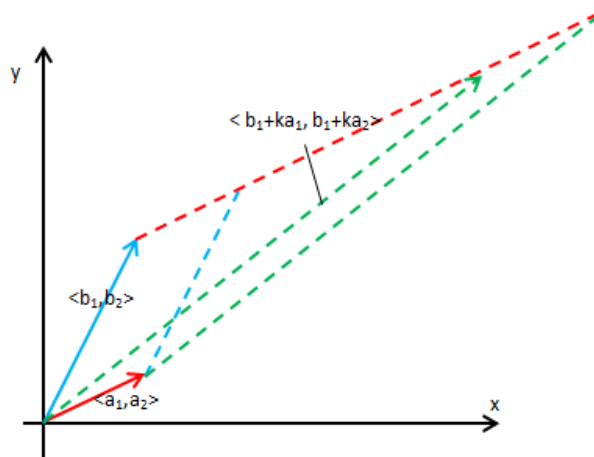
性质5，倍加性质

$$\begin{vmatrix} a_1 & a_2 \\ b_1 & b_2 \end{vmatrix} = \begin{vmatrix} a_1 & a_2 \\ b_1 + ka_1 & b_2 + ka_2 \end{vmatrix}$$

对于更高阶的行列式也一样。下图平行四边形的斜边展示了一个向量加上另一个向量的k倍：



两个平行四边形的面积是相同的，所以倍加公式成立。



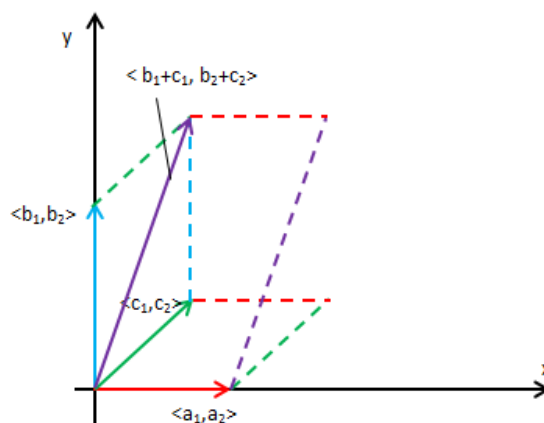
性质6, 单行可拆 (加) 性

$$\begin{vmatrix} * & & & \\ a_{i_1} & a_{i_2} & \cdots & a_{i_n} \\ * & & & \end{vmatrix} + \begin{vmatrix} * & & & \\ b_{i_1} & b_{i_2} & \cdots & b_{i_n} \\ * & & & \end{vmatrix} = \begin{vmatrix} * & & & \\ a_{i_1} + b_{i_1} & a_{i_2} + b_{i_2} & \cdots & a_{i_n} + b_{i_n} \\ * & & & \end{vmatrix}$$

其中*号表元素完全相同, 从左到右叫加, 从右到左叫拆。以二阶行列式为例:

$$\begin{vmatrix} a_1 & a_2 \\ b_1 & b_2 \end{vmatrix} + \begin{vmatrix} a_1 & a_2 \\ c_1 & c_2 \end{vmatrix} = \begin{vmatrix} a_1 & a_2 \\ c_1 + b_1 & c_2 + b_2 \end{vmatrix}$$

为了简单, 将<b₁, b₂>和<a₁, a₂>分别设置在两个坐标轴上, 如下图所示:



<a₁, a₂> <b₁, b₂>所围平行四边形面积是a₂ b₂, <a₁, a₂> <c₁, c₂>所围平行四边形面积是a₂ c₂, <a₁, a₂> <b₁+c₁, b₂+c₂>所围平行四边形面积是a₂(b₂+c₂), 由此可见性质6成立。

性质7, 以上所有作用于行的性质也可以作用于列上, 即|A| = |A^T|

性质8, 两个矩阵相乘的行列式, 等于这个两个矩阵的行列式相乘, |AB| = |A||B|

当两个矩阵相等时, 矩阵平方的行列式等于矩阵行列式的平方:

$$|A^2| = |A|^2$$

可以借助性质8计算A⁻¹的行列式:

$$|A^{-1}A| = |A^{-1}||A| = |I| = 1$$

$$|A^{-1}| = \frac{1}{|A|}$$

如果 $1/|A|$ 有意义, 则 $|A| \neq 0$, A 有逆矩阵; 反之, 如果 $|A| = 0$, A 是奇异矩阵。这就是性质9。

性质9, 如果 $|A| = 0$, A 是奇异矩阵。

行列式的意义

行列式是由向量组成的, 当 $D_n = |A| \neq 0$ 时, 意味着组成 $|A|$ 的向量全部独立。所谓独立, 就是向量围成的 n 维空间中图形的 n 维体积不为0。这似乎没有太大价值, 但是如果把行列式转换为方程组就意义重大了, 以二阶行列式为例:

$$\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} \neq 0 \leftrightarrow \begin{cases} x_1 + 2x_2 = 0 \\ 3x_1 + 4x_2 = 0 \end{cases}$$

$$\begin{vmatrix} 1 & 2 \\ 2 & 4 \end{vmatrix} = 0 \leftrightarrow \begin{cases} x_1 + 2x_2 = 0 \\ 2x_1 + 4x_2 = 0 \end{cases}$$

可以看到, 对于全部独立的向量, 方程组有唯一解, 否则方程组无解或有无数解。当 $|A| \neq 0$ 时, 说明至少有一个向量是“多余”的, 正是这个多余的向量使得 n 维体积为0。以阶行列式为例, 当体积为0时, 说明三个向量在同一平面内, 这意味着, 一定可以通过倍乘和倍加性质用另外两个向量表示第三个向量, 从而完全消除第三个向量。 N 元一次方程组需要 N 个完全不同的等式, 现在少了一个等式, 所以无法得到唯一解。

线性代数研究的是向量之间的关系, 向量间最重要的关系就是独立或不独立, 行列式是否等于0正是这种关系的有效描述。

行列式的计算

上三角矩阵的行列式等于主对角元素的乘积:

$$U = \begin{bmatrix} d_1 & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ 0 & d_2 & \blacksquare & \blacksquare & \blacksquare \\ 0 & 0 & d_3 & \blacksquare & \blacksquare \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & d_n \end{bmatrix}$$

$$|U| = d_1 d_2 d_3 \dots d_n = \prod_{i=1}^n d_i$$

对于更多阶的行列式, 一种有效的计算方法是先将其消元, 转换为上三角行列式, 然后在计算这个上三角行列式的值。以二阶行列式为例, 我们已经知道它的结果:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad \det(A) = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - cb$$

利用消元法将 A 转换为上三角矩阵:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \xrightarrow{-\frac{c}{a}r_1 + r_2} \begin{bmatrix} a & b \\ 0 & d - \frac{c}{a}b \end{bmatrix}$$

现在可以直接利用主对角线的元素相乘:

$$\det(A) = a \left(d - \frac{c}{a} b \right) - b \times 0 = ad - cb$$

行列式的公式

行列式的性质也可以用于计算行列式的值，以二阶行列式为例：

$$|A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = \underbrace{\begin{vmatrix} a & 0 \\ c & d \end{vmatrix}}_{\text{单行可拆性}} + \underbrace{\begin{vmatrix} 0 & b \\ c & d \end{vmatrix}}_{\alpha} = \underbrace{\begin{vmatrix} a & 0 \\ c & 0 \end{vmatrix}}_{\beta} + \underbrace{\begin{vmatrix} a & 0 \\ 0 & d \end{vmatrix}}_{\gamma} + \underbrace{\begin{vmatrix} 0 & b \\ c & 0 \end{vmatrix}}_{\delta} + \underbrace{\begin{vmatrix} 0 & b \\ 0 & d \end{vmatrix}}_{\delta}$$

在反复利用行列式的单行可拆性后，A分解成4项，每一行只有一个非零元素。二阶行列式计算的是图形的面积，对于 α 来说，由于构成行列式的两个向量 $\langle a, 0 \rangle$ 和 $\langle c, 0 \rangle$ 是在同一个维度上的直线，所以二者围成的面积是0；同理， δ 也一样。

B是上三角矩阵，它的值是主对角线的乘积 ad 。 γ 可以使用行列式的行互换性质形成一个新的上三角矩阵：

$$\gamma = \begin{vmatrix} 0 & b \\ c & 0 \end{vmatrix} = - \begin{vmatrix} c & 0 \\ 0 & b \end{vmatrix} = -bc$$

最终可以得到 $|A|$ 的值：

$$|A| = \alpha + \beta + \gamma + \delta = ad - bc$$

这种方法对于更高阶的行列式也同样适用，三阶行列式按照每一行只有一个非零元素的原则全部展开后将长达 3^3 项，这将占用长长的篇幅，可以考虑一个能够缩减展开式的办法。根据行列式的几何意义，行列式计算的是 n 维图形在 n 维空间中的 n 维体积，3阶行列式计算的自然是三维空间的体积，如此一来，只有三个向量分别指向三个不同维度时，才能保证体积不等于0，因此三阶行列式可以展开成：

$$|A| = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = \begin{vmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{vmatrix} + \begin{vmatrix} a_{11} & 0 & 0 \\ 0 & 0 & a_{23} \\ 0 & a_{32} & 0 \end{vmatrix} \\ + \begin{vmatrix} 0 & a_{12} & 0 \\ a_{21} & 0 & 0 \\ 0 & 0 & a_{33} \end{vmatrix} + \begin{vmatrix} 0 & a_{12} & 0 \\ 0 & 0 & a_{23} \\ a_{31} & 0 & 0 \end{vmatrix} + \begin{vmatrix} 0 & 0 & a_{13} \\ a_{21} & 0 & 0 \\ 0 & a_{32} & 0 \end{vmatrix} + \begin{vmatrix} 0 & 0 & a_{13} \\ 0 & a_{22} & 0 \\ a_{31} & 0 & 0 \end{vmatrix}$$

现在只剩下 $3! = 6$ 项，每一项都可以通过行列式的行交换性质变成上三角行列式（或者本身就是上三角行列式），这样就可以得到行列式的最终值：

$$|A| = a_{11}a_{22}a_{33} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31}$$

现在可以归纳出 n 阶行列式的公式：

$$|A| = \sum_{n!} \pm a_{1\alpha} a_{2\beta} a_{3\gamma} \dots a_{n\omega}$$

下标的数字项表示行号，希腊字母表示列号（实际数量可能远超过希腊字母的数量，暂且用希腊字母代替）。这相当于是列号的排列，在每一项中， n 个列标都各用一次。负号的目的是为了应对行交换的情况。

根据公式，对于 n 阶单位矩阵来说，只有主对角线的一项不是0，所以单位矩阵的行列式的值是1。

示例 计算A的行列式：

$$|A| = \begin{vmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{vmatrix}$$

通过消元法计算是正确的选择，通常也应该这么做，实际上不难看出这个A是一个奇异矩阵，所以它的行列式等于0，现在用行列式的公式来验证这个结论。根据公式，|A|的大多数展开项都等0，没有被淘汰的只有两项，二者相加等于0：

$$\begin{vmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = -1, \quad \begin{vmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix} = 1$$

第一个行列式是负值，因为它需要用1、3行进行一次行交换来变成上三角矩阵：

$$\begin{vmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = - \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = -1$$

代数余子式

代数余子式是从行列式的公式中提取出来的，它的作用是把n阶行列式化简为n-1阶行列式。我们以三阶行列式为例，看看代数余子式是什么。根据行列式的公式，3阶行列式展开，将得到：

$$|A| = a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31})$$

这实际上式选定第一行的一列，然后考虑各种可能的排列，为了突出重点，写成下面这样：

$$|A| = a_{11}(\dots) + a_{12}(\dots) + a_{13}(\dots)$$

注：根据第任何一行展开都将得到同样的结果，下面是根据第二行展开：

$$|A| = -a_{21}(a_{12}a_{33} - a_{13}a_{32}) + a_{22}(a_{11}a_{33} - a_{13}a_{31}) - a_{23}(a_{11}a_{32} - a_{12}a_{31})$$

按列展开也可以，比如按第一列展开：

$$|A| = a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{21}(a_{12}a_{33} - a_{13}a_{32}) + a_{31}(a_{12}a_{23} - a_{13}a_{22})$$

括号中由剩余因子组成的表达式就是代数余子式（第二项把符号移到了括号中，下节会说明原因），比如 $a_{22}a_{23} - a_{23}a_{32}$ 是 a_{11} 的代数余子式。可以用更直观的方式表达 $a_{11}(a_{22}a_{23} - a_{23}a_{32})$ ：

$$a_{11}(a_{22}a_{33} - a_{23}a_{32}) = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} = \begin{vmatrix} a_{11} & \blacksquare & \blacksquare \\ \blacksquare & a_{22} & a_{23} \\ \blacksquare & a_{32} & a_{33} \end{vmatrix}$$

代数余子式的符号

$-a_{12}(a_{22}a_{23} - a_{23}a_{32})$ 可以表示成:

$$-a_{12}(a_{21}a_{33} - a_{23}a_{31}) = -a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} = \begin{vmatrix} \blacksquare & a_{12} & \blacksquare \\ a_{21} & \blacksquare & a_{23} \\ a_{31} & \blacksquare & a_{33} \end{vmatrix}$$

注意到上式有一个负号, 我们一般不需要 $-a_{12}$ 的代数余子式, 所以 a_{12} 的代数余子式需要把符号移到括号中:

$$-a_{12}(a_{21}a_{33} - a_{23}a_{31}) = a_{12}(-a_{21}a_{33} + a_{23}a_{31})$$

代数余子式本身就是行列式, 只是它的正负号需要单独判断, 判断方法是根据选定元素行号和列号之和的奇偶性。用 C_{ij} 表示 a_{ij} 的代数余子式, 当 $i + j$ 是偶数时, 行列式取正号, 是奇数则取符号。比如三阶行列式中, C_{12} 的行列号之和是3, 它对应的代数余子式取符号。如果有一个五阶行列式, 它的每一项的代数余子式的符号是这样分布的:

$$\begin{vmatrix} + & - & + & - & + \\ - & + & - & + & - \\ + & - & + & - & + \\ - & + & - & + & - \\ + & - & + & - & + \end{vmatrix}$$

行列式的代数余子式展开

把某个行列式的用代数余子式展开实际上也是求行列式的另一种方法, 可以表示成:

$$|A| = a_{11}C_{11} + a_{12}C_{12} + \cdots a_{1n}C_{1n} = \sum_{i=1}^n a_{1i}C_{1i}$$

代数余子式本身是 $n - 1$ 阶行列式, 它可以继续展开成 $n - 2$ 阶行列式.....如此展开下去, 直到1阶行列式为止, 其核心思想是把一个复杂的高阶行列式转换成多个简单的低阶行列式。

由于行列式根据任何一行展开都可以, 所以:

$$|A| = a_{k1}C_{k1} + a_{k2}C_{k2} + \cdots a_{kn}C_{kn} = \sum_{i=1}^n a_{ki}C_{ki}$$

$$|A| = a_{1k}C_{1k} + a_{2k}C_{2k} + \cdots a_{nk}C_{nk} = \sum_{i=1}^n a_{ik}C_{ik}$$

二阶行列式的代数余子式

用代数余子式可以解释二阶行列式的计算公式。二阶行列式可以用一阶代数余子式展开:

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = a \begin{vmatrix} \blacksquare & \blacksquare \\ \blacksquare & d \end{vmatrix} - b \begin{vmatrix} \blacksquare & \blacksquare \\ c & \blacksquare \end{vmatrix} = ad - bc$$

由于 b 是第1行第2列, $1 + 2 = 3$ 是奇数, 所以 b 对应的代数余子式 C_{12} 是以符号开头的。

作者：我是8位的

出处：<http://www.cnblogs.com/bigmonkey>

本文以学习、研究和分享为主，如需转载，请联系本人，标明作者和出处，非商业用途！

扫描二维码关注公众号“我是8位的”



随笔

分类: 线性代数

标签: 行列式, 行列式计算公式, 代数余子式

[好文要顶](#)[关注我](#)[收藏该文](#)



我是8位的

关注 - 5

粉丝 - 288

[+加关注](#)

0

0

推荐

反对

« 上一篇: [线性代数笔记19——格拉姆-施密特正交化](#)
» 下一篇: [寻找“最好” \(4\) ——不等约束和KKT条件](#)

posted on 2018-11-21 16:39 我是8位的 阅读(6618) 评论(2) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 登录后才能查看或发表评论，立即 [登录](#) 或者 [逛逛](#) 博客园首页

【推荐】华为 HWD 2022 故事征集，分享最打动你的科技女性故事
【推荐】华为开发者专区，与开发者一起构建万物互联的智能世界

广告 X

yo zodcs.com

永中DCS_文档在线
预览解决方案

兼容不同版本的Office、PDF等，特定格式的合作，支持Windows、信创环境下一键部署

打开 >

- 编辑推荐:
- 革命性创新，动画杀手锏 @scroll-timeline
 - 戏说领域驱动设计（十二）—— 服务
 - ASP.NET Core 6框架揭秘实例演示[16]：内存缓存与分布式缓存的使用
 - .Net Core 中无处不在的 Async/Await 是如何提升性能的？
 - 分布式系统改造方案 —— 老旧系统改造篇



最新新闻:

- 乔布斯的创业搭档：他缺乏工程师才能，不得不锻炼营销能力来弥补
 - 美国大厂码农薪资曝光：年薪18万美元，够养家，不够买海景房
 - 两张照片就能转视频！Google提出FLIM帧插值模型
 - Android 再推 “杀手级” 功能，可回收 60% 存储空间
 - 溺在理财暴雷潮的投资人：本金63万，月兑25元不够卖菜
- » 更多新闻...

Powered by:

博客园

Copyright © 2022 我是8位的

Powered by .NET 6 on Kubernetes