目录

矩阵代数与行列式

1.概念与矩阵运算

若A是$\mathrm{m}\times\mathrm{n}$矩阵，即有$m$行$n$列的矩阵，$A$的第$i$行第$j$列的元素用$a_{ij}$表示，称为$A$的$(i,j)$元素，如下图所示

设A,B,C是相同维数的矩阵，r与s为数，则有

a. $A+B=B+A$
b. $(A+B)+C=A+(B+C)$
c. $(A+0)=A$
d. $r(A+B)=rA+rB$
e. $(r+s)A=\mathrm{r}A+\mathrm{s}A$
f. $r(sA)=(rs)A$
g. $AB\neq BA$

若乘积AB有定义，AB的第i行第j列的元素是A的第i行与B的第j列对应元素乘积之和。若$(AB)_{ij}$表示AB的(i,j)元素，A为$\mathrm{m}\times\mathrm{n}$矩阵，则

$$(AB)_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj}$$

矩阵乘法的理解可以参照前面有关矩阵方程的介绍。

矩阵的乘幂表示为A^k，其中A是$\mathrm{n} \times \mathrm{n}$矩阵，$A^k$表示$k$个$A$的乘积。$A^0$为单位矩阵。

$$A^k = A \cdots A$$

矩阵的转置表示为A^T，其中A是一个$\mathrm{m} \times \mathrm{n}$矩阵，则$A$的转置是一个$\mathrm{n} \times \mathrm{m}$矩阵，它的列是由$A$对应行构成的。

设A与B表示矩阵，其维数使下列和与积有定义，则

a. $(A^T)^T = A$
b. $(A+B)^T = A^T + B^T$
c. 对任意实数r，$(rA)^T = rA^T$
d. $(AB)^T = B^T A^T$

2.矩阵的逆

一个$\mathrm{n} \times \mathrm{n}$矩阵$A$是可逆的，若存在一个$\mathrm{n} \times \mathrm{n}$矩阵$C$使

$$AC = I \text{ 且 } CA = I$$

这时称C是A的**逆阵**，A可逆，且它的逆矩阵唯一。我们将C记为A^{-1}，于是

$$AA^{-1} = I \text{ 且 } A^{-1}A = I$$

不可逆矩阵也称为**奇异矩阵**，可逆矩阵也称为**非奇异矩阵**。

我们从这个定义可可以知道，可逆矩阵一定是方阵。后面还会引入左逆和右逆。下面是有关可逆矩阵的一些性质。

2.1 伴随矩阵求逆矩阵

将A和I排在一起构成增广矩阵$[A\ I]$，则对此矩阵进行行变换时，A和I受到同一变换。要么有一系列行变换把A变成I，同时I变成A^{-1}；要么A是不可逆的。例如

如求 $A = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 1 & 4 \\ 2 & -1 & 0 \end{bmatrix}$ 的逆矩阵A^{-1}。

$$B = [A|I] = \begin{bmatrix} 0 & 1 & 2 & 1 & 0 & 0 \\ 1 & 1 & 4 & 0 & 1 & 0 \\ 2 & -1 & 0 & 0 & 0 & 1 \end{bmatrix} \to \begin{bmatrix} 1 & 0 & 0 & 2 & -1 & 1 \\ 0 & 1 & 0 & 4 & -2 & 1 \\ 0 & 0 & 1 & -\frac{3}{2} & 1 & -\frac{1}{2} \end{bmatrix}$$

故A可逆并且，由右一半可得逆矩阵 $A^{-1} = \begin{bmatrix} 2 & -1 & 1 \\ 4 & -2 & 1 \\ -\frac{3}{2} & 1 & -\frac{1}{2} \end{bmatrix}$

伴随矩阵求A^{-1}时间复杂度为$O(n^4)$，开销大，更高效的算法可以应用矩阵的因式分解，请参考 3.4 应用。

2.2 逆矩阵的应用-方程求解

可逆矩阵的一个重要性质就是用于求解方程组。

若A可逆，则对每一个$\mathbb{R}^n$中的$\vec{b}$，方程$A\vec{x} = \vec{b}$有唯一解$\vec{x} = A^{-1}\vec{b}$。

我们举一个例子来说明可逆矩阵的作用。比如我们要解如下方程

$$\begin{cases} 3x_1 + 4x_2 = 3 \\ 5x_1 + 6x_2 = 7 \end{cases}$$

解 该方程组就是$A\vec{x} = \vec{b}$，可以看成

$$\begin{bmatrix} 3 & 4 \\ 5 & 6 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 7 \end{bmatrix}$$

所以有

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 5 & 6 \end{bmatrix}^{-1}\begin{bmatrix} 3 \\ 7 \end{bmatrix} = \begin{bmatrix} -3 & 2 \\ 5/2 & -3/2 \end{bmatrix}\begin{bmatrix} 3 \\ 7 \end{bmatrix} = \begin{bmatrix} 5 \\ -3 \end{bmatrix}$$

2.3 可逆矩阵的性质、判断是否可逆

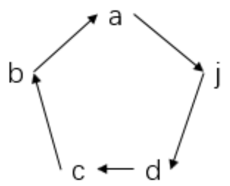
下面是有关可逆矩阵的两个基本定理：

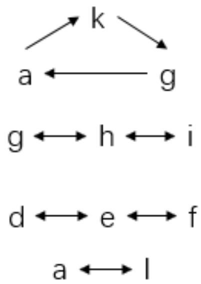
a. $(A^{-1})^{-1} = A$
b. 若A, B可逆，则有$(AB)^{-1} = B^{-1}A^{-1}$

接着是可逆矩阵定理。<u>设A为$\mathrm{n} \times \mathrm{n}$矩阵</u>，则下列命题是等价的，即对某一特定的$A$，它们同时为真或同时为假。

a. A是可逆矩阵
b. A等价于$\mathrm{n} \times \mathrm{n}$单位矩阵
c. A有n个主元位置
d. 方程$A\vec{x} = \vec{0}$仅有平凡解
e. A的各列线性无关
f. 线性变换$\vec{x} \mapsto A\vec{x}$是一对一的
g. 对$\mathbb{R}^n$中任意$\vec{b}$，方程$A\vec{x} = \vec{b}$至少有一个解
h. A的各列生成$\mathbb{R}^n$
i. 线性变换$\vec{x} \mapsto A\vec{x}$把$\mathbb{R}^n$映上$\mathbb{R}^n$上
j. 存在$\mathrm{n} \times \mathrm{n}$矩阵$C$使$CA = I$
k. 存在$\mathrm{n} \times \mathrm{n}$矩阵$D$使$DA = I$
l. A^T是可逆矩阵

它们的推导关系如下所示





c $\rightarrow$ b 是因为在矩阵有n个主元就说明它的简化阶梯形是I。

关于 d，可有 j 推出，即 $CA\vec{x}=\vec{0}$ 推得 $\vec{x}=\vec{0}$.

关于 e，其实就是 d，可参照 4.1 节。

利用这些定理，我们可以用于判断一个矩阵是否可逆。例如我们有一个矩阵A，如下所示。

$$A=\begin{bmatrix}1 & 0 & -2\\3 & 1 & -2\\-5 & -1 & 9\end{bmatrix}$$

解

$$A\sim\begin{bmatrix}1 & 0 & -2\\0 & 1 & 4\\0 & -1 & -1\end{bmatrix}\sim\begin{bmatrix}1 & 0 & -2\\0 & 1 & 4\\0 & 0 & 3\end{bmatrix}$$

所以A有 3 个主元位置，根据可逆矩阵定理 c，A是可逆的。

3.矩阵因式分解-$\boldsymbol{LU}$分解

矩阵的因式分解是把A表示为两个或更多矩阵的乘积，矩阵乘法是数据的综合（把两个或更多线性变换的作用结合成一个矩阵），矩阵的因式分解是数据的分解，把数据组成两个或者更多，这种结构可能更有用，或者更便于计算。

LU分解，是在工业与商业问题中常见的，解一系列具有相同系数矩阵中的线性方程

$$A\vec{x}=\overrightarrow{b_1},A\vec{x}=\overrightarrow{b_2},A\vec{x}=\overrightarrow{b_3},\dots,A\vec{x}=\overrightarrow{b_p}$$

我们解决这个问题是可以通过计算A^{-1}来求解方程，但是若A不可逆呢？

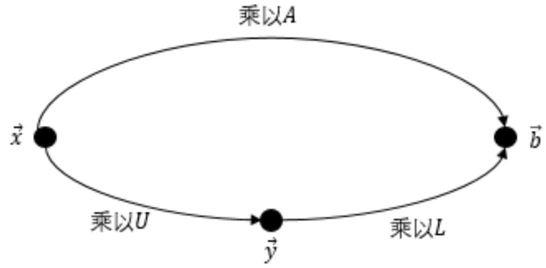
首先，我们假设A是$m\times n$矩阵可以行化简为阶梯形而不必行对换，则A可以写成形式$A=LU$，L是$m\times n$下三角矩阵，主对角线元素全是 1，U是A的一个等价的$m\times n$阶梯形矩阵，例如下图所示，这样一个分解称为LU分解，矩阵L是可逆的，称为单位下三角矩阵。

$$A=\underbrace{\begin{bmatrix}1 & 0 & 0 & 0\\ * & 1 & 0 & 0\\ * & * & 1 & 0\\ * & * & * & 1\end{bmatrix}}_{L}\underbrace{\begin{bmatrix}* & * & * & * & *\\0 & * & * & * & *\\0 & 0 & 0 & * & *\\0 & 0 & 0 & 0 & 0\end{bmatrix}}_{U}$$

求解$A\vec{x}=\vec{b}$可写成$L(U\vec{x})=\vec{b}$，因此可以由解下列方程来求解$\vec{x}$：

$$\begin{cases} L\vec{y}=\vec{b} & (1) \\ U\vec{x}=\vec{y} & (2) \end{cases}$$

先解(1)再解(2)即可得到$\vec{x}$，每个方程都很容易解，因为L和U都是三角矩阵。



映射$\vec{x}\mapsto A\vec{x}$的分解

下面我们举一个例子来应用LU分解。

已知

$$A=\begin{bmatrix}3 & -7 & -2 & 2\\ -3 & 5 & 1 & 0\\ 6 & -4 & 0 & -5\\ -9 & 5 & -5 & 12\end{bmatrix}=\begin{bmatrix}1 & 0 & 0 & 0\\ -1 & 1 & 0 & 0\\ 2 & -5 & 1 & 0\\ -3 & 8 & 3 & 1\end{bmatrix}\begin{bmatrix}3 & -7 & -2 & 2\\ 0 & -2 & -1 & 2\\ 0 & 0 & -1 & 1\\ 0 & 0 & 0 & -1\end{bmatrix}=LU$$

应用A的LU分解来解$A\vec{x}=\vec{b}$，其中$\vec{b}=\begin{bmatrix}-9\\5\\7\\11\end{bmatrix}$.

解

$$[L\ \vec{b}]=\begin{bmatrix}1 & 0 & 0 & 0 & -9\\ -1 & 1 & 0 & 0 & 5\\ 2 & -5 & 1 & 0 & 7\\ -3 & 8 & 3 & 1 & 11\end{bmatrix}\sim\begin{bmatrix}1 & 0 & 0 & 0 & -9\\ 0 & 1 & 0 & 0 & -4\\ 0 & 0 & 1 & 0 & 5\\ 0 & 0 & 0 & 1 & 1\end{bmatrix}=[I\ \vec{y}]$$

$$[U\ \vec{y}]=\begin{bmatrix}3 & -7 & -2 & 2 & -9\\ 0 & -2 & -1 & 2 & -4\\ 0 & 0 & -1 & 1 & 5\\ 0 & 0 & 0 & -1 & 1\end{bmatrix}\sim\begin{bmatrix}1 & 0 & 0 & 0 & 3\\ 0 & 1 & 0 & 0 & 4\\ 0 & 0 & 1 & 0 & -6\\ 0 & 0 & 0 & 1 & -1\end{bmatrix}=[I\ \vec{x}]$$

$$\text{所以}\vec{x}=\begin{bmatrix}3\\4\\-6\\-1\end{bmatrix}.$$

3.1 LU分解算法

算法并不复杂，总的来说就是

1. 矩阵A经过一系列行变换变成阶梯形矩阵，即为U
2. 变换过程在每个主元列, 把主元以下的元素除以主元即为L在该列主元以下的元素。

以一个例子来讲解。求下列矩阵的LU分解

$$A=\begin{bmatrix}2 & 4 & -1 & 5 & -2\\ -4 & -5 & 3 & -8 & 1\\ 2 & -5 & -4 & 1 & 8\\ -6 & 0 & 7 & -3 & 1\end{bmatrix}$$

解

$$A=\begin{bmatrix}2&4&-1&5&-2\\-4&-5&3&-8&1\\2&-5&-4&1&8\\-6&0&7&-3&1\end{bmatrix}\sim\begin{bmatrix}2&4&-1&5&-2\\0&3&1&2&-3\\0&-9&-3&-4&10\\0&12&4&12&-5\end{bmatrix}=A_1$$

$$\sim A_2=\begin{bmatrix}2&4&-1&5&-2\\0&3&1&2&-3\\0&0&0&2&1\\0&0&0&4&7\end{bmatrix}\sim\begin{bmatrix}2&4&-1&5&-2\\0&3&1&2&-3\\0&0&0&2&1\\0&0&0&0&5\end{bmatrix}=U$$

接着是在变换过程中求出L矩阵。上式中标出的元素确定了A化为U的行变换，在每个主元列，把标出的元素除以主元后将结果放入L。(提问，如果处理主元为 0 的情况？)

$$\begin{bmatrix}2\\-4\\2\\6\end{bmatrix}\begin{bmatrix}3\\-9\\12\end{bmatrix}\begin{bmatrix}2\\4\end{bmatrix}[5]$$

$\div 2 \quad \div 3 \quad \div 2 \div 5$

$$\begin{bmatrix}1&0&0&0\\-2&1&0&0\\1&-3&1&0\\3&4&2&1\end{bmatrix}=L$$

容易证明，所求出的L和U满足$LU=A$.

3.2 算法性能分析

下列运算次数的计算适用于$n\times n$稠密矩阵A（大部分元素非零），n相当大，例如$n\geq 30$。

1. 计算A的LU分解大约需要$2n^3/3$浮算，而求A^{-1}大约需要$2n^3$浮算。
2. 解$L\vec{y}=\vec{b}$和$U\vec{x}=\vec{y}$大约需要$2n^3$浮算，因为任意$n\times n$三角方程组可以用大约n^3浮算接触。
3. 把$\vec{b}$乘以A^{-1}也需要$2n^3$浮算，但结果可能不如L和U得出的精确(由于计算A^{-1}及$A^{-1}\vec{b}$的舍入误差)
4. 若A是稀疏矩阵（大部分元素为 0），则L和U可能也是稀疏的，然而A^{-1}很可能是稠密的，显然用LU分解来解$A\vec{x}=\vec{b}$很可能比用A^{-1}快很多。

3.3 LUP分解 and 置换矩阵

考虑下面一个矩阵的LU分解。

$$A=\begin{bmatrix}2&4&-1\\0&0&3\\0&-5&-4\end{bmatrix}$$

根据之前所述算法，先将矩阵A化为阶梯形矩阵，并在变换的过程中求出L。但是很明显，在这种情况下，会出现除数为 0 的情况，这当然是灾难性的。

除了除数为 0 外，还有除数很小的情况，这会产生数值不稳定，因此我们希望尽可能选一个较大的主元。所以有的时候我们必须在矩阵的非对角线元素中选主元。

我们先引入一个置换矩阵的概念，考虑下面两个矩阵的相乘。

$$已知 P=\begin{bmatrix}1 & 0 & 0\\0 & 0 & 1\\0 & 1 & 0\end{bmatrix}，A=\begin{bmatrix}2 & 4 & -1\\0 & 0 & 3\\0 & -5 & -4\end{bmatrix}，计算 P\cdot A$$

根据我们之前的所学知识很容易解出来。求得

$$A=\begin{bmatrix}2 & 4 & -1\\0 & -5 & -4\\0 & 0 & 3\end{bmatrix}$$

可以发现，矩阵A似乎没什么变化，只是行的位置改变了而已。我们把这样的矩阵称之为置换矩阵。

置换矩阵 一种系数只由 0 和 1 组成的方阵。矩阵的每一行/列的第i个元素为 1，表示原矩阵的该行/列为第i行/列。具体置换的行 or 列取决于是左乘还是右乘。

- 当矩阵A左乘一个置换矩阵，交换的是矩阵的行。即$P\cdot A$
- 当矩阵A右乘一个置换矩阵，交换的是矩阵的列。即$A\cdot P$

让我们回到LUP分解。借助这个置换矩阵，我们可以将矩阵A的行进行置换，每步重新选取较大的主元进行行替换。

原来的$A=LU$我们改写为$PA=LU$，其中P是一个置换矩阵。该式子意为先对A进行行置换，再对行置换后的A进行LU分解。

当我们要求解$A\vec{x}=\vec{b}$，可做如下变换：

$$A\vec{x}=\overrightarrow{b_1}\rightarrow PA\vec{x}=P\overrightarrow{b_1}\rightarrow LU\vec{x}=P\overrightarrow{b_1}$$

于是问题可以化为

$$\begin{cases}L\vec{y}=P\vec{b} \quad (1)\\ U\vec{x}=\vec{y} \quad (2)\end{cases}$$

求解得到的$\vec{x}$就是$A\vec{x}=\vec{b}$的解，证明如下:

$$A\vec{x}=P^{-1}LU\vec{x}=P^{-1}L\vec{y}=P^{-1}Pb=b$$

其实LU分解是LUP分解的一种，当$P=I$时，LUP分解就成为LU分解。可以理解成是A矩阵不进行置换的情况下LUP分解就成为了LU分解。

3.4 应用

利用好矩阵的因式分解，不只是在求解方程组，在求逆矩阵、最小二乘逼近方面也能大大提高效率。

在求逆矩阵时，我们可以有两种思路：

- $A^{-1} = (LU)^{-1}P = U^{-1}L^{-1}P$
- 通过LUP分解解方程$AX = I$，X即为A^{-1}

通过LUP分解来求逆矩阵的过程中，分解的过程需要$O(n^3)$，求解三角矩阵的过程需要$O(n^2)$，避免了主元素为 0 的情况。可提高求解一系列具有相同系数矩阵中的线性方程的效率，3.矩阵因式分解-***LU***分解。

LUP分解在 Python 中可以使用 scipy 库的 lu 方法直接进行求解，使用示例如下:

```
import numpy as np
from scipy.linalg import lu

if __name__ == "__main__":
    a = np.asarray([[2, 4, -1],
                    [0, 0, 3],
                    [0, -5, -4]])
    p, l, u = lu(a)
```

4.行列式

首先行列式是只对方阵才有定义的，即**只有方阵才能计算行列式**，可逆矩阵也类似。

当行列式不为 0 时，矩阵才可逆；当系数矩阵行列式不为 0 时，系数矩阵才线性无关，方程组才有解。

接下来我们会先介绍行列式是如何计算的。

4.1 行列式的计算

我们先给出2×2矩阵和3×3矩阵的行列式。

$$detA = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

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

$$-a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} - a_{13}a_{22}a_{31}$$

余子式 在n阶行列式中，把元素a_{ij}所在的第i行第j列划去后，余下的$n-1$阶行列式叫做(i,j)元素的a_{ij}的余子式，记作M_{ij}，同时$A_{ij} = (-1)^{i+j}M_{ij}$称为$a_{ij}$的代数余子式。

要注意的是，余子式与代数余子式都是行列式，行列式的阶越低越容易计算，它存在的意义就是为了更好的计算行列式。

下面给出行列式的计算公式。

$$detA = \begin{vmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{vmatrix} = a_{11}M_{11} - a_{12}M_{12} + \cdots + (-1)^{1+n}a_{1n}M_{1n}$$

$$= a_{11}A_{11} + a_{12}A_{12} + \cdots + a_{1n}A_{1n}$$

这是按照第一行来计算行列式，更一般地，我们可以按照第i行或者第j列来计算。通过观察我们也可以知道，**三角形矩阵的行列式等于对角线上元素的乘积**。

$$detA = \begin{vmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{vmatrix} = a_{i1}A_{i1} + a_{i2}A_{i2} + \cdots + a_{in}A_{in}$$

$$= a_{1j}A_{1j} + a_{2j}A_{2j} + \cdots + a_{nj}A_{nj}$$

让我们举个例子来加深理解。

计算$detA$，这里

$$A = \begin{bmatrix} 3 & -7 & 8 & 9 & -6 \\ 0 & 2 & -5 & 7 & 3 \\ 0 & 0 & 1 & 5 & 0 \\ 0 & 0 & 2 & 4 & -1 \\ 0 & 0 & 0 & -2 & 0 \end{bmatrix}$$

解 按A的第一列来进行计算，有

$$detA = 3 \cdot \begin{vmatrix} 2 & -5 & 7 & 3 \\ 0 & 1 & 5 & 0 \\ 0 & 2 & 4 & -1 \\ 0 & 0 & -2 & 0 \end{vmatrix} - 0 + 0 - 0 + 0$$

再对第一列进行展开，有

$$detA = 2 \cdot \begin{vmatrix} 1 & 5 & 0 \\ 2 & 4 & -1 \\ 0 & -2 & 0 \end{vmatrix} = -12$$

具体按照哪一列来展开要看情况。

4.2 行列式的性质

我们给出几个行列式的性质。

A是一个方阵。

a. 若A的某一行的倍数加到另一行得矩阵B，则 $detB{=}detA$
b. 若A的两行互换得矩阵B，则 $detB{=}{-}detA$
c. 若A的某行乘以k倍得到矩阵B，则 $detB{=}k \cdot detA$
d. $detA = detA^T$，这证明了行列式的列变换和行变换具有相同效果
e. 若A与B均为$n \times n$矩阵，则 $detAB = detA \cdot detB$

这几个性质可以帮我们有效地计算行列式。我们举一个例子来说明。

计算$detA$，其中$A = \begin{bmatrix} 1 & -4 & 2 \\ -2 & 8 & -9 \\ -1 & 7 & 0 \end{bmatrix}$

解 思路是先将A化简成阶梯形，再利用三角形矩阵的行列式等于对角线上元素的乘积来计算。

$$detA=\begin{vmatrix}1&-4&2\\-2&8&-9\\-1&7&0\end{vmatrix}=\begin{vmatrix}1&-4&2\\0&0&-5\\-1&7&0\end{vmatrix}=\begin{vmatrix}1&-4&2\\0&0&-5\\0&3&2\end{vmatrix}=-\begin{vmatrix}1&-4&2\\0&3&2\\0&0&-5\end{vmatrix}$$
$$=-(1)(3)(-5)=15$$

4.3 逆矩阵公式

下面给出一个利用行列式来计算矩阵的逆的公式。

定理 设A是一个可逆的$n\times n$矩阵，则$A^{-1}=\frac{1}{detA}adjA$.

用一个例子来说明。求矩阵$A=\begin{bmatrix}2&1&3\\1&-1&1\\1&4&-2\end{bmatrix}$的逆。

解 九个代数余子式为

$$A_{11}=+\begin{vmatrix}-1&1\\4&-2\end{vmatrix}=-2\quad A_{12}=3\quad A_{13}=5$$
$$A_{21}=14\quad A_{22}=-7\quad A_{23}=-7$$
$$A_{31}=4\qquad A_{32}=1\qquad A_{33}=-3$$

$adjA$为代数余子式的矩阵的转置(例如A_{21}到(1,2)的位置)，从而有

$$adjA=\begin{bmatrix}-2&14&4\\3&-7&1\\5&-7&-3\end{bmatrix}$$

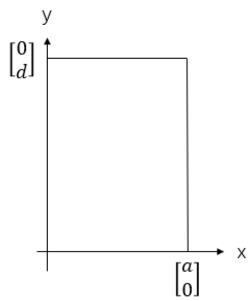
我们可以直接计算$detA$，但下列计算对上面的结果提供了一个检验并计算出$detA$的方法。

$$(adjA)\cdot A=\begin{bmatrix}-2&14&4\\3&-7&1\\5&-7&-3\end{bmatrix}\begin{bmatrix}2&1&3\\1&-1&1\\1&4&-2\end{bmatrix}=\begin{bmatrix}14&0&0\\0&14&0\\0&0&14\end{bmatrix}=14I$$

我们根据定理可知，$A^{-1}A=\frac{1}{detA}adjA\cdot A=\frac{1}{detA}14I$，可知 $detA=14$.

所以，$A^{-1}=\frac{1}{14}\begin{bmatrix}-2&14&4\\3&-7&1\\5&-7&-3\end{bmatrix}=\begin{bmatrix}-1/7&1&2/7\\3/14&-1/2&1/14\\5/14&-1/2&-3/14\end{bmatrix}$

4.4 用行列式表示面积或体积



若A是一个2×2矩阵，则由A的列确定的平行四边形的面积为$|detA|$；若A是一个3×3矩阵，则由A的列确定的平行六面体的体积也为$n\times n$。可引申到$n\times n$矩阵的情形。这也就是**行列式的几何意义**。

例如若A为 2 阶对角矩阵，定理显然成立。

$$\left|det\begin{bmatrix}a&0\\0&d\end{bmatrix}\right|=|ad|=\text{矩阵面积}$$

所以可以得出，判断一个线性方程组是否有唯一解，即判断系数矩阵的列向量所构成的体积是否为 0，即判断列向量是否两两位于不同平面上。