用高斯消元法解线性方程组

北京景山学校 何江舟

GPA排名系统（CTSC2001）

高等院校往往采用GPA来评价学生的学术表现。传统的排名方式是求每一个学生的平均成绩，以平均成绩作为依据进行排名。对于不同的课程，选课学生的平均成绩会受到课程的难易程度等因素的影响，因此这种排名方式不够合理。

为此，我们需要对排名系统进行这样的改进：对第i门课的每一个学生的成绩加上一个特定的修正值di（调整后的成绩不按照百分制），使得经过调整后，该课的平均分等于选该课的所有学生的所有课的平均分。对每一门课都这样调整，使得上述条件对所有课程都满足。

你的任务是根据一个年级学生某学年的成绩，通过上述调整，得出他们的排名。

简要分析

A_i：选修第i门课的学生的集合
B_j：第j个学生选修课程的集合
$G_{i,j}$：第j个学生第I门课的成绩
d_i：第i门课的修正值
对于第p门课，可列出如下关系式：

$$\frac{1}{|A_p|}\sum_{j\in A_p} G_{p,j} + d_p = \frac{1}{\sum_{j\in A_p}|B_j|}\sum_{j\in A_p}\sum_{i\in B_j}(G_{i,j} + d_i)$$

这是关于d_i（i=1,2,…,n）的线性方程，我们可以整理出n个这样的方程。

线性方程组的一般形式

下面是n元线性方程组的一般形式：

$$
\begin{cases}
a_{1,1}x_1+a_{1,2}x_2+\ldots\ldots+a_{1,n}x_n=b_1 \\
a_{2,1}x_1+a_{2,2}x_2+\ldots\ldots+a_{2,n}x_n=b_2 \\
\qquad\qquad\ldots\ldots \\
a_{n,1}x_1+a_{n,2}x_2+\ldots\ldots+a_{n,n}x_n=b_n
\end{cases}
$$

我们可以把它表示为增广矩阵的形式：

$$
\begin{pmatrix}
a_{1,1} & a_{1,2} & \ldots\ldots & a_{1,n} & b_1 \\
a_{2,1} & a_{2,2} & \ldots\ldots & a_{2,n} & b_2 \\
 & & \ldots\ldots & & \\
a_{n,1} & a_{n,2} & \ldots\ldots & a_{n,n} & b_n
\end{pmatrix}
$$

先看一个例子

$$\begin{pmatrix} 2 & -1 & 3 & 1 \\ 4 & 2 & 5 & 4 \\ 1 & 2 & 0 & 7 \end{pmatrix} \times 2 \ \times 0.5$$

$$\downarrow$$

$$\begin{pmatrix} 2 & -1 & 3 & 1 \\ & 4 & -1 & 2 \\ & 2.5 & -1.5 & 6.5 \end{pmatrix} \times 2.5$$

$$\downarrow$$

$$\begin{pmatrix} 2 & -1 & 3 & 1 \\ & 4 & -1 & 2 \\ & & -0.875 & 5.25 \end{pmatrix} \rightarrow$$

得出：

$x_3=5.25/(-0.875)=-6$

$x_2=(2-(-1)x_3)/4=-1$

$x_1=(1-(-1)x_2-3x_3)/2=9$

消元过程

$$\begin{pmatrix} a_{1,1}^{(1)} & a_{1,2}^{(1)} & \cdots\cdots & a_{1,n}^{(1)} & b_1^{(1)} \\ a_{2,1}^{(1)} & a_{2,2}^{(1)} & \cdots\cdots & a_{2,n}^{(1)} & b_2^{(1)} \\ & & \cdots\cdots & & \\ a_{n,1}^{(1)} & a_{n,2}^{(1)} & \cdots\cdots & a_{n,n}^{(1)} & b_n^{(1)} \end{pmatrix}$$

注：用上标(k)表示第k次消元前的状态

第1次消元，第1行的乘数：（i=2,3,…,n）

$$\boxed{m_{i,1} = \frac{a_{i,1}^{(1)}}{a_{1,1}^{(1)}}}$$

得到新的增广矩阵：

$$\begin{pmatrix} a_{1,1}^{(1)} & a_{1,2}^{(1)} & \cdots\cdots & a_{1,n}^{(1)} & b_1^{(1)} \\ & a_{2,2}^{(2)} & \cdots\cdots & a_{2,n}^{(2)} & b_2^{(2)} \\ & & \cdots\cdots & & \\ & a_{n,2}^{(2)} & \cdots\cdots & a_{n,n}^{(2)} & b_n^{(2)} \end{pmatrix}$$

$$\boxed{\begin{cases} a_{i,j}^{(2)}=a_{i,j}^{(1)}-m_{i,1}a_{1,j}^{(1)} \\ b_i^{(2)}=b_i^{(1)}-m_{i,1}b_1^{(1)} \end{cases}}$$

（i,j=2,3,…,n）

消元过程

第k次消元前的增广矩阵：

$$\begin{pmatrix}
a_{1,1}^{(1)} & a_{1,2}^{(1)} & \cdots\cdots\cdots & & a_{1,n}^{(1)} & b_1^{(1)} \\
 & a_{2,2}^{(2)} & \cdots\cdots\cdots & & a_{2,n}^{(2)} & b_2^{(2)} \\
 & & \cdots\cdots\cdots & & & \\
 & & a_{k,k}^{(k)} & \cdots\cdots & a_{k,n}^{(k)} & b_k^{(k)} \\
 & & & \cdots\cdots & & \\
 & & a_{n,k}^{(k)} & \cdots\cdots & a_{n,n}^{(k)} & b_n^{(k)}
\end{pmatrix}$$

第k步消元的主元素

第k步消元的主行

第k次消元，第k行的乘数：（i=k+1,k+2,…,n）

$$\boxed{m_{i,k} = \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}}}$$

增广矩阵的变化：（i,j=k+1,k+2,…,n）

$$\begin{cases} a_{i,j}^{(k+1)} = a_{i,j}^{(k)} - m_{i,k} a_{k,j}^{(k)} \\ b_i^{(k+1)} = b_i^{(k)} - m_{i,k} b_k^{(k)} \end{cases}$$

回代过程

最后得到的增广矩阵：

$$
\begin{pmatrix}
a_{1,1}^{(1)} & a_{1,2}^{(1)} & \cdots\cdots & a_{1,n}^{(1)} & b_1^{(1)} \\
 & a_{2,2}^{(2)} & \cdots\cdots & a_{2,n}^{(2)} & b_2^{(2)} \\
 & & \cdots\cdots & & \\
 & & & a_{n,n}^{(n)} & b_n^{(n)}
\end{pmatrix}
$$

最终结果的计算：

$$
\boxed{x_i = \frac{b_i^{(i)} - \sum\limits_{j=i+1}^{n} a_{i,j}^{(i)} x_j}{a_{i,i}^{(i)}}}
$$

为什么要选主元素

前面介绍的消元法都是按照自然顺序，即x_1、x_2、……、x_n的顺序消元的。有：

$$m_{i,k}=\frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}}$$

所以每一次消元的主元素都不能为0。如果按照自然顺序消元的过程中出现的$a_{k,k}^{(k)}=0$，那么消元无法继续进行下去。或者$|a_{k,k}^{(k)}|$很小，也会严重影响计算精度。

为什么要选主元素

例如（假设运算过程中使用单精度实数）：

$$\begin{bmatrix} 10^{-10} & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix} \longrightarrow \begin{bmatrix} 10^{-10} & 1 & 1 \\ & -10^{10} & -10^{10} \end{bmatrix}$$

解得：$x_1=0$，$x_2=1$

这个解与第二个方程差异很大。究其原因，因为消元过程中第一个方程所乘的系数过大，使得上式“吃掉”了下式，所以在结果中根本无法体现下式。

但如果调整一下顺序：

$$\begin{bmatrix} 1 & 1 & 2 \\ 10^{-10} & 1 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 1 & 2 \\ & 1 & 1 \end{bmatrix}$$

解得：$x_1=1$，$x_2=1$，这个解基本符合原方程

所以每次消元的主元素的绝对值应该尽可能大，使得与主行相乘的乘数尽可能小。

选主元素

$$
\begin{pmatrix}
a_{1,1}^{(1)} & a_{1,2}^{(1)} & \cdots\cdots\cdots & a_{1,n}^{(1)} & b_1^{(1)} \\
 & a_{2,2}^{(2)} & \cdots\cdots\cdots & a_{2,n}^{(2)} & b_2^{(2)} \\
 & & \cdots\cdots\cdots & & \\
 & & a_{k,k}^{(k)} \cdots\cdots & a_{k,n}^{(k)} & b_k^{(k)} \\
 & & \cdots\cdots & & \\
 & & a_{l,k}^{(k)} \cdots\cdots & a_{l,n}^{(k)} & b_l^{(k)} \\
 & & \cdots\cdots & & \\
 & & a_{n,k}^{(k)} \cdots\cdots & a_{n,n}^{(k)} & b_n^{(k)}
\end{pmatrix}
$$

进行第k次消元时，将$a_{k,k}$一下各元素（包括$a_{k,k}$）进行比较，将其中的最大者所在行与第k行交换。

无解的情况

如果在消元的过程中，增广矩阵出现这样一行：左侧各未知数的系数都为0，而右侧的常数项不为0，则意味着方程组无解。

无数组解的情况

在消元过程中，出现这样一行：各未知数的系数和常数项都为0。这相当于少了一个方程，也就是接下来的消元过程中，方程的个数少于未知数的个数，方程要么无解，要么有无数组解。下面讨论对于这样的方程，如何得到一组解。先看这样一个方程：

$$\begin{pmatrix} 4 & 2 & 3 & 9 \\ 2 & 1 & 1 & 4 \\ 2 & 1 & 2 & 5 \end{pmatrix} \longrightarrow \begin{pmatrix} 4 & 2 & 3 & 9 \\ 0 & 0 & -0.5 & -0.5 \\ 0 & 0 & 0.5 & 0.5 \end{pmatrix}$$

如果继续消元（消第2列），必须保证$a_{2,2}\neq 0$，可是第2列中不存在非0的项。

无数组解的情况

只能够把第3列的元素作为第2次消元的主元素，进行消元：

$$\begin{pmatrix} 4 & 2 & 3 & 9 \\ 0 & 0 & -0.5 & -0.5 \\ 0 & 0 & 0.5 & 0.5 \end{pmatrix} \longrightarrow \begin{pmatrix} 4 & 2 & 3 & 9 \\ 0 & 0 & -0.5 & -0.5 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

第2次消元得到的元素全部为0，所以第三行元素已失去意义。x_2没有做过主元素，可随意取值，再进行回代，得到一组可行解。如令$x_2=0$，$x_3=1$，$x_1=1.5$。

对于一般的线性方程组，先进行消元，每次消元前找到系数不完全为0的列，相应的元素作为此次消元的主元素，直至第k次消元时，得到的新元素全部为0，这时把各未知数分为两种：第k+1列至第n列对应的未知数，可以将这些未知数随意取值；第1列至第k列对应的未知数，这些未知数的值在回代过程中确定。

性能分析

时间复杂度： $O(n^3)$

消元 $O(n^3)$

选主元素： $O(n^2)$

回代 $O(n^2)$

空间复杂度： $O(n^2)$

增广矩阵 $O(n^2)$

如使用全选主元素，还需一个存储列与元素对应信息的表，为 $O(n)$

精度：

由于采用实数运算，另外每一次（第一次除外）消元都要使用以前消元产生的结果，每一次回代都要使用消元结果和其它回代结果，所以累积误差比较严重，该方法只能够求得近似解。但是可以根据具体需要进行相应改进。

整数线性方程组的精确解法

前面讨论了对于一般线性方程组通过实数运算得到近似解的算法。而在一些问题中，常常要求精确解，这里讨论一下系数、常数项和解均为整数的线性方程组的精确解法。

前面是用这种方法消元的：

$$\boxed{m_{i,k} = \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}}}$$

$$\boxed{\begin{cases} a_{i,j}^{(k+1)}=a_{i,j}^{(k)} - m_{i,k}a_{k,j}^{(k)} \\ b_i^{(k+1)}=b_i^{(k)} - m_{i,k}b_k^{(k)} \end{cases}}$$

显然这里进行的是实数运算。

整数线性方程组的精确解法

由于不能够保证$a_{i,k}^{(k)}$是$a_{k,k}^{(k)}$的倍数，要想消元，必须使两行分别乘以一个乘数。

$$m_{i,k} = \frac{[a_{i,k}^{(k)}, a_{k,k}^{(k)}]}{a_{k,k}^{(k)}}$$

$$m'_{i,k} = \frac{[a_{i,k}^{(k)}, a_{k,k}^{(k)}]}{a_{i,k}^{(k)}}$$

$$\begin{cases} a_{i,j}^{(k+1)} = m'_{i,k} a_{i,j}^{(k)} - m_{i,k} a_{k,j}^{(k)} \\ b_i^{(k+1)} = m'_{i,k} b_i^{(k)} - m_{i,k} b_k^{(k)} \end{cases}$$

方程较多时，系数有可能越来越大，到一定程度有可能导致系数越界，因此要随时对各行化简，即把这一行中所有元素除以这些元素的最大公约数。

但是，无论如何，这也保证不了不会发生越界，因此这种算法一般适用于系数、未知数范围较小，未知数个数较少的方程。

齿轮

你有一套玩具，包括许多不同尺寸的齿轮（至多20种，假定每一种齿轮有无限多个），每个齿轮最多100齿。你希望用它们构造不同比例的传动装置。一个传动装置包括偶数个齿轮，这些齿轮两两一组互相咬合，每一组齿轮都与下一组用轴承相连。用c_1、c_2、……、c_m表示每组第一个齿轮的齿数，用d_1、d_2、……、d_m表示每组第二个齿轮的齿数。

$$c:d=\prod_{i=1}^{m}\frac{C_i}{D_i}$$

例如你有3种齿轮：6齿、12齿、30齿，你需要实现4:5的传动比例，一种可行的方案是：使用4个齿轮，分2组，第1组的两个分别为12齿、6齿，第2组的两个分别为12齿、30齿。

简要分析

把这些齿轮的齿数设为a1、a2、……、an，设它们作为C类齿轮的数量分别为e1、e2、……、en，作为D类齿轮的数量分别为f1、f2、……、fn。有如下关系：

$$c:d=\prod_{i=1}^{n}\frac{a_i^{e_i}}{a_i^{f_i}}=\prod_{i=1}^{n}a_i^{e_i-f_i}$$

这时候我们不难发现，一种齿轮同时当作C类、D类使用是一种浪费。设$x_i=e_i-f_i$，$x_i>0$表示这种齿轮只作为C类，$x_i<0$表示这种齿轮只作为D类。这就转化为解x_i问题。

我们可以将c、d、a_i这些值分解质因数。由于a_i不超过100，所以$a_1 \ldots a_n$能够分解为的质因数不超过25种。另外，如果c或d中包括这以外的质因数，显然问题无解。

简要分析

设$g_{r,i}$为质数r在a_i的质因数分解中的指数，c_r、d_r分别为质数r在c、d中的质因数分解中的指数。有如下关系：

$$2\wedge(x_1g_{2,1}+x_2g_{2,2}+\ldots\ldots+x_ng_{2,n})=2\wedge(c_2-d_2)$$

$$3\wedge(x_1g_{3,1}+x_2g_{3,2}+\ldots\ldots+x_ng_{3,n})=3\wedge(c_3-d_3)$$

…………

这完全可以表示为关于指数的等式，即：

$$g_{2,1}x_1+g_{2,2}x_2+\ldots\ldots+g_{2,n}x_n=c_2-d_2$$

$$g_{3,1}x_1+g_{3,2}x_2+\ldots\ldots+g_{3,n}x_n=c_3-d_3$$

…………

$$g_{97,1}x_1+g_{97,2}x_2+\ldots\ldots+g_{97,n}x_n=c_{97}-d_{97}$$

当然还有一个约束条件：

$$x_1+x_2+\ldots\ldots+x_n=0$$

这就完全转化为了解线性方程组的问题，而且这需要精确地解出这个整数线性方程组，并且还要面临不定方程的问题，如果能够得出整数解，则问题有解。

小结

　　高斯消元法是一种比较简单、适用范围较广的有效算法，但在实际应用中，我们往往需要具体问题具体分析，对这样的标准算法进行改进，才能满足我们的需要。

谢谢

请多提宝贵意见