对于方程组Ax=b，A为n×m矩阵，如果A列满秩，且n>m。则方程组没有精确解，此时称方程组为超定方程组。

最常用的方法是最小二乘法

1. 广义逆（伪逆）

假定G为A($M \times N$)的广义逆矩阵，满足：

$AGA = A$

- 当A为方阵时：$G = A^{-1}$
- $M > N$时（行大于列）：$G = (A'A)^{-1}A'$
- $M < N$时（列大于行）：$G = A'(AA')^{-1}$
- 当**min(M,N)**满秩时，上述逆矩阵存在，对应的线性方程组解的二范数最小

```
x = np.linalg.pinv(A).dot(b)
```

参考：Python如何求解非方阵矩阵的线性方程组通解？ - 知乎 (zhihu.com)

病态矩阵

取决于条件数：

$cond(a) = \frac{\lambda_{max}}{\lambda_{min}}$

即最大特征值与最小特征值的比值

病态的根源

过大的条件数会导致矩阵的病态性，并不是根本原因，最根本的原因是矩阵的列之间的相关性过大

2. 最小二乘法

对于这样一个优化问题$min(||Ax-b||_2^2)$

- 如果A是一个full rank的矩阵。那Ax-b=0一定有非零解。所以，这个函数的最小值就是0，对应的x就是Ax-b=0的解
- 如果A是一个非方阵，或者非full rank。
  - 这个需要展开平方，使用最小二乘的方法求得最小值对应的x：x=$(A^TA)^{-1}A^Tb$
  - $(A^TA)^{-1}A^Tb$刚好也是非线性最小二乘每次迭代更新量的表达式。只是A换成了非线性方程的雅克比矩阵
  - 并且$(A^TA)^{-1}A^T$也是矩阵A的广义逆的表达式。
- 总结来说，因为定义了广义逆，我们可以认为最小二乘的解就是对应的线性方程组的解。

参考：[线性最小二乘，线性方程组以及广义逆的关系_ziliwangmoe的博客-程序员宅基地 - 程序员宅基地 (cxyzjd.com)]

1. 直接分解

高斯消去法的变体，包括SVD，QR分解等

SVD

包括丢弃最小特征值、加入规则项来求解病态矩阵

参考: [线性代数——最小二乘法——矩阵分解]

2. 稀疏算法

`scipy.sparse.linalg.lsqr`(*A*, *b*[, *damp=0.0*, *atol=1e-08*, *btol=1e-08*, *conlim=100000000.0*, *iter_lim=None*, *show=False*, *calc_var=False*, *x0=None*])[source]

Poor scaling of the rows or columns of A should therefore be avoided where possible（什么叫poor scaling）

比如说，如果有某一行（row行相对于其他列非常小或非常大，那么对应行应该scaled up or down

`damp`参数是用来regularize 病态系统，防止真解过大，另外一个regularization是`acond`，

预条件是另一种减少迭代次数的方法，如果求解相关系统$Mx = b$更有效率，那么M近似A很有帮助。

如果**A**是对称的，不那么**lsqr**不好用

`scipy.sparse.linalg.``lsmr`(*A*, *b*[, *damp=0.0*[, *atol=1e-06*[, *btol=1e-06*[, *conlim=100000000.0*[, *maxiter=None*[, *show=False*[, *x0=None*]]]]]]])[source]¶

lsqr可以求解三种问题：满秩，最小二乘和 正则化最小二乘，而**lsmr**只用于最后一种

2. 预条件

```python
import numpy as np
import scipy.sparse.linalg as spla

A = np.array([[ 0.4445,  0.4444, -0.2222],
              [ 0.4444,  0.4445, -0.2222],
              [-0.2222, -0.2222,  0.1112]])

b = np.array([[ 0.6667],
              [ 0.6667],
              [-0.3332]])
M2 = spla.spilu(A)
M = spla.LinearOperator((3,3), M2.solve)

x = spla.gmres(A,b,M=M)
```

预条件技术有以下几种：

- Jacobi — 雅可比预优因子：M是一个对角矩阵，其对角线上的元素为矩阵A 对角线元素的倒数
- ILU — 不完全LU分解预优因子：对大型稀疏矩阵A，它的完全LU分解的因子L和U的下三角与上三角部分都是满的矩阵。若取预优因子为M = LU，则从理论上来说是最优的，但从实际操作来说，面对大型的稀疏线性方程组来说，是难以实现且得不偿失的。为此，采用一种不完全的LU分解(Incomplete LU decomposition)方法，将矩阵A分解为L˜和U˜ ，它们的非零元素与完全分解的矩阵L和U对应的非零元素相同，非零元素的分布则与矩阵A的下三角和上三角部分一致[60]。将M = L˜ · U˜ 作为预优因子即为ILU预优因子。
- LIN(Low Induction Number) — 低感应数预优因子[61–63]：它用Helmholtz定理将电场分解为无旋场𝜙和无散场Ψ两个部分，再近似求解无散场部分Ψ的扩散方程的解，将其解作为预优因子。
- 其他：还有多项式法、不完全 Choleski 分解法等。在这些预优因 子中，低感应数法对 QMR 迭代收敛速度的加速效果最明显，可以将迭代次数从未做预优处理的上千次降低到几十次。不完全LU分解法的效果次之，可以将迭代次数降低到几百次。雅可比预优因子的效果再次之，其主要作用为提高迭代过程的稳定性