最小二乘法的三种数值计算方法

一、最小二乘法

对于一般线性回归问题的矩阵形式如下：

$$Y = Xw$$

其中X为自变量，y为因变量，w为回归参数。
该类问题又分了三种情况：

- $m = n$且X为非奇异矩阵，此时$Xw = y$有唯一解：$w = X^{-1}Y$
- $m < n$，即自变量的个数比样本数还多，此时$Xw = Y$有无穷解。
- $m > n$，约束个数比样本数多，此时$Xw = Y$有无解。

在实际情况中，绝大多数的情况都是第三种情况，即样本数比特征数大，因此在本文中只讨论第三种情况。上面第三种情况的问题的又名超定问题，并且一般情况下为非一致方程，因此方程无解，但可以转向求解最小二乘问题，其基本思想是最小化误差（损失）函数：

$$\min Loss(w) = \|Xw - Y\|_2^2$$

其中$X \in R^{m\times n}, w \in R^n, y \in R^m$。

对于其回归参数的求解又有多种求解算法，比如梯度下降法、qr 分解，正定矩阵法，奇异值分解等。而对于后三种方法既有联系又有区别，可以归类到数值分析研究去。以下针对这三种分别介绍。

二、最小二乘法的三种求解方法

2.1 正定矩阵法

损失函数：

$$Loss(w) = \sum_{i=1}^{m}\left(x^{(i)}w - y^{(i)}\right)^2 = \|Xw - Y\|_2^2$$

其中$x^{(i)}$为第 i个样本回归自变量向量；
$y^{(i)}$为第 i个样本回归因变量；
w为回归参数。

对上面损失函数乘以$\frac{1}{2}$，不影响求参数的求解，并用矩阵形式表示：

$$Loss(w) = \frac{1}{2}(Xw - Y)^T(Xw - Y)$$

$Loss(w)$对回归参数 w求导，并令结果取 0：

$$\begin{aligned}\nabla_w Loss(w) &= \tfrac{1}{2}\nabla_w (Xw - Y)^T(Xw - Y)\\ &= \tfrac{1}{2}\nabla_w (w^T X^T Xw - w^T X^T Y - Y^T Xw + Y^T Y)\\ &= X^T Xw - X^T Y\\ &= 0\\ \Rightarrow \quad w &= (X^T X)^{-1} X^T\end{aligned}$$

通过正规化方程的求解，可以利用公式算得使损失函数最小的情况下回归参数的值。但是在实际操作中可能会遇到一些情况，如多重共线性问题。即$X^T X$非正定，那么无法求逆，此时求解公式失灵。又或者$X^T X$趋于 0，导致$(X^T X)^{-1}$，继而影响到结果的稳定性、灵敏度。对于以上问题的解决方法一般是通过引入正则项来避免。此时正规方程变成：

$$w = (X^T X + \lambda I)^{-1} X^T Y$$

其中λ为大于 0 的正则系数，I为单位矩阵，矩阵大小与$X^T X$一致。

在正则化的现行回归模型中$(X^T X + \lambda I)$可以证明其不为奇异矩阵或退化矩阵。在此不详细证明。

2.2 SVD 奇异值分解法

在进行求解最小二乘回归前，先了解 SVD 奇异值分解的原理。

定理[1]**：**

对于任一矩阵$A \in R^{m\times n}$，都可以分解为：

$$A = U \cdot S \cdot V^T$$

其中，$U \in R^{m\times m}$*，*U为正交矩阵，U的列向量为AA^T的特征向量；

$S \in R^{m\times n}$*，*S*可以分解成*$[\sum,\ \mathbf{0}]^T$；$\sum$ 为对角矩阵，其值为矩阵A的奇异值，也即是AA^T的特征值；

$V^T \in R^{n\times n}$, V^T为正交矩阵，V^T的列向量为$A^T A$的特征向量；

最回到最小二乘法的求解上，损失函数：

$$
\begin{aligned}
Loss(w) &= \|Xw - Y\|_2^2 \\
&= \| U \cdot S \cdot V^T \cdot w - Y \|_2^2 \\
&= \| S \cdot V^T \cdot w - U^T \cdot Y \|_2^2 \\
&= \left\| \left[\Sigma , 0 \right]^T \cdot V^T \cdot w - [U_n, U_{m-n}]^T \cdot Y \right\|_2^2 \\
&= \left\| \begin{matrix} \Sigma \cdot V^T \cdot w - U_n^T \cdot Y \\ 0 - U_{m-n}^T \cdot Y \end{matrix} \right\|_2^2 \\
&= \left\| \Sigma \cdot V^T \cdot w - U_n^T \cdot Y \right\|_2^2 + \left\| 0 - U_{m-n}^T \cdot Y \right\|_2^2 \\
&\geq \left\| -U_{m-n}^T \cdot Y \right\|_2^2
\end{aligned}
$$

Note：上面第 4 行减号左边为S*可以分解成*$[\Sigma,\ \mathbf{0}]^T$；

上面第 4 行减号右边边为把$U = [u_1, u_2, \ldots, \quad _n, u_{n+1}, \ldots, u_m]$*可以分解成* $U = [U_n, U_{m-n}]$，其中$U_n = [u_1, u_2, \ldots, u_n]$；$U_{m-n} = [u_{n+1}, \ldots, u_m]$

其中，$\| -U_{m-n}^T \cdot Y\|_2^2$与回归参数$w$无关，因此，当有

$$\left\| \Sigma \cdot V^T \cdot w - U_n^T \cdot Y \right\|_2^2 = 0$$

在上面损失函数最后一行可以取等号，此时$\|Xw - Y\|_2^2$取最小值，仿照 2.1 正定矩阵法的求导求解得：

$$w = V \cdot \Sigma^{-1} \cdot U_n^T \cdot Y$$

2.3 QR 矩阵分解法

定理：

任意一个满秩矩阵矩阵A，都可唯一地分解$A = Q \cdot R$，其中Q为正交矩阵，R为具有正对角元的上三角矩阵[2]。

再回顾回归的原模型：$Xw = Y$，两边同坐乘X^T，得到：

$$\begin{aligned} & X^T X w = X^T Y \\ \Rightarrow\quad & (QR)^T \cdot QR \cdot w = (QR)^T \cdot Y \\ \Rightarrow\quad & R^T Q^T \cdot QR \cdot w = R^T Q^T \cdot Y \\ \Rightarrow\quad & R^T R \cdot w = R^T Q^T \cdot Y \\ \Rightarrow\quad & R \cdot w = Q^T \cdot Y \\ \Rightarrow\quad & w = R^{-1} \cdot Q^T \cdot Y \end{aligned}$$

三、三种方法的对比

3.1 鲁棒性对比

- 正定矩阵法：$X^T X$必须正定，也不适宜过小。
- SVD 奇异值分解法：任意输入矩阵，没限制条件。
- QR 矩阵分解法：输入矩阵为满秩矩阵，否则失灵。

3.2 计算效率及精度对比

对于正定矩阵法，需要计算$X^T X$的逆，由于$X^T X$为$n \times n$矩阵，实际计算开销相对大。在 Matlab 软件中，用正定矩阵法的常规求解代码为：

```
W = inv(A’ * A) * A’ * b
```

但可以改进为如下：

```
W = (A’ * A) \ A’ * b
```

虽然它和直接求逆再相乘的计算复杂度都是立方复杂度，但后者运用“\”进行倒除实际上进行了 LU 矩阵分解，其时间复杂度为前者的直接求逆的三分之一。

而先对输入矩阵X进行 SVD 分解后，公式中Σ 为对角矩阵，求逆相对简单，且U与V都为正交矩阵，其逆矩阵与其转置相等。计算开销较小。通常情况下Σ是按照奇异值由大到小排列的，且衰减速度特别快，一般前 10%的奇异值之和就占到 95%以上。因此在适当牺牲计算精度的情况下，可以将小于某个阈值的奇异值及其对应的左右奇异值对应的特征向量全部舍弃掉，进而对矩阵的规模缩减。

对于 QR 矩阵分解法的计算效率更是比前两者更快。为了更优雅地比较 QR 矩阵分解法与 SVD 奇异值分解法的复杂程度。定义矩阵$\boldsymbol{A}$的条件数[3]（condition number)：

$$cond\ \boldsymbol{A} = \frac{\max \sigma_i}{\min \sigma_i}$$

其中σ_i为矩阵的奇异值。

考虑两种特殊情况：

- $\boldsymbol{A}$为单位矩阵$\boldsymbol{I}$，那么矩阵$\boldsymbol{A}$的所有奇异值都为 1，$cond\ \boldsymbol{A} = 1$；
- $\boldsymbol{A}$为奇异矩阵，那么$\boldsymbol{A}$有等于 0 的奇异值，$cond\ \boldsymbol{A} = \infty$。

对于一般矩阵，其条件输都为 1 和∞之间。通常情况下，矩阵的条件数越低越好。由于计算机计算是看浮点精度多，有时候计算呢并不是那么精确，因此条件数越高，计算精度的误差对解的影响也越大。对于$w = (X^T X)^{-1} X^T$ ，如果对X进行 SVD 分解，其前半部分的条件数为：

$$cond\quad {}^T A = cond\,(UEV^T)^T \cdot (UEV^T) = cond\ VE^2 V = (cond\ A)^2$$

也就是说矩阵$A^T A$是矩阵A的平方，即是如果进行 SVD 分解，其复杂程度为矩阵A的平方倍。但进行 QR 分解的话，可以绕过了条件数被平方的这样的操作。

更深入地，QR 分解一般由三种分解方法，分别为 Schmidt 正交化变换、Householder 变换、Givens 变换。三种方法都有所区别，对于 Schmidt 正交化变换则是限制输入矩阵A为可逆矩阵。而对于 Householder、Givens 变换则没限制。而从时间复杂度来看，利用 Givens 矩阵分解需要作$\frac{n(n-1)}{2}$个初等旋转针的连乘积，当 n 比较大时开销较大。因此通常用 Householder 变换进行 QR 分解。另外，在 Matlab 中默认的 QR 分解方法为 Givens 变换。

参考文献

[1] 鲁铁定， 宁津生， 周世健,等. 最小二乘配置的SVD分解解法[J]. 测绘科学， 2008, 33(3):47-51.

[2] 鲁铁定， 宁津生， 周世健,等. 最小二乘配置的QR分解解法[J]. 辽宁工程技术大学学报:自然科学版， 2009， 28(4):550-553.

[3]杨大地. 矩阵条件数的新定义—矩阵的非正交度[J]. 重庆大学学报， 1996， 19(6):61-65.