深 圳 大 学 实 验 报 告

实验课程名称：　　　　最优化方法

实验项目名称：　　　　最小二乘法实验

学院：　计算机与软件学院　　专业：　软件工程（腾班）

报告人：　黄亮铭　学号：　**202215502**　班级：　腾班

同组人：

指导教师：　　　　李炎然

实验时间：　**2024** 年 **12** 月 **6** 日 **- 2024** 年 **12** 月 **26** 日

实验报告提交时间：　**2024** 年 **12** 月 **26** 日

教务处制

实验报告包含内容

1 实验目的与要求

1） 熟练最小二乘法优化模型的意义和求解手段；

2） 掌握最小二乘法的正规方程，能实现代码对其求解；

3） 掌握最速梯度下降法求解无约束最小二乘法问题。

2 问题

读取附件“MatrixA_b.mat”文件中的矩阵A和向量b。建立关于矩阵 $A \in R^{m\times n}$， 向量 $b \in R^{m}$，未知向量 $x \in R^{n}$ 最小二乘优化模型：

$$\min_x \|Ax-b\|_2^2$$

1）通过最小二乘法的正规方程，求出优化模型的准确解；

2）利用梯度下降法迭代求出模型“近似解”，通过设置迭代停止条件，分析“近似解”与“准确解”之间的误差。

3 模型建立及求解

3.1 最小二乘法求解

由于各种误差，在某些情况下难以求解出一组满足问题条件的解。而最小二乘法给出的解决方案就是寻找该问题的近似解，并尽可能地逼近原问题的解，使$r = Ax - b$尽可能的小。

最小二乘法的数学模型如下所示：

$$\min_x \|r\|_2^2 = \min_x \|Ax - b\|_2^2$$

3.1.2 正规方程推导

目标函数为：

$$\hat{x} = \underset{x}{\operatorname{argmin}} \|Ax - b\|_2^2$$

结合最小二乘法的原理，对上述目标函数进行求解，得到的$\hat{x}$应该满足如下的条件。

$$\|A\hat{x} - b\|_2^2 \le \|Ax - b\|_2^2$$

由于目标函数$f(x)=\min_{x}\|Ax-b\|_2^2$为可微函数，因此最优解$\hat{x}$满足梯度$\nabla f(x)=0$。

对$f(x)$求导得到：

$$\nabla f(x)=2A^T(Ax-b)$$

即：

$$2A^T(Ax-b)=0$$

也即：

$$A^TAx=A^Tb$$

由此，我们通过最小二乘法得到了正规方程。如果A^TA为非奇异矩阵，则此时正规方程（原问题）有唯一解。

3.1.3 正规方程求解

对于正规方程，我这里提供了两种求解方法。一种是使用高斯消元法直接对正规方程进行求解，另一种是使用 QR 分解进行求解。

这里重点讲解如何使用 QR 分解进行求解。

首先我们对A进行 QR 分解，得到$A=QR$。然后代入正规方程$A^TAx=A^Tb$，化简后得到$x=R^{-1}Q^Tb$。

3.1.4 代码实践

在代码实现上，对于 QR 分解方面，我们可以选择使用实验 3 中自己编写的 QR 分解代码，也可以选择 matlab 库函数进行 QR 分解。为了结果的准确性，我选择使用 matlab 的库函数进行 QR 分解。

对于矩阵求逆方面，我们可以选择实验 1 中自己编写的高斯消元法求矩阵的逆的代码进行求解，也可以选择 matlab 的库函数进行求解。同样是为了准确性，我也选择了使用 matlab 的库函数进行矩阵求逆的操作。

通过 QR 分解和矩阵求逆的步骤之后，我们就可以很轻松地得到最优解的精确解（图 1）。

1	-0.4106	21	0.1366
2	-0.4014	22	0.4923
3	-0.2396	23	0.6429
4	0.1844	24	-0.1005
5	0.0198	25	-0.3836
6	0.1678	26	0.3558
7	0.0910	27	0.1383
8	0.4264	28	-0.1622
9	0.1890	29	0.0608
10	-0.4568	30	-0.3770
11	0.0710	31	0.3445
12	0.2867	32	0.1276
13	0.3330	33	-0.2797
14	-0.4083	34	0.0727
15	-0.1434	35	0.4069
16	-0.3200	36	-0.1634
17	-0.3985	37	-0.0721
18	0.0116	38	0.1655
19	-0.5994	39	0.0888
20	0.6976	40	0.1581

图 1 精确解

3.2 梯度下降法求解

除了最小二乘法，梯度下降法也常用于优化问题最优解的逼近。梯度下降法的核心原理为：函数在某点的梯度方向，是函数值增长最快的方向。那么，梯度下降法就反其道而行之，沿着梯度的反方向去更新自变量，使得函数值逐步减小。

3.2.1 迭代公式推导

设函数$f(x)$可微，根据泰勒公式，在$x^{(k)}$的一阶公式为：

$$f(x^{(k+1)}) = f(x^{(k)}) + \langle \nabla f(x^{(k)}), x^{(k+1)} - x^{(k)} \rangle + o(\| x^{(k+1)} - x^{(k)} \|)$$

如果$\| x^{(k+1)} - x^{(k)} \|_2$足够小，则有

$$f(x^{(k+1)}) - f(x^{(k)}) \approx \langle \nabla f(x^{(k)}), x^{(k+1)} - x^{(k)} \rangle$$

$$|\langle \nabla f(x^{(k)}), x^{(k+1)} - x^{(k)} \rangle| \leq \| \nabla f(x^{(k)}) \|_2 \| x^{(k+1)} - x^{(k)} \|_2$$

即：

$$\left\langle \nabla f\left(x^{(k)}\right), x^{(k+1)} - x^{(k)} \right\rangle \geq -\| \nabla f(x^{(k)}) \|_2 \| x^{(k+1)} - x^{(k)} \|_2$$

当$x^{(k+1)} - x^{(k)} = -\alpha_k \nabla f\left(x^{(k)}\right), \alpha_k > 0$时，等式成立。

由此我们可以推出迭代公式：$x^{(k+1)} - x^{(k)} = -\alpha_k \nabla f\left(x^{(k)}\right)$ $f(x^{(k+1)}) < f(x^{(k)})$

3.2.2 求解过程

为了消去求导之后的常数，我们可以在目标函数的前面乘以一个常量，该常量不会影响目标函数的最优解，即我们将目标函数从$\min\limits_{x}\|Ax-b\|_2^2$乘以一个常量变为$\min\limits_{x}\frac{1}{2}\|Ax-b\|_2^2$。

令$f(x)=\frac{1}{2}\|Ax-b\|_2^2$，我们可以得到

$$\nabla f(x)=A^T(Ax-b)$$

通过迭代下降求解：

$$x^{(k+1)}-x^{(k)}=-\alpha^{(k)}A^T(Ax^{(k)}-b)$$

此外，我们还可以通过线性搜索估计$\alpha^{(k)}$：

$$\alpha^{(k)}=\underset{\alpha\in R}{\operatorname{argmin}} f(x^{(k)}-\alpha^{(k)}A^T(Ax^{(k)}-b))$$

即$\alpha^{(k)}$是最佳步长。令$g(a)=f(x^{(k)}-\alpha^{(k)}A^T(Ax^{(k)}-b))$是关于$\alpha$的凸函数，则有

$$\min_{\alpha} g(a) => g'(a)=0 => \alpha^{(k)}=\frac{\left\|A^T(Ax^{(k)}-b)\right\|_2^2}{\|AA^T(Ax^{(k)}-b\|_2^2}$$

上述就是我们完整的求解过程。经过上述步骤，我们成功得到了迭代求解公式$x^{(k+1)}-x^{(k)}=-\alpha^{(k)}A^T(Ax^{(k)}-b)$和$\alpha^{(k)}$的最佳步长$\frac{\left\|A^T(Ax^{(k)}-b)\right\|_2^2}{\left\|AA^T(Ax^{(k)}-b\right\|_2^2}$。

3.2.3 代码实践

代码编写思路如下：

1) 初始化$x^{(0)}$。
2) 循环：①求解$\nabla f(x)=A^T(Ax-b)$；②求解$\alpha^{(k)}=\frac{\left\|A^T(Ax^{(k)}-b)\right\|_2^2}{\left\|AA^T(Ax^{(k)}-b\right\|_2^2}$；③使用迭代公式进行更新$x^{(k+1)}-x^{(k)}=-\alpha^{(k)}A^T(Ax^{(k)}-b)$。
3) 重复步骤 2)，直到满足迭代条件。

在迭代条件方面，我选择了双迭代条件判断循环是否结束。第一个迭代条件是迭代次数，第二个迭代条件则是相邻迭代解之间的“相对接近程度”$\left\|x^k-x^{k+1}\right\|_2/\left\|x^k\right\|_2$。

为探究最优迭代次数，应观察分析不同迭代次数对于目标求解的影响。本次实验中，迭代次数与目标函数值之间的关系如图 2 所示。

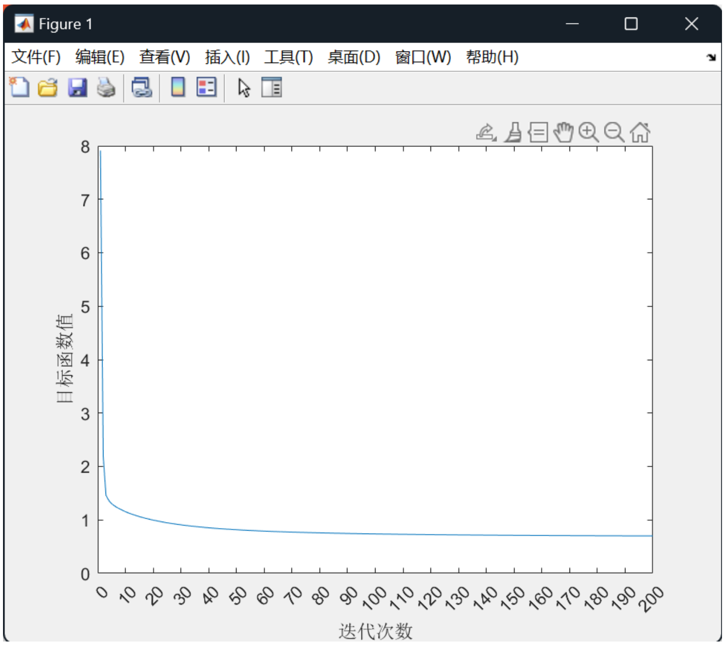

图 2 迭代次数与目标函数值

从图中我们可以看出，迭代次数在 40 之后，随迭代次数增大，目标函数趋于稳定。故在本实验中，选取迭代次数为 40 作为循环停止条件是一个比较好的选择。

同理，为探究最优阈值，应观察分析不同阈值对于目标求解的影响。本次实验中，迭代次数与相邻迭代解之间的“相对接近程度”之间的关系如图 3 所示。

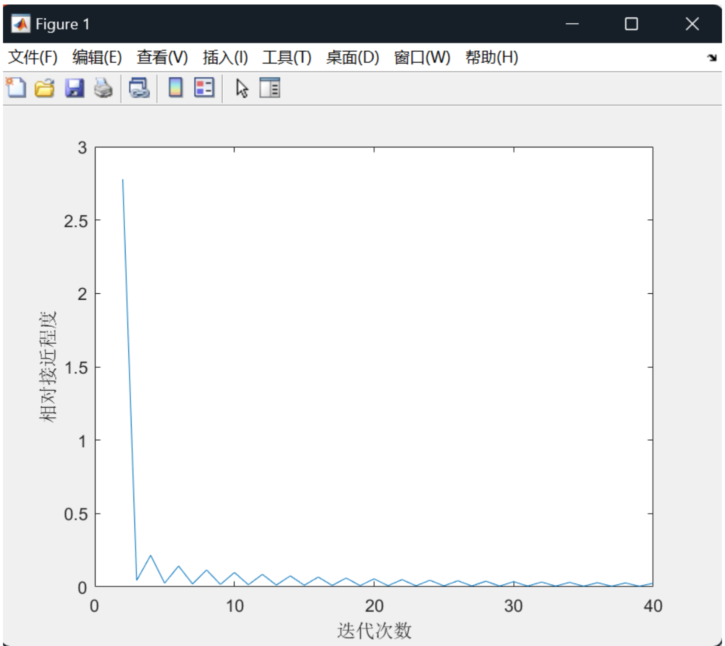

图 3 迭代次数与相邻迭代解之间的“相对接近程度”之间的关系

从图 3 中我们可以看出，随迭代次数增加，相邻迭代解之间的“相对接近程度”波动下降。经统计分析后，我选择相邻迭代解之间的“相对接近程度”的阈值为 0.001,作为梯度下降法的

循环停止条件。

3.3 误差分析

首先我们需要比较最小二乘法和梯度下降法之间的区别。

1) **原理**：最小二乘法基于数学推导，通过对误差平方和函数求导，令导数为零得出正规方程，直接求解得到使误差平方和最小的解析解。梯度下降法是迭代式优化算法，依据目标函数当前的梯度方向，朝着函数值下降最快的反方向，按设定的学习率更新参数，持续逼近目标函数最小值。

2) **收敛性**：最小二乘法对于这类凸优化问题，只要相关矩阵（本次实验为A^TA）可逆，就能直接获取全局最优解；而梯度下降法收敛速度受学习率、数据特征以及初始化等因素影响，在非凸函数场景下可能陷入局部最优解，不过凸优化场景（本实验）理论上也能收敛到全局最优解。

由上述最小二乘法和梯度下降法之间的区别分析，我们可以知道“精确解”和“近似解”之间的误差主要来源于梯度下降法的初始化的初始解。这个解往往与“准确解”的距离较远，所以每一次迭代的步长的方向和长度都是尽量“减小”误差，但是由于步长的问题，最后得到的解可能还是会与“准确解”存在一定的误差。

图 4 是比较不同迭代次数下（忽略“相对接近程度”这一循环停止条件）的“近似解”和“精确解”之间的误差，误差的计算方法为$error = \|x_{similar} - x_{least}\|_2$。

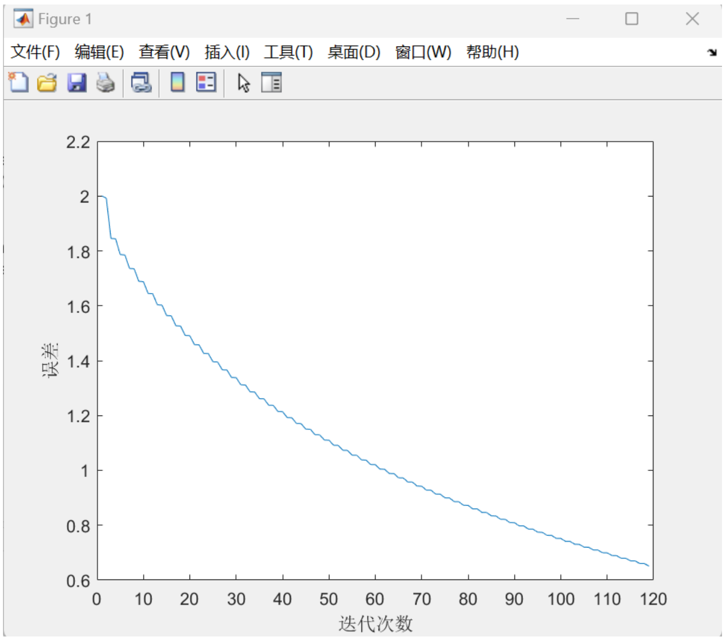

图 4“近似解”和“精确解”之间的误差

由图 4 可以看出，即使迭代次数再多，它们之间仍然存在误差。迭代次数为 120 的误差近似为 0.65139。

4 小结（可含个人心得体会）

1) 经过本次实验，我熟悉了如何通过最小二乘法的正规方程求解最小二乘优化模型。
2) 通过本次实验，我熟悉了如何使用梯度下降法迭代求解问题。通过不断迭代，解可以不断接近最优解，但是仍然存在误差。
3) 在本实验中，“精确解”和“最优解”之间存在误差。这个误差可能是由于初始值和步长引起的，在其他非凸优化问题中，误差还可能是因为陷入了局部最优问题。一个解决办法为使用模拟退火算法，即算法有概率选择次优解而非每次都选择最优解。

指导教师批阅意见：   成绩评定：   指导教师签字： 2024　年　12　月　　日
备注：

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后 10 日内。