AU7021 Optimization in Learning and Control (Fall 2021) **Homework: 4**

4. Solving Algorithm

Lecturer: Xiaolin Huang xiaolinhuang@sjtu.edu.cn

Student: Jincheng Wang jc.wang@sjtu.edu.cn

Problem 1

Consider the following LS problem

$$\min_x \quad \|Ax-b\|_2^2. \tag{1}$$

The data are generated by the following steps:

- randomly generate the true answer $\bar{x} \in R^n$ from $x_i \sim N(0,1)$;
- randomly generate $A \in R^{m\times n}$ from $a_{ij} \sim N(0,1)$;
- calculate $b = Ax + \varepsilon$ with $\varepsilon \sim N(0,\sigma^2)$, where the σ is set such that the signal-to-noise ratio is 20.

1. set $n = 50$ and $m = 200$ and implement GD algorithm to solve (1). You could try exact line search, backtracking line search, or fixed learning rate. Observe and discuss the solving procedure by plotting the objective value, $\|x^k - \bar{x}\|$, the difference to the pseudo-inverse result, and the length found step in each iteration.
2. modify your mode to SGD and change $n = 300, 500, 1000, 2000$ (m is fixed to 200). Try to use different starting points and report your result. Compare it with $\hat{x} = A^\top(AA^\top)^{-1}b$ and discuss the expected over-fitting phenomenon.

Answer. 1. 利用梯度下降法，对目标函数求梯度得 $\nabla||Ax-b||_2^2 = 2A^T(Ax-b)$，使用精确线搜索求步长 γ，变量更新公式为

$$x^{k+1} = x^k - \gamma \cdot 2A^T(Ax-b)$$

初始值为正态分布的随机值，迭代停止准则为若 $|x^{(k+1)} - x^{(k)}|$ 小于 10^{-8} 则退出迭代，步长 γ 使用黄金搜索法确定。最终算法得到的结果如图 1 和图 2所示。其中，目标函数 $||Ax-b||_2^2$ 采用了**半对数图**绘制，可以看到迭代次数在 15-20 次时，已经接近最优解了。

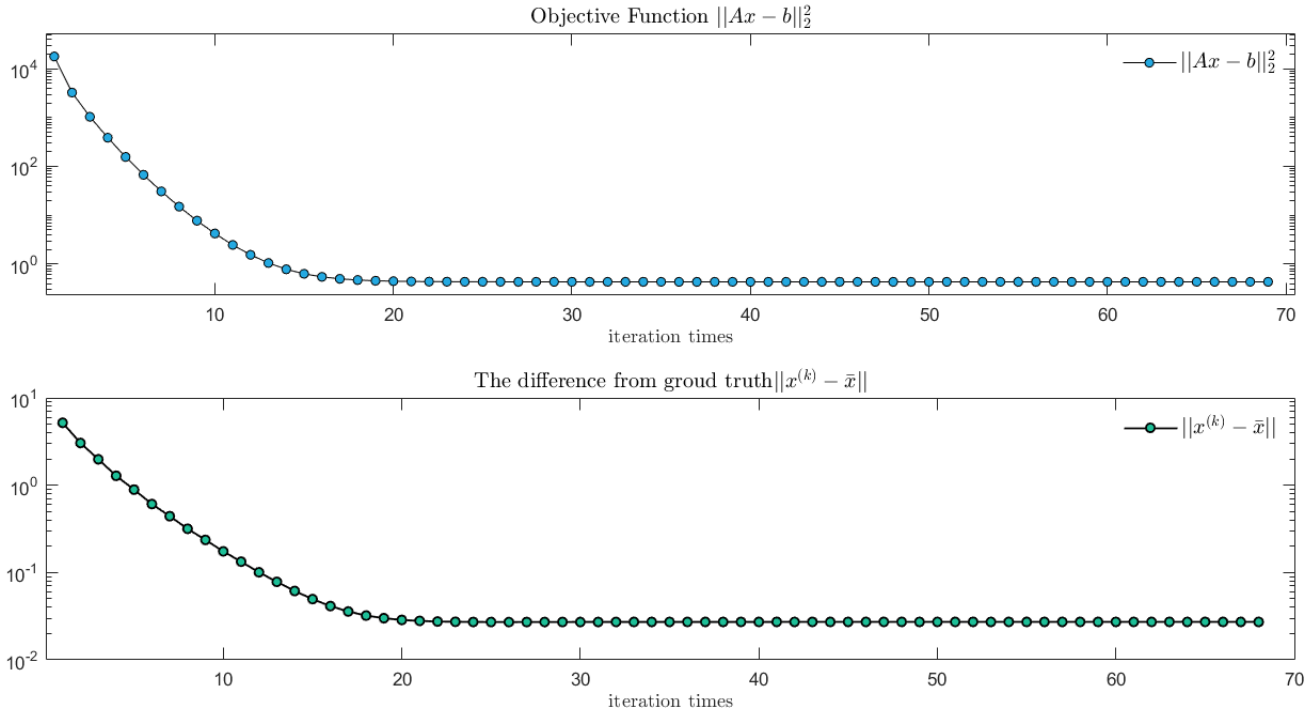

图 1: 目标函数和 $||x^k - \bar{x}||$ 随着迭代次数的下降图

利用梯度下降算法得到的估计结果 x 和使用矩阵伪逆 $x_{pseudo} = (A^TA)^{-1}A^Tb$ 计算结果与真值 $\bar{x}$ 比较如图 2 所示，分别使用粉色，墨绿色和绿色表示，三者计算的结果非常接近，在图中几乎看不出数值上的区别。事实上，**使用伪逆计算出来的数值解就是在求该问题的最小二乘解**，因此其与梯度下降法得到的解是几乎一致的。通过比较二者的值，发现 $|x - x_{pseudo}|$ 的量级在 10^{-6} 左右，而由于 ϵ 噪声的存在，$|x - \bar{x}|$ 的量级在 10^{-3} 左右。

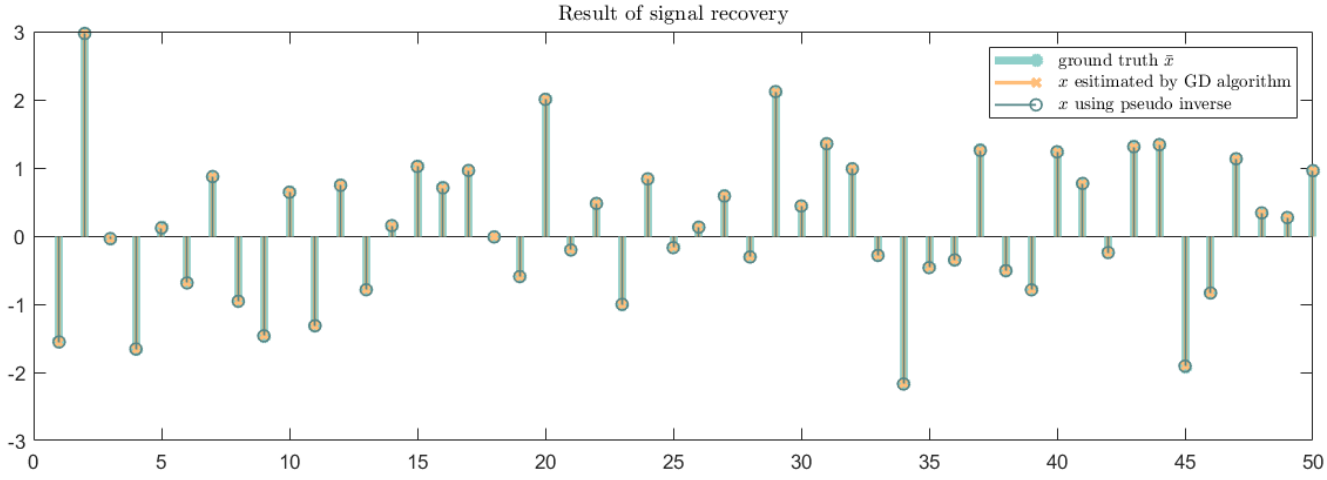

图 2: 估计值和矩阵伪逆计算的值与**真值**的比较

对于步长，本次算法使用黄金分割法作为线搜索的方法，阈值 ϵ 设为 0.0001，即若选取的左值和右值的差小于该阈值，则停止搜索算法。其步长曲线图如图 3 所示，可见其步长是震荡的，事实上，由于**梯度下降法连续两**

次的下降方向是相互正交的，因此这种趋势大致是可以预料到的。

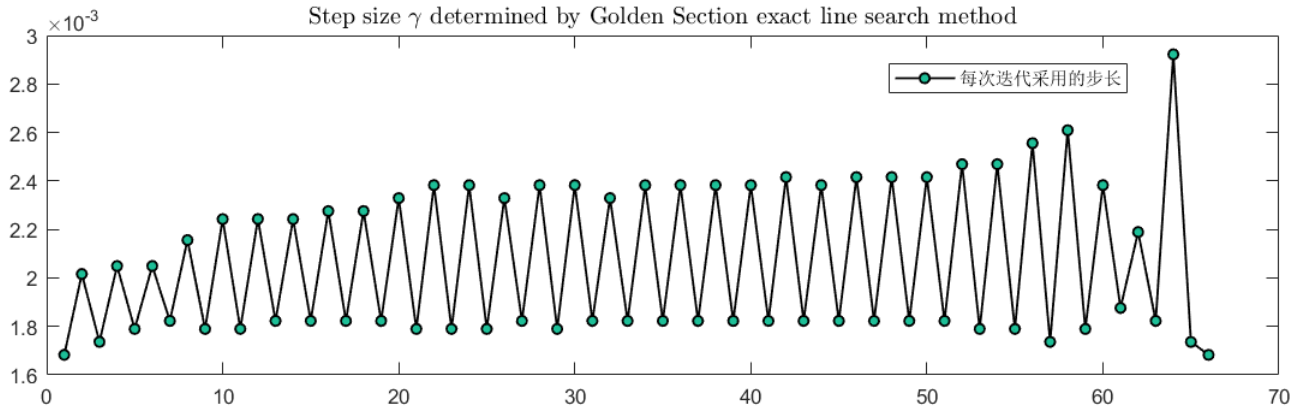

图 3: 通过**黄金分割法**进行精确线搜索所得到的每次迭代使用的步长

2. 修改第 (1) 问的代码，使用随机梯度下降法 (SGD) 求解。在观测样本较多的情况下 (对应 m 较大的情况)，对于每次更新，如果仍然利用所有观测值进行梯度的计算，那么效率会较低。随机梯度下降法选取随机的部分观测值以代替全部观测值进行梯度的更新。对于本次作业，选取 1 个随机样本代替所有样本的梯度进行梯度更新，其结果如图 4 所示, 可以发现，虽然目标函数下降到一个较小的值，**但是其与真实值仍然有较大的偏差**。

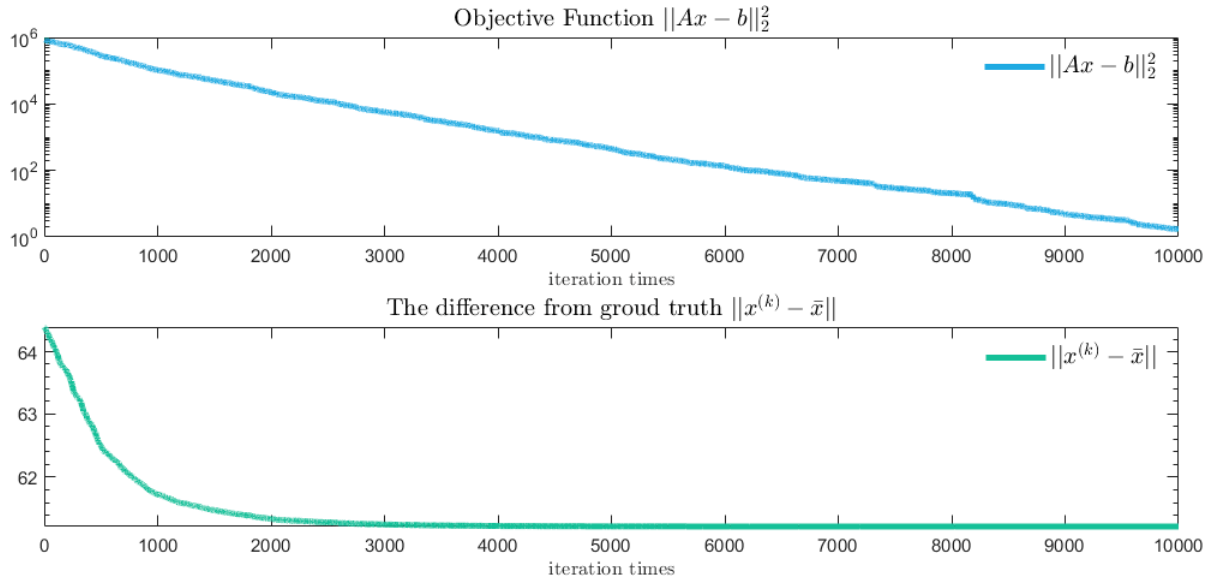

图 4: 目标函数和 $||x^{(k)}-\bar{x}||$ 随着迭代次数的下降图

如图 5所示，利用伪逆计算出来的值和用梯度下降法计算出来的值都不能很好的拟合原来的真实值，该图是在初始值为 $x \sim N(0,1)$,$n = 2000$ 时生成的。

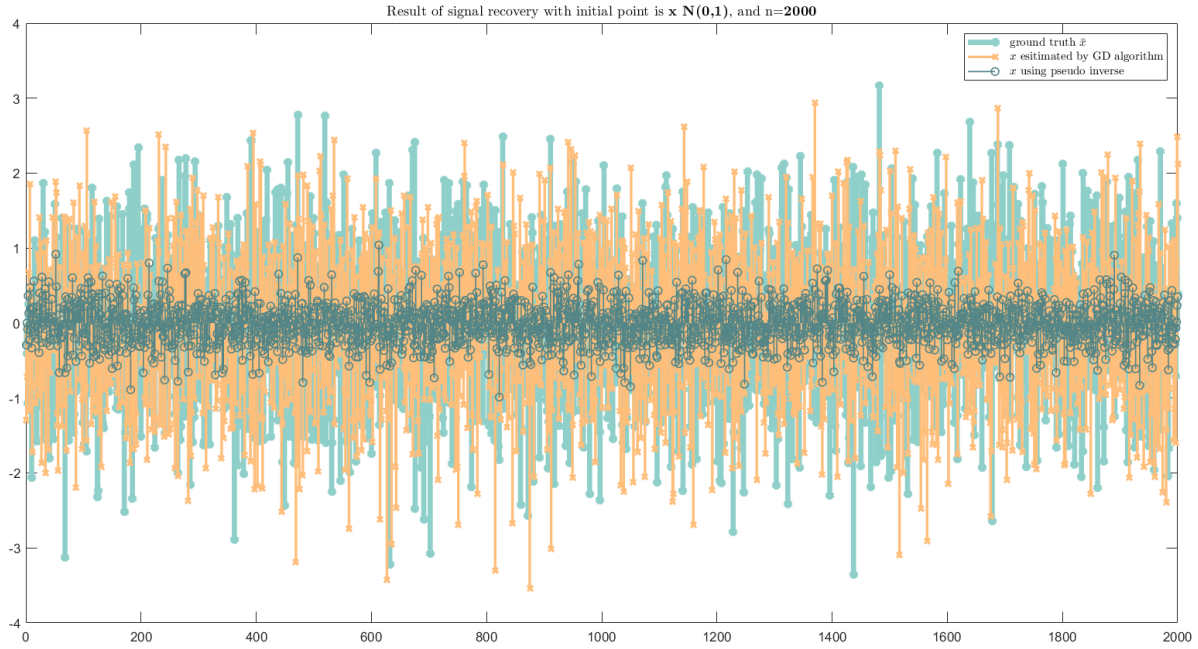

图 5: 估计值和矩阵伪逆计算的值与**真值**的比较

下面讨论梯度下降算法在不同初始值 (分别是正态分布、全零和全一) 和不同 n 值时信号复原的情况并与使用矩阵伪逆计算得到的值进行比较，如图 6所示。

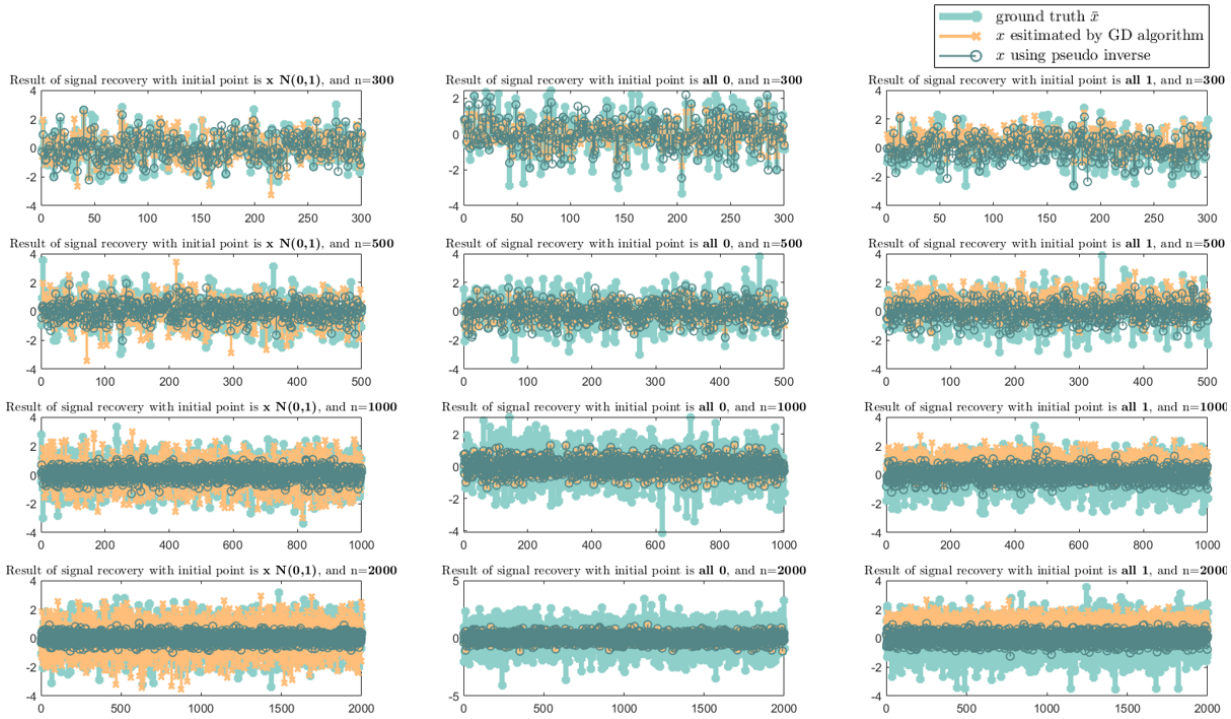

图 6: 梯度下降法的估计值和使用矩阵伪逆计算的值在不同初始值和不同 n 值时与**真值**的比较

可以看出:

- 在初始值为正态分布的情况下，算法生成的值和使用矩阵伪逆生成的值与真值都有很大的区别;
- **在初始值全为 0 的情况下，n 值较大的情况下，梯度下降法复原的结果和使用矩阵伪逆复原的结果非常相似**;
- 在初始值全为 1 的情况下，n 值较大的情况下，复原出来的信号大多是大于零的，这与初始值的设定有关。

事实上，通过矩阵伪逆计算出来的解 $x_{pseudo} = A^T(AA^T)^{-1}b$ 是等式 $Ax = b$ 的无穷多解中模最小的解，而当梯度下降法取初始值为 0 时，得到的解也是无穷多解中模最小的解，二者结果的近似如图 7所示。

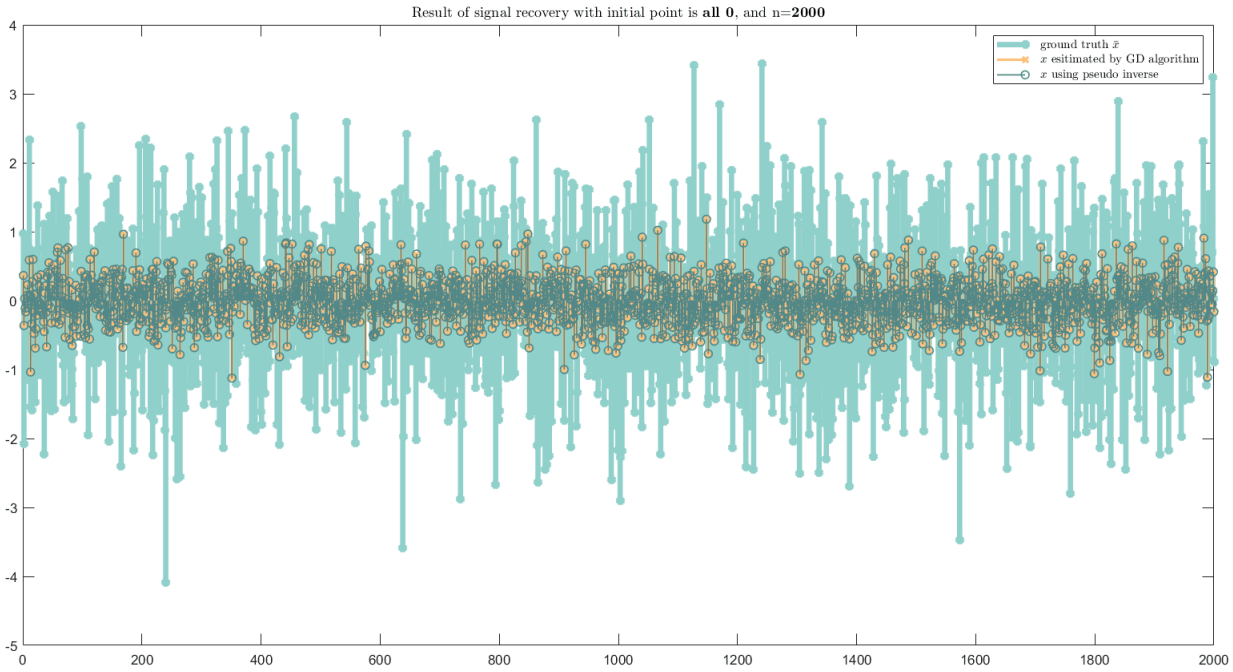

图 7: 初值为 0 时，梯度下降法和矩阵伪逆算法得到的结果非常相似

本次作业中，m 可以看做是观测量，固定为 200，而 n 是我们的待优化变量，在第 (2) 问中，$n > m$, 此时使用梯度下降法无法复原真实的信号，这是因为发生了**过拟合现象**，此时 $Ax = b$ 有无穷个解，而 $\bar{x}$ 只是其中的一个解，利用**矩阵伪逆计算出来的解是模最小的解**而不能保证是真实的解；利用**梯度下降法也只是求在当前目标函数中能使目标函数最小的解**，也不能保证是真实的解。因此对于这种现象，我们应当利用其他先验知识，例如根据数值的结构信息而在目标函数中引入正则化项进行优化，这样才能得到一个较好的结果。