

Homework 5

陈远洋 23307130322

2025 年 4 月 7 日

Problem 1

Consider the smoothed LASSO (Least Absolute Shrinkage and Selection Operator) problem:

$$\min_x \frac{1}{2} \|Ax - b\|_2^2 + \mu L_\sigma(x),$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, μ is the regularization parameter. $L_\sigma(x)$ is defined as

$$L_\sigma(x) = \sum_{i=1}^n l_\sigma(x_i),$$
$$l_\sigma(x_i) = \begin{cases} \frac{1}{2\sigma} x_i^2, & |x_i| < \sigma, \\ |x_i| - \frac{\sigma}{2}, & \text{otherwise.} \end{cases}$$

Let $\sigma = 0.5$ and $\mu = 1$. Please use the Newton method to numerically solve above problem with A and b provided in homework 3, where $m = 512, n = 1024$. Then write a report to illustrate the method you used and present your experiment results.

Note:

- To ensure the Hessian matrix is non-singular, you can add a scaled identity matrix (e.g. $1 \times 10^{-8} I_n$) to the original matrix.
- Please write your method as an independent function in your code.

Solution

解存在性分析

首先，该问题一定是有解的。类似于上一次的作业，我们只讨论 $x \in A$ 的零空间的正交补情况（其他情况可以转化）。记 $t^2 = \min_{\|x\|_2=1} \|Ax - b\|_2^2$ ，则有当 $\|x\|_2 \geq \frac{2\|b\|_2 + 1}{t}$ 时。由于 $\frac{\|x_i\|_2^2}{2\sigma} - \|x_i\|_2 + \frac{\sigma}{2} = \frac{1}{2\sigma}(\|x_i\|_2 - \sigma)^2 \geq 0$ 。且 $\|Ax - b\|_2 \geq \|Ax\|_2 - \|b\|_2 \geq t\|x\|_2 - \|b\|_2 \geq 0$ ，此时有：原式 $\geq \frac{1}{2}(\|x\|_2 t - \|b\|_2)^2 + \mu(\sum_{i=1}^n |x_i| - \frac{n\sigma}{2}) \geq \frac{1}{2}t^2\|x\|_2^2 + (-2tb + \mu)\|x\|_2 + O(1)$ 。这是一个关于 $\|x\|_2$ 的二次函数，可知在 $\|x\|_2 \rightarrow \infty$ 时，原式一定也区域无穷。又由于在 $\|x_i\|_2 = \sigma$ 处，两个分段函数均趋近于 $\frac{\sigma}{2}$ ，故原式在小范围内连续。所以原式一定有全局最优解。

代码构成分析

牛顿法求解优化问题的几个步骤：

Algorithm 1 Newton method(LM)

```
1: Input  $f, x_0, tolerance$ ,  
2:  $k \leftarrow 0$   
3: while  $\nabla f(x_k) \geq tolerance$  do  
4:   let the descent direction  $p_k$  be  $-(\nabla^2 f(x_k) + \lambda I)^{-1} \nabla f(x_k)$ .  
5:   pick a step size  $\alpha_k$  (initially may not be 1, finally always 1).  
6:   update  $x_k : x_{k+1} \leftarrow x_k + \alpha_k p_k$ .  
7:    $k \leftarrow k + 1$ ;  
8: end while
```

其中目标函数为 LASSO，在我的代码中实现在最后，名字为： $res = lasso(A, b, \sigma, \mu, x)$ 对应的梯度函数为 $dfres = dflasso(A, b, \sigma, \mu, x)$ 。由于问题关心的最优解是 x ，所以我在代码文件的最前面将目标函数与梯度函数定义为 f, df 两个关于 x 单一变量的函数句柄（带入了预设值得参数值，可以改变）。

而下降方向的选择上，采用牛顿法对应的方向。即 $-(\nabla^2 f(x_k) + \lambda I)^{-1} \nabla f(x_k)$ 。其中由于该问题对应的 Hessian 阵 $A^T A + diag(0 \text{ or } \frac{\mu}{\sigma})$ 可能奇异，所以加上正则项 λI 使得矩阵非奇异。对 λ 的选择，既不能太大，也不能太小。过大会导致下降方向与牛顿方向偏差过大，使下降效果变差。而 λ 过小则会使矩阵接近奇异。对此我的试验中 $\lambda \in [10^{-10}, 10^{-4}]$ 的效果较好。我的代码以 10^{-6} 为例。这部分对应逻辑在 $d = direction(df, x, A, sig, miu)$ 函数。

对于线搜索上，采用 armijo 方法要求每步至少下降一点，即 $f(x + \alpha d) \leq f(x) + c_1 \alpha \nabla f(x)^T d$ 。我们知道 $\nabla f(x_k)^T d = -\nabla f(x_k)^T H \nabla f(x_k) < 0$ 。H 是加上正则项后的 Hessian 阵的逆，正定。所以该方向一定能够在小范围内下降。原始代码如下：

其中在参数选择上，初始步长 $a0 = 1$ ，下降斜率 $c1 = 0.1$ ，后退步长 $beta = 0.6$ 。但是参数设

置其实对结果影响不大。最终在 5 ~ 6 步之内，结果已经收敛，并且每步步长为 1 时都能过关。所以只要将 $c1$ 设置得足够小即可。

```

1      function a0 = armijo(f, df, d, x, a0, c1, beta)
2          f0 = f(x); df0 = dot(df(x), d);
3          fx = f(x + a0 * d);
4          while (fx > f0 + c1 * a0 * df0)
5              a0 = a0 * beta; fx = f(x + a0 * d);
6          end
7      end

```

整体上，将线搜索、方向选择函数实现封装为 ls 与 dir 句柄，供主函数调用。通过改变句柄对应的函数可以更换线搜索以及方向选择方法。设置梯度精度要求以及最大迭代次数用作停机条件。整个主函数 *NewtonChen* 的实现逻辑与算法一一致。

首先，主函数被传入参数最大迭代次数与梯度精度要求。我们导入数据 A, b 并设置参数 $\sigma = 0.5, \mu = 1$ 。并设置初始值解 $x_0 = \vec{0}$ 。根据需求，设置线搜索方法参数为上述值以及将数据导入方向选择方法（更改 ls, dir 句柄即可）。最后进入算法一所描述的流程。

实验结果

本实验中，最终收敛效率是令人满意的，在几步之内精度陡然增加。6 步之内梯度值已经来到 3.9975×10^{-12} 级别。每一步进行后的梯度值如下表所示：但之后的步长已然变为可忽略的小量，即之后的点列并没有改变。所以我们认为此时精度在 10^{-10} 级别已经收敛，如果将要求再拉高无法继续进行。但应该可以通过改变 Hessian 阵 H 的正则化项系数来调节。

表 1: 迭代过程中的梯度值

迭代次数	梯度模长
1	4.5860×10^0
2	2.0880×10^0
3	2.7100×10^{-2}
4	1.0005×10^{-8}
5	3.9975×10^{-12}

在迭代终点处，函数值约为 50.701564。每步迭代后函数值减去最小值取对数的图像以及步长的变化如下图所示：明显看到每步步长恰能取到 1，完全为 Newton 法的步长。并且函数光滑，局部二次收敛。

最终结果 $x0$ 放在 $x0.mat$ 文件里面。

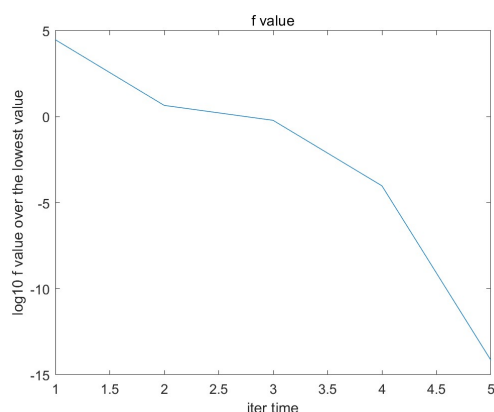


图 1: 函数值收敛历史

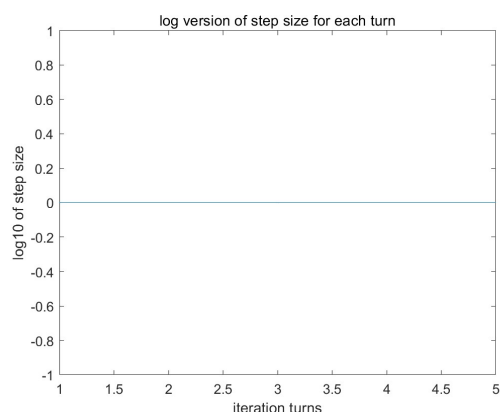


图 2: 步长变化

在 *test.m* 文件中我将我自己实现的 Newton 方法（NewtonChen）与上次作业的最速下降法（SteepestDescent）进行了比较。可以发现 Newton 法收敛远远快于最速下降法，而且精度远远高于最速下降法。运行结果如下：

```
>> test

Chen's code for Newton Method starts...
time spent: 0.121213 sec
Number of iterations: 6
f value at Optimal solution: 50.701564
log10-version of norm of gradient at Optimal solution:-11.40
Chen's code for Newton Method ends...

Steepest descent algorithm starts...
gradient level 1e-3 reached
iteration 4169 times
gradient level 1e-4 reached
iteration 5621 times
gradient level 1e-5 reached
iteration 7198 times
time spent: 2.353760 sec
Number of iterations: 7198
f value at Optimal solution: 50.701564
norm of gradient at Optimal solution: 0.000010
Steepest descent algorithm ends...
```

图 3: Newton 法与最速下降法的比较