最优化算法-程序作业

PB22010379　吴佳骏

本次程序作业我选择的题目是问题一

1　算法原理描述

我们的优化目标是稀疏二分类回归模型

$$\min_x \ell(x) \triangleq \frac{1}{m}\sum_{i=1}^{m}\ln(1+\exp(-b_i a_i^T x)) + \lambda\|x\|_2^2 + \mu\|x\|_1, \tag{1}$$

我们再定义

$$f(x) \triangleq \frac{1}{m}\sum_{i=1}^{m}\ln(1+\exp(-b_i a_i^T x)) + \lambda\|x\|_2^2 \tag{2}$$

$f(x)$ 的梯度是

$$\nabla f(x) = -\frac{1}{m}\sum_{i=1}^{m}(1-p_i(x))b_i a_i + 2\lambda x \tag{3}$$

$$\text{其中 } p_i(x) = \frac{1}{1+exp(-b_i a_i^T x)}$$

再设 $h(x) = \mu||x||_1$, 则

$$proh_{tf}(x)_i = \begin{cases} x_i - t\mu \quad if \quad x_i > t\mu \\ \ 0 \qquad\ |x_i| < t\mu \\ x_i + t\mu \quad if \quad x_i < -t\mu \end{cases} \tag{4}$$

1.1　近似点梯度法

我们取 x_0 为各分量均是 1 的向量。

则迭代公式为 $x^{k+1} = prox_{t_k h}(x^k - t_k\nabla f(x^k))$ 其中 x 梯度和近似点映射的计算由公式 (3)、公式 (4) 给出

t_k 的选取我们使用线搜索，从某个 $t_k = t^{'}$ 开始进行回溯 ($t \leftarrow \beta t$)，直到满足不等式

$$f(x^{k+1}) \leq f(x^k) + \nabla f(x^k)^T(x^{k+1} - x^k) + \frac{1}{2t_k}||x^{k+1} - x^k||_2^2$$

1.2　fisit 算法

我们取 x_0 和 y_0 均是全为 1 的向量。

则迭代公式为

$$y^k = x^{k-1} + \frac{k-2}{k+1}(x^{k-1} - x^{k-2})$$

$$x^k = prox_{t_k h}(y^k - t_k\nabla f(y^k))$$

t_k 的选取我们也使用回溯法，从某个 $t_k = t^{'}$ 和相应的 x^k 开始回溯

重复

$$\begin{cases} t_k \leftarrow \beta t_k \\ x^k \leftarrow prox_{t_k h}(y^k - t_k \nabla f(y^k)) \end{cases}$$

直到满足

$$f(x^k) \leq f(y^k) + \nabla f(y^k)^T (x^k - y^k) + \frac{1}{2t_k} ||x^k - y^k||_2^2$$

2 参数选取

在 $\lambda = \frac{1}{2m}$,$\mu = 10^{-2}$ 时。最优点 x^* 是根据近似点算法运行 5000 次得到的结果。得到最优点的程序为“optimal_point_prox.m”。最优点数据保存在”x_optimal.mat” 中。

设 ∇f 是 L-连续的。近似点梯度法和 fisit 算法均要求在固定步长时步长 $\leq \frac{1}{L}$, 经过作业 3 第一题，可得本题中 ∇f 是 $\frac{|A|^2}{4m} + \frac{1}{m}$ 连续的。经过计算可知固定步长时，每次步长均要小于大约 0.6361。为了减少线搜索的次数，我们把近似点算法和 fisit 算法的线搜索初始的步长 $t^{'}$ 设为 2.

在 2 种算法的线搜索迭代 $t \leftarrow \beta t$ 时，我们把 β 取为 0.8

3 2 种算法收敛性对比结果

我们首先使用”project.m” 程序来进行 2 种算法收敛性的对比。运行之后会直接给出关于 2 种算法条件数和函数误差与迭代次数关系的对比图。

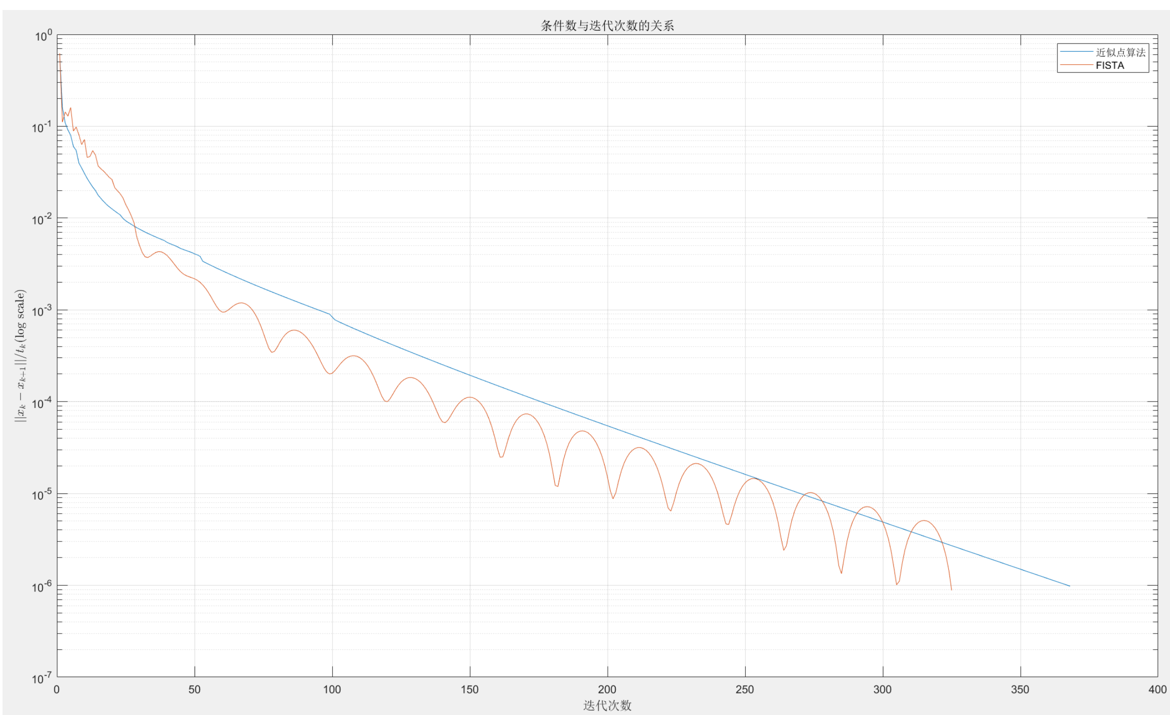

图 1: 条件数与迭代次数的关系

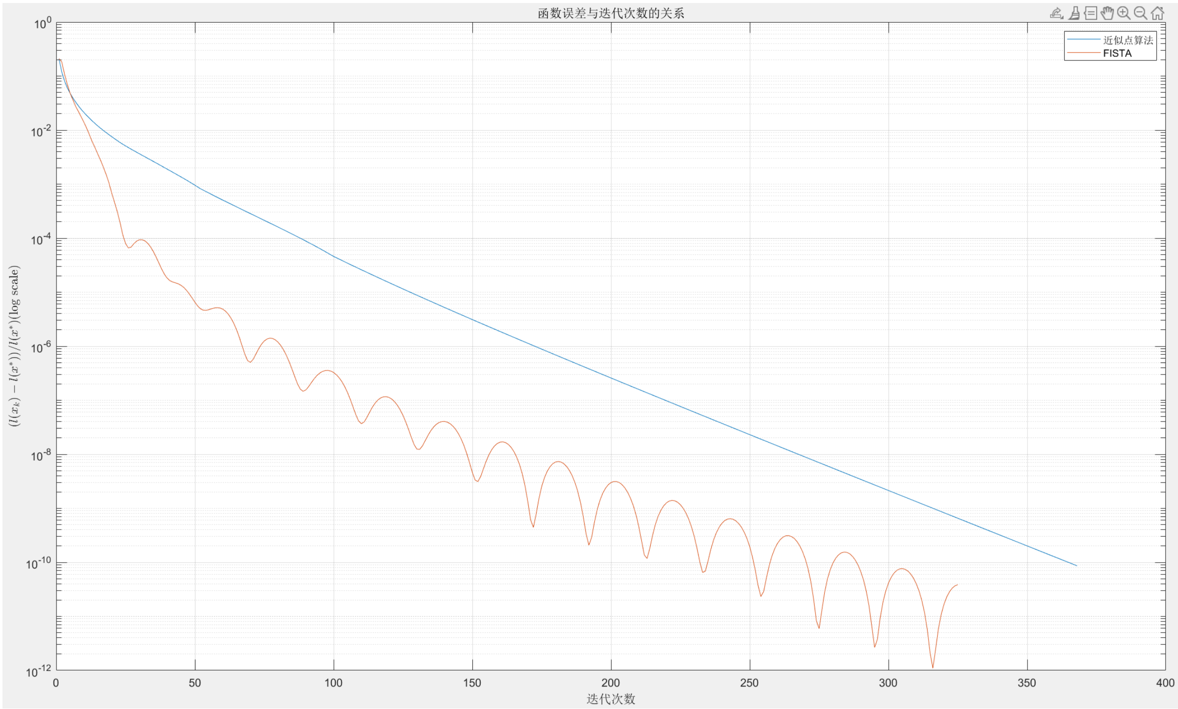

图 2: 函数误差与迭代次数的关系

其中图 1 和图 2 的纵坐标均使用了 log scale

从图中可以看到，近似点梯度算法是严格下降的算法，但 fisit 算法由于添加了动量项并不是严格下降的算法。但 fisit 算法的收敛速度要明显快于近似点梯度法。

3.1 稀疏度计算

稀疏度的计算使用程序“sparsity.m”, 算法与“project,m”完全一致。删去了作图的代码，添加了打印稀疏度等信息的代码。

我们修改 μ 依次为 $10^{-3}, 10^{-2}, 0.05, 0.1$, 计算稀疏度，然后列出表格。

表 1: 近似点梯度法中不同 μ 与稀疏度之间的关系

μ	稀疏度	迭代步数
1^{-3}	0.5285	460
1^{-2}	0.8455	368
0.05	0.9675	195
0.1	0.9837	190

表 2: Fisit 法中不同 μ 与稀疏度之间的关系

μ	稀疏度	迭代步数
1^{-3}	0.5285	455
1^{-2}	0.8455	325
0.05	0.9675	189
0.1	0.9837	68

从表格中可以看出，随着 μ 的增加，稀疏度越来越高，且迭代步数越来越少。