最优化实验报告

左一泓 PB22000116

日期：June 11, 2024

1 问题背景

本文考虑用近似点梯度法和 FISTA 算法 (Fast Iterative Shrinkage Thresholding Algorithm) 求解如下稀疏二分类逻辑回归问题：

$$\begin{aligned}\min_x \quad \ell(x) &\overset{def}{=} \frac{1}{m}\sum_{i=1}^{m}\ln\left(1+\exp\left(-b_i a_i^{\mathrm{T}}x\right)\right)+\lambda\|x\|_2^2+\mu\|x\|_1 \\ &\overset{def}{=} f(x)+\mu\|x\|_1\end{aligned} \tag{1}$$

并比较它们的收敛速率以及参数选择对于收敛性和解的影响。

1.1 定义与记号

$\#$ 表示集合元素个数,$\|.\|$ 在没有角标时都表示二范数，$\|.\|_p$ 表示 p 范数。

近似点映射：

$$\begin{aligned}\mathrm{Prox}_{\mu\|x\|_1}(x) &= \operatorname*{argmin}_t\left(\frac{1}{2}\|t-x\|^2+\mu\|t\|_1\right) \\ &= \mathrm{sign}(x)\max\{|x|-\mu,0\}\end{aligned}$$

f 的微分具体表达式如下：

$$\begin{aligned}\nabla f(x) &= -\frac{1}{m}\sum_{i=1}^{m}\left(1-p_i(x)\right)b_i a_i+2\lambda x \\ p_i(x) &= \frac{1}{1+\exp\left(-b_i a_i^{\mathrm{T}}x\right)}\end{aligned}$$

1.2 算法

近似点梯度法是常用的解决带有非光滑部分优化问题的方法，其基本思想为对于光滑部分进行显式梯度下降，对于非光滑部分利用近似点映射进行隐式梯度下降。针对问题（1）, 下给出近似点梯度法的伪代码。

Algorithm 1 近似点梯度法

Require: 最大迭代次数 n，精确度 ε

Ensure: $k = 1$, 随机初始化 x_0

1: **while** $k \leq n$ and $\|x^{k-1} - x^{k-2}\|^2 / t^{k-1} > 10^{-\varepsilon}$ **do**
2: 调用 3 选取合适的 t^k
3: $x^k \leftarrow \mathrm{Prox}_{t^k \mu \|x\|_1}(x^{k-1} - t^k \nabla f(x^{k-1}))$
4: $k \leftarrow k + 1$
5: **end while**

FISTA 算法为近似点梯度法的一个加速算法，其基本思想为对于光滑部分在进行梯度下降时使用 Nesterov 加速算法。使得在光滑部分 L 光滑的假设下有更好的理论收敛速度。针对问题（1）, 下给出近似点梯度法的伪代码。

Algorithm 2 FISTA 算法

Require: 最大迭代次数 n，精确度 ε

Ensure: $k = 1$, 随机初始化 x_0

1: **while** $k \leq n$ and $\|x^{k-1} - x^{k-2}\|^2 / t^{k-1} > 10^{-\varepsilon}$ **do**
2: $y^k \leftarrow x^{k-1} + \frac{k-2}{k-1}(x^{k-1} - x^{k-2})$
3: 调用 3 选取合适的 t^k
4: $x^k \leftarrow \mathrm{Prox}_{t^k \mu \|x\|_1}(y^k - t^k \nabla f(y^k))$
5: $k \leftarrow k + 1$
6: **end while**

对于迭代系数 t_k 的选取，由于 $\frac{1}{t^k} \leq L$ 时收敛速度有理论保证，其中 L 为 ∇f 的 Lipschitz 常数。我们采用线搜索，搜索的条件为：

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

Algorithm 3 线搜索

Require: $t^k = t^{k-1}$，参照点 y^k 以及其梯度 $\nabla f(y^k)$, 步长收缩的系数 γ

1: $x^k \leftarrow \mathrm{prox}_{\mu \|.\|_1}(y^k - t^k \nabla f(y^k))$
2: **while** x^k 与 y^k 不满足 2 式 **do**
3: $t^k \leftarrow \gamma t^k$
4: $x^k \leftarrow \mathrm{prox}_{\mu \|.\|_1}(y^k - t^k \nabla f(y^k))$
5: **end while**

2 实验结果

2.1 参数设置

本实验的数据集数据个数为 $m = 32561$ 个，每个数据为 $p = 123$ 维向量，均带有标签。

两个算法的最多迭代次数 $n = 10000$, 精确度 $\varepsilon = 6$, 其含义为当 $\|x_k - x_{k-1}/t_k\| \leq 1e^{-\varepsilon}$ 时停止迭代, 随机化种子为 Seed = 3, 线搜索步长收缩系数为 $\gamma = 0.1$,(1) 中的 $\lambda = 1/(2 * m)$,$\mu = 0.001$。

2.2 两算法的收敛速率

收敛条件 $\log(\|x_k - x_{k-1}\|^2/t_k)$ 与迭代次数关系图为：

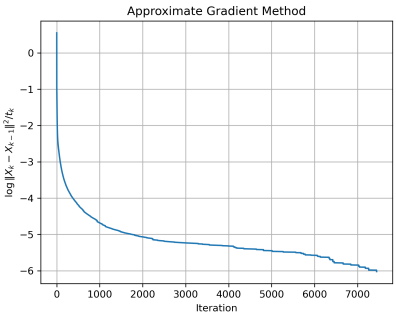


图 1: 近似点梯度法

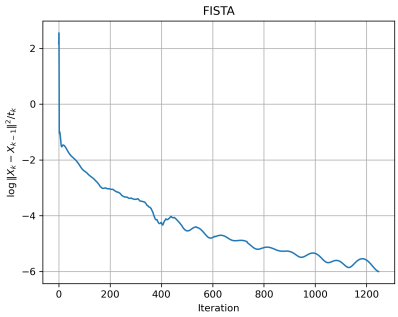


图 2: FISTA 算法

收敛条件 $\log(L(x^k) - L(x^*))$ 与迭代次数关系图为：

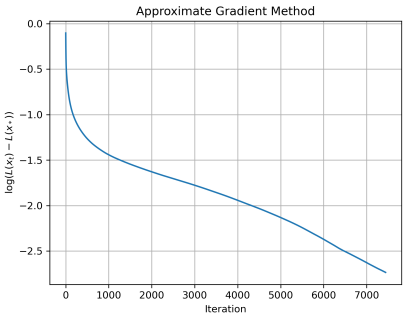


图 3: 近似点梯度法

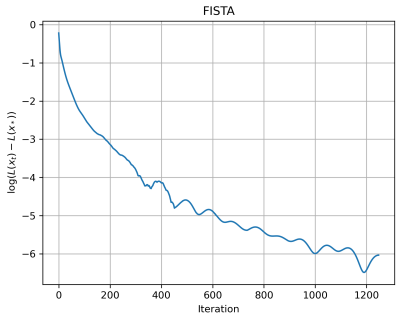
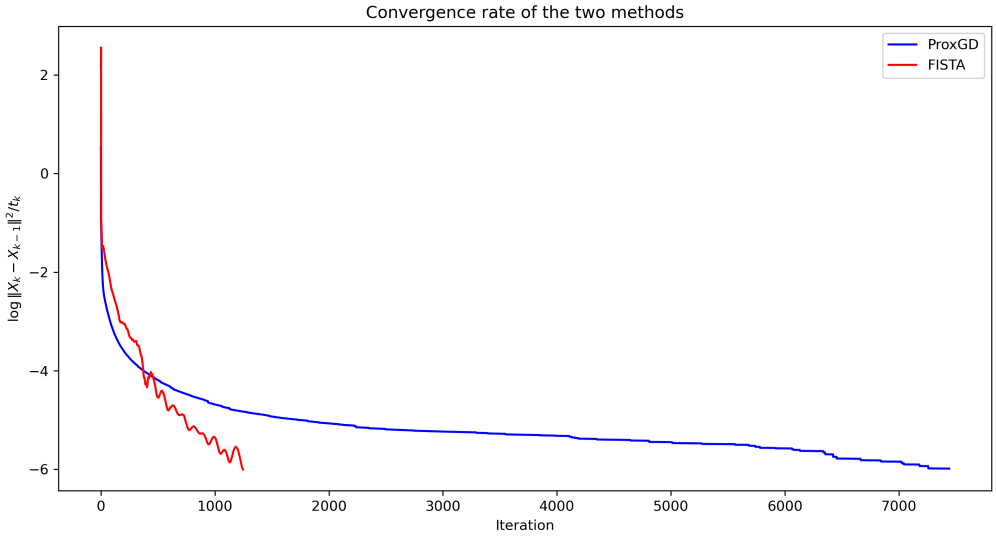
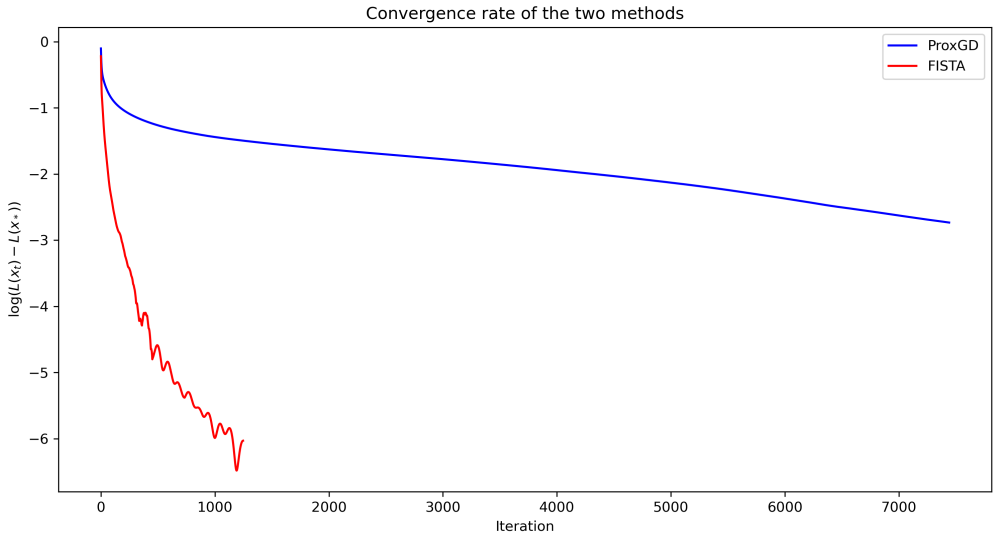


图 4: FISTA 算法

其中 x^* 用提前解出的一个高精度解来近似。





可以看到近似点梯度法用了 7434 次迭代停止，而 FISTA 算法只用了 1243 次迭代，且最后解的精度更高。

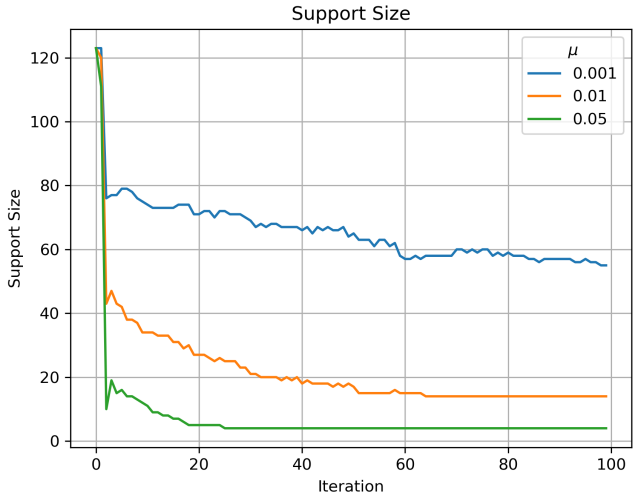
2.3 参数 μ 的选取对结果的影响

使用 FISTA 算法，其余参数设置不变，改变一范数正则项的系数 μ 为 $0, 1e^{-5}, 0.0001, 0.001, 0.01, 0.05$ 稀疏性计算公式为 $\text{sparsity} = \frac{\#\{i:x_i=0\}}{\dim(x)}$。得到如下实验结果：

μ	稀疏性	迭代次数
0.0	0.0	1635
1e-05	0.10569105691056911	1414
0.0001	0.37398373983739835	976
0.001	0.6829268292682927	492
0.01	0.8861788617886179	222
0.05	0.967479674796748	134

表 1: μ 的选取对解稀疏性的影响

可以看到随着 μ 的增大，对于非稀疏解的惩罚变大，因此最优解的稀疏度不断增大，达到最优解的迭代次数减少。



上图为 x_k 的支撑大小（即非零元素个数）与迭代次数的关系。随着迭代次数增加 x 的非零个数先快速下降，然后经过一段时间的缓慢下降，最后趋于稳定。这表明 1 范数的正则项实现了对于解的稀疏性的限制，越大的 μ 倾向于选择更稀疏的解。

3 总结

通过对稀疏逻辑回归问题 (2) 分别使用近似点梯度法和 FISTA 算法进行优化，结果表明 FISTA 算法确实在该问题中具有更好的收敛速度，且并未增加空间复杂度。通过对取不同的 μ 正则项稀疏进行实验，结果表明一范数正则项实现了对于解的稀疏性的限制，越大的 μ 倾向于选择更稀疏的解。