背景

问题介绍

本次作业中，我会用两种算法来求解下面的问题：

$$\min_{w\in\mathbf{R}^d}\frac{1}{n}\sum_{i=1}^{n}f_i(w)+\lambda\|w\|_1,$$

其中 $f_i(w)=\log(1+\exp(-y^iw^Tx^i)),\lambda>0$. 并使用 MNIST 和 Covertype 数据集对其进行检验。

数据集描述

MNIST 关于 MNIST 的介绍内容来自于 CSDN 博客https://blog.csdn.net/simple_the_best/article/details/75267863。

MNIST 数据集可在 http://yann.lecun.com/exdb/mnist/ 获取, 它包含了四个部分:

Training set images: train-images-idx3-ubyte.gz (9.9 MB, 解压后 47 MB, 包含 60,000 个样本)

Training set labels: train-labels-idx1-ubyte.gz (29 KB, 解压后 60 KB, 包含 60,000 个标签)

Test set images: t10k-images-idx3-ubyte.gz (1.6 MB, 解压后 7.8 MB, 包含 10,000 个样本)

Test set labels: t10k-labels-idx1-ubyte.gz (5KB, 解压后 10 KB, 包含 10,000 个标签)

MNIST 数据集来自美国国家标准与技术研究所, National Institute of Standards and Technology (NIST). 训练集 (training set) 由来自 250 个不同人手写的数字构成, 其中 50% 是高中学生, 50% 来自人口普查局 (the Census Bureau) 的工作人员. 测试集 (test set) 也是同样比例的手写数字数据.

Covertype Covertype 则是一个植被分类的数据集。

算法

首先，我们发现，目标函数中有不可微的部分：l_1 范数。而 $\frac{1}{n}\sum_{i=1}^{n}f_i(w)$ 满足 Lipschitz 连续条件，所以虽然目标函数是不可微的，但是我们可以用近端梯度来作为梯度的替代。

我们得到近端更新过程：

$$x_{k+1} = \text{prox}_{\tau\lambda\|x\|_1}(x_k - \tau\nabla f(x_k)) = \text{shrinkage}(x_k - \tau\nabla f(x_k), \tau\lambda)$$

我会对两种算法进行 python 实现。

adam

我对每个程序运行三次数值试验，并记录了相关信息于下面的表格中：

time	nrm1	res
1.05	5.51e+01	8.18e-13
1.07	4.87e+01	3.44e-7
0.99	5.15e+01	2.56e-07

CVXPY

time	nrm1	res	comparing to cvx
0.55	5.51e+01	4.18e+03	3.74e+01
0.56	4.87e+01	4.09e+3	3.70e+01
0.54	5.15e+01	4.27e+03	3.72e+01

CALL-MOSEK

time	nrm1	res	comparing to cvx
0.55	5.51e+01	8.18e-13	0
0.94	4.87e+01	3.44e-7	0
0.96	5.15e+01	2.56e-07	0

CVX-MOSEK

CVX-GUROBI

time	nrm1	res	comparing to cvx
14.61	5.51e+01	1.47e-10	1.86e-07
14.32	4.87e+01	5.43e-12	2.08e-6
14.31	5.15e+01	2.93e-11	1.00e-06

分析结果如下：

1. 运行速度最快的是直接调用 MOSEK 程序包的算法。GUROBI 的由于 license 调用很麻烦，所以要耽误很久。
2. 各种方法的相对误差和绝对误差都很小。精度都很高。

2.3

要解决的问题是基查找问题：

$$\min_u \{||u||_1, Au = f\}.$$

这次作业中，我们要解决的是其变体：

$$\min_x ||x||_1 + \frac{1}{2}||Ax - b||_2.$$

本文中，我们要使用 Bregman 迭代的方法求解。

Algorithm 1 Bregman Iterative Algorithm

Require: 函数 $J(u)$ 中的参数 μ，预先设定好的误差阈值 $\epsilon = 1 \times 10^{-5}$

1: $u_0 = 0, p_0 = 0;$
2: 迭代 $u_{k+1} = \arg\min_u (D_J^{p_k} + \|Au_k - b\|_2);$
3: 迭代 $p_{k+1} = p_k - A^T(Au_{k+1} - b)/\|Au_{k+1} - b\|_2;$
4: 计算误差 $error_k = \dfrac{\|Au_k - b\|}{\|b\|}$，如果 $error_k > \epsilon$ 则返回第二步继续进行迭代;

其中函数 $J(u) = \mu\|u\|_1$，参数 μ 根据论文中数值实验结果设置为

$$\mu = \frac{0.02}{\sqrt{\|u\|_0}}$$

得到的数值结果如下：

time	nrm1	res	error to cvx-mosek
0.91	4.81e+01	2.59e-07	1.11e+00
0.94	4.81e+01	3.02e-07	1.15e+00
0.92	5.00e+01	4.54e-07	1.20e+00

表 1: Bregman Iteration Algorithm

可见其运行效果稳定。相对误差和绝对误差都比较小。运行时间与前几种算法相当。考虑到我们没有对其数据结构进行任何优化，所以其达到现在的效率可以被认为是十分优秀的表现。

2.4

本文选取了第一篇文章中的算法进行了实现。

ADM 算法的迭代格式如下所示。

Algorithm 2 Alternating Direction Multiplier Algorithm

Require: β, γ，矩阵 A, 预先设定好的误差阈值 $\epsilon = 0.001$

1: 设定初值 $x_0 = \mathbf{0}, y_0 = \mathbf{0}, z_0 = \mathbf{0}$;

2: 迭代 $z_{k+1} = \mathcal{P}_{B_1^\infty}(A'y_k + \dfrac{x_k}{\beta})$;

3: 计算 g_k, α_k;

4: 迭代 $y_{k+1} = y_k - \alpha_k' g_k$;

5: 迭代 $z_{k+1} = x_k - \gamma\beta(z_{k+1} - A'y_{k+1})$;

6: 计算 $error_k$，如果 $error_k > \epsilon$ 则返回第二步进行迭代;

Ensure:

模型的最优解 X，取最优解时对应的最优值 out;

其中按照论文中的数值实验结果设置参数 $\beta = \dfrac{2m}{\|b\|_1}, \gamma \in (0, \dfrac{1+\sqrt{5}}{2}), \mu = 0$，同时 g_k, α_k 的表达式如下

$$g_k = \mu y_k + Ax_k - b + \beta A(A'y_k - z_{k+1})$$
$$\alpha_k = \frac{g_k' g_k}{g_k'(\mu I + \beta AA')g_k}$$

time	nrm1	res	error to cvx-mosek
1.51	4.81e+01	1.23e-01	1.15e+00
0.49	4.82e+01	1.13e-01	1.11e+00
0.58	5.01e+01	1.30e-01	1.20e+00

表 2: Caption

误差的计算公式为

$$error = \max\{\frac{\|r_p\|}{\|b\|}, \frac{\|r_d\|}{\sqrt{m}}, \frac{\Delta}{f_p(x,r)}\}$$

其中

$$\begin{cases} r_p = Ax + r - b = Ax + \mu y - b \\ r_d = A'y - z \\ \Delta = f_d(y) - f_p(x,r) = \mathbf{Re}(b'y) - \mu\|y\|^2 - \|x\|_1 \end{cases}$$

得到的运行结果如下：

Problem 3

3.1.a

结果请参考代码文件 3.1。

3.1.b

在这一部分，将原始的优化问题转换为对偶问题。

原始问题的形式为

$$\max_{X \succeq 0} \log\det X - Tr(SX) - \rho\|X\|_1$$

由 l_1 范数的性质，

$$\|X\|_1 = \max_{\|Z\|_\infty \le 1} tr(ZX)$$

该最大值可以进行如下转化

$$\begin{aligned}
\max_{X \succeq 0} \log\det X - Tr(SX) - \rho\|X\|_1 &= \max_{X \succeq 0}(\log\det X - Tr(SX) - \max_{\|Z\|_\infty \le 1} \rho Tr(ZX)) \\
&= \max_{X \succeq 0} \min_{\|Z\|_\infty \le 1} (\log\det X - Tr(SX) - \rho Tr(ZX)) \\
&= \max_{X \succeq 0} \min_{\|Z\|_\infty \le 1} (\log\det X - Tr((S + \rho Z)X)) \\
&= \max_{X \succeq 0} \min_{\|Z\|_\infty \le \rho} (\log\det X - Tr((S + Z)X))
\end{aligned}$$

由于 S 是一个观测样本的样本方差矩阵，因而它以概率 1 是正定的，可以通过对惩罚系数 ρ 的设定使得矩阵 $S+Z$ 是正定的（即只需让 ρ 取值不大于 S 的最小特征值，由于 S 半正定，因而它的最小特征值非负）。

对于正定矩阵的迹有如下引理成立。

引理：设 B 为 p 阶正定矩阵，则有

$$tr(B) - ln|B| \ge p$$

等号成立的充分必要条件是 $B = I_p$

引理证明：

因为 $B \succ 0$，于是 B 的全部特征值 $\lambda_1, \lambda_2, \cdots, \lambda_p > 0$。且有

$$|B| = \lambda_1\lambda_2 \cdots \lambda_p$$

由于 $ln(1+x) \le x$，所以有

$$\begin{aligned}
ln|B| = \sum_{i=1}^p ln\lambda_i = \sum_{i=1}^p ln(1 + \lambda_i - 1) \\
\le \sum_{i=1}^p (\lambda_i - 1) = tr(B) - p
\end{aligned}$$

所以有 $trB - ln|B| \ge p$，等号成立当且仅当 $ln(1+x) \le x$ 中的等号成立，即 $x = 0$。在这里即为 $\lambda_i - 1 = 0$，即为 $B = I_p$。

回到原题，由于要求原问题的对偶问题，根据论文 [OLA2008] 有对于这个 maximum worst case log-likelihood 问题，交换求最大值号与求最小值号即可得到原问题的对偶问题。

同时，根据引理有如下不等式成立

$$
\begin{aligned}
\log\det X - Tr((S+Z)X) &= \log\det X - Tr(X^{\frac{1}{2}}(S+Z)X^{\frac{1}{2}}) \\
&\leq \log\det X - (\log\det(X^{\frac{1}{2}}(S+Z)X^{\frac{1}{2}}) + n) \\
&= -\log\det((S+Z)) - n
\end{aligned}
$$

等号取到当且仅当 $X^{\frac{1}{2}}(S+Z)X^{\frac{1}{2}} = I_n$ 也即 $X = (S+Z)^{-1}$。进而有对偶问题为

$$
\min_{\|Z\|_\infty \leq \rho} -\log\det((S+Z)) - n
$$

更正式的，可以使用 $Y = S + Z$ 带入为

$$
\begin{aligned}
&minimize \ \ -\log\det(Y) - n \\
&subject\ to \ \|Y - S\|_\infty \leq \rho
\end{aligned}
$$

其中以 Y 为变量。

3.1.c

结果请参见代码文件 3.1.c。

对于 $\rho = 10^{-1}, 10^{-3}, 10^{-7}, 10^{-10}$, 最优值分别是 $-47.3675, -40.3091, -40.1341, -40.1341$.

3.1.d

结果请参加代码文件 3.1.d。