程序作业

曾燎原　　浙江工业大学

邮箱：zengly@zjut.edu.cn

致谢：本教案内容取自北京大学文再文教授的最优化书籍

提 纲

图像去噪问题

LASSO 问题

逻辑回归问题

程序作业

假设用 x 表示一幅真实图像，它是一个 $m \times n$ 维的矩阵, 其中 x_{ij} 代表图像在划定的位置 (i, j) 处的灰度值. 用 y 表示图像 x 受到噪声干扰后的图像，即

$$y = x + e,$$

这里假设 $e \in \mathbb{R}^{m \times n}$ 是高斯白噪声. 为了从有噪声的图像 y 中还原原始图像 x, 利用 Tikhonov 正则化的思想建立如下优化模型:

$$\min_{x \in \mathbb{R}^{m \times n}} \quad f(x) = \frac{1}{2}\|x - y\|_F^2 + \lambda(\|D_1 x\|_F^2 + \|D_2 x\|_F^2) \tag{1}$$

其中 $D_1 x$,$D_2 x$ 分别对 x 在水平方向和竖直方向上做向前差分, 即

$$(D_1 x)_{ij} = \frac{1}{h}(x_{i+1,j} - x_{ij}), \quad (D_2 x)_{ij} = \frac{1}{h}(x_{i,j+1} - x_{ij})$$

其中 h 为给定的离散间隔.

模型(1)由两项组成:

- 第一项为保真项, 即要求原始图像 x 和带噪声的图像 y 不要相差太大. 由于假设了噪声是高斯白噪声, 此时通常使用矩阵的 F 范数

  $$\|x-y\|_F=\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij}-y_{ij})^2}$$

  刻画两个图像矩阵 $x,y\in\mathbb{R}^{m\times n}$ 之间的距离.

- 第二项为 Tikhonov 正则项, 是对图像 x 的性质做出限制, 希望原始图像 x 的各个部分的变化不要太剧烈, 即水平和竖直方向的差分的平方和不要太大, 这种正则项会使得恢复的图像 x 具有比较好的光滑性.

模型(1)的目标函数是光滑的, 可以利用梯度法来求解. 目标函数 f 的梯度是

$$\nabla f(x) = x - y - 2\lambda\Delta x,$$

其中 Δ 是图像 x 的离散拉普拉斯算子, 即

$$(\Delta x)_{ij} = \frac{x_{i+1,j} + x_{i-1,j} + x_{i,j+1} + x_{i,j-1} - 4x_{ij}}{h^2},$$

因此梯度法的迭代格式为

$$x^{k+1} = x^k - \alpha_k(I - 2\lambda\Delta)x^k + \alpha_k y$$

其中 α_k 为步长.

梯度法算法框架回顾

Algorithm 1: 非单调线搜索的 BB 方法

给定 x^0, 选取初值 $\alpha > 0$, 整数 $M \geq 0$, $c_1, \beta, \varepsilon \in (0, 1)$, $k = 0$.
while $\|\nabla f(x^k)\| > \varepsilon$ **do**
    while $f(x^k - \alpha \nabla f(x^k)) \geq \max\limits_{0 \leq j \leq \min\{k, M-1\}} f(x^{k-j}) - c_1 \alpha \|\nabla f(x^k)\|^2$ **do**
        令 $\alpha \leftarrow \beta\alpha$.
    end while
    令 $x^{k+1} = x^k - \alpha \nabla f(x^k)$.
    根据 BB 步长公式之一计算 α, 并做截断使得 $\alpha \in [\alpha_m, \alpha_M]$.
    $k \leftarrow k + 1$.
end while

定义(非单调线搜索，见文献 (Zhang, Hager '04)) 设 d^k 是点 x^k 处的下降方向, 以下不等式可作为一种线搜索准则:

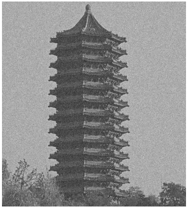
$$f(x^k + \alpha d^k) \le C_k + c_1 \alpha \nabla f(x^k)^{\mathrm{T}} d^k,$$

其中 C_k 满足递推式 $C_0 = f(x^0)$, $C_{k+1} = \frac{1}{Q_{k+1}}\left(\eta Q_k C_k + f(x^{k+1})\right)$, 序列 $\{Q_k\}$ 满足 $Q_0 = 1$, $Q_{k+1} = \eta Q_k + 1$, 参数 $\eta, c_1 \in (0,1)$.

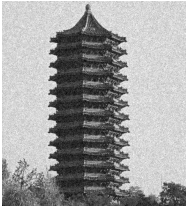
这个非单调线搜索准则中的 C_k 是本次线搜索中的参照函数值, 做为本次迭代充分下降的起始标准; 而下一步的参照标准 C_{k+1} 则是函数值 $f(x^{k+1})$ 和 C_k 的凸组合, 其中凸组合的系数由参数 η 决定. 当 $\eta = 0$ 时, 此准则即为 Armijo 线搜索准则.

Tikhonov 正则化模型求解结果

(a) The true one



(b) The noisy one



(c) $\lambda = 0.5$

(d) $\lambda = 2$

Figure: Tikhonov 正则化模型求解结果

提 纲

图像去噪问题

LASSO 问题

逻辑回归问题

程序作业

考虑 LASSO 问题:

$$\min_{x\in\mathbb{R}^n} \; f(x) = \frac{1}{2}\|Ax-b\|^2 + \mu\|x\|_1,$$

其中 $A \in \mathbb{R}^{m\times n}$,$b \in \mathbb{R}^n$ 且 $m < n$.

- LASSO 问题的目标函数 $f(x)$ 不光滑, 在某些点处无法求出梯度, 因此不能直接对原始问题使用梯度法求解
- 不光滑项为 $\|x\|_1$, 它实际上是 x 各个分量绝对值的和, 考虑如下一维光滑函数:

$$l_\delta(x) = \begin{cases} \frac{1}{2\delta}x^2, & |x| < \delta, \\ |x| - \frac{\delta}{2}, & \text{其他}. \end{cases}$$

- 上述定义实际上是 Huber 损失函数的一种变形, 当 $\delta \to 0$ 时, 光滑函数 $l_\delta(x)$ 和绝对值函数 $|x|$ 会越来越接近.

光滑化 LASSO 问题为

$$\min_{x \in \mathbb{R}^n} \quad f_\delta(x) = \frac{1}{2}\|Ax - b\|^2 + \mu L_\delta(x), \tag{2}$$

其中 $L_\delta(x) = \sum_{i=1}^n l_\delta(x_i)$，$\delta$ 为给定的光滑化参数.

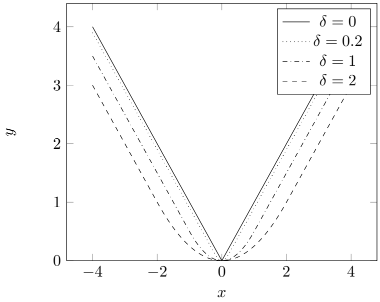


Figure: 当 δ 取不同值时 $l_\delta(x)$ 的图形

- $f_\delta(x)$ 的梯度为

$$\nabla f_\delta(x) = A^{\mathrm{T}}(Ax - b) + \mu \nabla L_\delta(x),$$

其中 $\nabla L_\delta(x)$ 是逐个分量定义的：

$$(\nabla L_\delta(x))_i = \begin{cases} \operatorname{sign}(x_i), & |x_i| > \delta, \\ \frac{x_i}{\delta}, & |x_i| \le \delta. \end{cases}$$

- $f_\delta(x)$ 的梯度是 Lipschitz 连续的, 其 Lipschitz 常数为 $L = \|A^{\mathrm{T}}A\|_2 + \frac{\mu}{\delta}$.
- 根据梯度法在凸函数上的收敛性定理, 固定步长需不超过 $\frac{1}{L}$ 才能保证算法收敛, 如果 δ 过小, 那么我们需要选取充分小的步长 α_k 使得梯度法收敛.

在 Matlab 中可以按照如下方式生成 LASSO 模型中的 A 和 b 以及真实解 xorig

```
m = 512; n = 1024;
A = randn(m,n);
xorig = sprandn(n, 1, r);
b = A*xorig;
```

第二行生成一个稀疏向量 xorig 做为真实解, 其中 $r \in (0,1)$ 是稀疏度, 即 xorig 中非零元个数与总元素个数的比例为 r.

也可以按照以下方式生成 A,b:

```
I = randperm(n);
J = I(1:k);  %随机选定k个非零元素的位置
xorig = zeros(n,1);
xorig(J) = randn(k,1);
A = randn(m,n);
b = A*xorig;
```

注:LASSO 中 $m < n$

提 纲

图像去噪问题

LASSO 问题

逻辑回归问题

程序作业

- 假设数据对 $\{a_i, b_i\}, i = 1, 2, \cdots, m$ 之间独立同分布，则在给定 $a_1, a_2, \cdots, a_m$ 情况下，$b_1, b_2, \cdots, b_m$ 的联合概率密度是

$$
\begin{aligned}
p(b_1, b_2, \cdots, b_m \mid a_1, a_2, \cdots, a_m; x) &= \prod_{i=1}^{m} p(b_i \mid a_i; x) \\
&= \frac{1}{\prod_{i=1}^{m}(1 + \exp(-b_i \cdot a_i^{\mathrm{T}} x))}.
\end{aligned}
$$

- 最大似然估计是求解如下最优化问题：

$$
\min_{x \in \mathbb{R}^n} \quad \sum_{i=1}^{m} \ln(1 + \exp(-b_i \cdot a_i^{\mathrm{T}} x)).
$$

- 考虑加 Tikhonov 正则项的逻辑回归模型：

$$
\min_{x \in \mathbb{R}^n} \ f(x) = \frac{1}{m} \sum_{i=1}^{m} \ln\left(1 + \exp\left(-b_i a_i^{\mathrm{T}} x\right)\right) + \mu \|x\|_2^2,
$$

其中 $\mu > 0$ 是正则化参数.

考虑如下光滑的无约束优化问题：

$$\min_{x\in\mathbb{R}^n} \quad f(x) = \frac{1}{m}\sum_{i=1}^{m} \ln\left(1+\exp\left(-b_i a_i^{\mathrm{T}} x\right)\right) + \mu\|x\|_2^2. \tag{3}$$

上述逻辑回归模型的目标函数是光滑的，可以使用梯度法求解。目标函数的梯度为

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

其中 $p_i(x) = \frac{1}{1+\exp\left(-b_i a_i^{\mathrm{T}} x\right)}$.

引入矩阵 $A = [a_1, a_2, \cdots, a_m]^{\mathrm{T}} \in \mathbb{R}^{m \times n}$, 向量 $b = (b_1, b_2, \cdots, b_m)^{\mathrm{T}}$, 以及

$$p(x) = (p_1(x), p_2(x), \cdots, p_m(x))^{\mathrm{T}},$$

则可重写梯度为

$$\nabla f(x) = -\frac{1}{m} A^{\mathrm{T}} (b - b \odot p(x)) + 2\mu x \tag{4}$$

其中 $\odot$ 指向量的点乘，对于两个相同维数的向量 $a, b \in \mathbb{R}^m$

$$a \odot b = (a_1 b_1, a_2 b_2, \cdots, a_m b_m)^{\mathrm{T}}.$$

则梯度法迭代格式可以写作:

$$x^{k+1} = x^k + \alpha_k \left(\frac{1}{m} A^{\mathrm{T}} \left(b - b \odot p\left(x^k\right)\right) - 2\mu x^k \right).$$

使用 libsvmread 导入数据集 ijcnn1.test 数据集，用到的 libsvmread.c 文件和 ijcnn1.test 已上传在学习通资料中. 同时，导入数据这部分代码已经写在 demo_lr_gradient.m 程序中.

由于 libsvmread 是一个.c 文件, 需要在运行你的 Matlab 程序前进行如下额外操作使得形成一个 Matlab 可执行的文件:

- Matlab 路径设置为 libsvmread.c 所在的文件夹
- 在 Matlab 的 command window 运行如下代码行:

```
mex libsvmread.c
```

作业内容

作业内容: 以下任务三选一

- 完成梯度法求解逻辑回归模型(3)的 Matlab 测试程序;
- 翻译图像去噪的 Matlab 程序为 Python 程序
- 用 Python 实现求解光滑化的 LASSO 模型(2)的梯度法; 其中线搜索使用非单调线搜索,BB 步长做为初始步长.

注: 此次程序作业满分 100; 选择编写逻辑回归模型 Matlab 测试程序 (第 1 个任务), 分数不高于 85 分; 选图像去噪的程序 (第 2 个任务) 翻译, 分数不高于 90;

作业要求: 需要写一个实验报告, 和程序打包上传学习通发布的分组任务.

实验报告内容要求

- 写明程序运行环境：比如电脑操作系统，电脑配置参数等
- 写清楚用到的 Python 的工具包及其在程序中的主要用途
- 写明各个.py 文件的主要功能；写明每个.py 文件中各个模块实现的是优化算法中的什么步骤
- 要有实验结果分析：可以根据以下几点或者其他方面进行讨论
  - 对不同的模型参数 (逻辑回归中取 $\mu = 1e-1/m, 1e-2/m, 1e-3/m$；图像去噪中选 $\lambda = 0.5, 2, 5$ ；LASSO 中取 $\mu = 1e-2, 1e-3, 1e-4$，光滑参数 $\delta = 0.2, 1, 2$ 等); 或者取不同终止准则，统计算法的迭代次数和运行时间
  - 每次迭代中非精确线搜索对应的内循环大概要多少步，BB 步长做为初始步长是否会被接受; 非精确线搜索中函数参考值如果用 $\max\limits_{0\le j\le \min\{k,M-1\}} f(x^{k-j})$, 迭代步数和运行时间会有什么变化; 如果使用常数步长应该取步长为多大才能保证算法收敛, 其迭代步数和运行时间相对线搜索的情况如何
  - 统计算法实现中最耗时的代码行，占整个运行时间的多少比例 (Matlab 中的可以用 profile 看到每一行代码被运行多少次及其总耗时)，根据 profile 考虑是否可以优化算法的效率
  - 记录每次迭代中目标函数的函数值和梯度的模，画出目标函数值和梯度随着迭代步数的变化图
  - LASSO 问题可以画出真实解和算法求出来的近似解的对比图；比如在同一个图中，用蓝色的 * 画真实解，用红色的 o 画算法求得的近似解这样的方式