凸优化作业 5

徐洋 1800010740

2021 年 1 月 5 日

1 问题描述

我们考虑 group LASSO 问题:

$$\min_{x\in\mathbb{R}^{n\times l}} \quad \frac{1}{2}\|Ax-b\|_F^2+\mu\|x\|_{1,2}$$

其中 $A\in\mathbb{R}^{m\times n}, b\in\mathbb{R}^{m\times l}, \mu>0$ 为给定, 且

$$\|x\|_{1,2}=\sum_{i=1}^{n}\|x(i,1:l)\|_2$$

2 通过 CVX 调用 Mosek 或 Gurobi

这一部分是平凡的.CVX 可以直接处理，不需要对目标函数做任何数学上的变换。
有关数值结果将会同其他方法下的数值结果，在后文特定章节统一给出。

3 直接调用 Mosek

我们将其转换成可直接调用 Mosek 的等价形式，更具体地，我们要将它转换为一个 SOCP 问题，我们为目标函数中所有涉及二范数的，添加额外的决策变量使之化为一个二次锥; 涉及二范数平方的, 我们将其化为 Rotated Quadratic Cone:

$$\begin{aligned}
\min \quad & z_1+\mu\sum_{i=1}^{n}y_i \\
s.t. \quad & u(i_1,i_2)= A(i_1,:)x(:,i_2)-b(i_1,i_2), \quad 1\le i_1\le m, 1\le i_2\le l \\
& z_2= 1 \\
& y_i\ge \sqrt{x(i,1)^2+\cdots+x(i,l)^2}, \quad i=1,2,3\ldots,n \\
& 2z_1z_2\ge \sum_{i_1,i_2}u(i_1,i_2)^2,
\end{aligned} \qquad .$$

其中 $A\in\mathbb{R}^{m\times n}, b\in\mathbb{R}^{m\times l}, \mu>0$ 为给定，$x\in\mathbb{R}^{n\times l}, u\in\mathbb{R}^{m\times l}, (y_1,\ldots,y_n)^T\in\mathbb{R}^n, (z_1,z_2)^T\in\mathbb{R}^2$ 为决策变量。此外我们还需将所有的决策变量进行变形，因为本问题中 Mosek 只接受一个向量形式的决策变量，我们具体的技术实现是：

$$\begin{aligned}\bar{x} = (&x_{1,1}, \ldots, x_{n,1}, x_{1,2}, \ldots, x_{n,2}, \ldots\ldots, x_{1,l}, \ldots, x_{n,l},\\ &u_{1,1}, \ldots, u_{m,1}, u_{1,2}, \ldots, u_{m,2}, \ldots\ldots, u_{1,l}, \ldots, u_{m,l},\\ &y_1, \ldots, y_n,\\ &z_1, z_2)\end{aligned}$$

对应的，线性约束矩阵 A_0 具有下面的形式

$$\begin{pmatrix} A_{m\times n} & & & -I_m & & & 0_{m\times(n+1)} \\ & A_{m\times n} & & & -I_m & & 0_{m\times(n+1)} \\ & & \ddots & & & \ddots & 0_{m\times(n+1)} \\ & & & A_{m\times n} & & -I_m & 0_{m\times(n+1)} \\ 0 & & & \cdots & & & \cdots \quad 1 \end{pmatrix}$$

线性约束右侧 b_0 为 b 的每一列拼接而成的一个列向量, 即 reshape(b,m*l,1)，末尾再补上一个 1. 此外还有 n 个二次锥约束和一个旋转锥约束，表述对应的元素在 $\bar{x}$ 中的角标即可。

4 直接调用 Gurobi

这一部分的本质和上一部分相似，只是在 Gurobi 中我们可以将二次锥约束写成二次约束：

$$\begin{aligned} \min \quad & \frac{1}{2}z_1 + \mu\sum_{i=1}^{n} y_i \\ s.t. \quad & A(i_1,:)x(:,i_2) - u(i_1,i_2) = b(i_1,i_2), \quad 1 \le i_1 \le m, 1 \le i_2 \le l \\ & x(i,1)^2 + \cdots + x(i,l)^2 - y_i^2 \le 0, \qquad i = 1,2,3\ldots,n \\ & \sum_{i_1,i_2} u(i_1,i_2)^2 - z_1 \le 0, \end{aligned} .$$

其中 $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^{m\times l}, \mu > 0$ 为给定，$x \in \mathbb{R}^{n\times l}, u \in \mathbb{R}^{m\times l}, (y_1, \ldots, y_n)^T \in \mathbb{R}^n, z_1 \in \mathbb{R}$ 为决策变量。类似上一节，我们同样需要将所有决策变量转换成一个决策向量，相比之下只是缺少了 z_2。线性约束矩阵也是类似的，这里便不重复叙述了。

5 次梯度法

5.1 基本原理

为求凸函数 f 的最优值，我们采取如下迭代：

$$x^{(k+1)} = x^{(k)} - t_k g^{(k)}$$

其中 $g^{(k)}$ 为目标函数在 $x^{(k)}$ 处的任一此梯度。对于 group LASSO 问题，目标函数第一项是可微的。第二项在 x 某一行均为 0 的点处是不可微的，我们选取的次梯度在对应分量上值为 0. 而对于不全为 0 的行，函数对于那些分量而言是可微的。

5.2 连续化策略

我们从较大的正则化参数 μ_0 逐渐减少到 μ，对于每个 μ_t，我们求解对应的子问题，然后设置

$$\mu_{t+1} = \max\{\mu_t \eta, \mu\}$$

其中 η 为缩小因子。每一个子问题的迭代初值取上一个子问题的最优值。

5.3 相关设定

步长取固定步长 $\alpha = \frac{1}{\lambda_{max}(A^T A)^2}$，在连续化策略的最后一个子问题中选取 $\alpha_k = \frac{\alpha}{\sqrt{\max(k,100)-99}}$.
停机准则为相对函数值变化量小于设定阈值或相对历史最佳函数值变化小于设定阈值。

5.4 数值结果与分析

测试代码依照 http://bicmr.pku.edu.cn/~wenzw/courses/Test_group_lasso.m
下图是函数值相对误差随迭代次数的变化，从中我们可以看到连续化方法的巨大作用

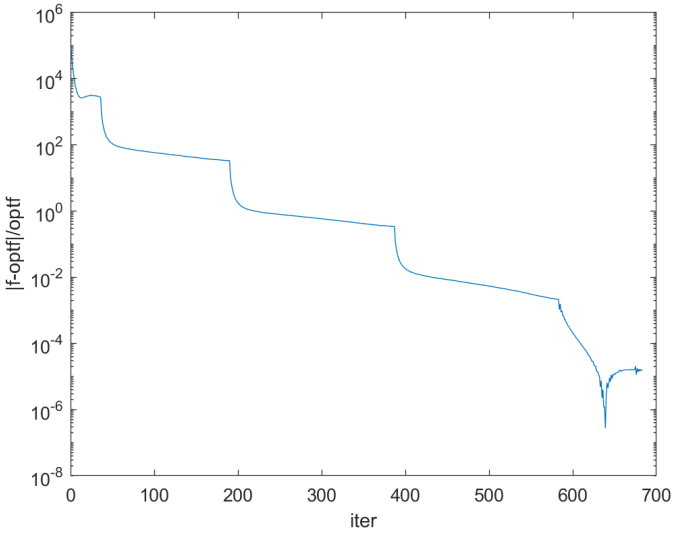

图 1: 函数值相对误差随迭代次数的变化

方法	cputime	迭代次数	最优点函数值	稀疏程度	与 u 的差
CVX_Mosek	0.82	-	5.80556E-01	0.105	3.78E-05
CVX_Gurobi	0.76	-	5.80556E-01	0.103	3.75E-05
SGD	0.10	683	5.80558E-01	0.100	5.00E-05

表 1: 不同方法下所得数值结果

可以看到，次梯度法在运算时间上很优异，得到的最优解精度令人满意。若想提高精确度，可以通过严格化停机条件来达到。

6 梯度法

6.1 基本原理

为求凸函数 f 的最优值，我们采取如下迭代：

$$x^{(k+1)} = x^{(k)} - t_k g^{(k)}$$

其中 g^k 为函数 f 光滑化 (见 6.3) 后的梯度.

对于 t_k 的选取我们使用 BB 步长, 令 $s^k = x^{k+1} - x^k$, $z^k = g^{k+1} - g^k$，在偶数与奇数步分别对应 $\frac{\langle s^k, s^k\rangle}{\langle s^k, z^k\rangle}$ 和 $\frac{\langle s^k, z^k\rangle}{\langle z^k, z^k\rangle}$ 两个 BB 步长。

得到 BB 步长后我们采用线搜索方法，计算 (Zhang & Hanger) 线搜索准则中的递推常数，其满足 $C_0 = f(x^0)$, $C_{k+1} = (\gamma Q_k C_k + f(x^{k+1}))/Q_{k+1}$ ，序列 Q_k 满足 $Q_0 = 1$, $Q_{k+1} = \gamma Q_k + 1$。当满足线搜索准则 (Zhang & Hanger) $f(x^k + \alpha d^k) \le C_k + \rho\alpha (g^k)^\top d^k$ 或进行超过 10 次步长衰减后退出线搜索。在当前步长不符合线搜索条件的情况下，对当前步长以 η 进行衰减，线搜索次数加一。

6.2 连续化策略

同 5.2

6.3 光滑化

由于 group LASSO 的目标函数第二项不是可微的，因此我们对此目标函数的第二项做光滑化，具体如下

$$\min_{x\in\mathbb{R}^{n\times l}} \quad \frac{1}{2}\|Ax - b\|_F^2 + \mu \sum_{i=1}^{n} l_\sigma\left(x(i,:)\right)$$

其中

$$\ell_\sigma(x) = \begin{cases} \frac{1}{2\sigma}\|x\|_2^2, & \|x\|_2 < \sigma \\ \|x\|_2 - \frac{\sigma}{2}, & \text{otherwise} \end{cases}$$

6.4 数值结果

方法	cputime	迭代次数	最优点函数值	稀疏程度	与 u 的差
CVX_Mosek	0.82	-	5.80556E-01	0.105	3.78E-05
CVX_Gurobi	0.76	-	5.80556E-01	0.103	3.75E-05
GD	0.11	640	5.80556E-01	0.106	3.90E-05

表 2: 不同方法下所得数值结果

可以看到，梯度法在运算时间上优异.

7 Fast Gradient Method

7.1 算法

此算法在框架上与梯度法相同。

梯度法迭代格式为

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

同样地我们使用 BB 步长，令 $s^k = y^{k+1} - y^k$, $z^k = g^{k+1} - g^k$，在偶数与奇数步分别对应 $\frac{\langle s^k, s^k\rangle}{\langle s^k, z^k\rangle}$ 和 $\frac{\langle s^k, z^k\rangle}{\langle z^k, z^k\rangle}$ 两个 BB 步长。

我们发现对于 FGD 方法，线搜索在性能提升方面不明显，因此我们没有使用线搜索。(但对于 GD 而言线搜索提升作用很明显, 不可或缺)

7.2 数值结果

方法	cputime	迭代次数	最优点函数值	稀疏程度	与 u 的差
CVX_Mosek	0.82	-	5.80556E-01	0.105	3.78E-05
CVX_Gurobi	0.76	-	5.80556E-01	0.103	3.75E-05
GD	0.11	640	5.80556E-01	0.106	3.90E-05
FGD	0.04	233	5.80556E-01	0.101	3.71E-05

表 3: 不同方法下所得数值结果

下图是函数值相对误差随迭代次数的变化

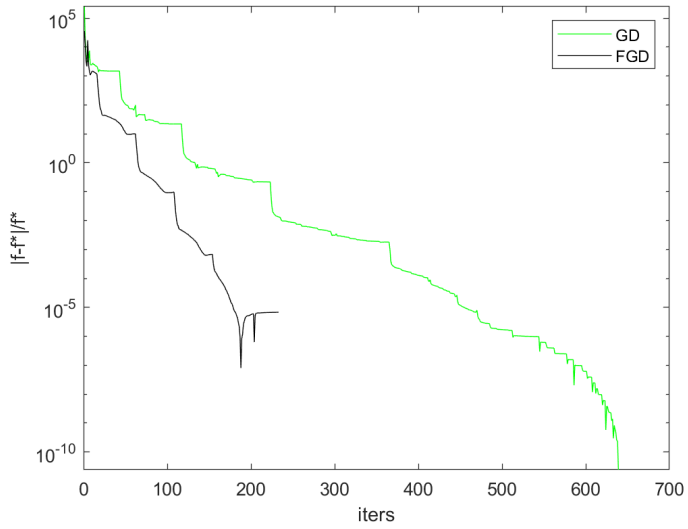

图 2: 函数值相对误差随迭代次数的变化

可以看到 FGD 在运算时间与迭代次数方面都显著优于 GD 算法。

8 近似点梯度法

8.1 算法

在本方法中我们仍然使用连续化方法 (见上文)。下面描述单步近似点梯度法。在下文中为表述方便，梯度视作与自变量 (是矩阵) 形状一致的矩阵。令 $\phi(x) = \frac{1}{2}\|Ax-b\|_F^2$, $h(x) = \mu\|x\|_{1,2}$, 近似点梯度法的迭代格式为

$$x^{k+1} = \mathrm{prox}_{\alpha_k h}(x^k - \alpha_k A^\top(Ax^k - b))$$

，其中近邻算子的计算如下 (逐行计算)：

$$\mathrm{prox}_{\mu h}(x(i,:)) = \begin{cases} 0, & \|x(i,:)\|_2 < \mu \\ \left(1 - \frac{\mu}{\|x(i,:)\|_2}\right) x(i,:), & \text{otherwise} \end{cases}$$

事实上，近似点梯度法的迭代格式根据定义可以写作

$$\begin{aligned} x^{k+1} &= \arg\min_u \left(\|u\|_{1,2} + \frac{1}{2\alpha_k}\|u - x^k + \alpha_k \nabla\phi(x^k)\|_F^2 \right) \\ &= \arg\min_u \left(\|u\|_{1,2} + \phi(x^k) + \langle \nabla\phi(x^k), u - x^k\rangle + \frac{1}{2\alpha_k}\|u - x^k\|_F^2 \right). \end{aligned}$$

令 $\psi_k(x) = \phi(x^k) + \langle \nabla\phi(x^k), x - x^k\rangle + \frac{1}{2\alpha_k}\|x - x^k\|_F^2$ ，针对 $\psi_k(x)$ 考虑线搜索准则，即为 $\psi_k(x^k + \alpha d^k) \le C_k + \rho\alpha\langle \nabla\psi_k(x^k), d^k\rangle$。直到满足条件或进行 5 次步长衰减后退出线搜索循环，得到更新的 x. C_s 为 (Zhang & Hanger) 线搜索准则中的量。

同时我们使用 BB 步长作为线搜索初始步长。令 $s^k = y^{k+1} - y^k$, $z^k = g^{k+1} - g^k$，在偶数与奇数步分别对应 $\frac{\langle s^k, s^k\rangle}{\langle s^k, z^k\rangle}$ 和 $\frac{\langle s^k, z^k\rangle}{\langle z^k, z^k\rangle}$ 两个 BB 步长。

当主循环达到最大迭代次数，或梯度或函数值的变化大于阈值时，退出迭代。特别地，除了每次迭代开始处的收敛条件外，如果连续 8 步的函数值最小值比 8 步之前的函数值超过阈值，则停止内层循环。

8.2 数值结果

本方法数值结果将在后文与 FISTA 方法一同展示并比较。

9 FISTA 方法

9.1 算法

记号沿用上文。在整体框架上与上一题相同，区别只是在于单步 FISTA 做了一步外推。

首先，计算辅助变量

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

在当前步长下进行一步迭代得到 $w^k = y^k - t_k A^\top (Ay^k - b)$ 和 $x^k = \mathrm{prox}_{t_k h}(w^k)$。其中 t_k 我们使用 BB 步长 (同 8.1)，并进行线搜索。线搜索条件为

$$\phi(x^k) \le \phi(y^k) + \langle \nabla\phi(y^k), (x^k - y^k)\rangle + \frac{1}{2t_k}\|x^k - y^k\|_F^2.$$

如果满足了线搜索条件或已经进行了 5 次步长衰减，则停止线搜索循环。

当主循环达到最大迭代次数，或梯度或函数值的变化大于阈值时，退出迭代。特别地，除了每次迭代开始处的收敛条件外，如果连续 8 步的函数值最小值比 8 步之前的函数值超过阈值，则停止内层循环。

9.2 数值结果

方法	cputime	迭代次数	最优点函数值	稀疏程度	与 u 的差
CVX_Mosek	0.82	-	5.80556E-01	0.105	3.78E-05
CVX_Gurobi	0.76	-	5.80556E-01	0.103	3.75E-05
ProxGD(fix)	0.22	1400	5.80556E-01	0.103	3.75E-05
FProxGD(fix)	0.04	180	5.80556E-01	0.104	3.75E-05
ProxGD(BB)	0.06	148	5.80556E-01	0.103	3.75E-05
FProxGD(BB)	0.03	133	5.80556E-01	0.103	3.76E-05

表 4: 不同方法下所得数值结果

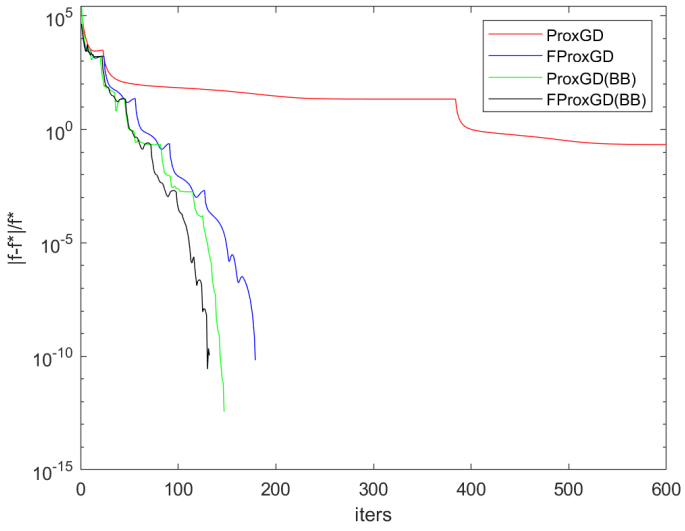

图 3: 函数值相对误差随迭代次数的变化

我们从中可以看到，无论是否使用 BB 步长，FProx 算法在各方面都优于 Prox 算法.BB 步长对于 Prox 算法而言提升明显，而对于 FProx 算法来说提升不那么明显。

但是此图像不能完全刻画各个算法的速度。事实上，非 BB 步长的 FProx 算法是快于 BB 步长 Prox 算法的，尽管迭代次数更多。其原因是后者在线搜索方面占用了一部分时间，而线搜索次数没有被我们统计到迭代次数中。因此总的来说，FProx 算法全面优异与 Prox 算法。

10 增广拉格朗日函数法解对偶问题

10.1 算法

首先将原问题等价转换到如下形式:

$$\begin{aligned} \min_{x\in\mathbb{R}^{n\times l}} \quad & \frac{1}{2}\|y-b\|_F^2+\mu\|x\|_{1,2} \\ \text{s.t.} \quad & Ax=y \end{aligned}$$

引入拉格朗日乘子 z，则拉格朗日函数为

$$L(x,y,z)=\frac{1}{2}\|y-b\|_F^2+\mu\|x\|_{1,2}+\langle Ax-y,z\rangle$$

固定 z 分别对 x,y 极小化可得对偶问题

$$\begin{aligned} \max_{z\in\mathbb{R}^{m\times l}} \quad & -\frac{1}{2}\|z\|_F^2-\langle b,z\rangle \\ \text{s.t.} \quad & \|A^\top z\|_{\infty,2}\le\mu \end{aligned}$$

等价于

$$\begin{aligned} \max_{z\in\mathbb{R}^{m\times l}} \quad & -\frac{1}{2}\|z\|_F^2-\langle b,z\rangle \\ \text{s.t.} \quad & A^\top z=w \\ & \|w\|_{\infty,2}\le\mu \end{aligned}$$

此时对于对偶问题考虑增广拉格朗日函数

$$\begin{aligned} L_\sigma(z,w,x) &= \tfrac{1}{2}\|z\|_F^2+\langle b,z\rangle-\langle A^\top z-w,x\rangle+\tfrac{\sigma}{2}\|A^\top z-w\|_F^2 \\ &= \tfrac{1}{2}\|z\|_F^2+\langle b,z\rangle+\tfrac{\sigma}{2}\|A^\top z-w-x/\sigma\|_F^2-\tfrac{1}{2\sigma}\|x\|_F^2 \end{aligned}$$

因此，ALM 迭代格式为

$$(z^{k+1},w^{k+1})=\arg\min_{z,\|w\|_{\infty,2}\le\mu} L_\sigma(z,w,x^k)$$

$$x^{k+1}=x^k-\sigma(A^\top z^{k+1}-w^{k+1})$$

由于其中的子问题没有显示解，因此我们采用梯度法。

10.2 数值结果

方法	cputime	迭代次数	最优点函数值	稀疏程度	与 u 的差
CVX_Mosek	0.82	-	5.80556E-01	0.105	3.78E-05
CVX_Gurobi	0.76	-	5.80556E-01	0.103	3.75E-05
ALM_Dual	0.24	86	5.80558E-01	0.100	4.18E-05

表 5: 不同方法下所得数值结果

11 ADMM 解对偶问题

11.1 算法

与上一节类似，不同的是当我们得到 $L_\sigma(z,w,x)$ 后，迭代格式为

$$z^{k+1} = \arg\min_z L_\sigma(z, w^k, x^k)$$

$$w^{k+1} = \arg\min_w L_\sigma(z^{k+1}, w, x^k)$$

$$x^{k+1} = x^k - \gamma\sigma(A^\top z^{k+1} - w^{k+1})$$

由于我们是交替地进行最优化求解，因此迭代格式有显式表达式

$$z^{k+1} = (I + \sigma AA^\top)^{-1}(A(\sigma w^k - x^k) - b)$$

$$w^{k+1} = \mathrm{Proj}_{\|\cdot\|_{\infty,2}\le\mu}(A^\top z^{k+1} - x^k/\sigma)$$

$$x^{k+1} = x^k - \gamma\sigma(A^\top z^{k+1} - w^{k+1})$$

11.2 数值结果

方法	cputime	迭代次数	最优点函数值	稀疏程度	与 u 的差
CVX_Mosek	0.82	-	5.80556E-01	0.105	3.78E-05
CVX_Gurobi	0.76	-	5.80556E-01	0.103	3.75E-05
ADMM_Dual	0.21	872	5.80560E-01	0.102	3.83E-05

表 6: 不同方法下所得数值结果

12 ADMM 解原问题

首先将原问题等价转换到如下形式:

$$\min_{x\in\mathbb{R}^{n\times l}} \quad \frac{1}{2}\|Ax-b\|_F^2 + \mu\|y\|_{1,2}$$

$$\text{s.t.} \quad x = y$$

增广拉格朗日函数为

$$\begin{aligned} L_\sigma(z,w,x) &= \tfrac{1}{2}\|Ax-b\|_F^2 + \mu\|y\|_{1,2} - \langle x-y, z\rangle + \tfrac{\sigma}{2}\|x-y\|_F^2 \\ &= \tfrac{1}{2}\|Ax-b\|_F^2 + \mu\|y\|_{1,2} + \tfrac{\sigma}{2}\|x-y-z/\sigma\|_F^2 - \tfrac{1}{2\sigma}\|z\|_F^2 \end{aligned}$$

于是，ADMM 迭代格式为

$$x^{k+1} = \arg\min_x L_\sigma(x, y^k, z^k) = (\sigma I + A^\top A)^{-1}(A^\top b + \sigma y^k + z^k)$$

$$y^{k+1} = \arg\min_y L_\sigma(x^{k+1}, y, z^k) = \mathrm{prox}_{\mu h/\sigma}(x^{k+1} - z^k/\sigma)$$

$$z^{k+1} = z^k - \gamma\sigma(x^{k+1} - y^{k+1})$$

12.1 数值结果

方法	cputime	迭代次数	最优点函数值	稀疏程度	与 u 的差
CVX_Mosek	0.82	-	5.80556E-01	0.105	3.78E-05
CVX_Gurobi	0.76	-	5.80556E-01	0.103	3.75E-05
ADMM_Primal	0.29	715	5.80559E-01	0.103	3.77E-05

表 7: 不同方法下所得数值结果

13 程序文件中的默认参数

程序文件中的默认参数如下:

* |opts.maxit| ：最大外层迭代次数
* |opts.maxit_inn| ：最大内层迭代次数
* |opts.ftol| ：针对函数值的停机判断条件
* |opts.gtol| ：针对梯度的停机判断条件
* |opts.factor| ：正则化系数的衰减率
* |opts.mu1| ：初始的正则化系数
* |opts.alpha0| ：初始步长
* |opts.opts1| ：结构体，用于向内层算法提供其它具体的参数
* |opts.thres| ：判断小量是否被认为为 0 的阈值
* |opts.bb| ：是否使用 BB 步长
* |opts1.maxit| ：最大 (内部) 迭代次数
* |opts.alpha0| ：步长的初始值
* |opts.verbose| ：不为 0 时输出每步迭代信息，否则不输出
* |opts.ls| ：标记是否线搜索
* |opts.sigma| ：Huber 光滑化参数 σ

14 声明与致谢

测试代码依照 http://bicmr.pku.edu.cn/~wenzw/courses/Test_group_lasso.m
代码中部分结构参考《最优化：建模、算法与理论/最优化计算方法》中配套代码。