参考文献

[1] Benjamin Rolfs, Bala Rajaratnam, Dominique Guillot, Ian Wong, and Arian Maleki. Iterative thresholding algorithm for sparse inverse covariance estimation. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1574–1582. Curran Associates, Inc., 2012.

大数据中的算法项目一

黄晃数院 1701210098

2018 年 4 月 17 日

1 问题一

原问题为

$$\begin{aligned} &min\ \|x\|_1 \\ &sub\ Ax = b \end{aligned} \tag{1}$$

1.1 cvx calling mosek or gurobi

因为primal 1是个凸问题,cvx下可以直接

$$minimize(norm(x, 1))$$

$$subAx = b$$

1.2 calling mosek or gurobi directly

1.2.1 问题转化

将primal转化为LP

$$\begin{aligned} &min\ \Sigma_{i=1}^{n} t_i \\ &sub\ \ -t_i \leq x_i \leq t_i \end{aligned} \tag{2}$$

引入$v_i = t_i - x_i$,则 2转化为

$$\begin{aligned} &min\ \Sigma_{i=1}^{n}(x_i + v_i) \\ &sub\ 2x_i + v_i \geq 0 \\ &v_i \geq 0 \end{aligned} \tag{3}$$

1.3 bregman iteration

Classical Augmented Lagrangian method (or Bregman method), where each augmented Lagrangian function is minimized by using the proximal gradient method

1.3.1 外部迭代

由所给参考文献,我们选择所给的Bregman method中的version2:

$$\begin{aligned} b_0 = 0, x_0 = 0 \\ fork = 1, 2, 3, \cdots do \\ b_{k+1} = b + b_k - Ax_k \\ x_{k+1} = argmin\ \mu\|x\|_1 + \frac{1}{2}\|Ax - b_{k+1}\|^2 \end{aligned} \tag{4}$$

1.3.2 子问题

更新x_k时,我们使用要求的proximal方法,所以子问题的迭代为

$$x_{m+1} = shrink(x_m - \tau\nabla f(x), \mu\tau)$$

其中$f(x) = \frac{1}{2}\|Ax - b_{k+1}\|^2$,k为外部迭代次数,m为内迭代次数

子问题终止条件 子问题通过$\|x_{m+1} - x_m\| < \epsilon\ or\ m > MAX$来终止

同伦 为了更好的求解子问题,对于给定的μ_0,采取了同伦的技巧来进行优化,具体为依次求解$\mu = 10000\mu, 1000\mu, 100\mu, 10\mu, \mu$对应的问题,然后用其解作为下一次的初始点,相应的终止条件ϵ也做适当放大

子问题参数 参数$\tau = \frac{1}{norm(A^TA)}\mu, \epsilon = [2e-4, 1e-5, 2e-6, 1e-6, 1e-6]$

1.3.3 外部终止条件以及参数的选择

根据文献,外部迭代的终止条件为

$$\frac{\|b_k - Ax_k\|}{\|b\|} < 1e-8$$

参数$\mu_0 = 0.001$

1.4 ADMM for dual

Alternating direction method of multipliers (ADMM) for the dual problem

1.4.1 外部迭代

对偶问题为

$$\begin{aligned} & max\ b^T y \\ & sub\ \|A^T y\|_\infty \le 1 \end{aligned} \tag{5}$$

由参考文献,ADMM迭代格式为

$$\begin{aligned} z_{k+1} =& P_{B_1^\infty}(A^T y_k + x_k/\beta) \\ y_{k+1} =& (A^T A)^{-1}(A z_{k+1} - (A x_k - b)/\beta) \\ x_{k+1} =& x_k - \gamma\beta(z_{k+1} - A^T y_{k+1}) \end{aligned} \tag{6}$$

1.4.2 终止条件以及参数的选择

根据文献,终止条件为

$$min\{\|r_p\|/\|b\|, \|r_d\|/\sqrt{m}, |\triangle|/\|x\|_1\} \le \epsilon$$

其中

$$\begin{aligned} r_p =& Ax - b \\ r_d =& A^T y - z \\ \triangle =& b^T y - \|x\|_1 \end{aligned} \tag{7}$$

参数$\gamma = 1, \beta = 2 * m/\|b\|_1, \epsilon = 1e - 8$

1.5 数值结果

method	nrm1fun	res	cpu	err to mosek
cvx-mosek	7.85e+01	1.90e-10	1.38	0
cvx-gurobi	7.85e+01	2.95e-09	0.86	3.29e-10
call-mosek	7.85e+01	1.62e-07	1.90	1.00e-09
call-mosek	7.85e+01	1.62e-07	1.92	1.00e-09
bregman	7.85e+01	3.95e-07	3.25	2.43e-09
admm	7.85e+01	4.49e-13	0.07	3.60e-09

1.5.1 结果分析

- 各个算法都达到了1e-9的精度
- 约束条件$Ax = b$的残量都达到了1e-7以上
- cpu时间方面,除了ADMM外均与cvx处在一个量级
- ADMM除了耗时极短(0.07)外,约束条件也满足的最好,达到了1e-13,高于cvx的1e-10

2 问题二

Algorithms For Sparse Inverse Covariance Estimation

2.1 数据产生

由参考文献,$S = \Sigma_n = \frac{1}{n}\sum_{k=1}^{n}(X_i - \overline{X})(X_i - \overline{X})^T$,其中$\overline{X} = \frac{k=1}{n}X_k$. $X \sim N(0, \Sigma_0)$, $\Sigma_0 = \Omega_1^{-1}$,所要求的两个模型只有$\Omega_0 = (\omega_{ij})$不同. 生成Ω_0后可得到对应的Σ_0,然后生成n=100组数据(X),然后求X的协方差矩阵作为Σ_n,注意到文献所给的定义与一般的定义有系数上的差别,所以用matlab的cov函数求得协方差后还需乘以$\frac{n}{n-1}$

2.1.1 model1

$\omega_{ij} = 0.6^{|i-j|}$

2.1.2 model2

$\Omega_0 = B + \delta I$,其中B非对角元满足两点分布,以0.1概率为0.5,0.9概率为0.然后δ使得Ω_0的条件数等于p,最后再单位化对角元.

δ的确定 要使得$\sigma_1/\sigma_p = p$,先求B最大特征与最小特征λ_1, λ_p,然后由于Ω的奇异值满足$\sigma_k = abs(\lambda_k + \delta)$,所以取$\delta = \frac{p\lambda_p - \lambda_1}{1-p}$即可

2.2 1

$$\max_{X \succeq 0} logdet(X) - Tr(SX) - \rho \|X\|_1 \tag{8}$$

其中

$$\|X\|_1 = \sum_{ij} |X_{ij}| \tag{9}$$

2.2.1 dual

注意到

$$\|X\|_1 = \sum_{ij} |X_{ij}| = \max_{\|U\|_\infty \leq 1} Tr(UX) \tag{10}$$

而

$$\begin{aligned} & \max_{X \succeq 0} logdet(X) - Tr(SX) - \rho \|X\|_1 \\ = & \max_{X \succeq 0} \min_{\|U\|_\infty \leq 1} logdet(X) - Tr(SX) - \rho Tr(UX) \\ = & \max_{X \succeq 0} \min_{\|U\|_\infty \leq \rho} logdet(X) - Tr(SX) - Tr(UX) \\ = & \max_{X \succeq 0} \min_{\|U\|_\infty \leq \rho} logdet(X) - Tr((S+U)X) \end{aligned} \tag{11}$$

交换max与min,对X求导$D(X) = X^{-1} - U - S$,由一阶条件,对偶函数为

$$\begin{aligned} g(U) &= \min_{\|U\|_\infty \leq \rho} \max_{X \succ 0} logdet(X) - Tr((S+U)X) \\ &= \min_{\|U\|_\infty \leq \rho} -logdet(U+S) - p \end{aligned} \tag{12}$$

等价于

$$\min_{U \in S_+} -logdet(U+S)-p \\ sub\|U\|_\infty \leq \rho \tag{13}$$

2.2.2 use cvx

只用考虑$\|X\|_1$的表示即可,这里选择构造全一矩阵$S1$,则$\|X\|_1 = trace(S1 * abs(X))$

2.2.3 method

选择了 [1]所描述的threating算法来求解该问题,详细见 [1]的3.2节,在此不再赘述

初值 初值选择参考 [1],$X_{ii} = (S_{ii}+\rho)^{-1}$

终止条件 终止条件为对偶间隙$\delta < \epsilon$,或者达到最大步数,或者是对给定的下降方向,线搜索找不到满足要求的α

2.2.4 数值结果

为了比较算法与cvx的计算结果,定义如下两个量

$$err_F = (F(X_2)-F(X_1))/(1+abs(F(X_1)))$$

$$err_X = norm(X_2-X_1)/(1+norm(X1))$$

其中X_1, X_2分别为cvx与threating的解

$F(X) = -logdet(X)+trace(SX)+\rho\|X\|_1$为目标函数

12分别展示了两个model的计算结果

此外为了观察终止条件的满足情况,下面图 1 2展示了两个model$\rho = 0.001$时对偶间隙δ随迭代的变化.

2.2.5 结果分析

- 由于model1比model2形式更简单,显示在计算结果上就是threaing方法对model1处理的更好

表 1: model1

method / ρ	cvx		threating				
	cpu	F	cpu	F	err_F	err_X	ite
10	2.02	1.05e+02	0.01	1.05e+02	3.07e-07	1.84e-05	1
1	2.67	6.63e+01	3.21	6.63e+01	-3.70e-07	2.88e-05	10000
0.1	2.19	4.90e+01	8.43	4.90e+01	1.23e-07	4.65e-05	10000
0.01	1.91	1.89e+02	41.92	1.89e+02	-2.84e-05	4.46e-05	10000
0.001	1.86	5.95e+01	16.77	5.95e+01	-9.69e-06	8.18e-05	10000

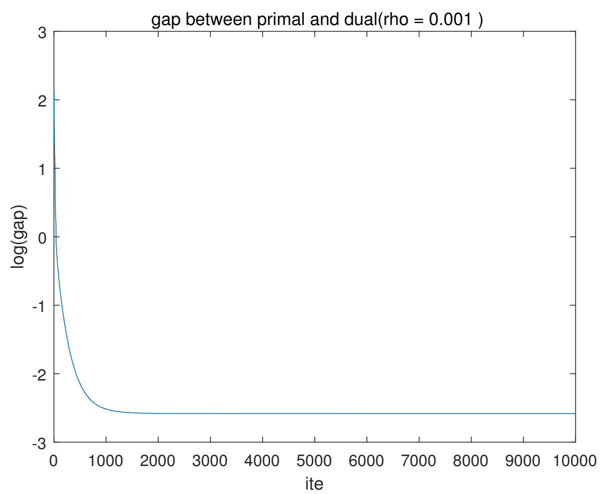

图 1: model1

表 2: model2

method / ρ	cvx		threating				
	cpu	F	cpu	F	err_F	err_X	ite
10	2.00	7.81e+01	0.01	7.81e+01	4.12e-06	2.50e-06	1
1	2.13	5.87e+01	0.00	5.87e+01	-5.85e-04	3.69e-02	1
0.1	2.21	8.42e+01	30.11	8.42e+01	-3.46e-06	5.97e-05	10000
0.01	1.86	4.43e+01	26.10	4.43e+01	5.11e-06	1.02e-04	10000
0.001	1.56	3.26e+01	4.76	3.26e+01	1.90e-05	8.67e-04	10000

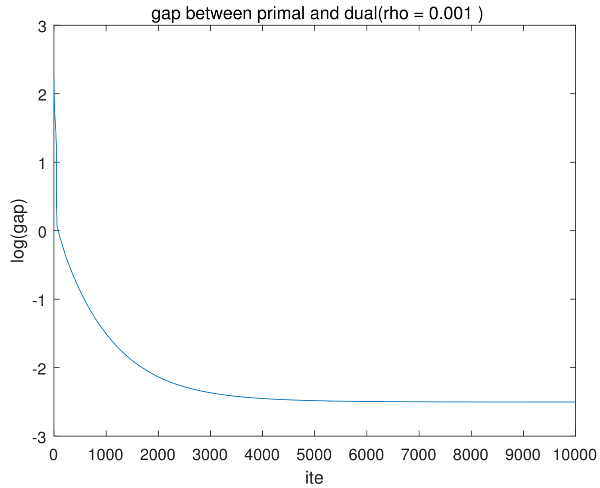

图 2: model2

- 观察到$\rho = 10, 1$时threating方法只进行了一次迭代,更换初值发现,这取决于初值的选择,更改为随机的一组初值,计算结果表现较差
- $\rho = 0.1, 0.01, 0.001$时,threating方法均正常运行,迭代10000步计算结果均到了1e-4以上
- 预设的终止条件$\delta < \epsilon$在ρ较小时均为达到,具体见12所示,在2000步左右趋于稳定,而不能达到预设的1e-4
- 时间方面,若只需达到表中所给精度,可将最大迭代次数改小至2000左右(因为对偶间隙条件达不到),所以threating方法的时间花费与cvx相当

2.3 2(extra-credit)

考虑问题

$$\begin{aligned} &\max_{X \succeq 0} \|X\|_1 \\ &sub \|SX_I\|_1 \leq \rho \end{aligned} \tag{14}$$

2.3.1 dual

定义集合$B = \{U \in S^p \mid U_{ij} \in \{1, -1\}\}$ 则有

$$\begin{aligned} \|X\|_1 = \sum_{ij} |X_{ij}| = \max_{\|U\|_\infty \leq 1} Tr(UX) \\ = \max_{U \in B} Tr(UX) \end{aligned} \tag{15}$$

则对应的lagrangian函数为 $L(X, \Lambda) = \|X\|_1 + \sum_{U_\alpha \in B} \lambda_\alpha (< U_\alpha, SX - I > -\rho)$ 一阶次梯度为

$$\partial L / \partial X = Z + \sum_{U_\alpha \in B} \lambda_\alpha U_\alpha S$$

，因为Z_{ij}可取遍$[-1, 1]$,所以当且仅当$\| \sum_{U_\alpha \in B} \lambda_\alpha U_\alpha S \|_\infty \leq 1$时导数可取到0,而且易见此时取X=0可使次梯度为0.因此问题转化成

$$\begin{aligned} \min \sum_{U_\alpha \in B} \lambda_\alpha + < \sum_{U_\alpha \in B} \lambda_\alpha U_\alpha > \\ sub \| \sum_{U_\alpha \in B} \lambda_\alpha U_\alpha S \|_\infty \leq 1 \end{aligned} \tag{16}$$

2.3.2 use cvx

类似于上一个问题,目标函数可写成trace(S1*abs(X)),而约束为$trace(S1 * abs(S * X2 - I)) <= rho$