Project 1 for "Algorithms for Big-Data Analysis"

Zaiwen Wen
Beijing International Center for Mathematical Research
Peking University

April 11, 2020

1 Submission Requirement

1. Prepare a report including
   - detailed answers to each question
   - numerical results and their iterpretation
2. The programming language can be either matlab, Python or c/c++.
3. Pack all of your codes named as "proj1mk-name-ID.zip" and upload the file to `https://file.admin.cluster-bicmr.com/u/d/045d80868f524d0bab11/`
   作业提交需要统一打包成压缩文件，命名格式为：proj1mk-学号-姓名，文件类型随意。文件名中不要出现空格，最好不要出现中文。
4. 请勿大量将代码粘在报告中，涉及到实际结果需要打表或者作图，不要截图或者直接从命令行拷贝结果。
5. 提交word 的同学需要提供word 原文件并将其转换成pdf 文件。
6. If you get significant help from others on one routine, write down the source of references at the beginning of this routine.

2 Algorithms for ℓ_1 minimization

Consider the problem

$$\min_x \quad \mu\|x\|_1 + \|Ax - b\|_1, \tag{2.1}$$

where $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ are given. Test data are as follows:

```
n = 1024;
m = 512;
A = randn(m,n);
u = sprandn(n,1,0.1);
```

```
b = A*u;
mu = 1e-2;
```

See `http://bicmr.pku.edu.cn/~wenzw/bigdata/Test_BP.m`

1. Solve (2.1) using CVX by calling different solvers mosek or gurobi.
   CVX, Mosek and Gurobi are available free at:
   CVX: `http://cvxr.com/cvx/`
   Mosek: `http://www.mosek.com/`
   Gurobi: `http://www.gurobi.com/`

2. Write down and implement one of the following algorithms in Matlab/Python:

   (a) Classical Augmented Lagrangian method (or Bregman method), where each augmented Lagrangian function is minimized by using the proximal gradient method
   Reference: Wotao Yin, Stanley Osher, Donald Goldfarb, Jerome Darbon, *Bregman Iterative Algorithms for l1-Minimization with Applications to Compressed Sensing*

   (b) Classical Augmented Lagrangian method (or Bregman method), where each augmented Lagrangian function is minimized by using the accelerated proximal gradient method (FISTA or Nesterov's method)
   Reference on FISTA: Amir Beck and Marc Teboulle, *A fast iterative shrinkage thresholding algorithm for linear inverse problems*

3. Write down and implement one of the following algorithms in Matlab/Python:

   (a) Alternating direction method of multipliers (ADMM) for the primal or dual problem
   Reference: Junfeng Yang, Yin Zhang, *Alternating direction algorithms for l1-problems in Compressed Sensing*, SIAM Journal on Scientific Computing, https://epubs.siam.org/doi/abs/10.1137/090777761

   (b) Alternating direction method of multipliers with linearization for the primal or dual problem
   Reference: Junfeng Yang, Yin Zhang, *Alternating direction algorithms for l1-problems in Compressed Sensing*, SIAM Journal on Scientific Computing, https://epubs.siam.org/doi/abs/10.1137/090777761

4. Requirement:

   (a) The interface of each method should be written in the following format

   ```
   [x, out] = method_name(x0, A, b, mu, opts);
   ```

   Here, x0 is a given input initial solution, A and b are given data, opts is a struct which stores the options of the algorithm, out is a struct which saves all other output information.

   (b) Compare the efficiency (cpu time) and accuracy (checking optimality condition) in the format as
   `http://bicmr.pku.edu.cn/~wenzw/bigdata/Test_BP.m`

3 Algorithms For Sparse Inverse Covariance Estimation

Let $S^n = \{X \in \mathbb{R}^{n\times n} \mid X^\top = X\}$. Let $S \in S^n$ be a given observation of covariance matrix.

1. Consider the model

$$\max_{X\succeq 0} \log\det X - \mathrm{Tr}(SX) - \rho\|X\|_1, \tag{3.1}$$

where $\|X\|_1 = \sum_{ij} |X_{ij}|$.

(a) data sets: set $n = 30$, generate models 1 and 2 in section 5.1 of page 599 at
`http://www-stat.wharton.upenn.edu/~tcai/paper/Precision-Matrix.pdf`

(b) Derive the dual problem of (3.1).

(c) Solve (3.1) using CVX using a few ρ, for example, $10, 0.1, 0.001$.

(d) Write down and implement a first-order type algorithm for solving (3.1) with the same ρ in (c).

2. (Optional, Extra-credit) Consider the model

$$\min_{X \succeq 0} \|X\|_1 + \frac{\sigma}{2}\|SX - I\|_F^2, \tag{3.2}$$

where I is the identity matrix, $\|X\|_1 = \sum_{ij} |X_{ij}|$ and $\|X\|_F^2 = \sum_{ij} X_{ij}^2$.

(a) data sets: set $n = 30$, generate models 1 and 2 in section 5.1 of page 599 at
`http://www-stat.wharton.upenn.edu/~tcai/paper/Precision-Matrix.pdf`

(b) Solve (3.2) using CVX using a few σ, for example, $10, 0.1, 0.001$.

(c) Write down and implement a first-order type algorithm for solving (3.2) with the same σ in (b).

Note that there is a constraint $X \succeq 0$ in (3.2).