A Mathematical Introduction to Data Science **April 5, 2016**

Homework 5. RPCA and SPCA

Instructor: Yuan Yao *Due: Tuesday April 12, 2016*

The problem below marked by * is optional with bonus credits.

1. *RPCA*: Construct a random rank-r matrix: let $A \in \mathbb{R}^{m\times n}$ with $a_{ij} \sim \mathcal{N}(0,1)$ whose top-r singular value/vector is λ_i, $u_i \in \mathbb{R}^m$ and $v_i \in \mathbb{R}^n$ $(i = 1, \ldots, r)$, define $L = \sum_{i=1}^r u_i v_i^T$. Construct a sparse matrix S with p percentage ($p \in [0,1]$) nonzero entries distributed uniformly. Then define
$$M = L + S.$$
   (a) Set $m = n = 20$, $r = 1$, and $p = 0.1$, use Matlab toolbox CVX to formulate a semi-definite program for Robust PCA of M:
$$\begin{aligned} \min \quad & \frac{1}{2}(\mathrm{trace}(W_1) + \mathrm{trace}(W_2)) + \lambda \|S\|_1 \\ s.t. \quad & L_{ij} + S_{ij} = X_{ij}, \quad (i,j) \in E \\ & \begin{bmatrix} W_1 & L \\ L^T & W_2 \end{bmatrix} \succeq 0, \end{aligned} \tag{1}$$
   where you can use the matlab implementation in lecture notes as a reference;
   (b) Choose different parameters $p \in [0,1]$ to explore the probability of successful recover;
   (c) Increase r to explore the probability of successful recover;
   (d) * Increase m and n to values beyond 50 will make CVX difficult to solve. In this case, use the Augmented Lagrange Multiplier method, e.g. in E. J. Candes, X. Li, Y. Ma, and J. Wright (2009) "Robust Principal Component Analysis?". Journal of ACM, 58(1), 1-37 (`http://www.math.pku.edu.cn/teachers/yaoy/Fall2011/rpca.pdf`). Make a code yourself (just a few lines of Matlab or R) and test it for $m = n = 1000$. A convergence criterion often used can be $\|M - \hat{L} - \hat{S}\|_F / \|M\|_F \leq \epsilon$ ($\epsilon = 10^{-6}$ for example).

2. *SPCA*: Define three hidden factors:
$$V_1 \sim \mathcal{N}(0, 290), \quad V_2 \sim \mathcal{N}(0, 300), \quad V_3 = -0.3V_1 + 0.925V_2 + \epsilon, \quad \epsilon \sim \mathcal{N}(0,1),$$
   where V_1, V_2, and ϵ are independent. Construct 10 observed variables as follows
$$X_i = V_j + \epsilon_i^j, \quad \epsilon_i^j \sim \mathcal{N}(0,1),$$
   with $j = 1$ for $i = 1, \ldots, 4$, $j = 2$ for $i = 5, \ldots, 8$, and $j = 3$ for $i = 9, 10$ and ϵ_i^j independent for $j = 1, 2, 3$, $i = 1, \ldots, 10$.

   The first two principal components should be concentrated on (X_1, X_2, X_3, X_4) and (X_5, X_6, X_7, X_8), respectively. This is an example given by H. Zou, T. Hastie, and R. Tibshirani, Sparse principal component analysis, J. Comput. Graphical Statist., 15 (2006), pp. 265-286.

(a) Compute the true covariance matrix Σ (and the sample covariance matrix with n examples, say $n = 1000$);

(b) Compute the top 4 principal components of Σ using eigenvector decomposition (by Matlab or R);

(c) Use Matlab CVX toolbox to compute the first *sparse* principal component by solving the SDP problem

$$\begin{aligned} \max \quad & \text{trace}(\Sigma X) - \lambda \|X\|_1 \\ s.t. \quad & \text{trace}(X) = 1 \\ & X \succeq 0 \end{aligned}$$

Choose $\lambda = 0$ and other positive numbers to compare your results with normal PCA;

(d) Remove the first sparse PCA from Σ and compute the second sparse PCA with the same code;

(e) Again compute the 3rd and the 4th sparse PCA of Σ and compare them against the normal PCAs.

(f) * Construct an example with 200 observed variables which is hard to deal with by CVX. In this case, use the Augmented Lagrange Multiplier method by Allen Yang et al. (UC Berkeley) whose Matlab codes can be found at `http://www.eecs.berkeley.edu/~yang/software/SPCA/SPCA_ALM.zip`.