DSA5103 Assignment 3

Instructions While all languages are acceptable, it is recommended that you code using Python or MATLAB. You must write your own code. Due on April 16, 11:59pm. Please submit a pdf file including answers for **every** question. In addition, you should submit the codes for Q2 (.m file or .ipynb file) that I can run for generating the numerical results reported in the pdf file. No programming is required for Q1.

1. **ADMM** Consider finding a point in intersection of two convex sets $C, D \subseteq \mathbb{R}^n$ via solving
$$\min_x \quad \delta_C(x) + \delta_D(x)$$
where δ_C is the indicator function of the set C.

   (a) Transform the above problem into the 2-block separable structure that ADMM can handle.

   (b) Give the explicit formulae of the ADMM iterations.

   (c) Set $\sigma = 1$, $\tau = 1$, $C = [1, 2] \subseteq \mathbb{R}$, $D = [1.5, 3] \subseteq \mathbb{R}$. Initialize every variable by zero. Compute the first two iterations of ADMM.

2. **Robust PCA** In this question, we will develop ADMM for Robust Principal Component Analysis and use it for the problem of foreground-background segmentation. Let $M \in \mathbb{R}^{m \times n}$ be the data matrix and we aim to decompose it into a low-rank matrix L and a sparse matrix S
$$\min_{L,S} \|L\|_* + \lambda \|S\|_1 \quad \text{s.t.} \quad L + S = M$$
where $\|\cdot\|_*$ denotes the nuclear norm and $\|\cdot\|_1$ denotes the ℓ_1 norm. The input of the ADMM should be M and λ and the output should be L and S. By default, we set $\lambda = \frac{1}{\sqrt{\max(m,n)}}$. We always use the zero matrix as the initial point. The method is terminated when
$$r^{(k)} := \max\left\{ \frac{\|L^{(k)} - L^{(k-1)}\|_F}{1 + \|L^{(k)}\|_F}, \frac{\|S^{(k)} - S^{(k-1)}\|_F}{1 + \|S^{(k)}\|_F} \right\} < 10^{-4}$$
or the number of iterations exceeds 200.

(a) Set $m = 500$, $n = 1000$. Generate an $m \times 10$ random matrix W and a $10 \times n$ random matrix H where each entry W_{ij} or H_{ij} is a random variable drawn from the standard normal distribution $N(0, 1)$. Let $L_0 = WH$. In addition generate a random sparse matrix $S_0 \in \mathbb{R}^{m \times n}$ with approximately $0.05mn$ nonzero entries (by "scipy.sparse.random" in Python or "sprandn" in MATLAB). Let $M = L_0 + S_0$. Run your algorithm and then report the differences between your approximate solutions L, S generated by ADMM and the truth L_0, S_0:

$$\|L - L_0\|_F,\ \|S - S_0\|_F.$$

In addition report the number of iterations and running time. (Your algorithm should converge very quickly)

(b) You may further speed up the codes by the following tricks:

- Use economy or reduced SVD instead of full SVD (most likely the full SVD cannot be stored for high-dimensional data)
- Tune parameter σ in every iteration via

$$\sigma^{(0)} = \frac{1}{\sigma_{\max}(M)} \qquad \text{inverse of the largest singular value of } M$$

$$\sigma^{(k+1)} = \rho\sigma^{(k)} \qquad \text{the ratio } \rho \geq 1 \text{ can be, e.g., } \rho = 1.1$$

(c) Apply your code for foreground-background segmentation of the Basketball player video: $n = 112$, $m = 918 \times 1374 = 1261332$. Download from
`https://1drv.ms/u/s!AgcVCsHBmAstgYwKdOhmNQ4AgMbJag?e=w8simu`
The file "BasketballPlayer.csv" is the data matrix $M \in \mathbb{R}^{m \times n}$ where each column is a video frame. Report the number of iteration, running time, the term $r^{(k)}$ defined above, the rank of your estimated matrix L, and the number of nonzero entries in your estimated matrix S.

(d) Visualize the 20-th frame: $M_{\cdot 20}$, $L_{\cdot 20}$, $S_{\cdot 20}$, namely reshape the 20-th column of the raw video frame $M_{\cdot 20}$, the 20-th column of the background $L_{\cdot 20}$, the 20-th column of the foreground $S_{\cdot 20}$ into 918×1374 images and show the images. Make sure that you first find the maximal and minimal values in L and S, then scale the entries into $[0, 1]$, otherwise the images will not look correct.

(e) Visualize L and S by making a background video and a moving object video.