Accelerated proximal gradient for robust PCA

(Due on Sep.-27-2019)

Instruction: Each student needs to submit the source code file and a report by Latex. In the report file, you can summarize what you observe or difficulties you have. **You will be evaluated based on correctness of your algorithms, accuracy and efficiency of algorithms, performance on applications, and also the report**. Compress your files into a single .zip file, name it as "MATP6960_Assignment1_YourName", and send it to `optimization.rpi@gmail.com`

1 Robust principal component analysis

Let X be composed of a sparse matrix S and a low-rank matrix L. The robust PCA aims at finding S and L, given X. Using ℓ_1 norm to promote sparsity and nuclear norm to promote low-rankness, robust PCA can be modeled as

$$\min_{L,S} \|L\|_* + \lambda\|S\|_1, \text{ s.t. } L + S = X. \tag{1}$$

Here, $\|L\|_*$ denotes the nuclear norm of L and equals the summation of its singular values, and $\|S\|_1 = \sum_{i,j} |s_{ij}|$. Penalizing the constraint into the objective, one can solve the penaty problem

$$\min_{L,S} \|L\|_* + \lambda\|S\|_1 + \frac{\beta}{2}\|L + S - X\|_F^2, \tag{2}$$

or a sequence of the above problem, where $\beta > 0$ is the penalty parameter.

2 Requirements

Include every item below in a single report and attach your code. Use the provided datasets to test your code.

1. Develop three solvers for (2): one is by the proximal gradient, and another is by the accelerated proximal gradient with two versions discussed in class. Treat λ and β as an input of your solvers.

2. Compare the two solvers by using the provided synthetic data and also Escalator video Dataset. Note that the video dataset is in 3D format. It contains 200 frames of size 130×160. You will need first reshape each frame into a column vector and form a 20800×200 matrix X. Report how these solvers decrease the objective values in terms of iteration number of also actual running time. Reshape each column of L and S into a 130×160 image and use `imshow` to show a few selected columns of L and S. You can also use `implay` to see how the foreground and background of the video are separated. Report what you observe.