ISyE 8803: Topics on High Dimensional Data Analytics

Homework 5

Question 1. Robust PCA (25 points)

Recently, robust PCA has become popular for many modern problems, including video surveillance (where the background objects appear in low rank matrix and foreground objects appear in sparse matrix), face recognition (eigenfaces are in low rank matrix and shadows, occlusions, etc. are in sparse matrix). In this problem, we want to use robust PCA on the Extended Yale Face Database to decompose some of its images to the low-rank component and occluded regions.

The Extended Yale Face Database consists of cropped and aligned images of 38 individuals (28 from the extended database, and 10 from the original database) under 9 poses and 64 lighting conditions. Each image is 192 pixels tall and 168 pixels wide. Each of the facial images in this data set have been reshaped into a large column vector with 192 * 168 = 32256 elements.

For this problem, consider the first 64 columns of the data set (corresponding to the first 64 images), then apply robust PCA method on that. Your response should include image numbers: 3, 4, 14, 15, 17, 18, 19, 20, 21, 32, and 43. Show each image along with its low rank image and sparse image. Comment on how effective RPCA is to fill in occluded regions of the image corresponding to the shadows.

Question 2. Matrix Recovery (25 points)

'ratings.mat' M_0 contains ratings on a 1 to 10 scale by 200 viewers for 100 movies. 50% of the ratings were randomly removed (i.e. replaced with 0) and stored in 'ratings_missing.mat' M_1. Recover the original matrix from 'ratings_missing.mat' by using the two methods below:

- Method 1: By directly solving the following optimization problem:

$$\min_{M} \|M\|_*$$

subject to $M(i,j) = M_1(i,j) \quad \forall (i,j) \in \{observed\ set\}$

- Method 2: By using the completion algorithm (i.e. using singular value thresholding to solve the above optimization problem)

Include the following results and commentary in your report:

(a) For Method 1, report the optimal objective function value and the relative reconstruction error. The relative reconstruction error is defined by:

$$\frac{\|M - M_0\|_F}{\|M_0\|_F}$$

(b) For Method 2, report the relative reconstruction error for the following values of δ and τ after (i) 1000 iterations and (ii) 2000 iterations. Comment on the results.

$\boldsymbol{\delta}$	$\boldsymbol{\tau}$
0.1	50
2	50
0.1	500
2	500

(c) Comment on the performance (i.e. execution time etc.) of Method 1 and Method 2. Conclude by providing your recommendation on which is a better method.

Question 3. Compressive Sensing for Color Images (25 points)

Color images are acquired in three channels - Red (R), Green (G) and Blue (B). The data (image) acquired by each of the channels is sparse in DCT or wavelet domain. i.e.,

$$C = \Phi^T \mathcal{X}_C,\ C \in \{R, G, B\}.$$

$$\mathcal{X}_C = \Phi C$$

Where C represent each of the three-color channels, Φ is the (sparsifying) transform matrix and $\mathcal{X}_C$ is the sparse transform coefficient for each color channel.

(1). In this problem, you are required to compress the color image (Lenna.png) to 70% of its original size by using compressive sensing matrix A_C (random sampling). The sensing matrix can be the same for all channels. Define your own sensing matrix A_C and plot the compressed color image.

(2). In this problem, use the provided code DWT.m to generate the transform matrix Φ and recover the color image. Compare the original color image with recovered color image. Compute the reconstruction error in terms of MSE.

Hint: in compact matrix-vector notation, $y_c = A_C C = A_C \Phi^T \mathcal{X}_C,\ C \in \{R, G, B\}$ can be expressed as

$$\begin{bmatrix} y_R \\ y_G \\ y_B \end{bmatrix} = \begin{bmatrix} A_R\Phi^T & 0 & 0 \\ 0 & A_G\Phi^T & 0 \\ 0 & 0 & A_B\Phi^T \end{bmatrix} \begin{bmatrix} \mathcal{X}_R \\ \mathcal{X}_G \\ \mathcal{X}_B \end{bmatrix}$$

In short,

$$y = A\mathcal{X}$$

Where $y = [{y_R}^T {y_G}^T {y_B}^T]^T, A = \text{BlockDiag}(A_R\Phi^T, A_G\Phi^T, A_B\Phi^T)$ and $\mathcal{X} = [{\mathcal{X}_R}^T {\mathcal{X}_G}^T {\mathcal{X}_B}^T]^T$

Question 4. Sparse Smooth Decomposition (25 points)

In this problem, we're going to use Sparse Smooth Decomposition to extract features rather than detect anomalies. Provided are two images of heatmaps of a GPU lid. One is at idle, and the other one is under load (training a CNN in to classify tensors of birds and cats.) We want to programmatically detect where heat spreads on the lid under load so engineers can design appropriate heatsinks and place thermal sensors on the die.

Unfortunately, the temperature sensor we have is very noisy when the GPU is idle due to the temperature differentials being quite small. Therefore, we can't solely rely on image processing techniques from Module 2, such as simply subtracting the at-idle image from the at-load image and doing edge detection.

A: 10%) Read in both images and convert to grayscale. To demonstrate why this would be an ugly problem with simple techniques, show a simple subtraction of the idle image from the load image as the deliverable for this part.

B: 40%) Implement 2-D SSD. For purposes of this part, use the default parameters, as in the example code. (Eg., delta = 0.2, x and y knots = 6, anomaly knots = length/4.) As deliverables, include in your report the same output as from the example code. That is, the:

- combined image used
- decomposed mean
- decomposed features – we are interested in heat generated under load, so set values <0 to 0!

C: 50%) The default parameters do quite well, but we can do better! Play with all available parameters to generate the best separation you can. Remember, your goal is to capture the load heat as sparse "anomaly" in the decomposition. In other words, you (probably) don't want the spots that are only hot under load to show up in the mean. For this part, include in your report your:

- combined image used (the "delta" used when combining the images is fair game)
- decomposed mean
- decomposed features – again, set values <0 to 0!
- a brief discussion of your methodology; say what you tried, what did (or didn't) work, and why you chose what you finally chose.