Assignment 4: CS 754, Advanced Image Processing

Due: 29th March before 11:55 pm

Remember the honor code while submitting this (and every other) assignment. All members of the group should work on and understand all parts of the assignment. We will adopt a zero-tolerance policy against any violation.

Submission instructions: You should ideally type out all the answers in Word (with the equation editor) or using Latex. In either case, prepare a pdf file. Create a single zip or rar file containing the report, code and sample outputs and name it as follows: A4-IdNumberOfFirstStudent-IdNumberOfSecondStudent.zip. (If you are doing the assignment alone, the name of the zip file is A4-IdNumber.zip). Upload the file on moodle BEFORE 11:55 pm on the due date. Late assignments will be assessed a penalty of 50% per day late. Note that only one student per group should upload their work on moodle. Please preserve a copy of all your work until the end of the semester. If you have difficulties, please do not hesitate to seek help from me.

1. In this task, you will implement the compressive KSVD algorithm that we did in class. The algorithm is based on the paper 'Compressive K-SVD' by Anaraki and Hughes. You should test your algorithm on the following two datasets and include the required results in your report.

   (a) Generate a dictionary $\boldsymbol{D}$ with $K = 20$ columns for signals of $p = 100$ elements. The dictionary entries should be drawn from $\mathcal{N}(0, 1)$, and the columns should subsequently be unit-normalized. Generate $N = 100$ signals where the i^{th} signal $\boldsymbol{x_i}$ is a random linear combination of any 5 dictionary columns. Different signals will have different supports, and the coefficients of the linear combination should be drawn from Uniform$[0, 10]$. Now for each $\boldsymbol{x_i}$, generate a compressive measurement vector $\boldsymbol{y_i} = \boldsymbol{\Phi_i x_i} + \boldsymbol{\eta_i}$ where each $\boldsymbol{\Phi_i}$ has size $m \times p$. Each entry of $\boldsymbol{\Phi_i}$ should be drawn independently (with 50% probability) from $\{-1/\sqrt{m}, +1/\sqrt{m}\}$. Also $\boldsymbol{\eta_i}$ represents noise from a zero-mean Gaussian distribution with standard deviation $\sigma = f \times \dfrac{1}{mN} \sum_{i=1}^{N} \|\boldsymbol{\Phi_i x_i}\|_1$ and f is a fractional value. For each $m \in \{10, 20, 30, 40, 50, 70, 90\}$ and each $f \in \{0.001, 0.01, 0.02, 0.05, 0.1, 0.3\}$, compute the average relative error for reconstruction of the N signals from their compressive measurements using the compressive KSVD algorithm. The average relative error is given as $\dfrac{1}{N} \sum_{i=1}^{N} \dfrac{\|\hat{\boldsymbol{x}}_{\boldsymbol{i}} - \boldsymbol{x_i}\|_2}{\|\boldsymbol{x_i}\|_2}$. Plot these error values as a chart on a scale from 0 to 1. Include a legend in the plot as well. State your observations or inferences from the chart. For the sparse coding part of the algorithm, you should run OMP either based on the Gaussian noise tail bound, or else with a fixed value of $T_0 = 5$ as mentioned in the paper, but you should be consistent (i.e. use the same method in all iterations of the algorithm).

   (b) In the second experiment, you should download the MNIST database from `http://yann.lecun.com/exdb/mnist/`. From the training file, take any 600 images per digit (each image has size 28×28 and downsample it using bilinear interpolation to size 16×16). For each such image $\boldsymbol{x_i}$, generate compressive measurements $\boldsymbol{y_i}$ as described in the previous part but with $f = 0.01$. Now learn a dictionary $\boldsymbol{D}$ with $K = 20$ columns from these compressive measurements. In your report, display the dictionary columns reshaped as images. Create a test set of noisy compressive measurements of any 10 images per digit, from the training set (again $f = 0.01$). Reconstruct the images in the test set from their compressive measurements using the OMP algorithm with the learned dictionary $\boldsymbol{D}$. Compare the performance to that of 2D-DCT for the OMP algorithm. For the comparison, report average relative errors (as defined earlier) for reconstructions using both algorithms. Display a sample reconstruction for each of the 10 digits for both dictionaries. Repeat the dictionary learning and reconstruction using $K = 128$. What do you observe?

(c) Continuing the previous part which involves digit images, create a test set of 20 images of each digit not from the training set. Report classification rates using a nearest neighbor search in the space of dictionary coefficients (i.e. the coefficient vector $\boldsymbol{\alpha_i}$ where $\boldsymbol{x_i} = \boldsymbol{D\alpha_i}$). Do this for $K = 80$ as well as $K = 128$ as well as for 2D-DCT. Note that classification rate = number of correctly classified test images / total number of test images. [20+20+10 = 50 points]

2. For our google search question, your job is to search for a recent paper which proposes a nice application of blind compressed sensing (i.e. where the dictionary is inferred from the compressive measurements directly). Very briefly summarize the application that the paper deals with. Write down the main objective function with the meaning of all terms clearly defined and state the optimization technique. Of course, you cannot choose the compressive KSVD paper here. You are also not allowed to use my paper on blind compressed sensing for CASSI from the SIAM journal, because I had briefly mentioned it in class. [15 points]

3. Consider a signal $\boldsymbol{f} = \boldsymbol{f_1} + \boldsymbol{f_2} + \boldsymbol{\eta}$ where $\boldsymbol{f_1}$ is a sparse linear combination of complex exponentials with integer frequencies (i.e. Discrete Fourier Bases) and $\boldsymbol{f_2}$ is a signal consisting of a small number of spikes. $\boldsymbol{\eta}$ represents noise from $\mathcal{N}(0, \sigma^2)$ where $\sigma = 0.05\times$ average value of $\boldsymbol{f_1} + \boldsymbol{f_2}$. Consider that $\boldsymbol{f}$ is a 1D discrete signal with 128 elements. Your job is to implement any technique of your choice to separate $\boldsymbol{f}$ into $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$. That is, you are given only $\boldsymbol{f}$ (which is noisy) and you want to estimate $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$. Experimentally study the quality of the estimation of both components (in terms of relative reconstruction error) with varying σ and varying sparsity level s for both $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$. (You may assume for simplicity that s is same for both.) Include all relevant plots in your report, and mention which technique you used. [35 points]