Assignment 4: CS 754, Advanced Image Processing

Due: 5th April 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. The cutoff is 10 am on 6th April after which no assignments will be accepted. 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. Consider a signal $\boldsymbol{f} = \boldsymbol{f_1} + \boldsymbol{f_2} + \boldsymbol{\eta}$ where $\boldsymbol{f_1}$ is a sparse linear combination of cosine waves with integer frequencies (i.e. sparse in DCT basis) 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 $\sigma = 0.01\times$ average value of $\boldsymbol{f_1} + \boldsymbol{f_2}$. Consider that $\boldsymbol{f}$ is a 1D discrete signal with 256 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 for both $\boldsymbol{f_1}$ and $\boldsymbol{f_2}$) with (a) varying σ and fixed s, and (b) varying sparsity level s with fixed σ. You may assume for simplicity that s is same for both. Include all relevant plots in your report, and mention which technique you used. Comment on these results.
   Now suppose that the magnitude of $\boldsymbol{f_2}$ was k times that of $\boldsymbol{f_1}$. Study the effect of varying k on the RMSE of both signals, on the same algorithm. Again, include the relevant plot and comments in your report. You may use any ready-made CS solver - examples are your own implementation of ISTA, OMP or solvers such as L1_LS, SPGL1, YALL1, L1-MAGIC (MATLAB codes for all are freely downloadable from the web). In all cases, state how you picked the relevant parameters for the solver or algorithm. [20 points]
   Solution: See code separation.m which uses the SPGL1 solver to run a quadratically constrained basis pursuit with $\varepsilon \triangleq \sigma\sqrt{n}$ where $n = 256$. The forward model is $\boldsymbol{y} = [\boldsymbol{U}|\boldsymbol{I}](\boldsymbol{\theta_1}|\boldsymbol{f_2})^t + \boldsymbol{\eta}$, where $\boldsymbol{f_1} = \boldsymbol{U\theta_1}$ where $\boldsymbol{U}$ is the DCT basis. If k is much larger than 1, we see that the RMSE of $\boldsymbol{f_1}$ worsens significantly. If k is much less than 1, then the RMSE of $\boldsymbol{f_2}$ worsens significantly.
   Marking scheme: 10 points for forward model correctly implemented. Plot of varying σ: 3 points, plot of varying s: 3 points. Both plots show an increase in RMSE for either signal. 2 points for stating the relevant parameters. 2 points for effect of k on RMSE.

2. We have studied two greedy algorithms for compressive recovery in class - MP and OMP. Your task is to do a google search and find out a research paper that proposes a greedy algorithm for CS recovery that is different from OMP and MP. Write down the algorithm in your report in the form of a simple pseudo-code. State the key theorem from the paper which presents performance bounds for the algorithm, and explain the meaning of the terms involved. If there are multiple theorems, pick the one that states the strongest result. [20 points]
   Solution: One example of such a greedy algorithm is Iterative Hard Thresholding. A good reference for this is `https://arxiv.org/pdf/0805.0510.pdf`, "Iterative Hard Thresholding for Compressed Sensing", by Thomas Blumensath and Mike E. Davies. The pseudo-code for this is summarized in a single equation -

Eqn. 11 of the aforementioned paper and the performance bounds are mentioned in Theorem 1.
Another greedy algorithm is CoSamp by Needell and Trop, from `https://arxiv.org/pdf/0803.2392.pdf`, 'CoSaMP: Iterative signal recovery from incomplete and inaccurate samples'. Algorithm 2.1 and Theorem A from the paper give the pseudo-code and performance bounds respectively.
Marking Scheme: 12 points for the pseudo-code and 8 points for the theorem. In all cases, meanings of all terms must be mentioned. Without meanings of terms clearly explained within the report, no points are to be awarded.

3. Consider that you learned a dictionary $\boldsymbol{D}$ to sparsely represent a certain class $\mathcal{S}$ of images - say handwritten alphabet or digit images. How will you convert $\boldsymbol{D}$ to another dictionary which will sparsely represent the following classes of images? Note that you are not allowed to learn the dictionary all over again, as it is time-consuming.

   (a) Class $\mathcal{S}_1$ which consists of images obtained by applying a known derivative filter to the images in $\mathcal{S}$.

   (b) Class $\mathcal{S}_2$ which consists of images obtained by rotating a subset of the images in class $\mathcal{S}$ by a known fixed angle α, and the other subset by another known fixed angle β.

   (c) Class $\mathcal{S}_3$ which consists of images obtained by applying an intensity transformation $I^i_{new}(x,y) = \alpha(I^i_{old}(x,y))^2 + \beta(I^i_{old}(x,y)) + \gamma$ to the images in $\mathcal{S}$, where α, β, γ are known.

   (d) Class $\mathcal{S}_4$ which consists of images obtained by applying a known blur kernel to the images in $\mathcal{S}$.

   (e) Class $\mathcal{S}_5$ which consists of images obtained by applying a blur kernel which is known to be a linear combination of blur kernels belonging to a known set $\mathcal{B}$, to the images in $\mathcal{S}$. [4+4+4+4+4=20 points]

   Solution:
   Part a: create a new dictionary $\boldsymbol{D_2}$ whose every column is obtained as follows: Reshape each column of $\boldsymbol{D}$ to form an image of the appropriate dimensions and convolve it with the derivative filter. Then reshape the resulting image to a vector in order to yield the column of $\boldsymbol{D_2}$.
   Part b: $\boldsymbol{D_2} = (\boldsymbol{D_{2\alpha}}|\boldsymbol{D_{2\beta}})$, where '—' indicates a column-wise concatenation, and where the columns of $\boldsymbol{D_{2\alpha}}$ are obtained by rotating the image-reshaped columns of $\boldsymbol{D}$ by α, followed by vectorization. You must take care of zero-padding since the image domain size may now increase. Another possible answer for this part is to pre-multiply the columns of D by a permutation matrix designed based on α or β. If you choose this route, you have to be careful to include appropriate interpolation in the design of the permutation matrix.
   Part c: If $\boldsymbol{x} = \boldsymbol{D\theta} = \sum_k \boldsymbol{d_k}\theta_k$, then $\boldsymbol{x}.^2 = \sum_{k,l} \boldsymbol{d_k} \cdot \boldsymbol{d_l}\theta_k\theta_l$. With this in mind, the new dictionary is $(\boldsymbol{D_2}|\boldsymbol{D}|\boldsymbol{1})$ where $\boldsymbol{D_2}$ is a dictionary containing K^2 columns (if $\boldsymbol{D}$ has K columns) containing pointwise multiplication of the columns of $\boldsymbol{D}$.
   Part d: This is same as the solution in part a, except that the blur kernel is used instead of the derivative kernel from part a.
   Part e: Let $L = |\mathcal{B}|$. Given dictionary $\boldsymbol{D}$, create a dictionary $\boldsymbol{D^l}$ by applying the operations in part d, with the l^{th} blur kernel from $\mathcal{B}$. The final dictionary $\boldsymbol{D_{final}}$ is obtained by a column-wise concatenation of all the dictionaries $\{\boldsymbol{D^l}\}_{l=1}^L$. Note that the coefficients of the linear combination are not important for creating $\boldsymbol{D^l}$ or $\boldsymbol{D_{final}}$. To see this, consider signal $\boldsymbol{g} = \sum_l \boldsymbol{d_l}\theta_l = \boldsymbol{D\theta}$. Let the L blur kernels in $\mathcal{B}$ be denoted as $\{h_k\}_{k=1}^L$. Then $sum_k\alpha_k h_k * g = sum_k\alpha_k h_k * \sum_l \boldsymbol{d_l}\theta_l = \sum_{k,l}(h_k * \boldsymbol{d_l})\alpha_k\theta_l$.

4. How will you solve for the minimum of the following objective functions: (1) $J(\boldsymbol{A_r}) = \|\boldsymbol{A} - \boldsymbol{A_r}\|_F^2$, where $\boldsymbol{A}$ is a known $m \times n$ matrix of rank greater than r, and $\boldsymbol{A_r}$ is a rank-r matrix, where $r < m, r < n$. (2) $J(\boldsymbol{R}) = \|\boldsymbol{A} - \boldsymbol{RB}\|_F^2$, where $\boldsymbol{A} \in \mathbb{R}^{n\times m}, \boldsymbol{B} \in \mathbb{R}^{n\times m}, \boldsymbol{R} \in \mathbb{R}^{n\times n}, m > n$ and $\boldsymbol{R}$ is constrained to be orthonormal. Note that $\boldsymbol{A}$ and $\boldsymbol{B}$ are both known.
   In both cases, explain briefly any one situation in image processing where the solution to such an optimization problem is required. [5+5+5+5=20 points]
   Solution: First part: by Eckart-Young theorem, you need to compute the SVD of $\boldsymbol{A} = \boldsymbol{USV}^T$. Then consider the r largest singular values and their corresponding singular vectors from $\boldsymbol{U}$ and $\boldsymbol{V}$ - respectively named as $\boldsymbol{U_r}, \boldsymbol{V_r}$. Then $\boldsymbol{A_r} = \boldsymbol{U_r S_r}(\boldsymbol{V_r})^T$.
   Application of first part: A rank-one approximation to the error matrix is computed in KSVD - see slides

110 and 111 of the lecture slides on Dictionary Learning. In PCA, we consider a rank-k approximation to the covariance matrix effectively. In orthographic structure from motion, a rank 3 approximation to the matrix of data-points across the frames is considered. (any one application is enough, but some more details of the matrix and its dimensions need to be mentioned in the answer.)

Second part: This is the orthogonal procrustes problem. The solution is obtained from the SVD of $\boldsymbol{BA}^T$. If $\boldsymbol{BA}^T = \boldsymbol{USV}^T$, then $\boldsymbol{R} = \boldsymbol{VU}^T$.
Application of second part: If $n = 3$, this is useful in determining a rotation/reflection transformation between two sets of points (assuming $m > 3$). We saw this in tomography (where exactly?). Another application is in the dictionary learning algorithm called union of orthonormal bases (any one application is enough, but some more details of the matrix and its dimensions need to be mentioned in the answer.)

5. Read the paper 'A Signal Processing Perspective on Hyperspectral Unmixing', a copy of which is placed in the homework folder. Answer the following questions:

   (a) What is hyperspectral unmixing? You may use an equation to support your answer with symbol meanings carefully explained.

   (b) In equation 40 of the paper, explain how non-negative matrix factorization is used for hyperspectral unmixing.

   (c) Explain the improvement to non-negative matrix factorization proposed in equation 41 of the paper. (You may explain any two forms each for g and h.) [6+6+8=20 points]

Solution:
Part a: Consider a hyperspectral datacube of size $N_1 \times N_2 \times M$ where M is the number of spectral bands. The vector $\boldsymbol{y_i}$ of M spectral responses at any location i is given as $\boldsymbol{y_i} = \boldsymbol{As_i}$ where $\boldsymbol{A}$ is a $M \times N$ matrix of N different 'endmembers' (one endmember per column) and $\boldsymbol{s_i}$ is the $N \times 1$ non-negative vector of endmember proportions all adding up to one. Essentially, one pixel of a hyperspectral datacube corresponds to a fairly large area of the scene being imaged. This is especially true in remote-sensing applications, where hyperspectral imaging is very common. In such a case, a pixel represents a mixture of different materials from the different locations in the area being imaged. The spectral responses of these different materials are nothing but the endmembers and their mixing coefficients are always non-negative and add up to 1. The task of obtaining $\boldsymbol{s_i}$ from $\boldsymbol{y_i}$ at each location i is called hyperspectral unmixing. If $\boldsymbol{A}$ is unknown, it is called blind hyperspectral unmixing.
Part b: In equation 41 of the paper, the matrix $\boldsymbol{Y}$ of size $M \times N_1N_2$ represents the spectral signatures at each pixel. Then we have $\boldsymbol{Y} = \boldsymbol{AS}$ where $\boldsymbol{S}$ is a matrix of size $N \times N_1N_2$ and it contains the endmember proportions at each of the pixels. As argued earlier, $\boldsymbol{S}$ is non-negative. Also, $\boldsymbol{A}$ contains all non-negative entries since these are spectral responses. The blind hyperspectral unmixing problem is therefore a natural case for non-negative matrix factorization, except that NMF does not impose the sum-to-one constraint in the columns of $\boldsymbol{S}$.
Part c: The different choices are presented in Table 1. Any two can be chosen and explained briefly. For example in the DL algorithm, there is an ℓ_1 penalty on the rows and columns of $\boldsymbol{S}$ since a given material may not contribute to too many pixels, and a single pixel will not receive contributions from too many materials. There is no constraint on $\boldsymbol{A}$ in this algorithm. The APS algorithm again has no constraint on $\boldsymbol{A}$ but forces the mixing proportions at neighboring locations to be similar to each other.
Note: All symbols meaning need to be explained carefully to get credit. Generous patial credit to be given for partially correct answers even if full explanation is absent.