

Assignment 3: CS 754, Advanced Image Processing

Due: 22nd 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: A3-IdNumberOfFirstStudent-IdNumberOfSecondStudent.zip. (If you are doing the assignment alone, the name of the zip file is A3-IdNumber.zip). Upload the file on moodle BEFORE 11:55 pm on 22nd March. No assignments will be accepted after a cutoff deadline of 10 am on 23rd March. 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. Download the book ‘Statistical Learning with Sparsity: The Lasso and Generalizations’ from https://web.stanford.edu/~hastie/StatLearnSparsity_files/SLS_corrected_1.4.16.pdf, which is the website of one of the authors. (The book is officially available free of cost). In chapter 11, there are theorems which show error bounds on the minimum of the following objective function: $J(\beta) = \frac{1}{2N} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_N \|\beta\|_1$ where λ_N is a regularization parameter, $\beta \in \mathbb{R}^p$ is the unknown sparse or compressible signal, $\mathbf{y} = \mathbf{X}\beta + \mathbf{w}$ is a measurement vector with N values, \mathbf{w} is a zero-mean i.i.d. Gaussian noise vector whose each element has standard deviation σ and $\mathbf{X} \in \mathbb{R}^{N \times p}$ is a sensing matrix whose every column is unit normalized. This particular estimator (i.e. minimizer of $J(\mathbf{x})$ for \mathbf{x}) is called the LASSO in the statistics literature. The theorems derive a statistical bound on λ also. The main result is theorem 11.1, parts (a) and (b), and its extension in equation (11.15). These are for sparse signals. For weakly sparse or compressible signals, the result is given in equation 11.16.

Your task is to answer the following questions:

- (a) Give a careful comparison of the bounds in equations 11.15 and 11.16 to Theorem 3 done in class, and derived in the previous assignment. For this, state how the error bounds vary w.r.t. the number of measurements, the signal sparsity, the noise standard deviation and the signal dimension. Comparing equations 11.15 and 11.16 on one hand to Theorem 3, which of these do you think provides a more intuitive result? Explain why. [2 + 2 + 2 + 2 + 2 = 10 points]
 - (b) Define the restricted eigenvalue condition (the answer’s there in the book and you are allowed to read it, but you also need to understand it). [3 points]
 - (c) Starting from equation 11.20 on page 309 - explain why $G(\hat{v}) \leq G(0)$. [3 points]
 - (d) Derive equation 11.21 starting from equation 11.20. [3 points]
 - (e) We will now prove theorem 11.2. For this, provide justification for the five equations in the proof of theorem 11.2, parts (a) and (b). [5 points]
2. In class, we studied a video compressive sensing architecture from the paper ‘Video from a single exposure coded snapshot’ published in ICCV 2011 (See http://www.cs.columbia.edu/CAVE/projects/single_shot_video/). Such a video camera acquires a ‘coded snapshot’ E_u in a single exposure time interval u . This coded snapshot is the superposition of the form $E_u = \sum_{t=1}^T C_t \cdot F_t$ where F_t is the image of the scene at instant t within the interval u and C_t is a randomly generated binary code at that time instant, which

modulates F_t . Note that E_u , F_t and C_t are all 2D arrays. Also, the binary code generation as well as the final summation all occur within the hardware of the camera. Your task here is as follows:

- (a) Read the ‘cars’ video in the homework folder in MATLAB using the ‘mmread’ function which has been provided in the homework folder and convert it to grayscale. Extract the first $T = 3$ frames of the video. You may use the following code snippet:
`A = mmread('cars.avi'); T = 3; for i=1:T, X(:,:,i) = double(rgb2gray(A.frames(i).cdata)); end; [H,W,T] = size(X);`
 - (b) Generate a $H \times W \times T$ random code pattern whose elements lie in $\{0, 1\}$. Compute a coded snapshot using the formula mentioned and add zero mean Gaussian random noise of standard deviation 2 to it. Display the coded snapshot in your report.
 - (c) Given the coded snapshot and assuming full knowledge of C_t for all t from 1 to T , your task is to estimate the original video sequence F_t . For this you should rewrite the aforementioned equation in the form $\mathbf{Ax} = \mathbf{b}$ where \mathbf{x} is an unknown vector (vectorized form of the video sequence). Mention clearly what \mathbf{A} and \mathbf{b} are, in your report.
 - (d) You should perform the reconstruction using either the ISTA algorithm or the OMP algorithm (the original paper used OMP). You can re-use your own code from a previous assignment. For computational efficiency, we will do this reconstruction patchwise. Write an equation of the form $\mathbf{Ax} = \mathbf{b}$ where \mathbf{x} represents the i th patch from the video and having size (say) $8 \times 8 \times T$ and mention in your report what \mathbf{A} and \mathbf{b} stand for. For perform the reconstruction, assume that each 8×8 slice in the patch is sparse or compressible in the 2D-DCT basis.
 - (e) Repeat the reconstruction for all overlapping patches and average across the overlapping pixels to yield the final reconstruction. Display the reconstruction and mention the relative mean squared error between reconstructed and original data, in your report as well as in the code.
 - (f) Repeat this exercise for $T = 5, T = 7$ and mention the relative mean squared error between reconstructed and original data again.
 - (g) **Note: To save time, extract a portion of about 120×240 around the lowermost car in the cars video and work entirely with it. In fact, you can show all your results just on this part. Some sample results are included in the homework folder.**
 - (h) Repeat the experiment with any consecutive 5 frames of the ‘flame’ video from the homework folder. [20 points = 12 points for correct implementation + 4 points for correct expressions for A, b ; 4 points for display results correctly.]
3. Consider the image ‘cryoem.png’ in the homework folder. It is a 2D slice of a 3D macromolecule in the well-known EMDB database. Generate N Radon projections of this 2D image at angles drawn uniformly at random from 0 to 360 degrees. Your job is to implement the Laplacian eigenmaps based algorithm for 2D tomographic reconstruction from unknown angles, given these projection vectors. You should test your algorithm for $N \in \{50, 100, 500, 1000, 2000, 5000, 10000\}$. In each case, display the reconstructed image, and compute its RMSE appropriately normalized for rotation. (Note that cryo-EM reconstructions are valid only up to a global rotation, so direct RMSE computation without compensating for the arbitrary rotation is meaningless). You can use the routines `radon`, `imrotate` in MATLAB. [30 points]
 4. Let $R_\theta f(\rho)$ be the Radon transform of the image $f(x, y)$ in the direction given by θ for bin index ρ . Let g be a version of f shifted by (x_0, y_0) . Then, prove that $R_\theta g(\rho) = R_\theta f(\rho - (x_0, y_0) \cdot (\cos \theta, \sin \theta))$. [8 points]
 5. Consider two observed particle images Q_1 and Q_2 corresponding to a 3D density map, each in different 3D orientations and 2D shifts. Let Q_1 be obtained by translating a zero-shift particle image P_1 by $(\delta_{x1}, \delta_{y1})$. Let Q_2 be obtained by translating a zero-shift particle image P_2 by $(\delta_{x2}, \delta_{y2})$. Note that Q_1, Q_2 are practically observed, whereas P_1, P_2 are not observed. Let the common line for the particle images P_1, P_2 pass through the origins of their respective coordinate systems at angles θ_1 and θ_2 with respect to their respective X axes. Derive a relationship between $\delta_{x1}, \delta_{y1}, \theta_1, \delta_{x2}, \delta_{y2}, \theta_2$ and some other observable property of the projection images. Explain how you will determine $\delta_{x1}, \delta_{y1}, \delta_{x2}, \delta_{y2}$ using this equation. Explain how you will extend

this relationship to determine the shifts $\{(\delta_{xi}, \delta_{yi})\}_{i=1}^N$ of the N different projection images, and mention the number of knowns and unknowns. [5+2+8+3=18 points]