AMATH 582: HOME WORK 5

RAPHAEL LIU

Department of Appliede Mathematics, University of Washington, Seattle, WA
raph651@uw.edu

Abstract. In this assignment, we are going to recover an image from limited observations of its pixels. The image used is a portion of René Magritte's "The Son of Man" along with its corrupted variant. A mysterious image is also given (a measurement matrix and a measurement outcome vector), and our task is to reproduce the original version. This is accomplished through Discrete Cosine Transformation and Convex Optimization.

1. Introduction and Overview

The original image of René Magritte's "The Son of Man" is an image with 292×228 pixels. We rescale it to 53×41 pixels to make the computation manageable. Otherwise, the optimization step can take a very long time. We specifically have three parts of tasks: Image Compression, Compressed Image Recovery, and Mysterious Image Recovery. Below is a list of detailed tasks.

(1) **(Image Compression)** Our first goal is to examine the compressibility of the discrete cosine transform (DCT) of "SonOfMan.png". Calculate and plot the vector form of DCT(F). Discuss the behavior of the transformation. Reconstruct and plot the image after thresholding its DCT coefficients to keep the top 5,10,20, and 40 percent.

(2) **(Compressed Image Recovery)** The goal in this step is to recover the image F from limited random observations of its pixels using the prior knowledge that the DCT(F) is nearly sparse. Randomly select M= 20%, 40%, and 60% of the (53×41) pixels. For each choice of M, obtain an approximation F^* to F and repeat the procedures 3 times by redrawing the random selection. Present a plot of these images and discuss the results

(3) (**A Mysterious Image**) Given measurement matrix B and outcome vector $\mathbf{y}$ for an unknown image of size 50×50 pixels. Reconstruct and visualize the unknown image.

2. Theoretical Background

2.1. Discrete Cosine Transform (DCT).

Given a discrete signal $\mathbf{f} \in \mathbb{R}^K$ we define its DCT as DCT($\mathbf{f}$)$\in \mathbb{R}^K$ where

$$\text{DCT}(\mathbf{f})_k = \sqrt{\frac{1}{K}} \left[f_0 cos(\frac{\pi k}{2K}) + \sqrt{2} \sum_{j=1}^{K-1} f_j cos(\frac{\pi k(2j+1)}{2K}) \right].$$

This transform is analogous to taking the real part of the FFT of $\mathbf{f}$, i.e., writing the signal as a sum of cosines (other definations of DCT exist in the literature). The inverse DCT (iDCT) transform

Date: March 18 2022.

reconstructs the signal $\mathbf{f}$ from DCT($\mathbf{f}$). The 2D DCT (resp. iDCT) is defined analogously to 2D FFT (resp. iFFT) by successively applying the 1D DCT (resp. iDCT) to the rows and columns of a 2D image.

2.2. Signal Recovery.

Consider a signal(or image) $f^+ : [0,1] \to \mathbb{R}$ along with some measurements of the signal $\mathbf{y} \in \mathbb{R}^M$. A simple example is

$$\mathbf{y}^+ = (y_0^+, .., y_{N-1}^+)$$

where $\mathbf{y}_j^+ = f^+(t_j)$ for some points $t_j \in [0,1]$. Of course we can simply solve this problem using for ex, polynomial interpolation. But we want to consider a harder problem specifically one where the measurements $\mathbf{y}^+ \in \mathbb{R}^M$ are of the the form:

$$y_j^+ = \xi_j^T f^+$$

where $\xi_j \sim \mathcal{N}(0, I),\ f^+ = (f^+(t_0), f^+(t_1), .., f^+(t_{N-1}))$.

Goal: reconstruct f^+ given $\mathbf{y}$ & $M < N$ Our first step is to assume a model for $\hat{f}$

$$f(t) = \sum_{m,n} \beta_{m,n} \Psi_{m,n}(t)$$

where $\Psi_{m,n}$ are the Haar wavelet basis. Reparametraize the wavelet coefficients into a 1D array, $f(t) = \sum_{j=0}^{J-1} \beta_j \psi_j(t)$ such that $\mathbf{f} = W\beta$. Finally for such an $\mathbf{f}$ we have the measurements

$$\mathbf{y} = S\mathbf{f},\ S = \begin{bmatrix} \xi_0^T \\ \xi_1^T \\ .. \\ \xi_{M-1}^T \end{bmatrix} \in \mathbb{R}^{M \times N}$$

In other words,

$$\mathbf{y} = A\beta$$

where $A = SW$. Then this is nothing but a supervised learning/regression problem- find $\hat{\beta}\ s.t.\ A\hat{\beta} \approx \mathbf{y}^+$. Typically, there is an important distinction $M < N$. In this assignment, our task is equivalent to solving the optimization problem

$$\text{minimize}_{\mathbf{x} \in \mathbb{R}^N}\ ||\mathbf{x}||_1,$$
$$\text{Subject to } A\mathbf{x} = \mathbf{y},$$

and denote the minimizer as $\mathbf{x}^*$. Clearly, this vector $\mathbf{x}^*$ is the DCT vector of an image F^* that, hopefully, resembles the original image F.

3. Algorithm Implementation and Development

The programming language used here is Python 3. The procedure for solving the tasks are given below, separately.

(1) **Image Compression:** First load the original "The Son of Man" image and rescale it to a 53×41 pixels image. Let N_y, N_x denote the number of rows and columns of the image with $N = N_y \times N_x$ denoting the number of pixels in the image. Let $F \in \mathbb{R}^{N_y \times N_x}$ denote the image and write $\text{vec}(F) \in \mathbb{R}^N$ for its vectorization. Use scipy's DCT package to construct matrices D, D^{-1} such that $D\text{vec}(F) = \text{vec}(\text{DCT}(F))$ and $D^{-1}\text{vec}(\text{DCT}(F)) = \text{vec}(F)$

(2) Plot DCT(F) and investigate its compressibility. See how many large coefficients are there.

(3) Iterating over k =5, 10, 20, and 40, sort the DCT coefficients by decreasing absolute value and store the index of sorted coefficients. Keep the first k% of the sorted DCT coefficients, set the remaining to 0, and reconstruct F_{thresh} with matrix multiplication of D^{-1}. Visualize the images.

(4) **Compressed Image Recovery:** Define an array of number of random pixels $M = [0.2, 0.4, 0.6] * N$. Construct an $N \times N$ identity matrix. Iterate over M, randomly select M number of rows in the identity matrix, and denote it as measurement matrix $B \in \mathbb{R}^{M \times N}$. Repeat this procedure 3 times to produce 9 different matrix B.

(5) Define operational matrix $A = BD^{-1}$. Now use cvx solver from cvxpy package. Set variable $\mathbf{x}$. Set objective to minimize the L_1 norm of x with constraint: $-$tolerance $\leq A\mathbf{x} - \mathbf{y} \leq$ tolerance, where tolerance is set to be 10^{-2}. The recovered image is obtained by matrix multiplication of $D^{-1}\mathbf{x}$.

(6) **A Mysterious Image:** Load the vector $\mathbf{y}$ and matrix B provided. The image will be of 50×50 pixels. Set N_x, N_y to be 50, and $N = N_x N_y$. Reconstruct D^{-1} and operational matrix $A = BD^{-1}$. Use cvx optimizer to solve for sparse DCT coefficients $\mathbf{x}$. Lastly, reconstruct the image as before and plot the image. Discuss the finding.

4. Computational Results

The image compression is performed by calculating the DFT coefficients. The resulting coefficients plot are as shown.1 We see that there are not a lot large coefficients. The first coefficient is about magnitude 26, dominating the others. A few coefficients are about magnitude 3 to 4, while the rest is close to 0. Keeping the top 5%, 10%, 20%, and 40% DCT coefficients and reconstruct the image.2 We see that these portions already contribute the majority and store much information about the image.

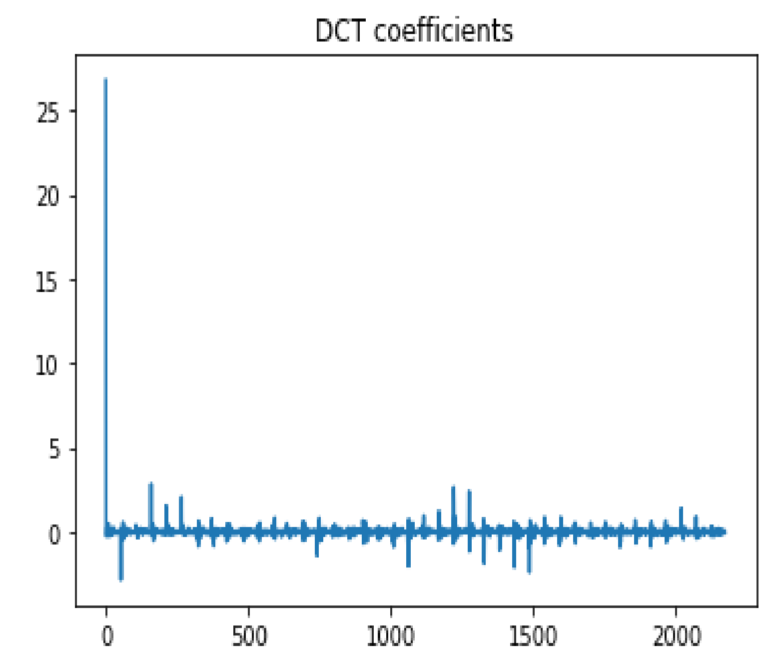

Figure 1. Plot of DCT coefficients

Now for the compressed image recovery part, we randomly select 20%, 40%, 60% pixels for measurement. Repeat the calculations three times by reselecting random pixels. This leads to a total of 9 images that approximate the original image F. The relevant results are shown below.3

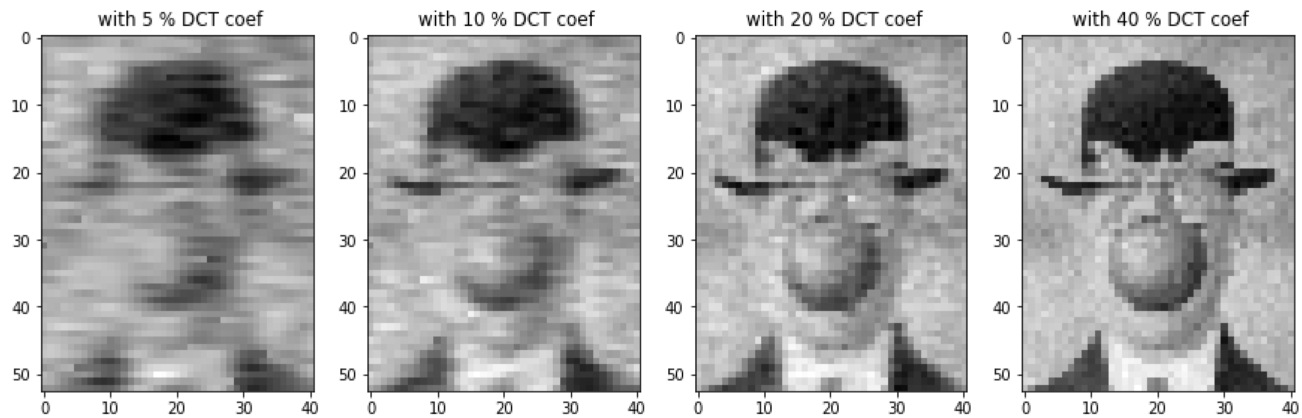

FIGURE 2. Image compression with portions of DCT coefficients.

The final task is done by first constructing a new iDCT matrix D^{-1} and operational matrix A. Solve the L_1 norm optimization problem and reconstruct the 50×50 pixels image. The recovered image is as shown.4

5. Summary and Conclusions

In order to perform Image Compression and Compressed Image Recovery, there are few steps most importantly to obtain the results. These steps include construct DCT and iDCT matrices, perform DCT and iDCT, analyze coefficients, define measurement matrix, build and solve convex optimization model, and reconstruct images.

Acknowledgements

The author is thankful to Prof.Hosseini for detailed discussion and his helper notebook for Sparse Signal and Image Recovery.

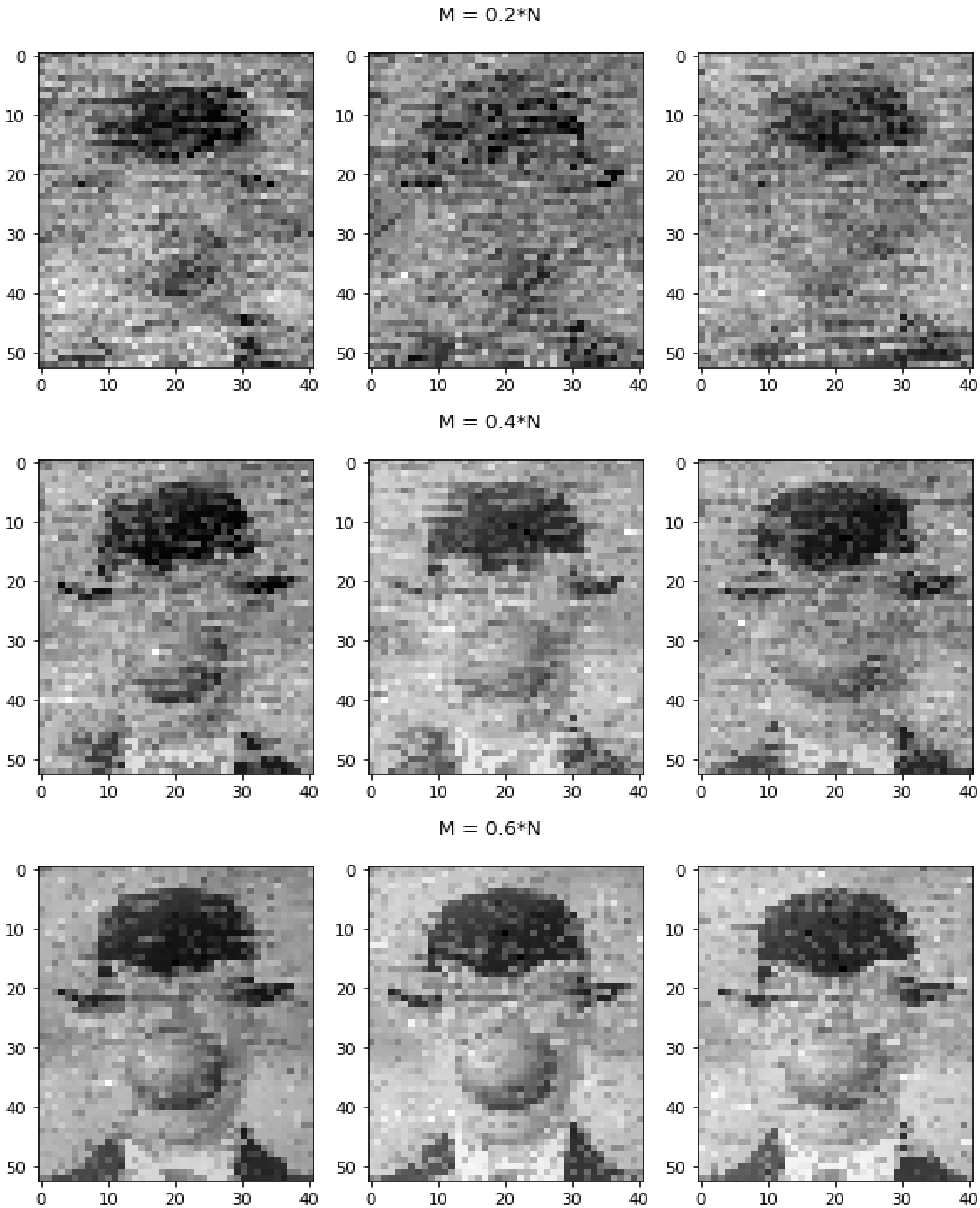

FIGURE 3. Compressed image recovery with different M.

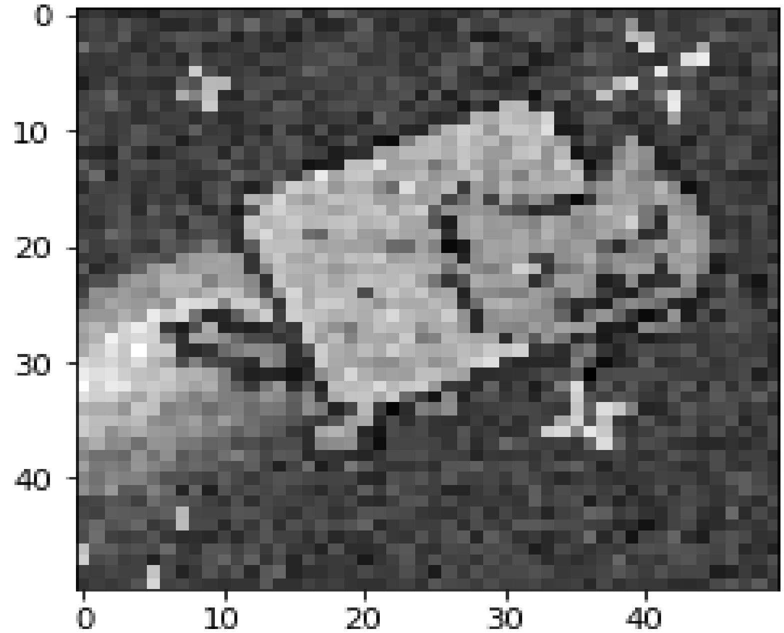

FIGURE 4. Recovered mysterious image