AMATH 482/582 HOMEWORK 5 - COMPRESSED IMAGE RECOVERY

PHILIP WESTPHAL

Department of Applied Mathematics, University of Washington, Seattle, WA
philw16@uw.edu

Abstract. This report explores the signal processing techniques of compression and sparse recovery using image data. A resized version of the Son of Man image was compressed using the Discrete Cosine Transform (DCT) with varying coefficients kept and the image quality explored. Sparse recovery of the resized image and the resulting quality was also explored with varying numbers of observed pixels, and the algorithms were tested on an unknown image.

1. Introduction and Overview

The goal of this problem was to explore compression and sparse recovery techniques using a resized version of the Son of Man painting. For the experiments in this problem, a large scale color version of the Son of Man painting was converted to a 53x41 pixel grayscale version. Working with the resized image is mainly to ease the computational burden of the sparse recovery problem, which can be large even with small images.

Both signal compression using the Discrete Cosine Transform (DCT) and sparse signal recovery are incredibly powerful tools in signal processing. Compression of a signal allows working with a smaller, yet representative dataset and enables accurate models to be generated with a much smaller data footprint. This also has the benefit of removing noise that may exist in the signal. Sparse recovery can generate a representation of a signal given only a small number of observations, which is often the case in real world scenarios. Due to the applicability of both compression and sparse recovery to every day problems, they are extremely important tools to know for data analysis.

2. Theoretical Background

2.1. Discrete Cosine Transform

In signal processing, the Discrete Cosine Transform (DCT) and its inverse transform can be used to convert signals in and out of the frequency domain. The DCT is equivalent to the real portion of the Fast Fourier Transform (FFT), and can be used for signals with no imaginary components. Within the frequency domain, certain signatures can be attenuated to remove noise or reduce the size of the signal. In one dimension the DCT of a signal [1] $f \in \mathbb{R}^K$ can be calculated as

$$F_k = \sqrt{\frac{2}{K}} C_k \sum_{j=0}^{K-1} \cos\left(\frac{\pi k(2j+1)}{2K}\right) f_j,\ C_j = \begin{cases} \frac{1}{\sqrt{2}} & k = 0 \\ 1 & \text{else} \end{cases} \tag{1}$$

and the inverse DCT as

$$iDCT(f) = \sqrt{\frac{2}{K}} \sum_{j=0}^{K-1} C_k \cos\left(\frac{\pi k(2j+1)}{2K}\right) F_k,\ C_j = \begin{cases} \frac{1}{\sqrt{2}} & k = 0 \\ 1 & \text{else} \end{cases} \tag{2}$$

For an image $X \in \mathbb{R}^{m,n}$ pixels, the two dimensional DCT and iDCT can be obtained by successively applying the one dimensional functions to the rows and columns of the data [2]. Mathematically, this is represented as [3]

$$DCT_{m,n} = DCT_m X DCT_n^T \tag{3}$$

Date: March 17, 2022.

Using the flattened image $X_{flat} \in \mathbb{R}^{m \times n}$, the DCT can be represented with the Kronecker matrix-vector product [3]

$$(DCT_n \otimes DCT_m)X_{flat} \tag{4}$$

Similar relations hold for the inverse DCT operations on both the unflattened and flattened images.

2.2. **Sparsity & Lasso Regression**

As defined in [4], the Ridge Regression problem is defined as

$$\beta_{MLE} = \underset{\beta}{\operatorname{argmin}} \frac{1}{2\sigma^2} \left\| A\beta - y \right\|_2^2 + \frac{\lambda}{2} \|\beta\|^2 \tag{5}$$

This is a least squares regression problem with an $L2$ norm penalty on the coefficient vector β. The advantage of the ridge regression problem is that it is fully differentiable, and thus a unique solution for β can often be obtained. However in underdetermined and sparse problems, ridge regression often cannot provide the optimal solution [5]. Consider the following problem, where the goal is to find an optimal vector β for

$$A\beta = y, \ \beta \in \mathbb{R}^N, y \in \mathbb{R}^M, A \in \mathbb{R}^{MxN}, M < N \tag{6}$$

The problem is underdetermined as the number of rows M is less than the number of columns N, and a unique solution cannot be obtained [5]. However, if β is S-sparse and only has $S << M$ non-zero entries, solutions can be obtained for specific matrices A. This is the fundamental logic behind the lasso regression problem, which is similar to the ridge regression problem but uses an $L1$ norm penalty on β instead of an $L2$ norm

$$\beta_{MLE} = \underset{\beta}{\operatorname{argmin}} \frac{1}{2\sigma^2} \left\| A\beta - y \right\|_2^2 + \lambda \|\beta\|_1 \tag{7}$$

The question can be asked as to why lasso regression models support sparse solutions better than ridge regression. The lasso regression problem can be equivalently rewritten as

$$\begin{aligned} \min_{\beta} \quad & \frac{1}{2\sigma^2} \|A\beta - y\|_2^2 \\ s.t. \quad & \|\beta\|_1 \leq t \end{aligned} \tag{8}$$

where t is a model parameter [6, pp. 79–91]. Similarly, the ridge regression problem could be written in a similar format using the $L2$ norm in the constraint. Figure 1 visualizes the constraints geometrically for a two dimensional problem. The ellipses represent contours of β centered at $\hat{\beta}$ that solve the optimization, and the gray areas represent values of β that satisfy the constraint. Due to the sharp edges on the $L1$ norm constraint range, it is more likely that the β countours will intersect with the sparse solutions on the axes [6, pp. 79–91]. As the $L1$ norm isn't differentiable at 0, the lasso regression is often solved using a convex

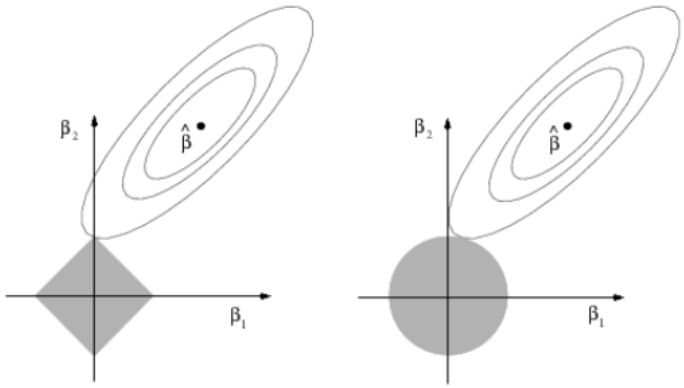

FIGURE 1. Optimal β for $L1$ and $L2$ norm penalties

optimization problem of the following format

$$\begin{aligned} \min_{\beta} \quad & \|\beta\|_1 \\ s.t. \quad & A\beta = y \end{aligned} \tag{9}$$

In practice, a value that exactly satisfies the problem may not be able to be obtained or may be computationally unfeasible. Typically, solvers modify the constraint as follows to stop when a solution within a certain tolerance has been reached

$$|A\beta - y| \leq tol$$

2.3. **Sparse Recovery**

Consider a signal $f : [0, 1] \to \mathbb{R}$ with a corresponding set of measurements $y \in \mathbb{R}^M$. It can be assumed that y is of the following form, where the vector c represent weights applied to all points of the signal f

$$y = c^T f \tag{10}$$

Specifically for a two dimensional image, the following structure can be assumed for f with ψ as a basis function and $x \in \mathbb{R}^N$ as the flattened image of $n_x \times n_y$ pixels

$$f(x) = \sum_{n=0}^{N-1} \beta_n \psi_n(x), \; N = n_x \times n_y \tag{11}$$

The M points in y represent specific pixels being sampled from the flattened image. Thus, a matrix Ψ can be obtained by evaluating each of the M samples points with all basis functions ψ_n [7]. The summation is then reduced to a matrix multiplication problem

$$f(x) = \Psi\beta, \; \Psi \in \mathbb{R}^{M,N} = \begin{bmatrix} \psi_0(m_0) & \dots & \psi_{N-1}(m_0) \\ \vdots & \ddots & \vdots \\ \psi_0(x_{M-1}) & \dots & \psi_{N-1}(x_{M-1}) \end{bmatrix} \tag{12}$$

Considering matrix $C \in \mathbb{R}^{M,N}$ representing all the one dimensional weight vectors c applied to Ψ, the signal recovery problem is reduced to finding the vector β that solves

$$y = A\beta, \; A = C\Psi \tag{13}$$

As the number of points N in the flattened image is greater than the number of sampled points M, this is an under determined optimization problem with a sparse solution [7]. As such, a lasso regression model can be used to recover the image. Specifically for the image recovery in this report, the problem is to find β such that

$$y = A\beta, \; A = CD^{-1} \tag{14}$$

where $D^{-1} \in \mathbb{R}^{N,N}$ is the inverse DCT transform of the flattened image, and matrix $C \in \mathbb{R}^{M,N}$ is the coefficient matrix generated by sampling random rows from an identity matrix $I \in \mathbb{R}^{N,N}$ [2]. The vector y is obtained as CX_{flat} to simulate sampling M observations from the full image. This problem formulation is equivalent to equation 9 and the solution β represents the DCT of a recovered X_{flat}.

3. Algorithm Implementation and Development

To explore the impact of image compression, the DCT of the flattened Son of Man image was taken and certain frequencies below four different thresholds filtered out. The inverse DCT was then taken such that the compressed image could be plotted in the problem space. This process is described in algorithm 1

To recover the Son of Man image from a sparse sampling of M pixels, a convex optimization problem was setup as described in section 2.3. The sparse recovery problem using the python `cvxpy` convex optimization package with the ECOS solver [8]. The CVXOPT solver was also tried, however feasability issues were experienced with the solver. Due to the randomness of generating the C matrix, three runs were done for each value of M rows to ensure a good result could be obtained. Algorithm 2 details the function used to solve the convex optimization and recover the image 9 using M observed pixels. This workflow was also used to recoveran unknown image that was provided.

Algorithm 1 Image Compression

```
% of coefficients in top [5%, 10%, 20%, 40%]      ▷ Frequency thresholds to keep in DCT
image DCT = DCT(flattened image)
for keep value in % of coefficients do
    threshold = percentile(image DCT, 1 - keep value)      ▷ Percentile threshold
    results = if image DCT ≥ threshold then keep coefficient else 0      ▷ Keep applicable coefficients
    compressed image = reshape inverse DCT(results) to 2D
end for
```

Algorithm 2 Image Compression

```
function recoverImage(M, flattened image, inverse DCT Matrix)
    N = length of flattened image
    C = sample M rows from N x N identity matrix
    A = C × inverse DCT matrix
    y = C × flattened image
    β = solve equation 9
    recovered flattened image = inverse DCT matrix × β
    return recovered flattened image
end function
```

4. Computational Results

Figure 3 shows the power of the DCT frequencies of the flattened Son of Man image. It can be seen that there are only a few spikes to frequencies with power greater than one or two, and that most are of an extremely low value. Figure 1 shows the original image as well as compressed images keeping the top $5\%, 10\%, 20\%$, and 40% of the DCT coefficients. While the compressed resized image is distorted and pixelated with the top 5% of the frequencies, a clear image is obtained when keeping the top 20% of the coefficients. While there is still some pixelation and some noticeable color differences in the background and hat, the quality of the image is still well preserved. Using 40% of the coefficients, the image is basically identical to the original.

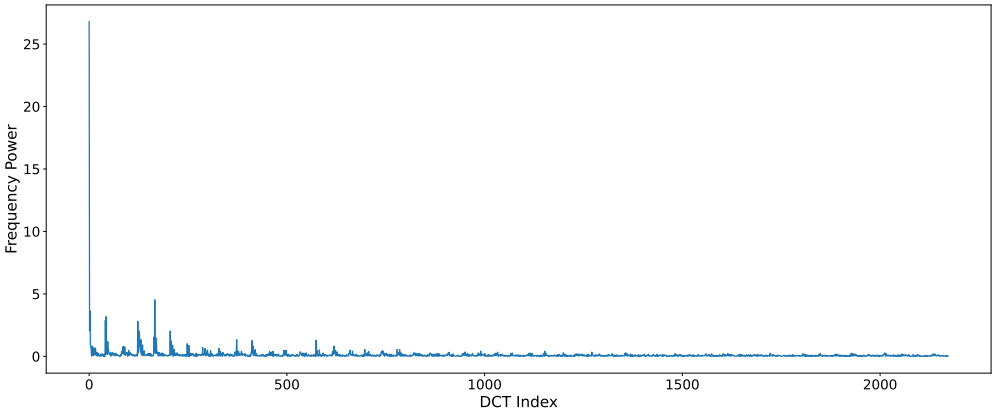

Figure 2. Frequency Power of Flattened Son of Man image

Figure 4 shows the sparsely recovered Son of Man images using $435, 870$, and 1304 observations from the original image. With only 435 observations, it is difficult to clearly identify that this is the Son of Man and there are a number of missing pixels. With 870 rows the image is identifiable, but it is difficult to distinguish between the face, apple, and background in the image. Interestingly, the first and second runs have a lot of missing pixels while third run has very few. Still, the results are impressive considering the number of observations is less than half of the total number of pixels in the image. Using 1304 observations (60% of

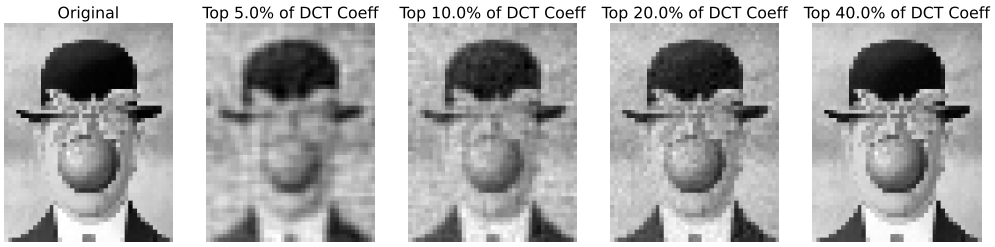

FIGURE 3. Compressed Son of Man images

the pixels), the second and third runs produce an image of similar clarity to the compressed image using the top 10% of the DCT frequencies. The color of the hat and suit have also become more pronounced, which helps provide location boundaries for the face in the image. The features of the apple can also be clearly distinguished, and there is more clear divide between the gray of the face and the image background.

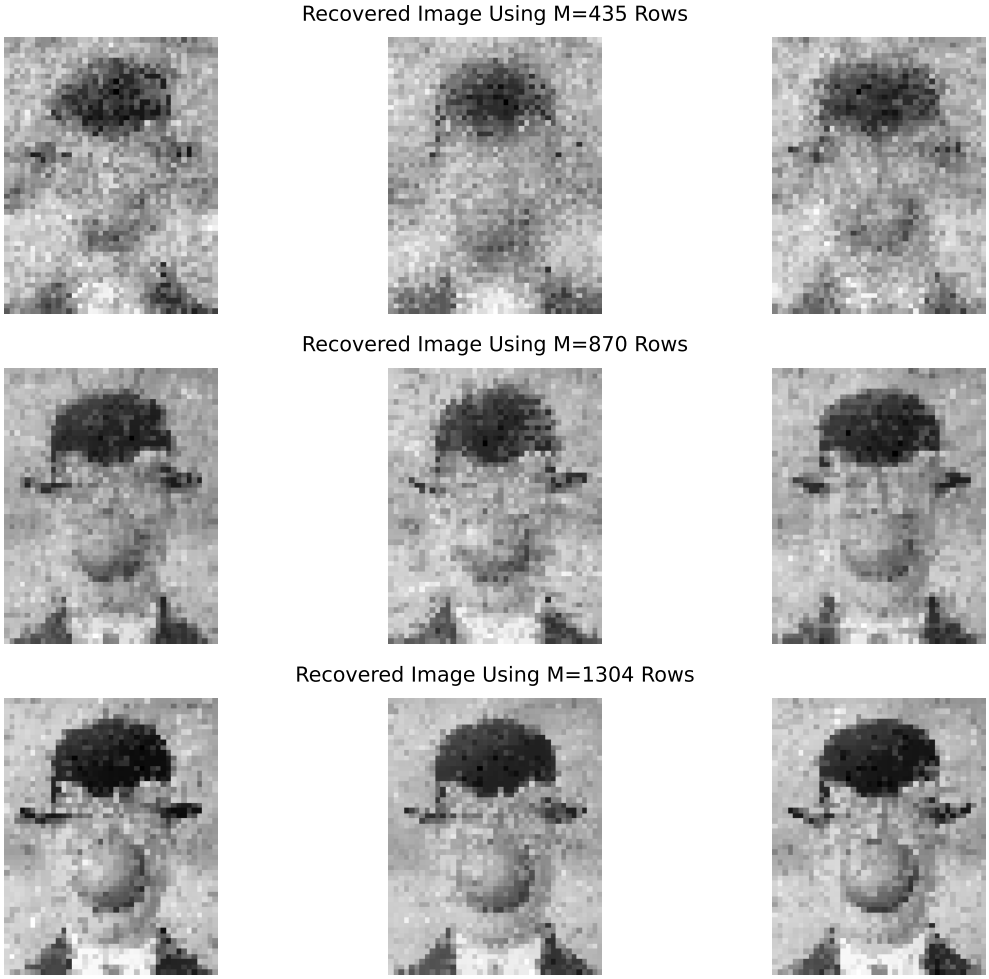

FIGURE 4. Recovered Son of Man images using varying observation counts M

Figure 5 shows the recovery using algorithm 2 for the unknown image provided. The image is Nyan Cat, a character from YouTube video in 2011 that became a meme.

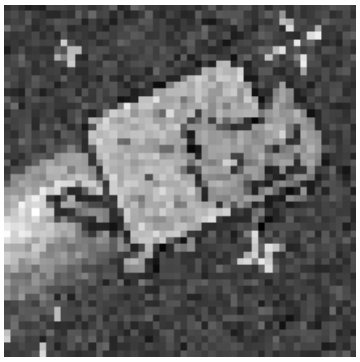

FIGURE 5. Recovered Unknown Image

5. Summary and Conclusions

In this problem, image compression and sparse recovery techniques were explored using a grayscale, resized image of the Son of Man painting. It was found that by filtering the frequencies of the imgage DCT, the image could be compressed with quality similar to the original. Using sparse recovery, the Son of Man image was recovered using a varying number of observations. With 435 observations the image is highly distorted and not identifiable, while increasing the number of observations to 1304 produced recovered an image of comparable quality to a highly compressed version. In future experiments, it would be interesting to test the recovery of signals other than images using the algorithms developed to see how accurate of a recovered signal is obtained. It could also be interesting to combine image recovery algorithms with things such as neural networks to see how well the recovered images would perform in a classification or segmentation model.

Acknowledgements

The author would like to thank Professor Bamdad Hosseini for covering the background concepts of Lasso Regression and Sparse Recovery in lecture. Additionally, discussion with classmates in Discord was useful for troubleshooting issues encountered when generating the algorithms for recovering sparse images. The numpy [9] and matplotlib [10] packages in python were also useful for processing data and generating visualizations.

References

[1] Oct 2001. [Online]. Available: https://users.cs.cf.ac.uk/Dave.Marshall/Multimedia/node231.html
[2] B. Hosseini, "Homework 5: Compressed image recovery," University of Washington, Mar 2022, aMATH 482/582.
[3] R. Taylor, "Compressed sensing in python," May 2016. [Online]. Available: http://www.pyrunner.com/weblog/2016/05/26/compressed-sensing-python/
[4] P. Westphal, "Amath 482/582 homework 2 - classifying digits," Feb 2022.
[5] B. Hosseini, "Introduction to sparse recovery," University of Washington, Feb 2022, aMATH 482/582.
[6] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman Hall/CRC, 2015.
[7] B. Hosseini, "Sparse signal/image recovery," University of Washington, Mar 2022, aMATH 482/582.
[8] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.
[9] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2
[10] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.