University of Toronto
Department of Electrical and Computer Engineering

ECE367 MATRIX ALGEBRA AND OPTIMIZATION

Problem Set #6

Autumn 2020

Prof. S. C. Draper **Due:** 5pm (Toronto time) Friday, 09 December 2020

Homework policy: Problem sets must be turned by the due date and time. Late problem sets will receive deductions for lateness. See the information sheet for futher details. The course text "Optimization Models" is abbreviated as "OptM". Also, see PS01 for details of the "Non-graded", "graded" and "optional" problem categories.

Note: In the categorization below Graded problems are highlighted in **red boldface**

Problem Set #6 problem categories: A quick categorization by topic of the problems in this problem set is as follows:

- Basic problems about convexity: problems 6.1-6.4
- Related optimization problems: problem 6.5
- Sparse coding: **problem 6.6**
- Optional problem (convexity and LPs): problem 6.7

NON-GRADED PROBLEMS

Problem 6.1 (Convex, affine, and conic hulls)

(a) Consider the set $\mathcal{S} = \{[1\ 1]^T, [1\ 2]^T\} \subseteq \mathbb{R}^2$. Sketch conv($\mathcal{S}$), affine($\mathcal{S}$) and conic($\mathcal{S}$), respectively the convex, affine, and conic hulls of the set $\mathcal{S}$. Each is the union of all combinations of the respective type (convex, affine or conic).

(b) Repeat part (a) for the set $\mathcal{S} = \{[1\ 1]^T, [1\ 2]^T, [0.5\ 0.25]^T\}$.

(c) Consider a set $\mathcal{S}$. What are the respective inclusion relations between the convex hull, the affine hull, and the conic hull of $\mathcal{S}$. I.e., which of these three sets are *always* subsets of the other, regardless of the original $\mathcal{S}$?

Problem 6.2 (Proving convexity-preserving operations)

(a) Consider any affine function $f : \mathbb{R}^n \to \mathbb{R}^m$ and convex set $\mathcal{S} \subseteq \mathbb{R}^n$. Prove that the image of $\mathcal{S}$ under f, i.e., $f(\mathcal{S}) = \{f(x) | x \in \mathcal{S}\}$, is a convex set.

(b) Consider any affine function $f : \mathbb{R}^n \to \mathbb{R}^m$ and convex set $\mathcal{S} \subseteq \mathbb{R}^m$. Prove that the *inverse* (or *pre*-) image of $\mathcal{S}$ under f, i.e., $f^{-1}(\mathcal{S}) = \{x | f(x) \in \mathcal{S}\}$, is a convex set.

Problem 6.3 (Is it convex?), from a previous exam
Consider the following optimization problem where $x \in \mathbb{R}$.

$$\begin{array}{ll} \text{minimize} & \sin(x) \\ \text{subject to} & \cos(x) \leq 0 \end{array}.$$

Answer the following questions.

(a) Is the problem a convex optimization problem? (If "yes" say why, if "no" say why not.)

(b) Is the problem stated above feasible? If your answer is that the problem is feasible then also: (i) what is the optimal value p^*, (ii) what is a value for the optimal variable x^*, (iii) is x^* unique (and, if not, specify one additional value for x^*)?

Problem 6.4 (Identifying convexity)
For each of the functions listed in parts (a)-(c) below identify whether the function is (i) convex, is (ii) quasi-convex, is (iii) concave, is (iv) quasi-concave.

(a) $f(x) = e^x - 1$ where $\text{dom} f = \mathbb{R}$.

(b) $f(x_1, x_2) = x_1 x_2$ where $\text{dom} f = \mathbb{R}^2_{++}$.

(c) $f(x) = 1/(x_1 x_2)$ where $\mathrm{dom} f = \mathbb{R}^2_{++}$.

Recall that a function f is "quasiconvex" if all its sublevel sets $\mathcal{S}_\alpha = \{x \in \mathrm{dom} f | f(x) \leq \alpha\}$ are convex sets, i.e., are convex sets for all $\alpha \in \mathbb{R}$. (Note the empty set is a convex set.) A function f is "concave" if the function $-f$ is convex. A function is "quasiconcave" if $-f$ is quasiconvex; equivalently, a function is quasiconcave if every "superlevel" set $\{x \in \mathrm{dom} f | f(x) \geq \alpha\}$ is a convex set.

Problem 6.5 (Quadratic Inequalities)
OptM Problem 8.1.

Problem 6.6 (Monotonicity and locality)
OptM Problem 8.6.

GRADED PROBLEM

Problem 6.7 (Sparse coding of images)
In this problem you will experiment with an image compression method that uses ℓ_1 regularization. To set the context for this problem, we recommend you to read Section 9.6.2 and Example 9.19 in OptM. For solving the numerical parts you may use `CVX`, `MATLAB` or any other computational tool of your choice. In Example 9.19 of OptM, a vectorized image is considered. In this problem we will consider an image without vectorizing it. To avoid possible confusion with the notation for vectorized version in OptM, we use a different notation in this problem. You will note that the two versions are quite similar.

Let $M \in \{0, \dots, 255\}^{n \times n}$ be a matrix that represents a grayscale image of dimensions $n \times n$. Similar to matrix A in Example 9.19, we use an orthonormal matrix $H \in \mathbb{R}^{n \times n}$, i.e., $H^T H = H H^T = I_n$, to perform a wavelet transform. Analogous to the relation $\tilde{y} = A^T y$ in Example 9.19, in our case the wavelet transform coefficients of M are obtained as $\tilde{M} = H M H^T$. This is called the "2-D wavelet transform" where we note that $\tilde{M}$ is a matrix, $\tilde{M} \in \mathbb{R}^{n \times n}$. The transform matrix H is designed so that, when applied to natural images. the entries in the $\tilde{M}$ matrix are typically quite "concentrated", i.e., most are of very small magnitude clustered about zero with a much smaller number of large coefficients. The particular H we use in this problem corresponds to the family of "Haar" wavelets.

As in OptM Ex. 9.19 we use ℓ_1 regularization to search for a sparse encoded $X \in \mathbb{R}^{n \times n}$ to represent M in the (Haar) wavelet domain. In this problem the matrix X play the role of the vector x in the OptM example. As the natural equivalence of the LASSO problem discussed in OptM, but now using our matrix notation, we define the optimization problem

$$f(\lambda) = \min_{X \in \mathbb{R}^{n \times n}} \frac{1}{2} \|H^T X H - M\|_F^2 + \lambda \|X\|_1 \tag{1}$$

and solve for best choice of the matrix X. Given the optimizer X^* the optimal (regularized) approximation to M is $\hat{M} = H^T X^* H$. In these expressions, $\|X\|_1 = \sum_{i\in[n]} \sum_{j\in[n]} |X_{i,j}|$ and $\|\cdot\|_F$ denotes the Frobenius norm defined as $\|V\|_F^2 = \sum_{i\in[n]} \sum_{j\in[n]} |V_{i,j}|^2$ for any real matrix V.

(a) Find an expression for the inverse wavelet transform, i.e., use H to express M in terms of $\tilde{M}$. From the point of view of the optimization problem stated in (1), what is the importance of selecting a specially designed transform matrix H, opposed to using a random orthonormal matrix?

(b) Show that $f(\lambda)$ is a *separable* problem. A separable problem reduces to a set of single variable problems. See OptM Example 9.19 for an example of a separable problem. After comparing your result to that in Example 9.19 you should be able to write down the solution to $f(\lambda)$. Do so. You may find following matrix properties useful: $\|V\|_F^2 = \text{trace}(VV^T)$ and $\text{trace}(ABC) = \text{trace}(CAB) = \text{trace}(BCA)$.

We now move to the numerical part of the problem. Download the file `sparseCoding.mat` from the course website. You can load the content of this file onto a `MATLAB` workspace by executing `load ‘sparseCoding’`. You will see two variables `M` and `H` which correspond to the matrices M and H. In this case $n = 256$. You can view the image by executing `imshow(M, [])`. One measure of the goodness of an approximation $\hat{M}$ to M is the "mean-squared-error" (MSE). The MSE is defined as

$$\text{MSE}(M, \hat{M}) = \frac{1}{n^2} \sum_{i\in[n]} \sum_{j\in[n]} |M_{i,j} - \hat{M}_{i,j}|^2.$$

(c) Produce three histograms. The first two are histograms of the values of M and $\tilde{M}$, and should be similar to Figure 9.19 and 9.20 in OptM. For the third histogram, zoom in on your histogram of $\tilde{M}$ around the origin of the x-axis. You should observe that most coefficients are clustered around zero. This indicates the Haar wavelet transform is doing what it was designed to do. Include all three plots in your solutions. Also indicate the number of non-zero coefficients in M and the number of non-zero coefficients in $\tilde{M}$. (To get a better sense of the difference between the distributions of M and $\tilde{M}$, you might try sorting vectorized versions of M and $\tilde{M}$ and plot the result versus index. You are not required to generate these plots, but if you do you might consider using a log-scale on the vertical axis.)

(d) Compute the optimal X^* for the problem of (1) when $\lambda = 30$. Also compute the corresponding compression factor and MSE. As is discussed in OptM Problem 9.19, the compression factor is the ratio between the number of non-zero components in X^* and the size of M (which is n^2). For the X^* you found, perform the inverse wavelet transform and plot the resulting approximation $\hat{M}$ alongside the original image. As in part (c), produce a histogram of non-zero components of X^*. Zoom in along the x-axis to see what is happening close to the origin, include a plot of the close-up of the histogram in that regime and comment on what you see.

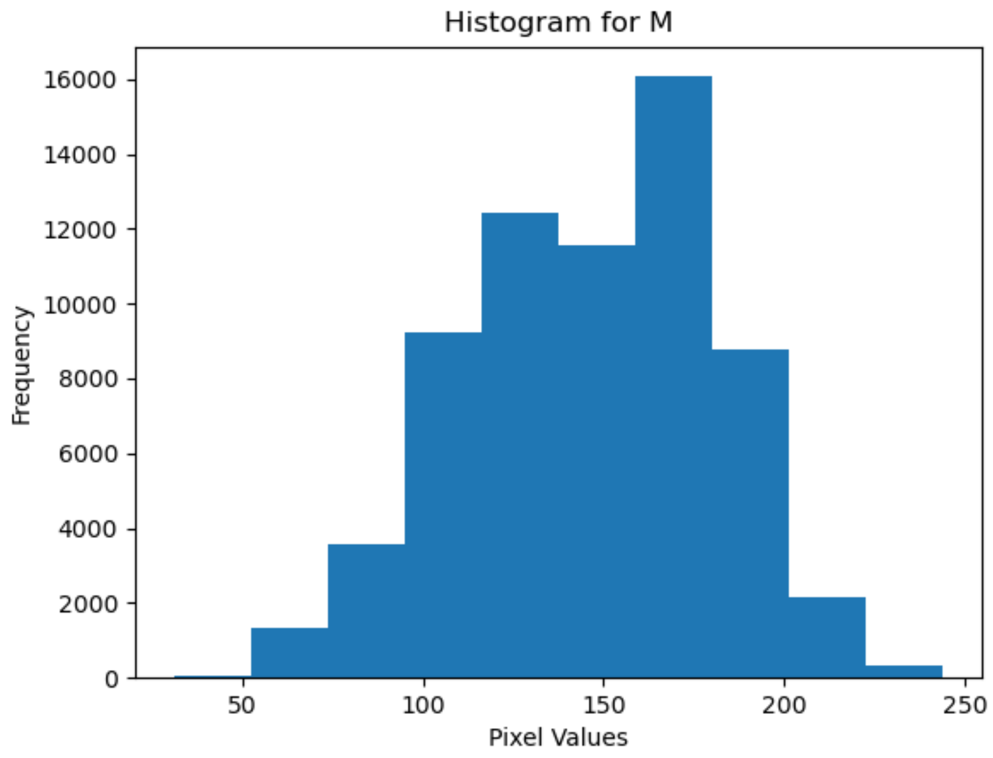
(e) Repeat part (d) but now trying increasing and decreasing λ by a factor of three, i.e., $\lambda = 10$ and $\lambda = 90$. What are the differences that you see between the results for these two λ values versus the $\lambda = 30$ case in terms of the respective $\hat{M}$, the histograms, and the approximate images produced?

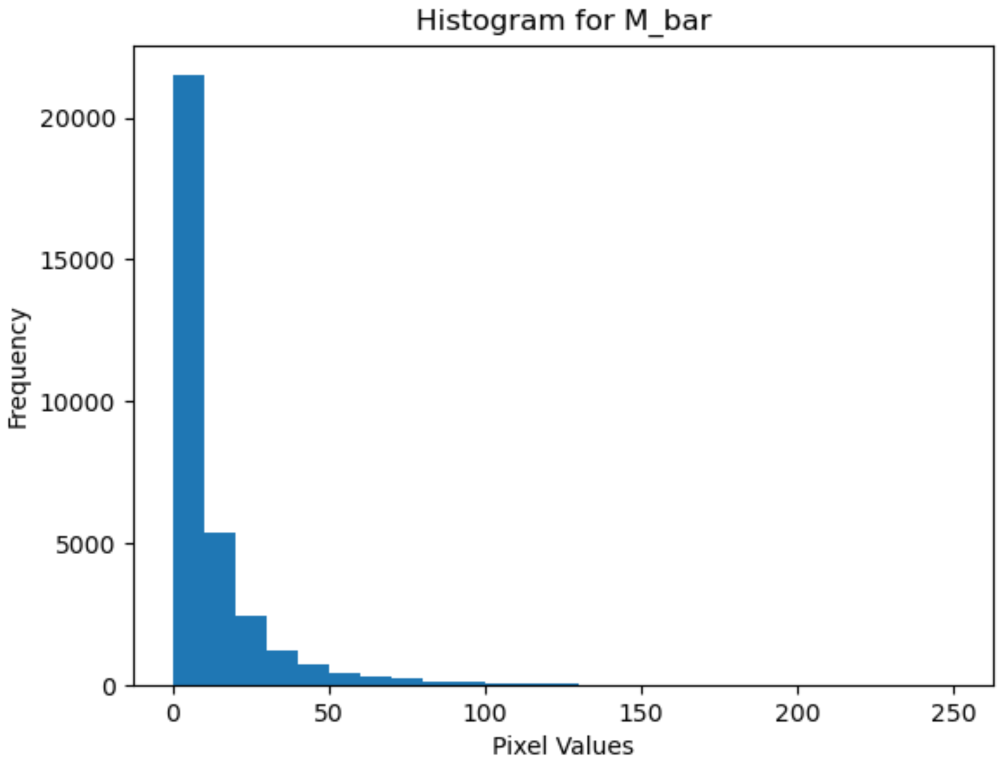
OPTIONAL PROBLEM

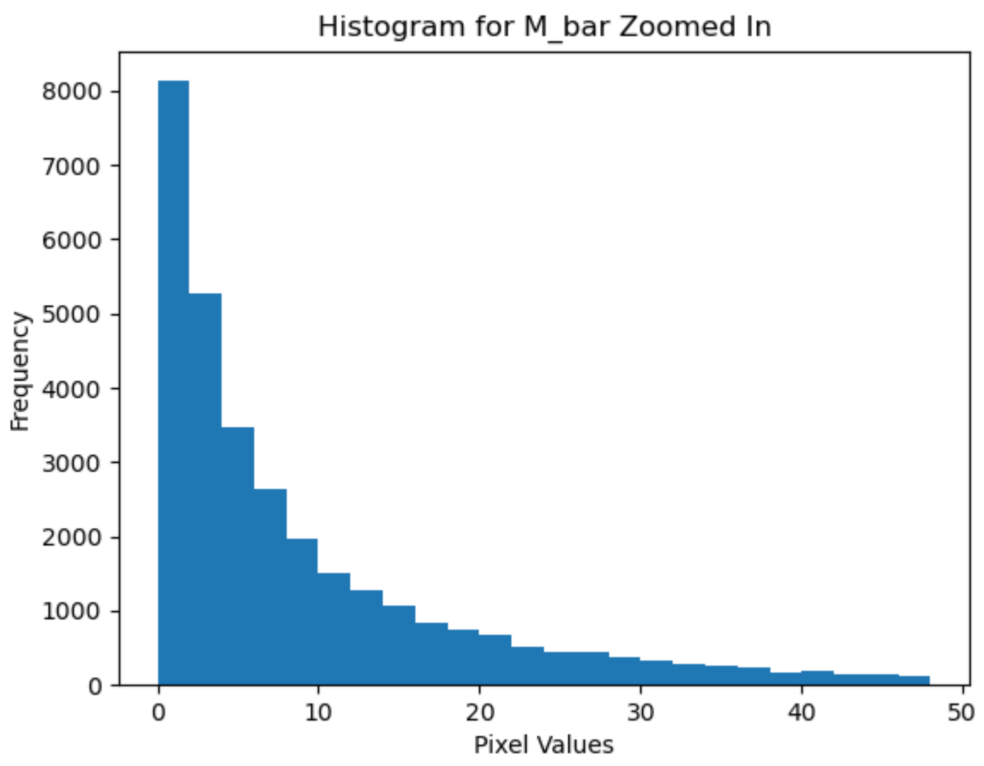
Problem 6.8 (Convexity and concavity of optimal value of an LP)
OptM Problem 9.8.

Hints: (i) For the first part think about re-expressing $\min_x c^T x = \min_x f_x(c)$ where $f_x : \mathbb{R}^n \to \mathbb{R}$ as $f_x(c) = c^T x$. Next, recall that f is concave if $-f$ is convex and apply the equivalent (for concave functions) of the "pointwise supremum or maximum" property of Sec. 8.2.2.4. For the second part one can use the epigraph property characterization of convexity. For the third think about scalar examples.)

Part c


Histogram for M
Frequency
Pixel Values
0
2000
4000
6000
8000
10000
12000
14000
16000
50
100
150
200
250

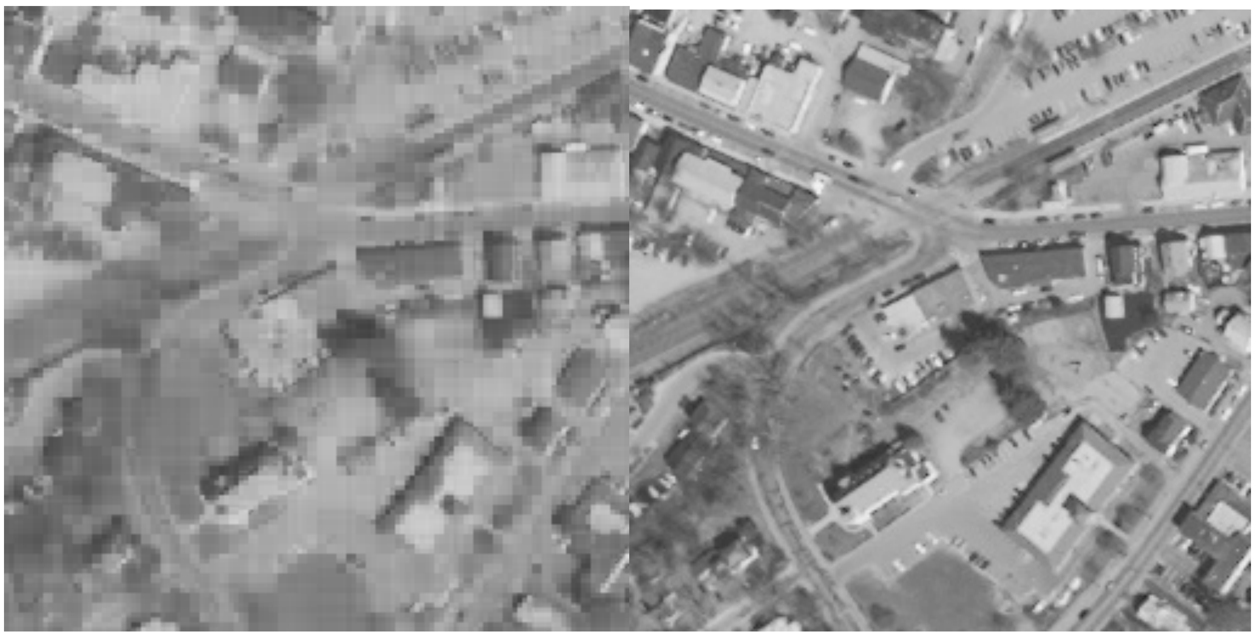

Histogram for M_bar
Frequency
Pixel Values
0
5000
10000
15000
20000
0
50
100
150
200
250

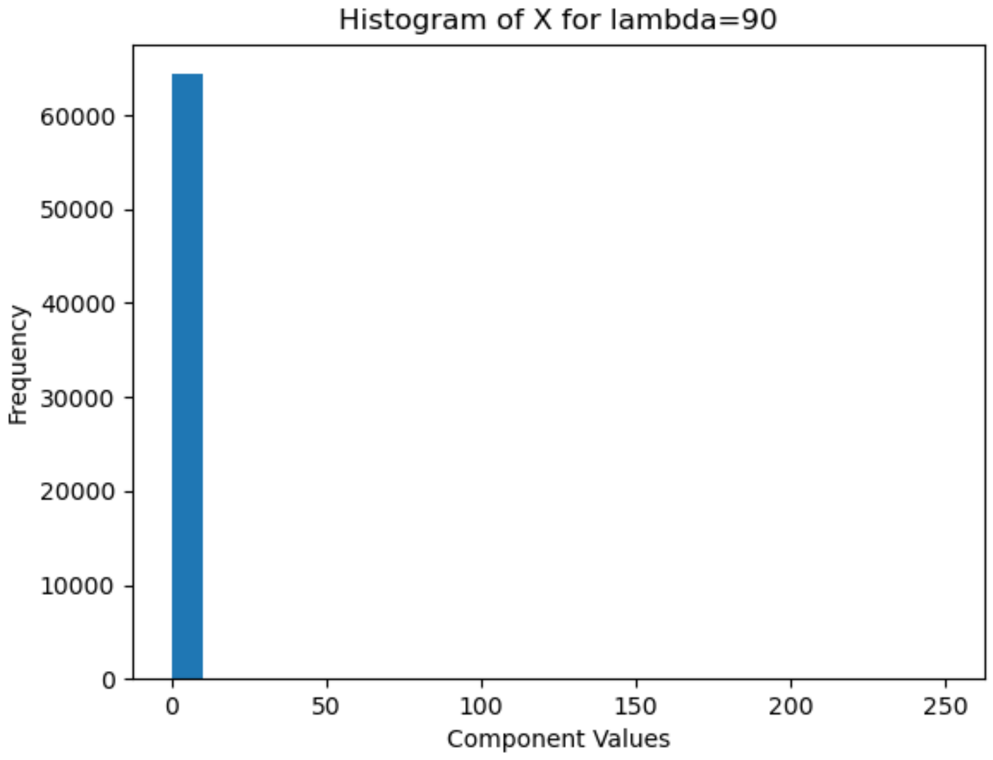


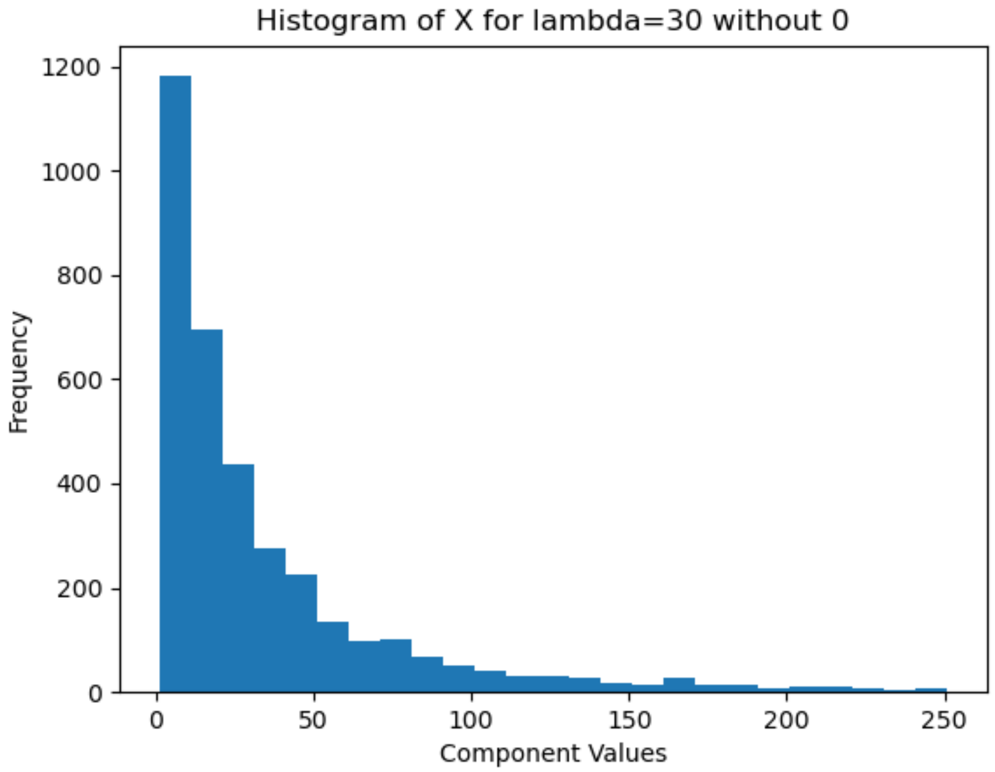
M nonzeros: 65536, M_bar nonzeros: 65248

Part d

Compression Factor: 0.12
Mean Squared Error: 195.36



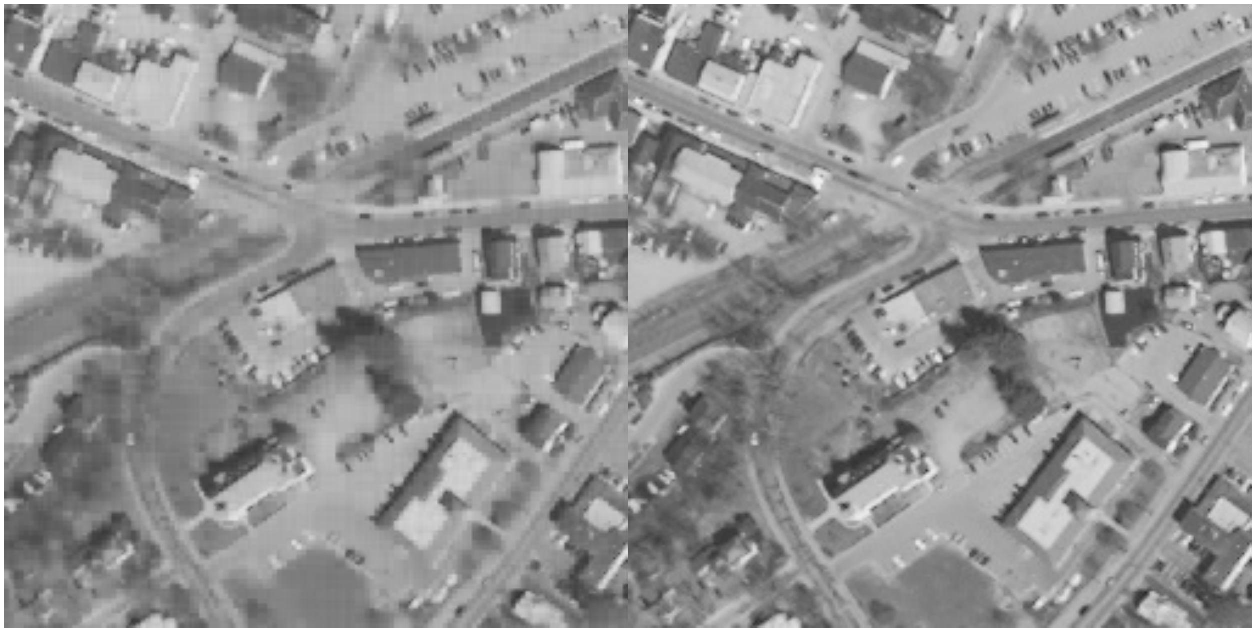

Histogram of X for lambda=90
Frequency
Component Values
0
10000
20000
30000
40000
50000
60000
0
50
100
150
200
250

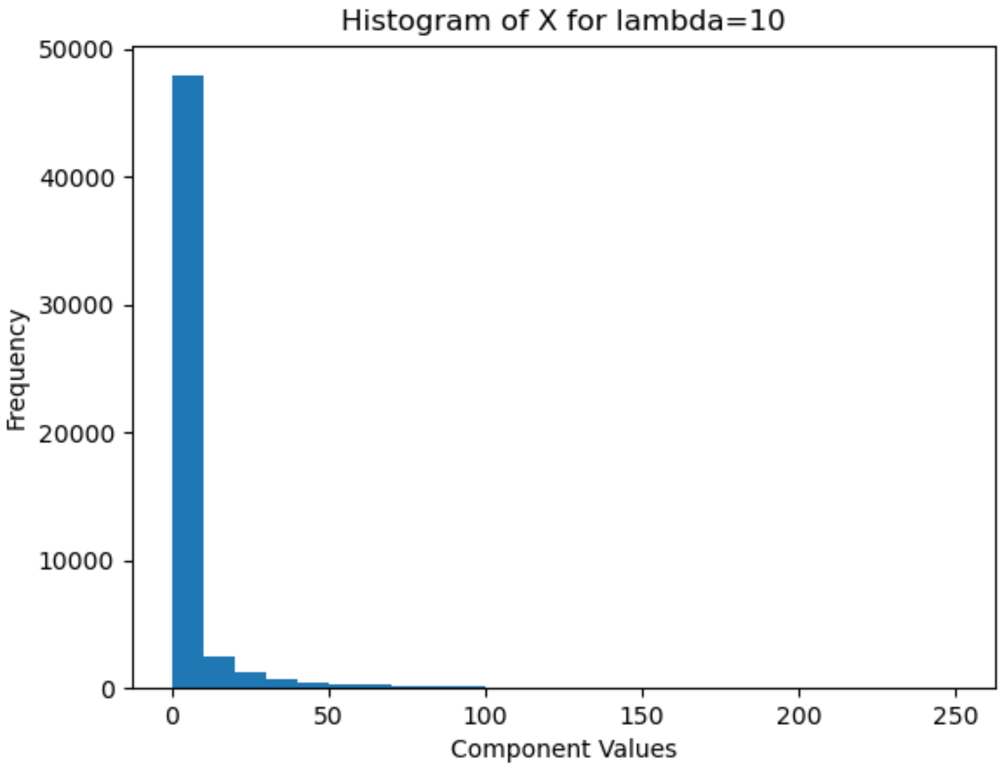

Histogram of X for lambda=30 without 0
Frequency
Component Values
0
200
400
600
800
1000
1200
0
50
100
150
200
250

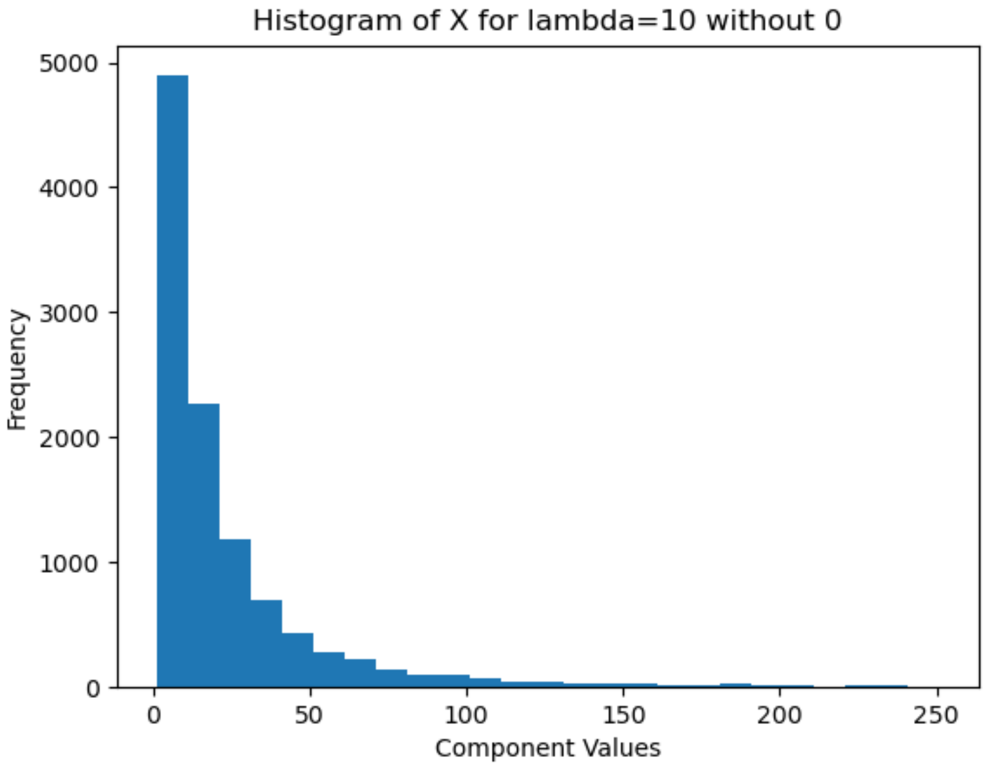
As shown in the histograms above, it becomes obvious that the solution X is extremely sparse. When viewing the first histogram where 0 is included, it is nearly impossible to recognize the existence of non-zero components of X primarily because it is dominated by 0. This is expected, however, since l1 regularization tends to lead to sparse solutions.

Part e

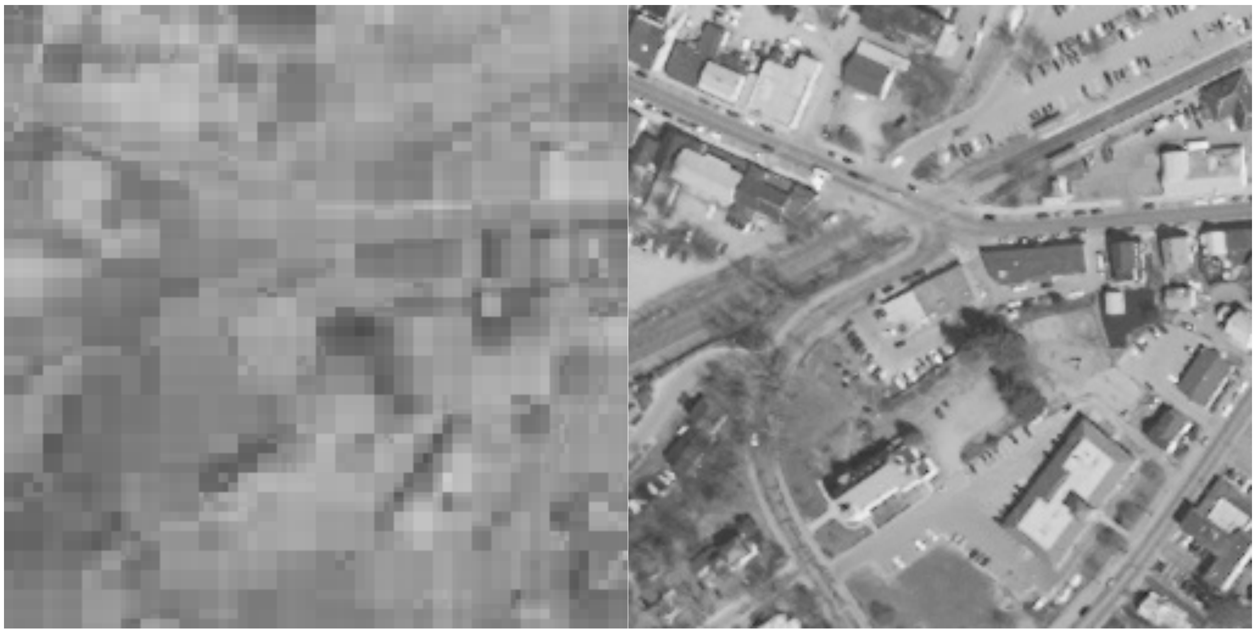
Approximation with lambda = 10
Compression Factor: 0.35
Mean Squared Error: 48.34

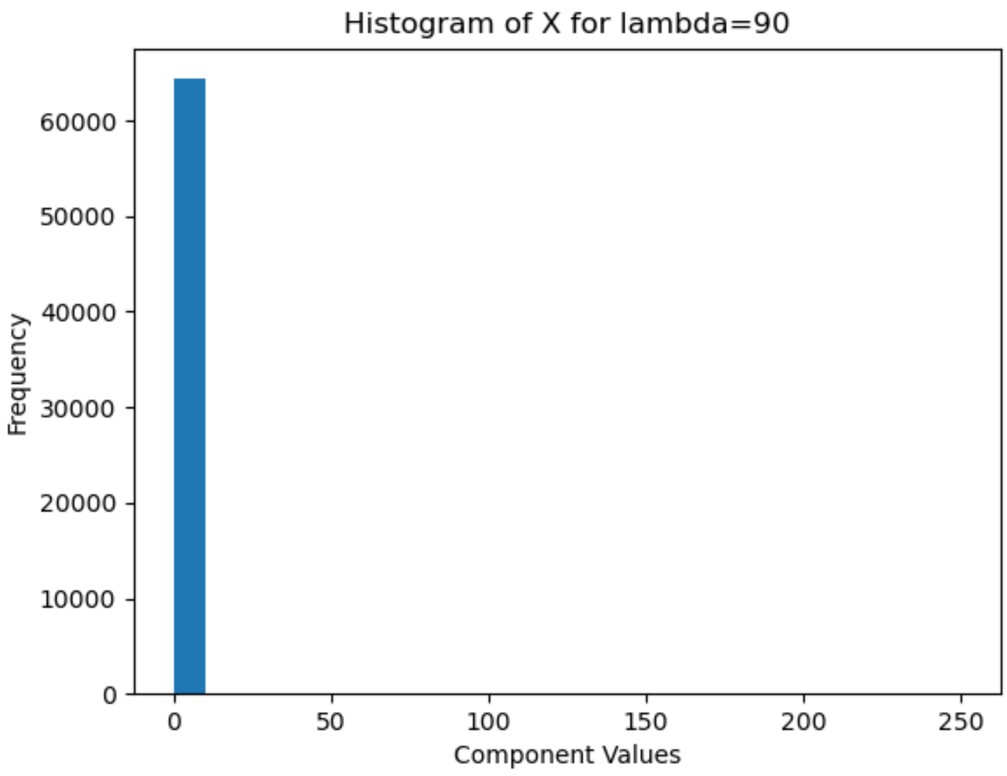


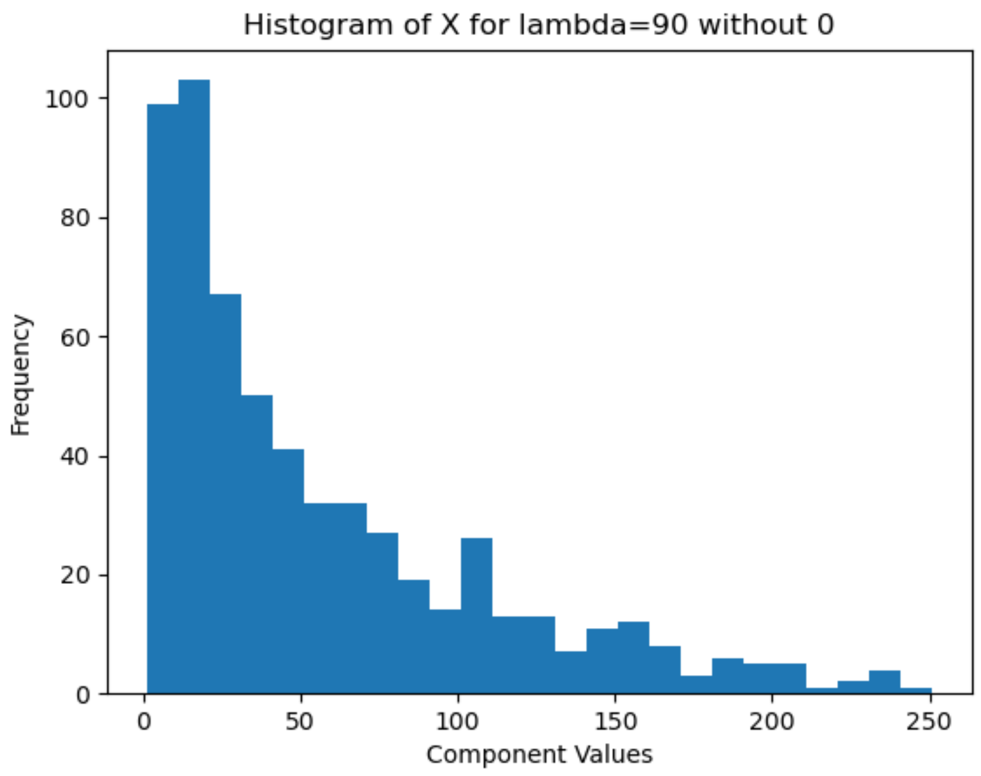

Histogram of X for lambda=10
Frequency
Component Values
0
10000
20000
30000
40000
50000
0
50
100
150
200
250


Histogram of X for lambda=10 without 0
Frequency
Component Values
0
1000
2000
3000
4000
5000
0
50
100
150
200
250

Approximation with lambda = 90
Compression Factor: 0.02
Mean Squared Error: 494.02







As displayed by the above images and graphs, it becomes apparent that the greater the value of lambda, the sparser X becomes and the worse the approximation of the image becomes. This is also directly reflected in the compression factor and mean squared error; a higher lambda leads to a smaller compression factor, but larger mean squared error. This is expected behaviour because the larger the regularization coefficient, the more zeros will show up in the approximation, which is shown by comparing the number of zeros when lambda is 10 compared to when it is 90. The image approximation becomes worse as the regularization coefficient grows because more pixels in the original image are ignored due to the zeros. A larger regularization coefficient directly corresponds to the magnitude of compression, which is why larger coefficients lead to worse image predictions.

Code

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error as mse
from PIL import Image

H = np.genfromtxt('H.csv',delimiter=',')
M = np.genfromtxt('M.csv',delimiter=',')

M_bar = np.matmul(np.matmul(H,M),H.T)
```

```python
def histogram(array, bins, title):
    plt.hist(array.flatten(), bins=bins)
    plt.title(title)
    plt.xlabel('Pixel Values')
    plt.ylabel('Frequency')
    plt.show()

def prediction(l):
    X = np.zeros(M.shape)
    X = np.where(abs(M_bar) <= l, 0, M_bar-l*np.sign(M_bar))

    M_approx = np.matmul(np.matmul(H.T,X),H)
    error = mse(M_approx,M)

    compression = np.count_nonzero(X)/(M.shape[0]*M.shape[1])

    print('Compression Factor: %.2f' %(compression))
    print('Mean Squared Error: %.2f' %(error))

    original = Image.fromarray(M)
    approx = Image.fromarray(M_approx)

    original.show()
    approx.show()
    return X, M_approx

histogram(M,10,'Histogram for M')
histogram(M_bar,range(0,260,10),'Histogram for M_bar')
histogram(M_bar,range(0,50,2),'Histogram of M_bar Zoomed In')

M_nonzero = np.count_nonzero(M)
M_bar_nonzero = np.count_nonzero(M_bar)
print('M nonzeros: %i, M_bar nonzeros: %i' %(M_nonzero, M_bar_nonzero))

l = 90
x, m = prediction(l)
histogram(x,range(0,255,10),'Histogram of X for lambda=%i' %(l))
histogram(x,range(1,255,10),'Histogram of X for lambda=%i without 0' %(l))
```