CS/ECE/ME532 Activity 15

Estimated Time: 15 minutes for Q1, 25 minutes for Q2.

a). $V = \begin{bmatrix} 1 & -2 & -4 \\ -1 & 2 & 4 \\ -1 & 2 & 4 \\ 1 & -2 & -4 \end{bmatrix}$

1. Consider the 4-by-3 matrix defined as $\boldsymbol{V} = \begin{bmatrix} 1 & \mathrm{X} & \mathrm{X} \\ \mathrm{X} & 2 & 4 \\ -1 & 2 & \mathrm{X} \\ \mathrm{X} & -2 & \mathrm{X} \end{bmatrix}$ where X denotes missing entries. Assume $\boldsymbol{V}$ is a rank-1 matrix.

a) Use what you know about the structure of rank-1 matrices to find the missing entries.

b) What is the minimum number of missing entries for which you cannot complete a 4-by-3 rank 1 matrix? Where are the missing entries in this case?

If 3 entries are missing from the same row, the matrix cannot be completed.

2. A data file is available that contains a rank-2, 16-by-16 matrix `Xtrue` with integer entries and three versions of this matrix (`Y1`, `Y2`, and `Y3`) with differing numbers of missing entries. The missing entries are indicated by `NaN`.

A script is provided to complete a matrix using iterative singular value thresholding. The script contains a function that requires two inputs: *i)* the matrix with missing entries, and *ii)* the rank.

a) Apply the iterative singular value thresholding function (provided in the script) to the three incomplete matrices assuming the rank is 2. You will first need to complete the line of code in the function. Compare your recovered completed matrices to `Xtrue` (Note: compare the output by subtracting the completed matrix from the original matrix, and then displaying them). Does the number of missing entries affect the accuracy of the completed matrix?

b) Now apply your routine to the three incomplete matrices assuming the rank is 3. Compare your recovered completed matrices to `Xtrue`. Comment on the impact of using the incorrect rank in the completion process.

a). Generally, the fewer missing values the higher the accuracy of the completed guesses. With 136 missing, the total difference is about 230, which decreases significantly to 0.0018 with 76 missing entries and -0.00018 with 16 missing entries

b). Error is quite a bit higher when rank is 3.
136 missing → 523 16 missing → 122
76 missing → 122

```
import numpy as np
from scipy.io import loadmat
```

Question 2

```
Xtrue = loadmat("incomplete.mat")["Xtrue"]
Y1 = loadmat("incomplete.mat")["Y1"]
Y2 = loadmat("incomplete.mat")["Y2"]
Y3 = loadmat("incomplete.mat")["Y3"]


def ItSingValThresh(Y, r):
    """
    Iterative Singular Value Thresholding function for Matrix Completion
    """
    tol = 10**(-3)  # difference between iterates at termination
    max_its = 100;
    n,p = Y.shape
    X = np.array(Y) #make a copy so operations do not mutate the original
    X[np.isnan(X)] = 0 # Fill in missing entries with zeros

    err = 10**6
    itt = 0

    while err > tol and itt < max_its:
        U,s,VT = np.linalg.svd(X, full_matrices=False)
        V, S = VT.T, np.diag(s)
        Xnew = U[:,0:r]@S[0:r,0:r]@VT[0:r,:] ### Complete this line
        for i in range(n):
            for j in range(p):
                if ~np.isnan(Y[i,j]):  #replace Xnew with known entries
                    Xnew[i,j] = Y[i,j]
        err = np.linalg.norm(X-Xnew,'fro')
        X = Xnew
        itt += 1
    return X


X1 = ItSingValThresh(Y1, 2)
print("Num of Missing: ", sum(sum(np.isnan(Y1))))
print("Compare: \n", Xtrue-X1)
print("Error: ", sum(sum(Xtrue-X1)))

X2 = ItSingValThresh(Y2, 2)
print("Num of Missing: ", sum(sum(np.isnan(Y2))))
print("Compare: \n", Xtrue-X2)
print("Error: ", sum(sum(Xtrue-X2)))
```

```
X3 = ItSingValThresh(Y3, 2)
print("Num of Missing: ", sum(sum(np.isnan(Y3))))
print("Compare: \n", Xtrue-X3)
print("Error: ", sum(sum(Xtrue-X3)))
```

```
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 2.51033270e-05  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  9.41627285e-05  0.00000000e+00
   0.00000000e+00  0.00000000e+00  1.77539919e-05  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  5.14112472e-06  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  2.52004533e-05
   1.33700956e-05  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [-2.01394180e-05  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00 -6.02194882e-05  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00

   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 4.89656768e-05  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
  -3.05136634e-04  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  4.21924336e-04  0.00000000e+00  0.00000000e+00
  -4.51478963e-04  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00 -8.47534783e-05  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
```

```
      [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00 -1.41677235e-06]]
     Error:  -0.00018252060126489056


X1 = ItSingValThresh(Y1, 3)
print("Num of Missing: ", sum(sum(np.isnan(Y1))))
print("Compare: \n", Xtrue-X1)
print("Error: ", sum(sum(Xtrue-X1)))

X2 = ItSingValThresh(Y2, 3)
print("Num of Missing: ", sum(sum(np.isnan(Y2))))
print("Compare: \n", Xtrue-X2)
print("Error: ", sum(sum(Xtrue-X2)))

X3 = ItSingValThresh(Y3, 3)
print("Num of Missing: ", sum(sum(np.isnan(Y3))))
print("Compare: \n", Xtrue-X3)
print("Error: ", sum(sum(Xtrue-X2)))
      [ 9.24413430e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
      [ 0.00000000e+00  0.00000000e+00 -4.65676240e-05  0.00000000e+00
        0.00000000e+00  0.00000000e+00 -8.33472246e-06  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]

      [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00 -3.00064311e-06  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  6.07257541e-05
       -1.65479369e-05  0.00000000e+00  0.00000000e+00  0.00000000e+00]
      [ 1.35202658e+01  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
      [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
      [ 0.00000000e+00  2.38104370e-05  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
      [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
      [ 1.27971686e+01  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
      [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
        0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
```

```
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   3.69213846e-04  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00 -2.00027710e-04  0.00000000e+00  0.00000000e+00
   2.61458261e-04  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  1.81949454e-05  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00  0.00000000e+00  3.71555723e-06]]
Error:  122.02310065259364
```