CS/ECE/ME532 Activity 15

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

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?

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.

1. Consider the 4-by-3 matrix defined as $\boldsymbol{V} = \begin{bmatrix} 1 & \text{X} & \text{X} \\ \text{X} & 2 & 4 \\ -1 & 2 & \text{X} \\ \text{X} & -2 & \text{X} \end{bmatrix}$ 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?

a.) If rank(V) = 1:

$$\underline{\tilde{V}} = \begin{bmatrix} 1 & -2 & -4 \\ -1 & 2 & 4 \\ -1 & 2 & 4 \\ 1 & -2 & -4 \end{bmatrix}$$

$2\cdot$

$-\frac{1}{2}\cdot$

b.) You need a full row or column basis vector, plus one value in every other row or column to act as a scale factor.

Looking at rows, you need: 3 + 3 = 6 known items

Looking at columns, you need 4 + 2 = 6 known items.

So you cannot complete a rank-1, 4x3 matrix with < 6 known values (or, equivalently, > 6 unknowns).

$$\begin{bmatrix} 1 & \bullet & \bullet \\ 2 & 4 & \bullet \\ 3 & \bullet & 6 \\ 4 & \bullet & \bullet \end{bmatrix}, \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & \bullet & \bullet \\ \bullet & 10 & \bullet \\ \bullet & \bullet & 12 \end{bmatrix}$$

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.

In [24]:
```
Xtrue = loadmat("incomplete.mat")["Xtrue"]
Y1 = loadmat("incomplete.mat")["Y1"]
Y2 = loadmat("incomplete.mat")["Y2"]
Y3 = loadmat("incomplete.mat")["Y3"]
```

In [61]:
```
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[:,:r] @ S[:r,:r] @ VT[:r,:]
        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
```

In [74]:
```
# Problem 2a

# compute frobenius norm of errors
errors = []
for item in [Y1, Y2, Y3]:
    error = np.linalg.norm(Xtrue-ItSingValThresh(item,2),'fro')
    errors.append(error)

print(errors)

print("Num zeros:\n" ,np.count_nonzero(np.isnan(Y1)), np.count_nonzero(np.isnan(Y2)), np.cou

# Comment: Yes, the number of missing entries impacts the accuracy
# of the filled in matrix.
```

```
[87.24667705099746, 0.0047355995274097045, 0.0007153218655240741]
Num zeros:
 136 76 16
```

In [75]:

```
# Problem 2b

# compute frobenius norm of errors
errors2b = []
for item in [Y1, Y2, Y3]:
    error = np.linalg.norm(Xtrue-ItSingValThresh(item,3),'fro')
    errors2b.append(error)

print(errors2b)

# Comment: Using the incorrect rank causes the error to
# substantially increase.
```

```
[128.77804846772054, 48.97940976510775, 20.785069891601793]
```