CSCI 3656: Numerical Computation **Pourna Sengupta**
Homework 4 October 2, 2020

https://www.geeksforgeeks.org/program-to-check-if-a-matrix-is-symmetric/
https://www.geeksforgeeks.org/program-check-diagonal-matrix-scalar-matrix/
https://www.geeksforgeeks.org/check-whether-given-matrix-orthogonal-not/

```
import numpy as np
import sys

#helper functions

#Tranpose Matrix
def transpose(mat, tran, length):
for i in range(length):
for j in range(length):
#tranpose matrix
tran[i][j] = mat [j][i]

#check if matrix is symmetric
def isSymmetric(mat, r, c):
#transpose matrix
tran = [[0 for a in range(c)] for b in range(r)]
#call helper function
transpose(mat, tran, c)

for i in range(c):
for j in range(c):
#compare matrix and tranpose
#if not equal, they are not symmetric
if(mat[i][j] != tran[i][j]):
return 0
return 1
#check if matrix is diagonal
def isDiagonal(mat, r, c):
for i in range(0, c):
for j in range(0, c):
if((i != j) and (mat[i][j] != 0)):
return 0
return 1


def isOrthogonal(mat, r, c):
if(r != c):
return 0
#transpose matrix
tran = [[0 for a in range(c)] for b in range(r)]
```

```
transpose(mat, tran, c)

#product matrix
matProduct = [[0 for k in range(c)] for l in range(c)]

#find the product matrix of the resultMatrix
#find the transpose of the product matrix
for i in range(0, c):
for j in range(0, c):
sum = 0

for x in range(0, c):
sum = sum + (mat[i][x] * mat[j][x])

matProduct[i][j] = sum

#check for identity matrix
for m in range(0, c):
for n in range(0, c):
if( m != n and matProduct[m][n] != 0):
return 0
if(m == n and matProduct[m][n] != 1):
return 0

return 1

def readMatrix(filename):
resultMatrix = []

#open text file
#name matrixFile
matrixFile = open(filename, 'r')

for lines in matrixFile:
#split matrixFile by line
lines = lines.rstrip('\n')

#then split by comma
sCells = lines.split(',')

#using numpy map, change strings to float
fCells = list(map(np.float32, sCells))

#append to resultMatrix
resultMatrix.append(fCells)
```

```
matrixFile.close()
#find matrix dimensions
rows = len(resultMatrix)
columns = len(resultMatrix[0])

#find number of nonzeros
nonzeros = np.count_nonzero(resultMatrix)

#is the matrix symmetric
symmetric = isSymmetric(resultMatrix, rows, columns)

#is the matrix diagonal
diagonal = isDiagonal(resultMatrix, rows, columns)

#is the matrix orthogonal
orthogonal = isOrthogonal(resultMatrix, rows, columns)

#rank
rank = np.linalg.matrix_rank(resultMatrix)

#minimum value
minVal = np.amin(resultMatrix)

#maximumvalue
maxVal = np.amax(resultMatrix)

#condition number
cNum = np.linalg.cond(resultMatrix)

print("--------------\n")
print("Matrix Report for " + filename + "\n")
print("--------------\n")

print("[" ,rows, "x ", columns, "] matrix")
print("Nonzeros: ", nonzeros)

if(symmetric == 1):
print("Symmetric: True")
else:
print("Symmetric: False")

if(diagonal == 1):
print("Diagonal: True")
else:
print("Diagonal: False")
```

```
if(orthogonal == 1):
print("Orthogonal: True")
else:
print("Orthogonal: False")
print("Rank: ", rank)
print("Smallest Value: ", minVal)
print("Largest value: ", maxVal)
print("Condition Number: ", cNum)

return np.asarray(resultMatrix, dtype=np.float32)

def main():
readMatrix("mat1.txt")
readMatrix("mat2.txt")
readMatrix("mat3.txt")
readMatrix("mat4.txt")
readMatrix("mat5.txt")


if __name__ == '__main__':
main()
```

Matrix Report for mat1.txt

[4 x 4] matrix
Nonzeros: 10
Symmetric: False
Diagonal: False
Orthogonal: False
Rank: 3
Smallest Value: 0.0
Largest value: 0.8762
Condition Number: 1.9963328e+16

Matrix Report for mat2.txt

[30 x 30] matrix
Nonzeros: 900
Symmetric: True
Diagonal: False

Orthogonal: False
Rank: 30
Smallest Value: -5.043682
Largest value: 12.633182
Condition Number: 206.6731

Matrix Report for mat3.txt

[400 x 400] matrix
Nonzeros: 800
Symmetric: False
Diagonal: False
Orthogonal: False
Rank: 399
Smallest Value: -1.0
Largest value: 1.0
Condition Number: 4.2814807e+16

Matrix Report for mat4.txt

[50 x 50] matrix
Nonzeros: 2500
Symmetric: True
Diagonal: False
Orthogonal: False
Rank: 50
Smallest Value: -0.41156825
Largest value: 0.9999995
Condition Number: 1.0

Matrix Report for mat5.txt

[625 x 625] matrix
Nonzeros: 3025
Symmetric: True
Diagonal: False
Orthogonal: False

Rank: 625
Smallest Value: -1.0
Largest value: 4.0
Condition Number: 273.30606