

COSC 343: Test 1

Micah Sherry

February 24, 2024

1 vector P-Norm

write a python function that will compute the p-norm of any vector it is passed

code:

Listing 1: vector p-norm

```
import numpy as np
import matplotlib
from random import randint

from Latex_generator import *

def p_norm(vec, p):
    sum = 0
    for element in vec:
        if not isinstance(element, (int, float)): # nonnumeric or wrong shape
            raise TypeError("elements of the vector must be numbers")

        sum += np.abs(element) ** p

    return sum ** (1/p)

if __name__ == "__main__":
    """testing and formatting the p-norm function"""
    for i in range(4,7):
        vec = []
        for j in range(i):
            vec.append(randint(-10, 10))
        p = randint(1,3)
        ans = round(p_norm(vec, p),3)
        latex = vector_to_latex(vec)
        print(pnorm_to_latex(latex, p, ans))
```

sample output:

$$\left\| \begin{pmatrix} 2 \\ -6 \\ 6 \\ -6 \end{pmatrix} \right\|_2 = 10.583$$
$$\left\| \begin{pmatrix} -10 \\ 0 \\ 7 \\ 9 \\ 3 \end{pmatrix} \right\|_3 = 12.804$$

$$\left\| \begin{pmatrix} -8 \\ -2 \\ -7 \\ -4 \\ -1 \\ -2 \end{pmatrix} \right\|_2 = 11.747$$

2 Matrix 1 norm

write a python function that will compute the 1-norm of a matrix

code:

Listing 2: matrix 1-norm

```
import numpy as np
import matplotlib
from random import randint

from Latex_generator import *

def one_norm(matrix):
    max = 0
    m = len(matrix)
    n = len(matrix[0])

    for j in range(n):
        sum = 0
        for i in range(m):
            if len(matrix[i]) != n: # wrong shape
                raise ValueError("Matrix must be rectangular")
            element = matrix[i][j]
            if not isinstance(element, (int, float)): # nonnumeric or wrong shape
                raise TypeError("elements of the vector must be numbers")

            sum += np.abs(element)
        if sum > max:
            max = sum
    return max

if __name__ == "__main__":
    """test one_norm """
    for i in range(3):
        m = randint(3,5)
        n = randint(3,5)
        matrix = []
        for i in range(m):
            row = []
            for j in range(n):
                row.append(randint(-10,10))
            matrix.append(row)
        p = 1
        ans = one_norm(matrix)
        matrix_latex = matrix_to_latex(matrix)
        print(pnorm_to_latex(matrix_latex, p, ans))
```

sample output:

$$\left\| \begin{pmatrix} -5 & 7 & 8 & -9 & 8 \\ 5 & -6 & -1 & -5 & 2 \\ -2 & 8 & -2 & -2 & 1 \end{pmatrix} \right\|_1 = 21$$

$$\left\| \begin{pmatrix} -3 & 8 & -7 & -3 & 4 \\ -10 & -7 & 2 & -5 & -7 \\ 0 & -8 & -9 & 10 & 5 \end{pmatrix} \right\|_1 = 23$$

$$\left\| \begin{pmatrix} -3 & 8 & -8 & -1 \\ 4 & -5 & -4 & -1 \\ 9 & 4 & -7 & 9 \\ 4 & 3 & 10 & 2 \\ 3 & -6 & -8 & -5 \end{pmatrix} \right\|_1 = 37$$

3 Matrix ∞ norm

write a python function that will compute the ∞ -norm of a matrix

code:

Listing 3: matrix ∞ norm

```
import numpy as np
import matplotlib
from random import randint

from Latex_generator import *

def inf_norm(matrix):
    max = 0
    m = len(matrix)
    n = len(matrix[0])

    for i in range(m):
        sum = 0
        if len(matrix[i]) != n: # wrong shape
            raise ValueError("Matrix must be rectangular")

        for j in range(n):
            element = matrix[i][j]
            if not isinstance(element, (int, float)): # nonnumeric or wrong shape
                raise TypeError("elements of the vector must be numbers")
            sum += np.abs(element)

        if sum > max:
            max = sum
    return max

if __name__ == "__main__":
    """ test inf_norm """

    for i in range(3):
        m = randint(3,5)
        n = randint(3,5)
        matrix = []
        for i in range(m):
```

```

row = []
for j in range(n):
    row.append(randint(-10,10))
matrix.append(row)
p = "\\infty"
ans = inf_norm(matrix)
matrix_latex = matrix_to_latex(matrix)
print(pnorm_to_latex(matrix_latex, p, ans))

```

sample output:

$$\left\| \begin{pmatrix} 1 & -2 & -9 & -5 & 8 \\ 3 & 10 & 2 & -8 & -9 \\ 1 & -10 & 5 & -2 & 8 \\ -5 & 8 & 0 & -10 & 1 \\ 0 & 0 & 7 & -10 & -7 \end{pmatrix} \right\|_{\infty} = 32$$

$$\left\| \begin{pmatrix} -8 & 7 & 2 \\ -3 & -6 & -2 \\ -9 & 6 & -4 \end{pmatrix} \right\|_{\infty} = 19$$

$$\left\| \begin{pmatrix} 2 & 0 & 5 & -1 \\ -6 & 9 & -2 & 0 \\ -4 & -2 & -4 & -2 \\ 9 & -10 & -8 & 7 \end{pmatrix} \right\|_{\infty} = 34$$

4 formatting code

This section is for code I created to format the vectors, matrices and norms. it does not contain any logic for calculating the norms and is only included for completeness sake.

Listing 4: latex code

```
def vector_to_latex(vec):
    """formats vectors to be added to a latex document"""
    latex_code = "\\begin{pmatrix}"
    for element in vec:
        latex_code += str(element) + "-\\\\\\"
    latex_code += "\\end{pmatrix}"

    return latex_code

def matrix_to_latex(matrix):
    """formats matrices to be added to a latex document"""
    latex_code = "\\begin{pmatrix}"
    for row in matrix:
        latex_code += "&".join(map(str, row)) + "-\\\\\\"
    latex_code += "\\end{pmatrix}"
    return latex_code

def pnorm_to_latex(matrix_latex ,p, ans):
    """ formats the norm to be added to the latex document """
    return "$$-\\left\\|\\|^{-"+matrix_latex+"\\right\\|\\|^{-\\{"+str(p)+"\\}}^{-}"+str(ans)+"$$"
```