Data Warehousing & Mining Lab Assignment

Lab - 3

Sub Code: CSE-326

Date: 02-02-2022

Vivek Kumar Ahirwar

191112419

CSE - 3

Department:

Computer Science and Engineering

Contents

A. Write a program to perform data reduction using wavelet (HAAR) transformation on input given by user. Also extend same program to perform inverse wavelet transform.

a. First take input from user.
b. Apply wavelet transform.
c. Print transformed data.
d. Ask user to decide threshold.
e. Apply inverse wavelet transform.
f. Plot original data, transformed data and reconstructed data on same plot to observe the changes.

Make this program in generalized way that it will take input of variable size.

i. Approach Used:

HAAR wavelet algorithm is a type of recursive algorithm. At first, we need to validate the input data. We need to have the input data of length be a power of 2. To do this, we append 0 at the end of the data to confirm this. After this, we implement the HAAR wavelet algorithm. We know, at each step we reduce the length of the data by 2. So, we can simply create a recursive function which sends half the length in the next call until the length we are left with is 1. At each call, we iterate for half the length and create a new array with the values of next transformation, which is then sent to the next recursive call. At the end we return the complete transformed array.

Same approach has been used in the reconstruction of the data. We now go from length 1 to full length of the data. But before this, we also have to take care of the threshold. Now, on each recursion call we expand the data (reconstruct) following the HAAR wavelet algorithm. When we reach the full length, we simply return the complete reconstructed data.

ii. Code

```
# importing matplotlib library
import matplotlib.pyplot as plt


# utility function to get the smallest power of 2
# greater than or equal to the given number
def power2(x):
    ans = 1
    if x and not(x & (x-1)):
        return x
    while ans < x:
        ans <<= 1
    return ans


# performs HAAR wavelet transformations on the input list
# the recursive function inside performs the wavelet transformation
def transform(inp):
    high = power2(len(inp))
```

```
    while len(inp) < high:
        inp.append(0)

    def util(arr, l):
        if l == 1:
            return arr
        mid = l//2
        new_arr = arr.copy()
        for i in range(mid):
            new_arr[i] = (arr[2*i] + arr[2*i+1])//2
            new_arr[i+mid] = arr[2*i] - new_arr[i]

        return util(new_arr, mid)

    transformed = util(inp, len(inp))
    return transformed


# reconstructs new data from the input list after discarding
# values less than the given threshold
def inverse(inp, threshold):
    high = power2(len(inp))

    while len(inp) < high:
        inp.append(0)

    inp = [0 if i < threshold else i for i in inp]

    def inverseUtil(inp, l):
        if l == len(inp):
            return inp

        mid = l*2
        new_arr = inp.copy()
        for i in range(l):
            new_arr[2*i] = inp[i] + inp[i+l]
            new_arr[2*i+1] = inp[i] - inp[i+l]

        return inverseUtil(new_arr, mid)

    reconstructed = inverseUtil(inp, 1)
    return reconstructed


# Example input:
# inp = [ ]
# transformed = transform(inp)
# threshold = 10
# reconstructed = inverse(transformed, threshold)
inp = input("Enter the data values of any size and length: ").split()
inp = [int(i) for i in inp]
```

```
transformed = transform(inp)
print("The transformed data is: ", transformed)

threshold = int(input("Enter the threshold value: "))
reconstructed = inverse(transformed, threshold)
print("The reconstructed data is: ", reconstructed)

# Plotting of all the data
plt.plot(inp, label="Input data")
plt.plot(transformed, label="Transformed data")
plt.plot(reconstructed, label="Reconstructed data")
plt.legend(loc="lower left")
plt.show()
```

iii. Code Description

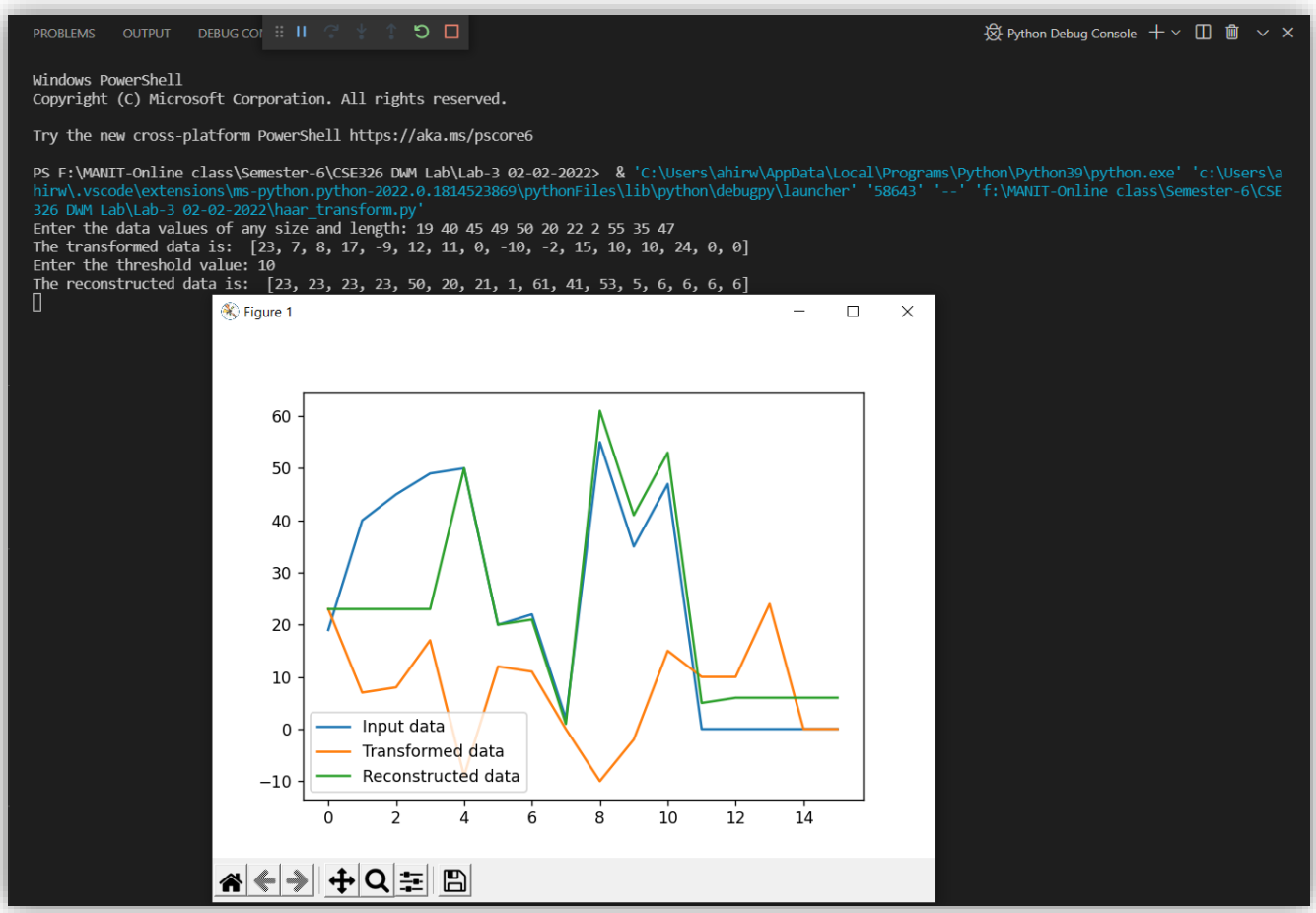
First, we have a utility function "***power2***" which takes an integer (actually the length of the data list) and returns the smallest integer which is a power of 2 and greater than or equal to the input argument x. This function will be used by the transformation functions to validate the length of the input list and append 0s if necessary.

The next function is "***transform***" which takes input the data list and returns a new list after performing the HAAR wavelet transformation in the data. We have implemented a function inside here, "***util***", which is a recursive function. It takes input the current state of the data and length (that is, the segment till which transformation has to be performed). At each step, it performs transformation suppose on length l, and calls itself again but this time only for the first half length. The base case is when the length reaches 1. It returns the complete array all the way to the calling statement.

Now, the user is prompted for the threshold value to be used for the reconstruction of the data. The inverse function is about the same. To reconstruct we first filter out the lists with the threshold value. The "***inverseUtil***" function is implemented to perform the reconstruction recursively. The methodology here is reverse of what was done in transformation. Here, we take 1 length data and move towards the full length. At each step we reconstruct the data by adding and subtracting values and positioning them at their respective locations. When, the length reaches the full array length. The new array is then returned all the way to the calling statement.

To plot these 3 lists, we are using the matplotlib library. It provides the function "***plot***" and "***show***" to easily and effectively perform plotting of data on Jupyter notebooks and normal python programs.

iv. Output


PROBLEMS
OUTPUT
DEBUG CO
Python Debug Console
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS F:\MANIT-Online class\Semester-6\CSE326 DWM Lab\Lab-3 02-02-2022> & 'C:\Users\ahirw\AppData\Local\Programs\Python\Python39\python.exe' 'c:\Users\ahirw\.vscode\extensions\ms-python.python-2022.0.1814523869\pythonFiles\lib\python\debugpy\launcher' '58643' '--' 'f:\MANIT-Online class\Semester-6\CSE326 DWM Lab\Lab-3 02-02-2022\haar_transform.py'
Enter the data values of any size and length: 19 40 45 49 50 20 22 2 55 35 47
The transformed data is: [23, 7, 8, 17, -9, 12, 11, 0, -10, -2, 15, 10, 10, 24, 0, 0]
Enter the threshold value: 10
The reconstructed data is: [23, 23, 23, 23, 50, 20, 21, 1, 61, 41, 53, 5, 6, 6, 6, 6]
Figure 1
60
50
40
30
20
10
0
−10
0
2
4
6
8
10
12
14
Input data
Transformed data
Reconstructed data