International Institute of Information Technology Bangalore

Basic Computational Topology
SM 402

BCT Implementation Assignment

April 26, 2022

Group 7
Aryan Bhatt (IMT2020020)
Balaji Sankapal (IMT2020090)
Anmol Shetty (IMT2020551)
Abhinav Mahajan (IMT2020553)

Problem Statement P7

Given a volumetric scalar data, compute a sub-level set corresponding to a given scalar value. Integrate with P1 and compute β_0 for an increasing sequence of scalar values.

Algorithm

- For a given data-set with length d_x, breadth d_y and height d_z, if $f(i,j,k) \leq c$ for a given threshold value of c it is considered as a vertex in the sub-level set.
- For every point (i,j,k), if either one of the neighbouring points $(i+1,j,k),(i,j+1,k),(i,j,k+1)$ belong to sub-level set, it is considered as an edge and the edge values are stored as a single unique number.
- To map (i,j,k) to a single number we use $i + jd_x + kd_xd_y$. Suppose the two edge numbers are x,y then they are stored as an array with opposite parity.
- Finally β_0 is calculated by taking the difference of the number of vertices with the rank of the above matrix.

  We have used the formula given below in our code to calculate β_0:

  $$\beta_0 = dim(H_0(K)) = dim(C_0(K)) - dim(Im(\partial_1)) \tag{1}$$

  In our program we obtain the vertices and edges as input and create a matrix ∂_1 and then compute $dim(Im(\partial_1))$ using the python matrix operations.

  As we also know that $C_0(K)$ is the vector space of 0-chains $\implies dim(C_0(K)) \implies$ Total No of vertices.
- By using all the above facts we can easily calculate β_0 for a simplicial complex.

Implementation Steps

1. First we ask for filename as input from user (.raw file). From here we generate vertices and edges from the above algorithm and also $dim(C_0(K))$.
2. In the next step we create the matrix corresponding to ∂_1 using the edges of the input.
3. In the last and the final step we calculate β_0 using 1 and using the knowledge of the total count of the vertices.

Steps to run the code

1. Open the terminal.
2. Enter the commmand **pip3 install numpy**.
3. Enter the command **python3 temp1.py**.
4. Enter the filename of the *.raw* file you want to take input from.
5. Specify the threshold value to get different values of β_0.
6. Press Enter to get the final result.

Python Code

Python code for calculating β_0 for increasing values of c

```
import numpy as np
import time
begin=time.time()
#INPUT - file
filename = "marschner_lobb_41x41x41_uint8.raw.txt"
# Hardcoding the dimensions
dim_x, dim_y, dim_z = 41, 41, 41
#Number of Vertices -> Mapping
v = dim_x*dim_y*dim_z
#Number of EDGES in SUB LEVEL SET
e = 0
# A is array of Mapped Inputs
A = np.fromfile(filename, dtype='uint8', sep="")
print("The Total Number of Vertices in raw file are :")
print(len(A))
A = A.reshape((dim_x, dim_y, dim_z))
#Taking C value
c=int(input("Enter the c value: "))
# Number of VERTICES in SUB LEVEL SET
a=0

#Traversal
for i in range(0,dim_x):
    for j in range(0,dim_y):
        for k in range(0,dim_z):
            if(A[i][j][k]<=c):
                a+=1
                if(i+1 < dim_x):
                    if(A[i+1][j][k]<=c):
                        e+=1

                if(j+1 < dim_y):
                    if(A[i][j+1][k]<=c):
                        e+=1

                if(k+1<dim_z):
                    if(A[i][j][k+1]<=c):
                        e+=1
print("The Total Number of Vertices in Sub-Level Set are :")
print(a)
print("The Total Number of Edges in Sub-Level Set are :")
print(e)

#For delta 1 matrix
t=0
k=0
arr=[ [0] * int(e) for i in range(int(v))]
for i in range(0,dim_x):
    for j in range(0,dim_y):
        for k in range(0,dim_z):
            if(A[i][j][k]<=c):
                if(i+1 < dim_x):
                    if(A[i+1][j][k]<=c):
                        x=i+j*dim_x+k*dim_x*dim_y
                        y=(i+1)+j*dim_x+k*dim_x*dim_y
                        arr[x][t]=-1
                        arr[y][t]=1
                        t+=1

                if(j+1 < dim_y):
                    if(A[i][j+1][k]<=c):
                        x=i+(j)*dim_x+k*dim_x*dim_y
                        y=(i)+(j+1)*dim_x+k*dim_x*dim_y
                        arr[x][t]=-1
                        arr[y][t]=1
                        t+=1
                if(k+1<dim_z):
                    if(A[i][j][k+1]<=c):
                        x=i+(j)*dim_x+k*dim_x*dim_y
                        y=(i)+(j)*dim_x+(k+1)*dim_x*dim_y
                        arr[x][t]=-1
                        arr[y][t]=1
                        t+=1
```

```
74
75
76 rank=np.linalg.matrix_rank(arr)
77 print("Rank: " + str(rank))
78 betti_0 = a - rank
79
80 print("==========")
81 print('| \N{GREEK SMALL LETTER BETA}\N{SUBSCRIPT ZERO} =',betti_0,'|')
82 print("==========")
83 end=time.time()
84 print("Total execution time(in sec) = ",end-begin)
```

Examples

```
(base) balajibabasahebsankapal@Balajis-MacBook-Air Topo % python3 p7.py
The Total Number of Vertices in raw file are :
68921
Enter the c value: 1
The Total Number of Vertices in Sub-Level Set are :
536
The Total Number of Edges in Sub-Level Set are :
444
Rank: 372
==========
| β₀ = 164 |
==========
```

Figure 1

```
(base) balajibabasahebsankapal@Balajis-MacBook-Air Topo % python3 p7.py
The Total Number of Vertices in raw file are :
68921
Enter the c value: 5
The Total Number of Vertices in Sub-Level Set are :
1456
The Total Number of Edges in Sub-Level Set are :
1562
Rank: 1222
==========
| β₀ = 234 |
==========
(base) balajibabasahebsankapal@Balajis-MacBook-Air Topo %
```

Figure 2

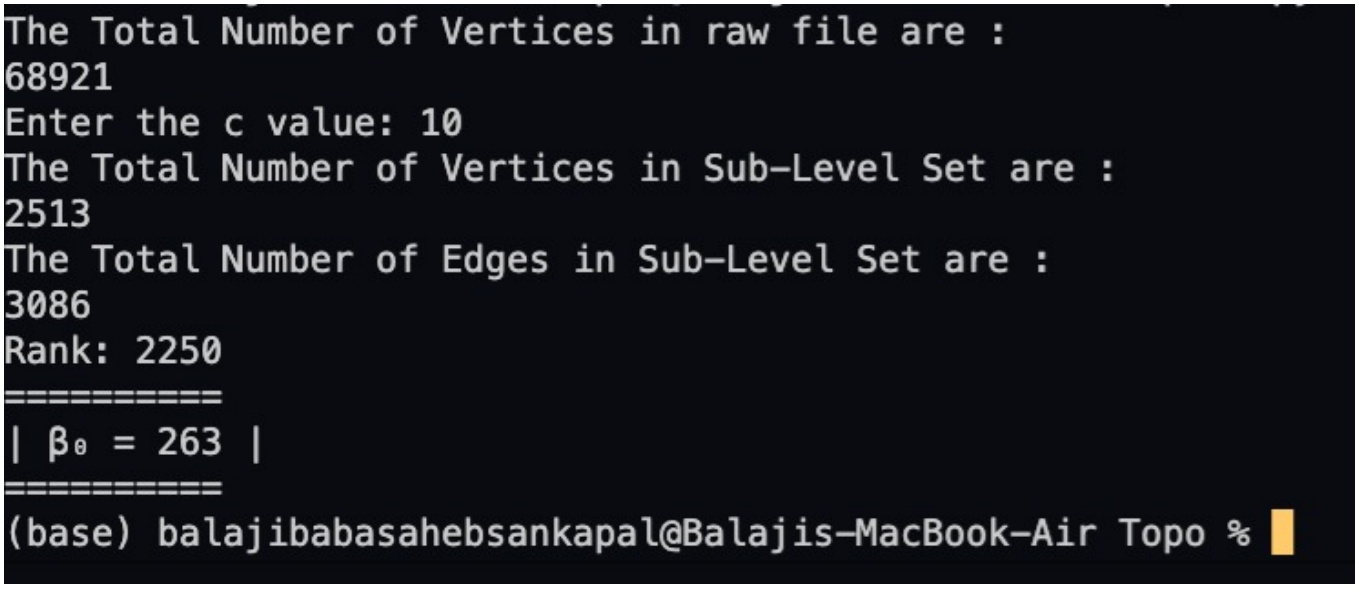

Figure 3

Note: The python script includes extra functionality of the run time of the code for a particular *.raw* file.

GitHub Link

Please visit this for the source code.
https://github.com/Anmol-S314/Topo-Proj-P7