

Finding the Missing Links

Asodariya Harsh (Entry no: 2023csb1103)

1 Introduction

At first, we have to create a graph from the file, and then we have to create an adjacency matrix from that graph of 0's and 1's using numpy. Now consider 0 at (i, j) , it will represent no direct edge from i to j and vice versa if 1 is there. Note '0' has 2 meanings:

1. The i th person is not impressed with the j th person.
2. The i th person never met j th person.

We will work on case 2, as many pairs didn't even talk to each other as people tend to remain in a friend circle. So, we will iterate through all the zeros in the given adjacency matrix and use the matrix method to find whether the link will exist or not.

2 Logic

Let's say there is '0' at position (i, j) ; after that, we will remove both the i th row and j th column from the matrix, and let's name it A .

$$\text{Dim}(A) = (n - 1) \times (n - 1)$$

Now let's take coefficient matrix $X = [x_1, x_2, x_3, \dots, x_{(n-1)}]$ and that i th row as R but without that '0' since we have to find that zero in the linear combination of that corresponding j th column matrix and take j th column as matrix C .

$$\text{Dim}(X) = (1) \times (n - 1)$$

$$\text{Dim}(R) = (1) \times (n - 1)$$

$$\text{Dim}(C) = (n - 1) \times (1)$$

We know, $value = [R][C]$. This value will tell us whether the link will exist or not, I have taken a threshold value of 0.5. If the value is greater than 0.5, then the link will exist from i to j and substitute 0 with 1.

To find X , we have to multiply X with A to get R .

$$[X][A] = [R]$$

Now in numpy, we can multiply the unknown matrix on the left side. So, we will take the transpose on both sides and using linear regression and *lstsq* function we can get a close to perfect value of X .

$$([X][A])^T = [B]^T$$

```
X = np.linalg.lstsq(small_matrix.T, row_mat.T, rcond=None)[0]
```

By this, we can get the X matrix. Therefore, $value = [X][C]$. So, this is how we will get the value of a missing link.

3 Code

1. I have created a graph from the file and found the pagerank before transformation and after transformation.
2. Defining the main function to get the missing link.
3. Using a nested loop, I am iterating every element in the given code.
4. If any entry is 0, then I apply the logic as mentioned earlier to find the actual value.
5. After the loop, I'm just rechecking my PageRank.

4 Observation

I am finding subtle differences in the pagerank of the top 10 individuals, which confirms the accuracy of the logic and identifies several missing links that were not present in the original graph.