CSC 466 Lab 7: PageRank

Kavya Karunakaran (kkarunak@calpoly.edu)

Implementation Overview

In order to implement the PageRank algorithm, I began by creating an adjacency matrix of the pages to represent the data as a graph. The adjacency matrix has size n x n, where n is the number of pages, and I chose to store it in a two-dimensional numpy array. An edge or link is represented with a value of 1 in the matrix, where the row represents the page on which the link is located and the column represents the page to which the link leads.

I then iterated through the PageRank algorithm, keeping track of the current pagerank scores in a dictionary mapping the page name to its score. In order to compute the term of the formula dealing with pages that link to the current page, I used the numpy array of the values of the column corresponding to the current page in the adjacency matrix. Then for each of those values, I computed the number of pages outgoing from the page by taking the sum of the row corresponding to the page. After each computation of a pagerank score, I took the difference between the new score and the page's previous score before I updated the score dictionary, and I used this difference to control the number of iterations.

Results

NCAA Football

I ran this dataset with an epsilon value of 0.0001 and a d value of 0.1.
Output:

```
1 North Dakota with pagerank: 0.002539872408293461
2 Weber State with pagerank: 0.002514809751651857
3 Montana with pagerank: 0.0024866142629300526
4 South Dakota with pagerank: 0.0023652882205513785
5 Northwestern State with pagerank: 0.002262426900584795
6 Florida with pagerank: 0.002194548872180451
7 Iona with pagerank: 0.0021929824561403508
8 Richmond with pagerank: 0.0021691381230854916
9 Utah with pagerank: 0.0021557365636313006
10 Oklahoma with pagerank: 0.0021273828510670616
11 Central Arkansas with pagerank: 0.002122145641882484
12 Sacred Heart with pagerank: 0.0021023201944254575
```

```
13 Savannah State with pagerank: 0.002101608187134503
14 Texas Tech with pagerank: 0.0020835232019442546
15 Texas with pagerank: 0.002066624895572264
```

I found the rankings on this dataset interesting because the rankings do not change much from the top rank to the bottom rank (not pictured here). For example, the highest rank, 0.0025 for North Dakota, is close in value to the 15th highest rank for Texas, even after multiple iterations. There were also many items in this dataset that yielded the same pagerank score. This suggests that PageRank may not have discovered the proper ranking of the items.

Dolphins

I ran this dataset with an epsilon value of 0.0001 and a d value of 0.5.
Output:

```
1 Trigger with pagerank: 0.02964391464013793
2 Jet with pagerank: 0.02873013049035356
3 Web with pagerank: 0.026117523018141686
4 Grin with pagerank: 0.024612157352501166
5 Scabs with pagerank: 0.02412458562036102
6 Patchback with pagerank: 0.023661042171934682
7 SN4 with pagerank: 0.023150131533344488
8 Topless with pagerank: 0.02304818351495977
9 SN63 with pagerank: 0.022239423190729288
10 Gallatin with pagerank: 0.021456651272487186
11 Beescratch with pagerank: 0.020953202227378225
12 Kringel with pagerank: 0.02043710938810147
13 Stripes with pagerank: 0.0201933316950657
14 Feather with pagerank: 0.02004877363335504
15 SN100 with pagerank: 0.01893941629482202
```

The rankings decreased more drastically from rank 1 to rank 15 for this dataset, as shown with the value at rank 1, 0.029, and the value at rank 15, 0.0189. Additionally, as the number of iterations increased, the pagerank scores had large changes. This suggests that the PageRank algorithm honed in on the correct values.

Les Mis

I ran this dataset with an epsilon value of 0.0001 and a d value of 0.9.
Output:

```
1 Valjean with pagerank: 0.07572356170960227
2 Myriel with pagerank: 0.039654962775574926
```

```
3 Gavroche with pagerank: 0.03741191833380349
4 Marius with pagerank: 0.03240121099669431
5 Javert with pagerank: 0.03135980598048133
6 Thenardier with pagerank: 0.02886104660002371
7 Fantine with pagerank: 0.02776773588502783
8 Enjolras with pagerank: 0.023511877322317538
9 Cosette with pagerank: 0.02099932031217025
10 Bossuet with pagerank: 0.020293856839998115
11 MmeThenardier with pagerank: 0.02009512441629902
12 Courfeyrac with pagerank: 0.019951633776383308
13 Eponine with pagerank: 0.01866207198242834
14 Bahorel with pagerank: 0.018424060551548858
15 Joly with pagerank: 0.018423807125602588
```

Similarly to the dolphins dataset, the pagerank scores decreased drastically, going from 0.075 at rank 1 to 0.018 at rank 2. This distinction between the pages may suggest that PageRank discovered the proper ranking of pages.

Overall Summary

PageRank works by taking into account the probability of a user clicking on pages with equal probability or by clicking through pages randomly, and using those factors to compute an overall probability of the user reaching a page. This implementation of PageRank produced good rankings for all three of the datasets. Based on my observations and conclusions in the Results section, I would say that the algorithm worked the best with datasets that produced undirected graphs, as shown by the results of the Les Miserables dataset. However, the algorithm did still yield good results for the datasets that yielded directed graphs.

Performance Evaluation

The table below shows the results of running PageRank for each dataset with different d values, and the number of iterations, processing time, and read time for each run. The general trend was that as the d value was increased, the processing time and number of iterations increased. The processing and read times were fairly low for the given datasets, so I would predict that the implementation would scale well to larger graphs.

Dataset	Epsilon Value	d value	Iterations	Processing Time	Read Time
NCAA Football	0.001	0.01	2	0.0401 sec	0.0713 sec

	0.001	0.1	3	0.0703 sec	0.0814 sec
	0.001	0.5	3	0.0556 sec	0.0699 sec
	0.001	0.75	3	0.0566 sec	0.0694 sec
	0.001	0.9	3	0.0556 sec	0.0700 sec
Dolphins	0.001	0.1	3	0.0021 sec	0.0178 sec
	0.001	0.5	7	0.0049 sec	0.0181 sec
	0.001	0.75	13	0.0092 sec	0.0179 sec
	0.001	0.9	29	0.0201 sec	0.0175 sec
Les Miserables	0.001	0.1	3	0.0033 sec	0.0247 sec
	0.001	0.5	7	0.0076 sec	0.0258 sec
	0.001	0.75	14	0.1573 sec	0.0262 sec
	0.001	0.9	32	0.0390 sec	0.0293 sec

Appendix: README

The files needed are pageRank.py and the data files (dolphins.csv, lesmis.csv, NCAA_football.csv)
The Page Rank program can be run with the following command:

```
python pageRank.py <filename>
```

To change the d-value to run with different datasets, modify line 6 of the main method to pass in the corresponding d value as the final parameter to the call to pageRank.
For example, to run NCAA Football with d value 0.1, modify line 6:

```
scores, node_scores, iterations, process_time =
pageRank(adjacency_matrix, adjacency_indices, 0.1)
```