CSC 466 Lab 7 Report: Applications of PageRank Analysis on Various Graph Representations

Martin Solomon Hsu: mshsu@calpoly.edu
Instructor: Dr. Alexander Dekhtyar
CSC 466: Knowledge Discovery from Data, Fall 2023

Abstract

In this lab, we will investigate PageRank graph traversal, scoring, and analysis as applied to three different datasets representing graphs: dolphin social networks, character co-appearances in *Les Miserables*, and NCAA football matches. After cleaning the datasets, constructing proprietary graph representations, and using matrix algebra principles to apply PageRank, we constructed rankings of each node. With these rankings, we drew inferences on individual node popularity, importance, and team quality respectively, and then analyzed these results and inferences for both correctness using domain knowledge and research, and performance in terms of speed and efficiency.

I. Introduction

PageRank is an innovative algorithm developed by Larry Page and Sergey Brin at Google in the 1990s.[1] This method of ranking site popularity remains a powerful method for investigating node importance in directed and undirected graphs. This lab report explores the PageRank method within the context of different datasets that are representations of directed or undirected graphs observable in different real world settings, through associative social networks or win-lose team dynamics.

II. Dataset Description

We utilized three different datasets. Each dataset is represented in tabular form, where each row is a pair of nodes connected by an edge. The edge can be directed, in which case a nonzero value will be associated with each node. The parent node, or the node from which the edge is outgoing, is the node with the lower value; and the child node. Or the node to which the edge is incoming, is the node with the higher value.

A. DOLPHINS (dolphinsDir.csv)

The Dolphins dataset represents "an undirected social network of frequent associations between 62 dolphins." which are specific to a pod in Doubtful Sound, New Zealand. The data were

[1] Sergey Brin, Lawrence Page, Reprint of: The anatomy of a large-scale hypertextual web search engine, Computer Networks, Volume 56, Issue 18, 2012, Pages 3825-3833, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2012.10.007.

collected and published in 2003 by ecologist Dr. David Lusseau.[2] Since this graph is undirected, each edge is only represented once, with each edge representing a symmetric association between dolphins. In this investigation, we want to utilize PageRank to understand which of the 62 dolphins is the most "popular," with "popular" being defined as "having the most associations with other individuals in the pod."

B. LES-MISERABLES (lesmisDir.csv)

The Les-Miserables file represents the undirected "weighted network of appearances of characters in Victor Hugo's novel 'Les Miserables.'" Each row represents an undirected edge between two nodes, where each node is a character in the novel (e.g. Jean Valjean), with an edge existing between two character nodes if they have appeared together at least once. Since this graph is undirected, each edge is only represented once, with each edge representing a co-appearance of two characters. This dataset was published in 1993 on *The Stanford GraphBase*.[3] Much like the Dolphins dataset, we want to investigate character "importance" using PageRank, with the assumption that characters that co-appear with the most other characters are more important, and characters that co-appear the least with other characters are less important to the novel.

C. NCAA-FOOTBALL (NCAA_football.csv)

The NCAA-Football dataset is a directed graph representing matches played between two NCAA football teams in the 2009 regular season. Each node is a single NCAA football team. As the graph is directed, the parent node is determined by the team with the lower score, and the child node is determined by the team with the higher score. In other words, each edge is a football match, with the parent node being the losing team and the child node being the winning team. The goal of applying PageRank to this dataset is to determine which team tends to win more often than the other teams, and thus rank the teams in order of "quality."

III. Methods

To apply PageRank to the data, we constructed a proprietary implementation within Python 3.10. The runtime of each step was also measured using the time tool, in addition the number of iterations it took for the PageRank algorithm to converge.

Importing and Cleaning Data

Using data management packages such as Pandas and Numpy, we imported the graph dataset and removed unnecessary columns. String and integer representations were cleaned of any unnecessary characters and converted to the proper data format. The graph was then transformed from a tabular representation of nodes to an adjacency list. This was represented by a dictionary,

[2] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations", *Behavioral Ecology and Sociobiology* 54, 396-405 (2003).

[3] D. E. Knuth, *The Stanford GraphBase: A Platform for Combinatorial Computing*, Addison-Wesley, Reading, MA (1993).

where each key was a hub node. We defined a hub node as any node with outgoing edges, also known as a parent node. In an undirected graph, any node connected to an edge is a hub node. Each key contained a list of the hub node's children.

PageRank Implementation

We used the following recursive matrix definition of PageRank as our final implementation:

$$\mathbf{A}\mathbf{p} + (1 - d) / n = \mathbf{p}$$

The constant d represents the probability of traversing any given adjacent edge from a node in a PageRank traversal, versus jumping spontaneously to another node (represented by the probability $1 - d$). The variable n represents the number of nodes in the graph.

A is a modified n x n square adjacency/transition matrix, with each row representing a node as a potential child, and each column representing a node as a potential hub or parent. The value of each entry, a_{ij}, was the probability of traveling along the edge from the parent node i to the child node j, calculated by dividing d by the outdegree of the parent node, d_i:

$$a_{ij} = \begin{cases} d / d_i & \text{if node } j \text{ is a child of node } i, \\ 0 & \text{otherwise} \end{cases}$$

This matrix was populated by iterating over each child of each "hub" or parent in the adjacency list.

The vector **p** represents the vector of PageRank scores/probabilities, which we sorted in descending order to obtain our final rankings.

After constructing **A**, we initialized the vector **p** to a vector of $1 / n$, and iteratively applied the operation $\mathbf{A}\mathbf{p} + (1 - d) / n$ to the vector until the vector **p** converged upon a relatively unchanging solution, within a margin of error of 1.0e-08. The constant d was arbitrarily assigned four different values in four trials; 0.25, 0.5, 0.75, and 1.00; to test the robustness of the rankings against a changing probability of edge traversal. Output for $d = 0.75$ is found in the appendix.

IV. Results

A. DOLPHINS

As seen in Table 4.1, we can observe that the rankings stay relatively stable as d, the probability of traversing an edge rather than jumping to a random node, increases. The dolphins that are consistently ranked highest by number of associations include Trigger, Jet, Web, Grin, and Scabs. From these PageRank results, we can infer that many dolphins associate with these few compared to other dolphins.

Table 4.1: DOLPHINS Graph PageRank Results

d	*Data Read Runtime (ms)*	*PageRank Runtime (ms)*	*Iterations*	*Rank 1*	*Rank 2*	*Rank 3*	*Rank 4*
0.25	151.618	24.798	8	Trigger (0.0247)	Jet (0.0236)	Web (0.0219)	Scabs (0.0206)
0.50	132.779	23.139	12	Trigger (0.0296)	Jet (0.0287)	Web (0.0261)	Grin (0.0246)
0.75	157.371	44.193	21	Jet (0.0315)	Trigger (0.0314)	Grin (0.0297)	Web (0.0292)
1.00	193.356	91.075	162	Grin (0.0377)	SN4, Topless (0.0346)	Scabs, Trigger (0.0314)	Web, Jet, Kringel (0.0283)

Interestingly, more ties between dolphin nodes appear in the ranking as *d* increases, particularly when *d* = 1.

B. LES-MISERABLES

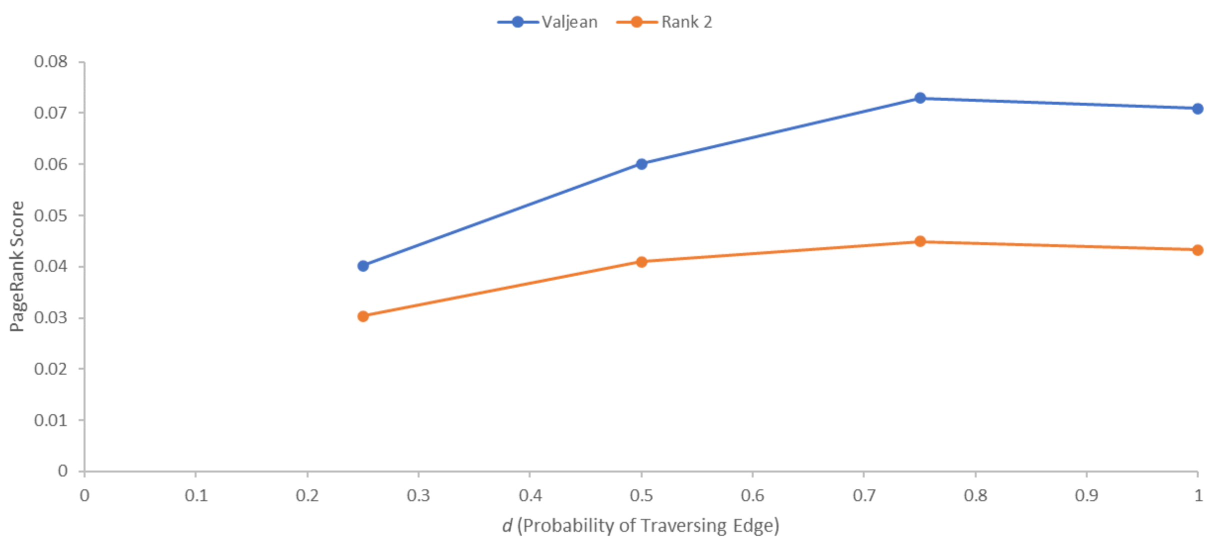
We can observe a similar phenomenon to the Dolphins graph in the Les-Miserables graph, as seen in Table 4.2. Valjean consistently appears as the top character in terms of co-appearances according to the PageRank algorithm, followed by some combination of Myriel, Gavroche, Javert, and Marius. Similarly to the Dolphins dataset, the number of iterations needed for PageRank to converge increases at a growing rate as *d* increases.

Table 4.2: LES-MISERABLES Graph PageRank Results

d	*Data Read Runtime (ms)*	*PageRank Runtime (ms)*	*Iterations*	*Rank 1*	*Rank 2*	*Rank 3*	*Rank 4*
0.25	220.894	40.100	10	Valjean (0.0402)	Myriel (0.0304)	Gavroche (0.0197)	Javert (0.0182)
0.50	229.809	58.626	16	Valjean (0.0601)	Myriel (0.0410)	Gavroche (0.0267)	Javert (0.0232)
0.75	247.714	60.983	30	Valjean (0.0729)	Myriel (0.0449)	Gavroche (0.0330)	Marius (0.0284)
1.00	240.632	136.651	104	Valjean (0.0709)	Gavroche (0.0433)	Marius (0.0374)	Javert (0.0335)

Intuitively, it makes sense that Jean Valjean overwhelmingly captures the first rank position in PageRank, as he is the primary protagonist of *Les Miserables*. Marius and Javert, as secondary protagonist and antagonist respectively, also make sense as topping the rankings. Gavroche, as a supporting character who commonly appears accompanying the French revolutionary characters in the plot, also may make sense as being a character with a high number of co-appearances, even though he may not necessarily be as influential to the plot as other characters topping the rankings.

Figure 4.2: PageRank Score of Valjean Compared to 2nd Rank Character, as a Factor of d



Interestingly, as *d* increases, two phenomena occur. Firstly, the disparity between primary protagonist Valjean's PageRank score and the other PageRank scores tends to increase, as seen in Figure 4.2. Secondly, between $d = 0.75$ and $d = 1.00$, Bishop Myriel's ranking drops from second to 18th. This may also make sense intuitively, as Myriel plays an influential role in the book on Jean Valjean's life, but when evaluated on pure co-appearances by setting $d = 1.00$, he is not as prevalent of a character.

C. NCAA-FOOTBALL

The PageRank analysis for the NCAA-Football graphs demonstrates significant volatility, with Mississippi, Florida, and Oklahoma consistently entering the top four rank positions. Additionally, as *d* increases, the PageRank score appears to approach 0.

Table 4.3: NCAA-FOOTBALL Graph PageRank Results

d	*Data Read Runtime (ms)*	*PageRank Runtime (ms)*	*Iterations*	*Rank 1*	*Rank 2*	*Rank 3*	*Rank 4*
0.25	868.828	146.919	9	Montana (0.0054)	Weber State (0.0051)	Florida (0.0051)	Richmond (0.0051)
0.50	777.636	116.965	15	Mississippi (0.0098)	Florida (0.0090)	Oklahoma (0.0083)	Richmond (0.0083)
0.75	1025.374	167.216	31	Mississippi (0.0021)	Florida (0.0017)	Oklahoma (0.0013)	Utah (0.0013)
1.00	1061.579	300.504	382	Mississippi (2.90e-7)	Florida (2.26e-7)	Wake Forest (1.54e-7)	Alabama (1.34e-7)

This ranking is interesting, but does not make much sense in terms of traditional preseason NCAA evaluations[4]. While some teams, such as Florida and Utah, make sense when they appear at the top of the rankings in this context, some universities, such as Montana, Weber State, Richmond, and Wake Forest, do not appear at all in traditional 2009 rankings. Additionally, Mississippi is not usually polled consistently as the number one ranking team, though it certainly ranks high in general in the preseason.

An explanation for this ranking may come from the ratio of outgoing versus incoming edges. It could be that though the top rankings in these schools are not as athletically rigorous compared to schools topping traditional rankings, they win much more games than they lose due to the nature of their regional division. Schools such as Montana, Richmond, and Utah have a particularly high count of wins compared to their losses. This underscores the importance of an approach that takes into account more than simply the number of wins or losses in determining college football rankings.

V. Conclusions

Overall, PageRank produced acceptable rankings for the three datasets. PageRank worked especially well when utilized for the context of popularity or importance for undirected graphs of social networks and associations between individuals, as observed in the Dolphins and

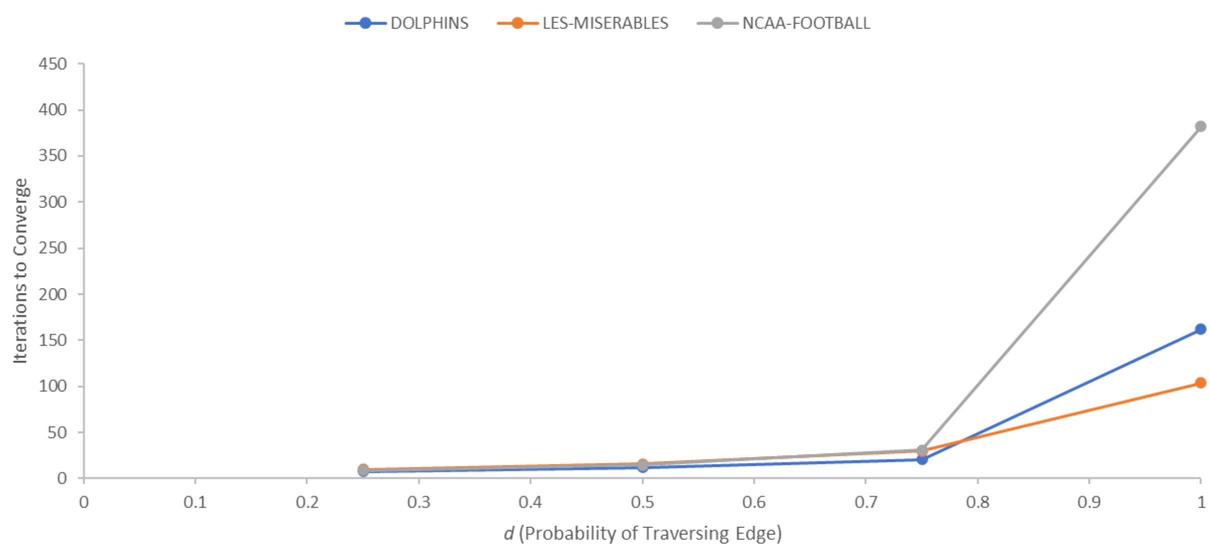
[4] https://web.archive.org/web/20090919022544/http://www.appollarchive.com/football/ap/seasons.cfm?seasonid=2009

Les-Miserables analyses. In directed graphs, especially the NCAA-Football dataset, where performance-based rankings can depend on more free parameters than simply the wins and losses used to determine direction in the graph, PageRank did not perform as well as widely accepted rankings. A more effective approach may involve weighting the probability of traversing one of the outgoing edges by a value factor, such as the number of co-appearances in the Les-Miserables graph or the winning score in the NCAA graph.

VI. Performance

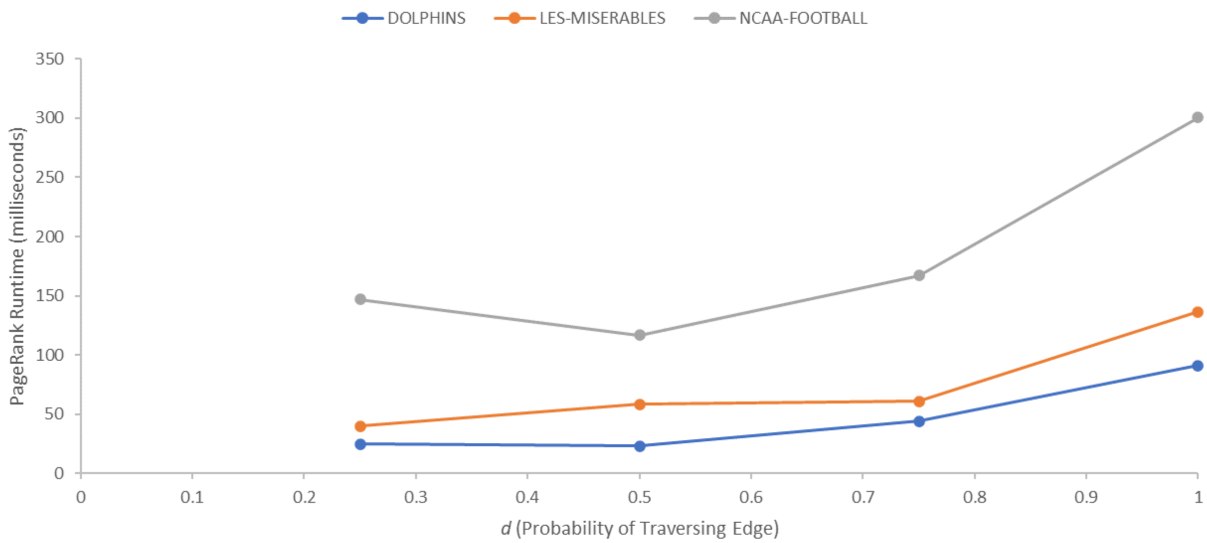
When examining the effect of a changing *d* on the number of iterations and the runtime, we can see that overall, the algorithm runs relatively quickly and converges in a few iterations. However, as *d* increases, the number of iterations needed for the algorithm to converge on a solution appears to tend to increase at an increasing rate, as seen in Figure 5.1.

Figure 5.1: PageRank Iterations before Convergence, as a Factor of d



This makes sense intuitively; when $d = 0$, the solution should instantly converge mathematically on a **p** vector where each element is equal to $1 / n$. As d increases, the $1 / n$ component decreases in influence, resulting in a solution that takes longer to converge upon. This phenomenon is reflected in the runtime, as a similar pattern can generally be found when examining runtime as a factor of *d*. As expected, overall average runtime also consistently increases relative to the size of the dataset.

Figure 5.2: PageRank Runtime, as a Factor of d



From this pattern we can make two inferences; firstly, that as *d* increases, PageRank's runtime and number of iterations also increase. Secondly, the runtime consistently increases as graph size increases as well, though the number of iterations does not consistently follow this pattern, especially with a lower *d* value. As a result, we may expect that as the number of nodes or edges increases in the graph, the runtime and number of iterations can be expected to increase. However, as demonstrated by implementations such as Google, the algorithm has a proven scalability across millions of nodes.

VII. Appendix

Please see README file for Python PageRank implementation details. Output for *d* = 0.75 PageRank implementation is shown below and is considered to be the best or most balanced PageRank implementation out of the four tested (*d* = 0.25, 0.50, 0.75, 1.00).

A. DOLPHINS PageRank Output, *d* = 0.75

See dolphins_results_0.75.txt for full output.

```
Page Rank, d=0.75

File: dolphinsDir.csv
Read Time:        157.371ms
Processing Time: 44.193ms
N Iterations:     21
```

Rank	Node	PageRank
1	Jet	0.031487
2	Trigger	0.031429
3	Grin	0.029726
4	Web	0.029234
5	SN4	0.027749
6	Topless	0.027420
7	Scabs	0.027207
8	Patchback	0.025824
9	Gallatin	0.024888
10	SN63	0.023611
11	Beescratch	0.023555
12	Kringel	0.023209
13	Feather	0.022597
14	Stripes	0.021516
15	SN9	0.020710
16	Upbang	0.020569
17	SN100	0.020054
18	DN21	0.019510
19	Haecksel	0.019307
20	Jonah	0.018494
21	TR99	0.018251
22	SN96	0.017351
23	Number1	0.017210
24	TR77	0.016961
25	Double	0.016689
26	Beak	0.016462
27	MN105	0.016325
28	MN83	0.016319
29	Shmuddel	0.016156
30	Hook	0.015926
31	SN90	0.015635
32	DN63	0.015503
33	PL	0.015298
34	Fish	0.015149
35	Zap	0.014753
36	DN16	0.014682
37	Oscar	0.014676
38	Ripplefluke	0.014537
39	Bumper	0.013965
40	Thumper	0.013222
41	Knit	0.013065
42	Mus	0.012206
43	TSN103	0.012197
44	Zipfel	0.012049
45	Notch	0.011874
46	MN60	0.010408
47	TR88	0.010277
48	TR120	0.010192
49	CCL	0.010189
50	TSN83	0.009350
51	Wave	0.009224
52	SN89	0.008617
53	Vau	0.008458
54	Zig	0.007666
55	Quasi	0.006656
56	MN23	0.006656
57	TR82	0.006468
58	Cross	0.006389
59	Five	0.006389
60	Whitetip	0.006246
61	SMN5	0.006184
62	Fork	0.006073

B. LES-MISERABLES PageRank Output, *d* = 0.75

See lesmis_results_0.75.txt for full output.

```
Page Rank, d=0.75

File: lesmisDir.csv
Read Time:       247.714ms
Processing Time: 60.983ms
N Iterations:    30
```

Rank	Node	PageRank
1	Valjean	0.072890
2	Myriel	0.044922
3	Gavroche	0.033045
4	Marius	0.028404
5	Javert	0.028231
6	Thenardier	0.026299
7	Fantine	0.025619
8	Cosette	0.019753
9	Enjolras	0.019586
10	MmeThenardier	0.018498
11	Bossuet	0.017178
12	MlleGillenormand	0.017090
13	Courfeyrac	0.016810
14	Mabeuf	0.016803
15	Eponine	0.016570
16	Bahorel	0.015671
17	Joly	0.015671
18	Babet	0.015475
19	Gueulemer	0.015475
20	Gillenormand	0.015350
21	Claquesous	0.015338
22	Bamatabois	0.015007
23	Tholomyes	0.014995
24	Feuilly	0.014557
25	Combeferre	0.014557
26	Montparnasse	0.014192
27	Grantaire	0.013380
28	Fauchelevent	0.012968
29	Dahlia	0.012443
30	Favourite	0.012443
31	Listolier	0.012443
32	Fameuil	0.012443
33	Blacheville	0.012443
34	Zephine	0.012443
35	Brevet	0.012344
36	Cochepaille	0.012344
37	Chenildieu	0.012344
38	Judge	0.012344
39	Champmathieu	0.012344
40	Prouvaire	0.012261
41	Brujon	0.011390
42	MmeMagloire	0.010846
43	MlleBaptistine	0.010846
44	MmeHucheloup	0.010276
45	Simplice	0.009669
46	MmeBurgon	0.009472
47	LtGillenormand	0.009190
48	Pontmercy	0.008281
49	Toussaint	0.007358
50	Woman2	0.007358
51	MotherInnocent	0.007197
52	MmePontmercy	0.007148
53	Child1	0.006997
54	Child2	0.006997
55	Anzelma	0.006870
56	Jondrette	0.006799
57	Count	0.006616
58	CountessDeLo	0.006616
59	OldMan	0.006616
60	Geborand	0.006616
61	Cravatte	0.006616
62	Napoleon	0.006616
63	Champtercier	0.006616
64	Perpetue	0.006341
65	Magnon	0.006153
66	Marguerite	0.006046
67	BaronessT	0.006013
68	Woman1	0.006011
69	Gribier	0.005678
70	MlleVaubois	0.005078
71	MmeDeR	0.004765
72	Scaufflaire	0.004765
73	Gervais	0.004765
74	Labarre	0.004765
75	Isabeau	0.004765
76	Boulatruelle	0.004480
77	MotherPlutarch	0.004392

C. NCAA-FOOTBALL PageRank Output, *d* = 0.75, top 100 only

See NCAA_football_results_0.75.txt for full output.

```
Page Rank, d=0.75

File: NCAA_football.csv
Read Time:        1025.374ms
Processing Time: 167.216ms
N Iterations:     31
```

Rank	Node	PageRank
1	Mississippi	0.021463
2	Florida	0.017664
3	Oklahoma	0.013269
4	Utah	0.013219
5	Texas Tech	0.013076
6	Richmond	0.011188
7	Texas	0.011103
8	Oregon State	0.010744
9	Montana	0.010439
10	James Madison	0.010303
11	Wake Forest	0.010282
12	Alabama	0.010068
13	Virginia Tech	0.010029
14	USC	0.008870
15	Vanderbilt	0.008530
16	Georgia Tech	0.008290
17	Virginia	0.008182
18	Boston College	0.008094
19	Weber State	0.008032
20	Villanova	0.007734
21	South Carolina	0.007616
22	North Carolina	0.007364
23	TCU	0.007291
24	Duke	0.007186
25	Florida State	0.006890
26	Maryland	0.006652
27	West Virginia	0.006561
28	Miami (FL)	0.006503
29	North Carolina State	0.006456
30	Clemson	0.006379
31	East Carolina	0.006373
32	Penn State	0.006286
33	Georgia	0.006247
34	Cincinnati	0.006167
35	Appalachian State	0.006042
36	Pittsburgh	0.006028
37	Iowa	0.005866
38	Tulsa	0.005797
39	LSU	0.005766
40	Northern Iowa	0.005254
41	New Hampshire	0.005201
42	Northwestern State	0.005098
43	Brown	0.005075
44	Jacksonville	0.005022
45	Oregon	0.005022
46	Central Arkansas	0.004908
47	Boise State	0.004855
48	California	0.004800
49	Northwestern	0.004680
50	Rutgers	0.004618
51	Grambling State	0.004466
52	Missouri	0.004447
53	Connecticut	0.004420
54	Houston	0.004326
55	Albany	0.004302
56	Ohio State	0.004282
57	South Carolina State	0.004277
58	Michigan State	0.004245
59	Dayton	0.004178
60	Southern Illinois	0.004119
61	Navy	0.004055
62	Harvard	0.003973
63	South Florida	0.003929
64	Liberty	0.003896
65	Arkansas	0.003782
66	Massachusetts	0.003764
67	Nevada	0.003722
68	Brigham Young	0.003655
69	Colgate	0.003652
70	Nebraska	0.003644
71	Holy Cross	0.003626
72	Stanford	0.003604
73	Maine	0.003601
74	Nicholls State	0.003597
75	McNeese State	0.003563
76	Wofford	0.003544
77	Texas State	0.003534
78	Oklahoma State	0.003522
79	San Diego	0.003516
80	Lafayette	0.003491
81	Eastern Washington	0.003429
82	Kansas	0.003412
83	Ball State	0.003395
84	North Dakota	0.003308
85	Hawaii	0.003291
86	Presbyterian	0.003274
87	Furman	0.003269
88	Florida A&M	0.003255
89	Monmouth	0.003246
90	Buffalo	0.003220
91	Kentucky	0.003215
92	Arizona	0.003209
93	Rice	0.003183
94	William & Mary	0.003171
95	Yale	0.003162
96	Eastern Kentucky	0.003146
97	Cal Poly	0.003117
98	Colorado	0.003111
99	Tennessee State	0.003085
100	Sacred Heart	0.003016
...	...	...

D. Other PageRank Output

See README for full descriptions.

1. dolphins_results_0.25.txt
2. dolphins_results_0.5.txt
3. dolphins_results_0.75.txt
4. dolphins_results_1.txt
5. lesmis_results_0.25.txt
6. lesmis_results_0.5.txt
7. lesmis_results_0.75.txt
8. lesmis_results_1.txt
9. NCAA_football_results_0.25.txt
10. NCAA_football_results_0.5.txt
11. NCAA_football_results_0.75.txt
12. NCAA_football_results_1.txt