CPE466
Lab3

Gilbert, Andrew
apgilber@calpoly.edu

Terrell, Josh
jmterrel@calpoly.edu

Abstract

When working with a graph of social data, where nodes represent actors and edges represent relationships, it can be useful to determine the most influential nodes. The PageRank algorithm provides a straightforward way to rank nodes based on their influence. We implemented the PageRank algorithm and observed its results on various datasets.

1 Introduction

The PageRank algorithm, developed by Larry Page and Sergey Brin, and initially used in what became the Google search engine, is a technique for calculating the importance of nodes in a graph. We wrote and tested an implementation of this algorithm to better understand how it works as well as observe its behavior on various datasets.

2 Implementation Overview

We implemented our reading and writing system in Python. We wrote the PageRank algorithm in c, and used Python's cffi library to interface between the two languages.

The ranking process starts by loading a given file into memory by parsing each line as an edge between two nodes. Each edge can have an optional weight. If the `--weighted` flag is specified, the weight is read from file. Otherwise each line is expected to be an edge between the left node pointing to the right node.

Our implementation computes the standard PageRank with options to customize its behavior:

`--help` — display all the below options and descriptions

`--dval <int>` — the `d` parameter of the PageRank algorithm; probability of following a link

`--epsilon <float>` — (per iteration) consider the algorithm to have converged if the maximum change in PageRank of each node in the graph is less than epsilon

`--maxiterations <int>` — the number of iterations the algorithm will run before before giving up and considering the graph converged

`--threads <int>` — the number of computation threads to use

`--batchsize <int>` — per iteration, each thread claims `batchsize` nodes at a time to compute PageRank for. Too small and the threads will spend all their time synchronizing around a mutex. Too large and all the threads may wait on one thread to finish so the algorithm can continue to the next iteration.

`--fmt string` — the data format; either csv (default) or snap

`--scale` — scale epsilon and the printed results by the size of the graph.

`--weighted` — specify if the graph includes weights per edge.

When the `--weighted` flag is enabled, our algorithm runs a modified version of PageRank which takes into account the weights of edges. The modified version computes PageRank such that the weight is the number of edges between the nodes. The result is that edges with higher weight indicate stronger relationships than edges with lower weights.

The in-memory representation of the graph, PageRank computation, and parallel execution all happen in C. In the parallelized PageRank computation, each thread asks the master thread for a block of nodes to work on. Each node has two page ranks—the page rank of the previous iteration and the page rank of this iteration. By having both page ranks, we were able to parallelize computation without forcing Nodes to be computed in a specific order.

The graph is composed of an array of Nodes, a c structure with six fields:

```
{
  unsigned int id;
  char active;
  double pageRank_a;
  double pageRank_b;
  int outDegree;
  struct LLNode *inNodes;
}
```

The PageRank fields allow us to switch between the page rank of the last iteration and this iteration. inNodes is a linked list of incomming edges. Each edge has a source node and weight.

Once the graph is built, many iterations of the PageRank are computed until the algorithm converges. On every iteration, worker threads calculate PageRank on blocks of nodes from the list of nodes that haven't been processed yet until there are no nodes left to process in the iteration.

For each node in the block, the current PageRank is read from either the `pageRank_a` or `pageRank_b` field and the new rank is written to the other. If the rank has changed more than the (configurable) epsilon, a flag is updated to inform the system that the algorithm has not converged. Once all the worker threads have completed their work,

if the "unconverged" flag is set, the master thread starts the workers again, this time telling them to read from the field they were just writing to and to write to the field they were reading from.

3 Results

3.1 STATES

MA	1.462869437832707	8.532877876499256e-09
PA	1.3524810393473186	7.338763104158819e-09
NY	1.3271898437852825	7.786156441813574e-09
TN	1.3246348667661838	5.411682039990762e-09
KY	1.2686518916839202	5.653326071164422e-09
VA	1.2149463457633922	5.519433868284018e-09
ID	1.1957384420210948	2.7722326170764866e-09
MO	1.1940810483446511	4.389067234655819e-09
AR	1.1802419201439789	4.378786742920138e-09
MD	1.1208570953850048	5.514879698742536e-09
GA	1.1115358233807457	3.888831097420553e-09
OK	1.100494937600193	3.6911694187291566e-09
NV	1.0631327348400668	2.5448008283968093e-09
TX	1.0454328357344842	3.6191985172684404e-09
NH	1.0356269847111486	5.2541955161689025e-09
OH	1.0345121503002406	4.7246054291072426e-09
WV	1.03418223392241	5.0745513693883915e-09
UT	0.9466970183172859	2.3233356211127543e-09
CO	0.9221848584696164	2.5064941441832467e-09
WY	0.9209460697948871	2.126906983868082e-09
OR	0.9207126832348004	2.0249385501713846e-09
NB	0.9198421526448283	2.550868544171081e-09
VT	0.9177377192802924	5.087968588113334e-09
SD	0.9152833803384534	2.0886851276324947e-09
IA	0.9032036994941208	2.6370548106280367e-09

3.2 NCAA-FOOTBALL

We observered that the NCAA data set is troublesome to run as unweighted because the nodes are in the opposite order from the other datasets. Typically edges in the dataset run from left (source) to right (destination), but for PageRank to do anything meaningful with this dataset, the edges must run from right to left. This is because the left team is always the winning team, and thus there should be an edge connecting from losing team (right) to winning team (left) to represent the winning team gaining prestige from the losing team. Our program only provides meaningful results for this dataset if the `--weighted` flag is provided, in which case the edges are flipped to go from right to left due to the weighting of edges.

Mississippi	15.35711933877105	6.565710225503452e-09
Florida	15.120842239804153	5.958453064280889e-09
USC	9.709094062122265	8.673411308612877e-09
Oregon State	9.138868090363559	9.268392678341009e-09
Oklahoma	9.093401455685925	2.0352699109382755e-09
Texas Tech	6.933583512895485	1.3122560954537121e-09
South Carolina	6.186403273609781	2.8357764469966185e-09
Texas	5.771265700405203	1.2049815728554236e-09
Alabama	5.289853932502011	2.11207079603426e-09
Penn State	5.107477363379916	4.5762501149626544e-09
Vanderbilt	4.91241475509652	2.353541389554792e-09
Utah	4.589133042096809	1.7619420089598492e-09
Oregon	3.959113846990245	4.055958388593517e-09
Wake Forest	3.4167626140316907	1.518014255963429e-09
North Carolina State	3.032703633203939	7.923217221206258e-10
North Carolina	2.9538761491670127	6.973040513058582e-10
Virginia	2.8493836410031554	5.648575808481215e-10
TCU	2.8280597269404333	8.340812832074995e-10
Clemson	2.821228451582217	7.64319538215652e-10
Florida State	2.812912146215853	9.765833107466815e-10
Richmond	2.5124588409373794	1.572353358625378e-14
West Virginia	2.4976151735654684	1.7437398747155441e-10
Maryland	2.453766427549545	1.178236914284314e-09
Houston	2.396438567629544	1.7840993266071692e-10
LSU	2.346431884334138	4.135507318536735e-10

3.3 KARATE

34	3.431252149966154	8.707609577829345e-09
1	3.2979076567614305	8.195399414656634e-09
33	2.4375696493652135	6.146548853225298e-09
3	1.940669293582341	5.122124044354415e-09
2	1.7978153919561228	4.609911521957777e-09
32	1.2633749429357202	3.0732737188454706e-09
4	1.2192351473227652	3.073274190690256e-09
24	1.0717654878943703	2.5610611964488328e-09
9	1.012045892241893	2.5610617862548146e-09
14	1.0042394946523805	2.561061904216011e-09
6	0.9897792474544365	2.048849971625355e-09
7	0.9897792474544365	2.048849971625355e-09
30	0.8938102700237597	2.0488492638581768e-09
28	0.8717520828066848	2.0488492638581768e-09
31	0.8360652668415772	2.0488492638581768e-09
8	0.8326768875894534	2.048849617741766e-09
11	0.747250371688433	1.5366375671899135e-09
5	0.747250371688433	1.5366375671899135e-09
25	0.716585132305966	1.5366369773839317e-09
26	0.7142107027051523	1.5366369773839317e-09
20	0.6665576263645843	1.536637095345128e-09
29	0.6654976130625588	1.5366369773839317e-09
17	0.5706561792973743	1.024425162754472e-09
27	0.5114972890075956	1.0244244549872938e-09
13	0.49792632259874825	1.024424808870883e-09

3.4 DOLPHINS

Grin	1.9929584993071667	9.35167862592401e-09
Jet	1.967144665571631	7.040288171888953e-09
Trigger	1.9405600839191688	7.792422068719773e-09
Web	1.865912985987976	7.0405458685307565e-09
SN4	1.8522709448116916	8.572964761699442e-09
Topless	1.8298805734742087	8.57237666268551e-09
Scabs	1.7622302792686884	7.792798718819771e-09
Patchback	1.640430072175734	7.0131839918774475e-09
Gallatin	1.6217262752147679	6.259463475322136e-09
Beescratch	1.5283444092422989	6.251719669725375e-09
Kringel	1.5277369287435327	7.01561124472061e-09
SN63	1.4842331174368502	6.2342287790451145e-09
Feather	1.4544256882987086	5.477406223031167e-09
SN9	1.3619146564136206	6.2369371750525815e-09
Stripes	1.3448497261403864	5.454560920992169e-09
Upbang	1.3423543472968191	5.473508507547464e-09
SN100	1.2780301140466968	5.460063970397222e-09
DN21	1.2433249899836387	4.694479728561962e-09
Haecksel	1.232751011081599	5.455429517853272e-09
Jonah	1.2025241103754618	5.454888742095765e-09
TR99	1.1923805400446332	5.4553137909807425e-09
SN96	1.092356294750833	4.677576721789922e-09
TR77	1.0750500910429874	4.678380571832008e-09
Number1	1.0620656262882313	3.908881406056164e-09
Double	1.0600946257920347	4.677003895281473e-09

3.5 LES-MISERABLES

Valjean	7.665974282581213	9.349708741598906e-09
Marius	3.9784442848480737	6.154232121309555e-09
Myriel	3.020831596182633	1.8344405538876785e-09
Cosette	2.8420371738890955	4.023922938346108e-09
Enjolras	2.8194934790338655	5.384945372423344e-09
Thenardier	2.7475371663125885	3.6096958469164497e-09
Courfeyrac	2.5409217654001015	4.970718280993686e-09
Gavroche	2.179302801471389	3.313812536875904e-09
Fantine	2.0915902723875295	2.781246382938668e-09
Javert	2.0653543190409005	2.7812416640571325e-09
Combeferre	2.0505992845473897	4.023914287063293e-09
Bossuet	2.0170107783106874	3.9055639516720475e-09
MmeThenardier	1.543019540501597	2.011961993493225e-09
MmeMagloire	1.5040372211824298	1.1243338226586685e-09
Gillenormand	1.4095121514104394	1.7160836644943e-09
Joly	1.3606420733536875	2.5445336527922524e-09
MlleBaptistine	1.340072283463428	1.0059829629472522e-09
MlleGillenormand	1.284394920231858	1.3610323961604776e-09
Bahorel	1.2668098475120204	2.307832982009761e-09
Babet	1.20889623728762	1.5977341155833102e-09
Feuilly	1.1930561303835443	2.2486579453941807e-09
Gueulemer	1.1309018254952004	1.479383255871894e-09
Favourite	1.0882621697317887	1.5385631424490523e-09
Tholomyes	1.0832720590924592	1.538562880288967e-09
Dahlia	1.0516761942505566	1.4793877125933441e-09

3.6 POLITICAL-BLOGS

155	14.33699779725792	3.184952301893418e-13
55	12.167448030347968	3.2911173786231984e-13
1051	10.087984834737082	2.526728826168778e-13
855	9.981406373574437	1.5287771049088406e-13
641	9.934803171129245	2.484262795476866e-13
1153	8.717819558526761	1.985286934846897e-13
963	8.55903698255803	9.236361675490912e-14
729	8.426207150886063	2.3993307340930414e-13
1245	7.139541800692389	1.613709166292665e-13
798	6.8826107180770135	1.5818596432737309e-13
323	6.801816765155126	1.794189796733292e-13
1112	6.774591406277854	1.5606266279277747e-13
1461	5.732336246128715	1.5818596432737309e-13
1306	5.593177860657325	1.411995520506082e-13
1463	5.455375831744107	1.603092658619687e-13
1179	5.399673672205255	1.3907625051601258e-13
1041	5.3512669970315105	1.242131397738433e-13
1437	5.14921968532767	8.493206138382448e-14
535	4.977002837558232	1.42261202817906e-13
990	4.815735281748969	1.0298012442788718e-13
180	4.448210464984683	1.263364413084389e-13
642	4.398969659166724	1.1731240978640756e-13
756	4.295354297845709	1.2739809207573671e-13
301	4.2541475426620625	9.501774367315363e-14
1067	4.197743464155556	8.015463293098435e-14

3.7 WIKI-VOTE

5000	0.15000000000000005	0.0
5001	0.15000000000000005	0.0
5002	0.15000000000000005	0.0
5003	0.15000000000000005	0.0
3	0.0	0.0
4	0.0	0.0
5	0.0	0.0
6	0.0	0.0
7	0.0	0.0
8	0.0	0.0
9	0.0	0.0
10	0.0	0.0
11	0.0	0.0
12	0.0	0.0
13	0.0	0.0
14	0.0	0.0
15	0.0	0.0
16	0.0	0.0
17	0.0	0.0
18	0.0	0.0
19	0.0	0.0
20	0.0	0.0
21	0.0	0.0
22	0.0	0.0
23	0.0	0.0

3.8 P2P-GNUTELLA05

1676	2.289326536963029	7.841079559625132e-09
1020	2.2403734160272597	7.066438398404401e-09
386	2.138792620909424	6.337140629473267e-09
222	2.1180519571356573	6.5605154186249535e-09
227	2.0587734837001266	6.805262942847102e-09
388	2.034446530890321	7.856065266527955e-09
389	2.024772893658017	6.180546226392837e-09
688	1.9477121731092966	5.783130314527488e-09
226	1.9082240862874207	4.066776062281764e-09
842	1.9043619099871956	4.8611213900235685e-09
876	1.8850775373543949	3.772987392620334e-09
223	1.7619395982432815	4.576491744649833e-09
31	1.7526146250970633	4.998516584962722e-09
391	1.7334015961053015	4.6258266099444105e-09
279	1.6981482949981928	4.81747366044144e-09
271	1.6915315317060065	4.333089577250593e-09
225	1.6806709215757638	4.4718840775575044e-09
277	1.6775938809818893	4.550289535844385e-09
274	1.6599153062361252	5.897778165473398e-09
272	1.643728843565932	5.081589191807418e-09
887	1.6308898566648484	4.116577762317778e-09
278	1.6155748529618141	4.044252904463787e-09
229	1.6115493616387802	3.730798911233575e-09
679	1.5637113774550506	8.713356797315688e-09
47	1.5298408382105524	3.1729529746346054e-09

3.9 SLASHDOT-ZOO-NOV6-2008

75	139.10869163006186	0.0
43	104.35655378302116	0.0
749	101.05897756485822	0.0
38	92.86216590392718	0.0
184	90.61875965881137	0.0
625	86.25195739493604	0.0
651	49.260104197988326	0.0
7262	34.626469309980244	0.0
74	33.97370811645962	0.0
877	33.806994726745295	0.0
163	32.7346809780974	0.0
57	29.460257181044028	0.0
28	28.913357515829233	0.0
1981	28.233941822149216	0.0
1240	28.150536496422063	4.1931519020876884e-15
1116	27.96799282036949	0.0
6487	27.3996761970114	0.0
34	26.887991365658717	0.0
46	26.768588413766594	0.0
11241	25.717756150265114	0.0
1397	25.27464343714724	0.0
1491	25.14467273185559	0.0
523	23.948384912093164	0.0
6246	23.672777601701213	4.1931519020876884e-15
165	23.65712952278909	0.0

3.10 AMAZON-MAY03

593	716.1350330208265	9.575464064888461e-09
595	595.6180810872537	9.45403946274459e-09
591	588.9014405022796	9.406448135164391e-09
89	528.5222626364321	1.6355404652140804e-09
590	421.90888647397355	6.6541571532366395e-09
972	409.18910845413495	6.958830605502397e-09
976	346.7322270803584	5.852043133067197e-09
974	338.6719026659318	5.744806973444899e-09
975	315.98100629724553	4.860631271606797e-09
978	310.40205320687534	5.79942580547152e-09
120	300.96452967132376	1.084059277674937e-09
977	294.1891785060549	2.490227335102957e-09
634	293.09074166740766	1.2090421099182624e-09
2612	284.5788411482617	2.2683716884019933e-10
598	258.3757870710722	4.114870720637459e-09
597	229.10405836743982	4.296874190224537e-09
585	222.55394269292776	3.2573817445452624e-09
162	217.58581615239163	9.72820109190753e-10
596	196.76295115954053	3.251955443643595e-09
4455	194.42224538691647	1.654354606864944e-10
88	182.55613792716736	4.686856209124707e-10
44	181.573800305272	2.893063460237405e-10
1196	179.52262153854838	7.012826747261613e-10
4458	177.25465267418588	1.456872836345241e-10
594	168.25616727047833	1.886284492534987e-09

3.11 LIVEJOURNAL1

8737	667.7378481886446	1.379633589987904e-10
2914	588.7797064482876	1.0150161412053865e-10
18964	411.67883873595986	1.0058185839388004e-10
1220	356.109611293974	7.653681582551943e-11
2409	336.3273934810311	8.330359010022201e-11
10029	327.4238324338779	1.5116842336010319e-10
214538	303.35145321800434	7.693099685123026e-11
7343	286.8930973744934	5.689346137759642e-11
39295	270.0845662115683	6.306896411373275e-11
38283	269.0910138021256	9.036600014420771e-11
18963	268.7649320414741	1.6305955096904655e-10
40509	239.19028450395746	3.54105954763562e-11
1918	236.0082507540417	3.081181684306319e-11
4494	235.22733404137333	4.628342210221325e-11
3407	227.1944549538492	2.562176667120393e-11
214406	223.48878544091093	9.953662170737016e-09
56913	220.85722026121175	5.544813094999004e-11
39633	218.738612486407	3.44908397496976e-11
1772	212.27053645376893	4.0009374109649214e-11
503	210.9315239444211	2.992490953521382e-11
1689	204.68175142268692	6.221490522469262e-11
33515	195.08870669755345	5.682776453997795e-11
96144	191.71010604439104	9.673859339319945e-11
16624	187.9257211584422	2.4242133081216026e-11
58404	186.77091266451208	5.0652261803841615e-11

4 Overall Summary

Before implementing edge weights, page rank appeared to do decently on the graphs we manually tested. We created our own fakeball dataset, and analyzed carefully the results of the NCAA football dataset before and after weighting. As noted in the NCAA football section of Results, NCAA football was a bit troublesome to run unweighted.

Once we implemented edge weights, we noticed different rankings of the football teams, and the change in rankings seemed a bit more inline with the rankings from ESPN's 2008 data, though the rankings were far from perfect. We also compared the PageRank results with a ranking the teams by total score. Weighted PageRank matched much more closely to the teams sorted by total season score.

5 Performance Evaluation

All datasets were successfully executed in under 30 minutes (combined) on a 4-core hyperthreaded processor using 8 threads. The execution time for each dataset is provided in table 1. A visual representation is also included in fig. 1. Note that that on

Dataset	Execution Time (seconds)
amazon0505	38.734575
dolphins	0.005477999999999997
karate	0.004890999999999993
lesmis	0.005057000000000006
NCAA_football	0.006725000000000009
p2p-Gnutella05	0.025919000000000025
polblogs	0.06062600000000001
soc-LiveJournal1	1121.060663
soc-sign-Slashdot081106	2.5643610000000003
stateborders	0.006414000000000003
wiki-Vote	0.0008580000000000254

Table 1: Execution Times on Datasets

the graph, both the x and y axis are drawn in log10 so the outliers do not croud out the trends. The visible trend on the graph suggests that the execution times increase roughly linearly as edge count increases to the max of our dataset.

6 Extra Credit

We added a weighted flag to our algorithm which takes into account a weight per edge when computing PageRank . We experimented with different ideas, but decided that simply using the weight as a multiplication factor for the importance of the edge would have the desired effect. We treated weights as edge counts. For instance, if the edge a->b had a weight of 7, we would compute page rank as if there were 7 edges from a to b. A weight of -7 would mean there were 7 edges in the opposite direction.

Increasing the edges also involved increasing the node count. We discovered that PageRank would not converge unless we added an appropriate amount of fake nodes to the graph so that outdegree and other variables of the PageRank computation were accurate.

We used our weighted modification to the PageRank algorithm to compute PageRanks for the NCAA_football, lesmis, and SlashDot datasets.

A README

Team: Andrew Gilbert, Josh Terrell

Important: Run everything from the project's root directory

A.1 Setup environment

```
pyvenv virtual
source virtual/bin/activate
```

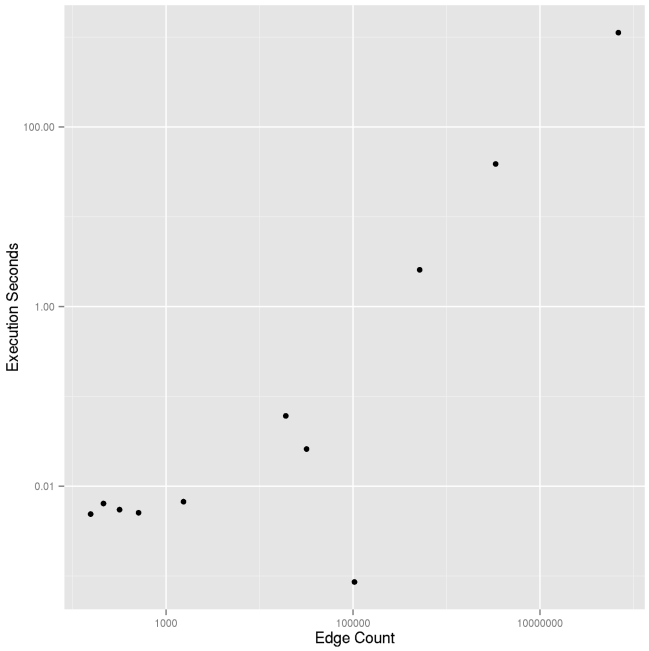

Figure 1: Graph of Execution Times on Datasets

```
pip install --upgrade -e .
```

A.2 Run all the tests

```
python3 -m unittest discover
```

A.3 Compute Page Rank

The ranker scripts assume the graph is directed.

Correctly parsed formats include:

SNAP (edges must be repeated if undirected) (weight is optional):

```
a b [weight]
```

CSVs, (edges must be repeated if undirected) (w_a and w_b are ignored unless `--weighted` flag is specified):

```
a,w_a,b,w_b
```

To run:

```
ranker <filename.csv>             # unweighted
ranker <filename.csv> --weighted # weighted
ranker <filename.txt> --fmt=snap # option specifiying input file is snap format
```

You can use `ranker --help` for more options such as setting the pagerank `d` parameter or tweaking parallel computation settings.

A.4 Deactivate environment

```
deactivate
```

A.5 (Optional) Manual Rebuild of Page Rank C-Implementation

```
python build_page_rank.py
```