Rank Vertices in Undirected Networks Based on the Mutually Reinforcing Relationship

Zirou Qiu, Yixian Li

School of Computing, Clemson University, Clemson SC, USA,

{zirouq, yixian}@clemson.edu

***Abstract*—Many real-world applications can be modeled by networks. Given a network, we often want to quantify the importance of vertices based on the structure of the network. In this report, we introduce a centrality measure which respects the mutually reinforcing relationship between vertices in undirected networks.**

Keywords—centrality, mutual reinforcement, PageRank, convergence, network science

I. Introduction

One nature of the undirected network is the mutual reinforcement of centralities between adjacent vertices such that vertex *u* distributes centrality to *v*, and in return, *v* distributes centrality back to *u*. In real-world scenarios, one nature of the mutual reinforcement relationship is that you should give the most amount of rewards to those who helped you the most.

Given a network, we often want to quantify the importance of vertices based on the structure of the network. Traditional measures such as the eigenvector centrality and the Katz centrality both have an undesirable feature: neighbors of some high-centrality vertex will also have high centrality[1]. PageRank solves this issue by distributing the centrality evenly to all neighbors, however, it does not respect the nature of the mutual reinforcement. Under the setting of PageRank, given a vertex *v*, all neighbors of *v* gets the same amount of *v*'s centrality regardless of their contribution to *v*[1]. In other words, the neighbor with the least amount of contribution is treated equally as the neighbor with the most amount of contribution. Such a measure is not fair and some intrinsic network information is missing. Therefore, we need a new measure which takes the nature of the mutual reinforcement into account.

In this project, we introduce a measure of centrality which respects the nature of mutually reinforcing relationships between vertices in undirected networks. The intuition behind this measure is that *the amount of centrality a vertex v gained from u should be proportional to the centrality v contribute to u.*

This report is organized as follows. *Section II* discusses the related works. *Section III* introduces the proposed measure. *Section IV* presents the experimental results. *Section V* suggests the future works. Instructions on running the code can be found in the Appendix.

II. Related Work

Many works have been conducted on the topic of weighted PageRank. Fiala et al. suggest a weighted PageRank such that centrality should be distributed unevenly [2]. In their work, given a vertex *v*, the fraction of centrality which *v* should distribute to neighbors is predefined for each neighbor and remains the same for every iteration. This is different from our approach. Also, they predefine such fractions based on non-network data, and our proposed measure relies on the nature of the mutual reinforcement. Last but not least, their work is very specific to bibliographic networks.

Haveliwala suggests a model which a set of scores are computed for each vertex based on some queries, and these scores are combined with the PageRank score[3]. This approach relies on the non-network data to distribute centralities and does not consider the mutually reinforcing relationship. Ding et al. also considered the weighted PageRank; however, they added the weight to β such that different vertices have different initial centrality. The weights in their model do not get reinforced. He et al. discuss how to rank vertices on the heterogeneous bipartite network[4]. They set the scaling factor as $w_{ij} / \sqrt{d_i}\sqrt{d_j}$, which is fixed. At last, Xing and Ghorbani[5] introduce another model which assign larger rank values to more important (popular) nodes, and the importance of a node is proportional to its indegree and outdegree. Under this setting, the importance of each vertex is fixed, and the author does not consider the mutually reinforcing relationship.

III. Proposed Measure

In the section, we introduce the base model. Given an undirected graph $G = (V, E)$ where $|V| = n$ and $|E| = m$, let C denote the n x n *contribution matrix* with real-valued entries such that C_{ij} is the amount of centrality vertex i should contribute to vertex j. C_{ij} equals to zero if $(i, j) \notin E$. C is asymmetric in general. Let R be the n by 1 vector which R_i is the centrality of vertex i. We have :

$$C_{ij} = (C_{ji} / \sum_k C_{ki}) R_i \qquad (1)$$

$$R_i = \alpha(\sum_j C_{ij} / \sum_k C_{kj} \cdot R_j) + \beta_i = \alpha \sum_j C_{ji} + \beta_i \qquad (2)$$

where α is the damping factor and β_i is the free centrality of vertex *i*. In matrix forms:

$$C = D'D^{-1}C^{T} \quad (3)$$

$$R = \alpha C^{T}1 + \beta \quad (4)$$

where *D*' is a diagonal matrix such that $D'_{ii} = R_i$, and *D* is a diagonal matrix where $D_{ii} = (C^T 1)_i$.

To compute the ranking, we assign each vertex an initial centrality and each pair of adjacency vertices two (one for each direction) initial contribution values. The proposed measure works better when we know the importance of each vertex from network-independent sources such that β_i is uniquely predefined for each *i*. We keep iterating (3) and (4) until *R* converges. At last, R_i is the final centrality of vertex *i*, and C_{ij} is the contribution relationship between *i* and *j*.

We can easily understand why the proposed measure produces rankings that is different from PageRank. Under the setting of our proposed measure, a vertex will not have high centralities simply because it is adjacent to some highly ranked vertices. For example, suppose vertex *u* is adjacent to a high-ranked vertex *v*, but if the amount of centrality *u* contributed to *v* is trivial compared with *v*'s other neighbors' contribution, then in return, *v* will not give much of its centrality back to *u*. Similarly, a highly-ranked vertex will not always get the most unit of centralities from all its neighbors.

IV. EXPERIMENTAL RESULT

The dataset we use is an email network provided by a European research institution. The network is unweighted and undirected with 1005 vertices and 25571 edges. Vertices are anonymized by replacing identities with numbers. The link to the dataset can be found in the appendix.

We modified the base model in *section III* by replacing *β* with *(1 - α)R*. The intuition is that centralities of vertices for next iteration should be accumulated on the centralities from the previous iteration. We ran the proposed measure against the PageRank on the email network and observed very different rankings of vertices. The Spearman coefficient between the rankings produced by the proposed measure and the PageRank is 0.269 for α = 0.2.

We tested convergence on different α values. Figure 1. and Figure 2. plot the number of iterations vs the spearmen coefficient between the ranking at the current iteration and the final ranking, for α = 0.2 and α = 0.5.

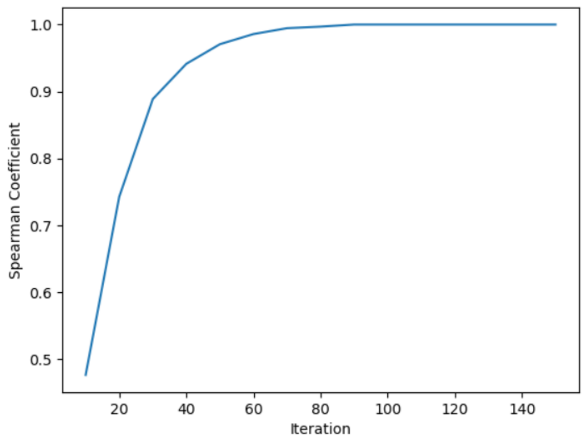


Figure 1. Convergence on α = 0.2

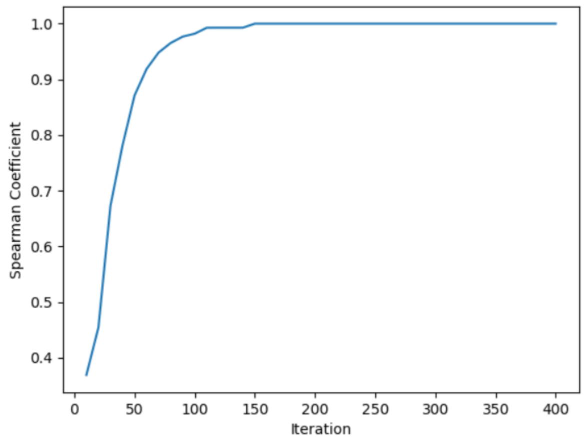


Figure 2. Convergence on α = 0.5

For different α, the number of iterations *k* required for *R* to converge such that the ranking produced at the k_{th} integration is the same as the ranking produced after the k_{th} iteration is presented in Table I.

TABLE I. NUMBER OF ITERATIONS REQUIRED TO CONVERGE

α	Number of iterations
0.2	100
0.3	110
0.4	125
0.5	157
0.6	218
0.7	235
0.8	352
0.9	707

As we expected, with the increase of α, the number of iterations required for *R* to converge also increases.

V. FUTURE WORK

The next step is to justify the ranking produced by the proposed measure and find real-world applications (datasets) which could explain the ranking. Currently, all we know is that the ranking is very different from PageRank since is it based on the completely different intuition, but we cannot identify the characteristics of high-ranked vertices under our proposed measure. Also, we believe that the based model can be further refined.

REFERENCES

[1] M. Newman, "*Networks: an Introduction*", Oxford: Oxford University Press, 2010.

[2] D. Fiala et al., "*PageRank for Bibliographic Networks*", Scientometrics, 2008.

[3] T. H. Haveliwala , "Topic-Sensitive PageRank," WWW, 2002.

[4] X. He et al., "*BiRank: Towards Ranking on Bipartite Graphs*", IEEE transactions on knowledge and data engineering, 2017.

[5] W. Xing and A. Ghorbani, "*Weighted PageRank algorithm*", Second Annual Conference on Communication Networks and Services Research, 2004.

APPENDIX

A. Dataset

The dataset *Email-Eu-core network* is included in the project folder. You can also download the dataset from: https://snap.stanford.edu/data/email-Eu-core.html

B. Required Python Version and Libraries

Python3, networkx, numpy, scipy, matplotlib.pyplot

C. How to run the code

cd to the project folder. The base command: *python3 main.py* (*python main.py* if only python3 is installed).

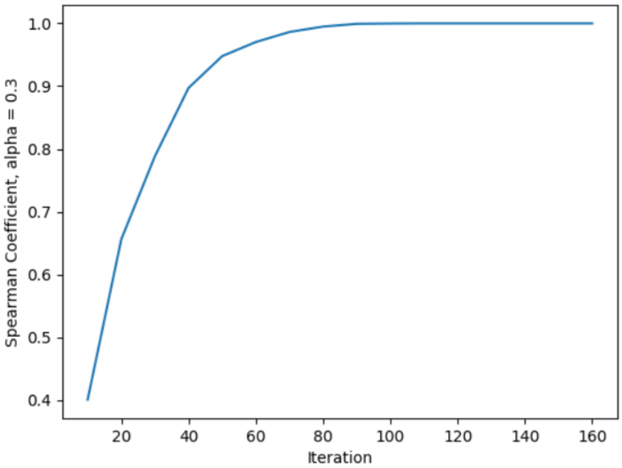
Command-line Arguments (order doesn't matter)	
No argument	run the proposed algorithm, print the top 20 vertices, save the full ranking to the file *proposed_ranking.txt*"
float	Specify the α. The default is 0.2 (recommended value).
-r	Same as no argument
-vs	run both the proposed algorithm and PageRank, save the full ranking to the file *vs.txt*, print the Spearman coefficient
-plot	construct the convergence diagram of the given α value (the default alpha is 0.2)
-?	Print the explanations of command-line arguments

D. Example

`Python3 main.py :` run the proposed algorithm, print the top 20 vertices, save the full ranking to the file *proposed_ranking.txt*"

```
 -----------------
| top 20 vertices | alpha = 0.2
------------------------------------------
vertex : 377, centrality: 3.8021071719264006
vertex : 107, centrality: 3.2214554859855316
vertex : 160, centrality: 3.189061405919984
vertex : 5, centrality: 3.12679672026704 5
vertex : 211, centrality: 3.04310418227513
vertex : 971, centrality: 2.8209108706671393
vertex : 121, centrality: 2.810393330052683
vertex : 411, centrality: 2.5829627999713445
vertex : 82, centrality: 2.5467762569514814
vertex : 65, centrality: 2.4910734096158658
vertex : 414, centrality: 2.3945538621597944
vertex : 96, centrality: 2.27419206552260 2
vertex : 84, centrality: 2.2645144359447973
vertex : 86, centrality: 2.1562109853612275
vertex : 258, centrality: 2.1023431633659606
vertex : 189, centrality: 2.095249036419088
vertex : 462, centrality: 2.0751115910156117
vertex : 191, centrality: 2.0689074965342678
vertex : 412, centrality: 2.05360106099305
vertex : 62, centrality: 1.971482204250692
------------------------------------------
```

`python3 main.py -plot 0.3 :` construct the convergence plot of $\alpha = 0.3$



`python main.py 0.25 -vs -r :` First, run the proposed algorithm, print the top 20 vertices, save the full ranking to the file *proposed_ranking.txt*", then, run the proposed algorithm and PageRank, save the full ranking to the file *vs.txt*, print the spearman coefficient.

```
 -----------------
| top 20 vertices | alpha = 0.25
------------------------------------------
vertex : 377, centrality: 4.028386861814834
vertex : 160, centrality: 3.5527194507702395
vertex : 107, centrality: 3.447200126924233
vertex : 5, centrality: 3.3434408569773435
vertex : 211, centrality: 3.2103287505352407
vertex : 121, centrality: 3.02952427072401
vertex : 971, centrality: 3.0082455000698456
vertex : 82, centrality: 2.761610515304673
vertex : 411, centrality: 2.7307202167302256
vertex : 65, centrality: 2.616397820070694
vertex : 414, centrality: 2.4610300884219
vertex : 84, centrality: 2.43572034983181
vertex : 96, centrality: 2.4042902625810227
vertex : 86, centrality: 2.350682006219941
vertex : 189, centrality: 2.2007243101693215
vertex : 258, centrality: 2.1894602974367086
vertex : 191, centrality: 2.1591355375709123
vertex : 412, centrality: 2.1355659100649005
vertex : 62, centrality: 2.13430183252956
vertex : 462, centrality: 2.096316186083566
------------------------------------------
```

```
 -------------------------------------
| top 20 vertices Proposed VS PageRank | alpha = 0.25
----------------------------------------------------
Proposed | PageRank
377 | 160
160 | 121
107 | 82
5 | 107
211 | 86
121 | 62
971 | 5
82 | 13
411 | 166
65 | 434
414 | 377
84 | 64
96 | 211
86 | 183
189 | 129
258 | 249
191 | 533
412 | 84
62 | 21
462 | 128
----------------------------------------------------

Spearman Coefficient: 0.3049147322227602
```