

AuthorEchelon Algorithm for Erdos1 Authors

March 25, 2014

ABSTRACT

We analyze coauthor and article networks populated by authors with an Erdos number of 1, and articles linked by citation respectively. We develop an algorithm to rank the authors in terms of relative influence, taking into account the quality of published works. When the only relationship between authors is coauthorship, only general centrality measures can be readily applied to attempt to distinguish influential network members. Simple models which only consider node degree measures fail to provide a purposeful way to weight links, limiting the detail of the analysis. We present a method to rank coauthors by first constructing an article network composed of articles written by members of the coauthor network. We require that the articles cite other papers in the network so that relative article rankings can be calculated. These article rankings are used to determine link weights in a symmetric coauthor(coendorsement) network, which can be thought of as the work quality of an author. Weighting coauthor links in this manner treats endorsement by a prominent author as more valuable than endorsement by a lesser one. With this furniture in place, a variation of a pagerank algorithm is applied to the coauthor network. We find the Perron-Frobenius eigenvector of the matrix composed of authorrank coefficients (quality weights), whose elements when normalized to one are the author ranks of the coauthor network. We discuss conclusions concerning influential members of the Erdos1 subnetwork and general applicability of this ranking method to other types of networks. We conclude with suggestions for further modifications to this approach, such as treating Erdos like a random web surfer in the Erdos1 network, where the pagerank damping coefficient is the probability that Erdos will coauthor with you given that he pays you a visit.

Keywords: Network, Node, Target, Centrality, Heimdall Node

Software Used: MATLAB, RStudio, Cytoscape

Contents

1	Introduction	1
1.1	Vocabulary	1
2	Construction of the Erdős1 Network	2
2.1	RStudio	2
2.2	The Data	3
3	Preliminary Algorithm	4
3.0.1	Cytoscape	4
3.1	Extracting the Erdos1 Subnetwork	4
3.2	Graphing Erdos1 Subnetwork	5
3.3	Limitations of this Model	5
4	Construction of Article Network 1	6
5	Pagerank Algorithm for Directed Graphs	6
5.1	Applicability	6
5.2	Implementation	7
5.2.1	Applying Pagerank to Article Network 1	8
5.3	Thorny Issues With Pageranking Articles	8
6	ArticleImport Algorithm for Ranking Articles	9
6.1	Implementation	9
6.2	Conclusions Regarding Influential Nodes in Article Network 1	9
7	Construction of Article Network 2	9
8	Modified Authorrank Algorithm	10
8.1	Implementation	10
9	Using Modified Authorrank Algorithm on Erdos1 Subnetwork	12
9.1	Dangling Nodes, Zero Indegree, and Perron-Frobenius Eigenvector Revisited	12
9.2	Applying Modified Authorrank Algorithm to Article Network 2 and Erdos1 Subnetwork	12
10	Utility of Modeling Influence and Impact Within Networks	12
11	Conclusion	12

1 Introduction

Mathematics has been a very powerful tool dating back thousands of years. At its roots, it is just another language, effectively able to communicate and describe ideas. In this way, mankind has been attempting to use mathematics to create descriptive models of our universe, from the cosmos to the atom. As our species has advanced, both in a technological and societal sense, we have been able to focus our efforts on more abstract ideas, exploring new fields such as interactions between humans rather than the motion of a rock hurled through the air. New tools have enabled us to see microscopic processes in the body we never knew existed, and the leaps we have made in advancing computing power now allow us to work with truly huge and complex systems. Network science can be seen as a branch of mathematics; a method to formally speak of these interactions between people, between the components of a metabolic process in the human body, or in the context of numerous other connected systems. Specifically, network science combines elements of linear algebra, graph theory, and computer science (specifically data mining and data preparation) in order to analyze and make predictions about a given system. It is a field with very direct practical applications spanning many different fields as already noted, and thus research to further this field stands a great chance of finding its way into real-world scenarios. For example, networking science is largely responsible for Google's success. Where other search engines simply searched web pages for key words, Google was able to rank pages according to their importance within the network of existing pages (citation). The recent expansion of this field and its applied nature make it a worthwhile area of research. This claim is supported through the actions of Consortium for Mathematics and its Applications (COMAP). The COMAP hosts the MCM/ICM contest, an international mathematical modeling competition featuring three problems to choose from each year, one of which is always an interdisciplinary prompt (citation? <http://www.comap.com/undergraduate/contests/>). Topics in the past have focused on clean energy research, queuing theory, and in two of the last three years, network science. Drawing upon the Interdisciplinary Contest for Modeling prompt from 2014 (see attached prompt), this paper will focus on the creation and analysis of a network of coauthors who share authorship with Paul Erdős, called the Erdős1 network from this point forward. Specifically, this paper will address the creation of Erdős1 and the AuthorRank algorithm, answering the question of which author in the Erdős1 network is most influential, and then discuss ways in which this algorithm could potentially be generalized to analyze the most influential node for any given network.

1.1 Vocabulary

Before continuing further, our vocabulary must be established, as it will be seen throughout the rest of the text.

Node, Edge: Each element of a network will be referred to as a node, with the connections between nodes being the edges. For example, each coauthor in the Erdős1 network is a node, with the coauthor relationship between them being edges.

Centrality: The centrality of a vertex measures its relative importance within a graph. A centrality index is a real-valued function on the nodes of a graph.

Shortest Path: The shortest path(s) $p_{min} \in \mathbb{Z}^+$ between two nodes s and t is defined to be the minimum number of edges connecting s and t .

Betweenness Centrality: A real-valued function measuring node A 's centrality in a network, equalling the number of shortest paths from all nodes to all other nodes that pass through A .

$$g(v) = \sum_{S \neq V \neq t} \frac{r_{st}(v)}{r_{st}}$$

r_{st} = total number of shortest paths from node s to node t

$r_{st}(v)$ = number of those paths which pass through node v

Closeness Centrality: The average shortest distance from node i to every other node. This value is lower for vertices which are more central in the graph.

Stress Centrality: The stress of a node is the number of shortest paths that travel through that node.

Heimdall (Gatekeeper) Node: We say a node V is a Heimdall node if for some other two nodes X and Y , every path from X to Y passes through X . The stress and betweenness centrality measures are good indicators of which nodes are Heimdall nodes.

2 Construction of the Erdős1 Network

2.1 RStudio

The R programming language is a free, open-source project that is used frequently when working with data and performing statistical analyses. One advantage to using R is that instead of performing button-click analysis, R is a language that is used by programmers to construct custom scripts and functions allowing for tailored, precise functionality and experimentation throughout the data-preparation process. R also provides an easy way to read in various file formats, such as .txt or .csv files (often seen in Microsoft Excel). RStudio specifically was chosen for data preparation, being an interactive development environment for the R language providing a text editor, console, and workspace for the user allowing for ease of use when composing functions and viewing data.

2.2 The Data

The raw data was retrieved from <https://files.oakland.edu/users/grossman/enp/Erdos1.html> as a list of the authors who have coauthored a paper with Paul Erdős with the date of the coauthored paper listed beside each name. Beneath each of these authors is a corresponding list of their coauthors, resulting in over 18,000 lines of data. The list was copied into a .txt file, and loaded into RStudio. Below is a sample of the data:

ABBOTT, HARVEY LESLIE 1974
 Aull, Charles E.
 Brown, Ezra A.
 Dierker, Paul F.
 Exoo, Geoffrey
 ...

ABBOTT, HARVEY LESLIE is the first Erdős coauthor in the dataset and the names below are those who have coauthored a paper with Abbott, who is said to have an *Erdős number* of one, and consequently each name beneath Abbott would have an Erdős number of two. It is possible for those with an Erdős number of one to appear again beneath others with degree of separation one and in fact, this happens quite often and is an essential characteristic of this data set. Only those authors with an Erdős number of one will be compiled into the Erdős1 network. Therefore all authors with degree of separation greater than one must be removed from the dataset. There are other characteristics throughout the raw data that must be taken into account when attempting to pluck out the relevant authors. These are:

MOSER, LEO* 3	* signifies deceased author
Mori, Kenichi^2	^n is nth unique person with same name
Hartman, Irith Ben-Arroyo	some have multiple middle names with hyphens
LOEBL, MARTIN	some are in all upper case
1944: 2	numbers after dates show how many coauthored papers

These can cause numerous problems when searching through data composed of nothing but characters. To do the required task requires the use of *regular expressions*. A regular expression is a sequence of characters forming a search pattern according to a specified set of syntax rules (source). For example, creating a regular expression in R that reads “[a-z]3,6” will search the dataset for any three lowercase letters in a row, but no more than six. This would return matches for any words of lengths 3, 4, 5, or 6. Regular expressions can quickly get complicated and more intricate, allowing for the

searching of specific types of strings in the dataset.

First, a list of all of the 511 coauthors of erdos number 1 was constructed by searching through the dataset for any names with a date beside it. After an initial viewing of the dataset it is clear that only the non-indented names have dates associated with them, and these names are indeed the members of the Erdos1 set. Once a list of names is compiled, it is necessary to search through the lists beneath each name, and remove the erdos2 authors. This is where the complications listed in the table above became factors. It is difficult to construct a search pattern when many of the names hold irregularities that break any attempt at establishing a pattern. A successful, though inefficient and tedious method was constructed in the `erdosnetwork.R` function to remove the unwanted authors, but revised *erdos.R* makes quick work of this process by realizing that the relevant authors case letters.

3 Preliminary Algorithm

3.0.1 Cytoscape

Cytoscape is also an open sourceware for visualizing data networks. While originally intended more for biological systems, it can be used for any general network (www.cytoscape.org/what_is_cytoscape.html). In addition to the base Cytoscape application, an application named Centiscape was downloaded. This application calculates various influence measures based on the structure of a network (defined below). Cytoscape appears to be a well-maintained site, reliable in its operation. The standards of centrality measures are well-defined through it, and the visualization options aid in understanding the details of a given network.

3.1 Extracting the Erdos1 Subnetwork

Data was supplied via a website or included by the COMAP competition and consisted of over 18,000 lines of data, including names of authors and coauthor relationships. An example of a few lines of the raw data format is:

ABBOTT, HARVEY LESLIE 1974

Aull, Charles E.

Brown, Ezra A.

Dierker, Paul F.

Exoo, Geoffrey

To create a network of authors only with Erdos numbers equal to one, this data set had to be modified with those authors of Erdos number equal to two removed. The function `erdosnetwork.R` was composed to handle the data extraction and formatting. After various attempts at constructing an appropriate pattern using regular expressions in R, it was noticed that all Erdos 1 author names were spelled in all uppercase let-

ters, leading to a relatively easy extraction of the relevant data while maintaining the coauthor relationships. The function takes the data saved as a .txt file and writes the relevant data into a new .txt file formatted into two columns, where horizontal rows represent a coauthor relationship. A sample of the returned data is:

```
> #newdata <- erdosnetwork("Erdos.txt")
> #head(newdata)
```

This data was then loaded into Cytoscape to construct the network and perform analyses via .

3.2 Graphing Erdos1 Subnetwork

Cytoscape accepts Microsoft Excel files with the data formatted in columns, one being the column of source nodes, and the other the corresponding target nodes. Microsoft Excel in turn can accept formatted text files. In this way the data assembled in R is turned into the input for the software to build the network. Once the network has been assembled, Cytoscape offers various options for controlling the directedness of the graph, removing duplicate nodes and edges, and with Centiscape performing network analysis calculations. Specifically, this software was used to calculate closeness, betweenness, stress, mean average path, and other structural-related influence measures.

3.3 Limitations of this Model

While this network contains all researchers in the Erdos1 set, the only determining factors are those from the inherent structure of the network. Therefore, we concluded that the node most central and well connected in the network was the most influential, with connectedness being determined by summing up the values of each node's closeness, betweenness, and stress. While this model served as a good beginning point, there are many factors to consider when considering the influence of a researcher. The influence was thought of as influence due to connectedness and influence due to importance of research. For example, if a researcher is connected to many others but their research isn't of much worth when thinking about advancing that field, that researcher's influence wouldn't be as high as somebody in a comparable situation with more significant research. A more thorough understanding of all of the structural influence measures, such as eccentricity, clustering, and others could give a more accurate depiction of which influence is most influential. This graph is also lacking direction, meaning that there is no apparent way to weight how each node will affect the relationship they share. From a practical standpoint, this model is somewhat limited when considering general application to other networks in the fact that the data extraction process requires a precise, network specific algorithm. If the data is presented in any other way than the Erdos1 set was presented, the model would require some alterations to be

able to reach the analysis stage. These limitations lead us to consider creating a more thorough model, one which will also take into account research importance.

4 Construction of Article Network 1

5 Pagerank Algorithm for Directed Graphs

5.1 Applicability

The pagerank algorithm is the reason why google was an important player among search engines early on. Instead of simply considering the web as an undirected graph and considering only total node (page) degree, the pagerank takes into consideration the direction of links between web pages and treats links conceptually as votes for the webpages they are directed to. Backlinks to a page are links to that page from other pages. Simply counting backlinks and using those values to rank web pages fails to account for the idea that important pages should contribute more than unimportant pages when votes are being tallied for a page which both categories of pages link to. Pagerank does consider such weighted votes, and is therefore far superior to simpler ranking algorithms. This approach is additionally advantageous because it prevents pages from gaining importance by simply creating links to more pages. It accomplishes this by dividing a nodes' vote by the number of outgoing links. In the context of an article network, one must declare what characteristic provides the link between articles. In the case where the goal is to apply knowledge of article rankings to weight links in a coauthorship network, it makes sense to treat citations as backlinks. As with ranking pages, one should avoid simply taking the sum of indegree (being cited) and outdegree (citing someone) and treating that as the rank. Doing so would not consider who is citing who, which wouldn't provide very good information about which articles are influential. Thus, a better approach is to divide an articles' rank among the articles it cites, distributing the vote equally among them. This will have the same effect as the basic pagerank algorithm described above, giving more weight to votes from more influential articles. Of course, this model is still simple in that it assumes equal distribution of vote, which implies that an author relied upon each source equally which is an unlikely event. Setting up the article network in this fashion allows a system of equations to be established which expresses the relationships between article ranks given the outdegree for each network member. Equal distribution does have a nice effect, namely that in the absence of dangling nodes (nodes with zero outdegree) and nodes with zero indegree (which will henceforth be referred to as anchored nodes), the rank coefficient matrix produced is non-negative and column stochastic. That is, each entry is greater than or equal to 0, and the sum of the values in any given column is equal to 1. In fact, this matrix will be guaranteed to have 1 as an eigenvalue. This matrix is a component of the eigen problem corresponding to the system of equations. Taking the eigenvector corresponding to eigenvalue 1 and normalizing its' components to 1 gives article rankings which take into account indegree, outdegree, and weighted

importance of vote (citation). Difficulties arise when dangling nodes (Articles which do not cite) or anchored nodes (articles which are not cited by anyone in the network) appear. When such things occur, patches can sometimes be made, but one may find an alternative approach is preferable.

5.2 Implementation

This is the method described by K. Bryan and T. Leise in *THE \$ 25,000,000,000* EIGENVECTOR THE LINEAR ALGEBRA BEHIND GOOGLE.*//

Let r_n be the rank of a node n in a directed network. Define $r_n = \sum_{j \in L_n} \frac{r_j}{k_j}$ where

$L_n = \{ \text{nodes with links to node } n \}$ and
 $k_j = \text{number of out citations for the } j\text{th node.}$

The system of equations produced from this can be written as follows:

$$\begin{aligned} r_1 &= c_{12}r_2 + c_{13}r_3 + \dots + c_{1n}r_n \\ r_2 &= c_{21}r_1 + c_{23}r_3 + \dots + c_{2n}r_n \\ r_3 &= c_{31}r_1 + c_{32}r_2 + \dots + c_{3n}r_n \\ \vdots &\quad \vdots \quad \quad \quad \vdots \quad \quad \dots \quad \quad \vdots \\ r_n &= c_{n1}r_1 + c_{n2}r_2 + \dots + c_{nn}r_n \end{aligned}$$

Note that

$$c_{ij} = \begin{cases} \frac{1}{k_j} & , j \in L_n \\ 0 & , j \notin L_n \end{cases}$$

The above system of equations is equivalent to the eigensystem:

$$Cr = \begin{pmatrix} 0 & c_{12} & c_{13} & \dots & c_{1n} \\ c_{21} & 0 & c_{23} & \dots & c_{2n} \\ c_{31} & c_{32} & 0 & \dots & c_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & c_{n3} & \dots & 0 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_n \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_n \end{pmatrix}$$

Note that all values along the diagonal are zero because articles (or webpages) will not be given credit for self-reference.

Recall that for a matrix A with eigenvector X and eigenvalue λ ,

$$AX = \lambda X$$

Therefore, the column vector r of the ranks of network members is the eigenvector corresponding to eigenvalue $\lambda = 1$, which C is guaranteed to have as an eigenvalue because C is column stochastic (the column values sum to one in each column). C will be column-stochastic as long as there are not any dangling or anchored nodes. Calculate the eigenvector of C corresponding to eigenvalue 1, normalize the eigenvector to 1, and the ranks are given by the components of the normalized vector.

5.2.1 Applying Pagerank to Article Network 1

As described above, a small database of articles was compiled, each citing one of the others in the network. There were a total of nine nodes, and eighteen directed edges, and this data was represented by an excel data table which was exported to MATLAB for analysis. Many attempts were made trying to implement the pagerank algorithm to this network, but it was realized that the presence of dangling nodes and those with indegree 0 were throwing a wrench into the algorithm.

5.3 Thorny Issues With Pageranking Articles

As mentioned earlier, dangling and anchored nodes create problem for this article pagerank algorithm. In the case of dangling nodes, the matrix is no longer column stochastic. Instead, it is column substochastic, meaning the sum of the values in any given column is less than or equal to 1. This removes the guarantee that the matrix will have 1 as an eigenvalue. One may definitely cry alas! However, hope is not yet lost thanks to a striking theorem from linear algebra independently discovered by Oskar Perron and Georg Frobenius in 1907 and 1912 respectively. The theorem will be stated and applied in a later algorithm.

Anchored nodes lead to a more troubling problem. In this model, such nodes receive zero votes due to their not being cited. This causes their corresponding rows and columns in the rank coefficient matrix to zero out, making it futile to perform the eigenvector analysis as before. When this situation is encountered, one must find a way to modify the algorithm, or develop a scheme to artificially assign equal ranks to such nodes from the onset. This allows for a similar algorithm to be carried out which preserves informative relative rankings of the remaining nodes in the graph. After the ranks are computed, one can then reassign the rank of the zero indegree nodes using an appropriate scheme given the other rankings. Admittedly, this is not an ideal patch, but it does provide meaningful rankings. Furthermore, it makes sense that anchored articles will have equal or similar rankings given that neither is receiving a citation. They could be interpreted as new editions to the network and therefore have not been around long enough to be cited, or as articles with less than significant value. Thus, their true rank is indeterminate, but should be considered non-zero because they did manage to get published. Another approach aimed at eliminating this issue is the introduction of a damping coefficient in the pagerank algorithm which allows for

a baseline rank to be assigned to every node in the network. Anchored nodes then receive a nonzero rank and the aforementioned method can be applied in largely the same manner.

6 ArticleImport Algorithm for Ranking Articles

This algorithm was an early attempt to account for the occurrence of anchored nodes. It is meant to be replace the basic pagerank implementation. This algorithm deals with the presence of dangling and anchored nodes(articles) by assigning the latter a rank of 1. Therefore, anchored nodes have a unit rank to distribute evenly among the nodes they cite. This action uniquely defines the ranks for the rest of the network, thereby giving ranks to the dangling nodes, and allowing the matrix corresponding to the system of equations expressing node relationships to be placed in reduced-row echelon form. Solving this system of equations gives the article rankings, in this case before rank reassignment of zero indegree nodes. The rankings of nodes which were not artificially set to 1 allow determination of which articles are influential in the network. Note that this algorithm relies on solving a system of equations outright as opposed to the eigensystem discussed in the previous method. MATLAB was used to create ArticleImport. To create the matrix, MATLAB had to loop through the data in order to create the ordered pairs representing directed edges between two nodes. These ordered pairs were then used to create a matrix representing the system of equations based on the number of edges for each node. Unit rank was assigned to anchored nodes, the system solved, and the appropriate eigenvector normalized to 1 reported as the set of ranks of corresponding to each article.

6.1 Implementation

6.2 Conclusions Regarding Influential Nodes in Article Network 1

7 Construction of Article Network 2

In much the same way, a database of articles was created for a subnetwork of Erdos1. The search was laborious, seeking articles citing other articles authored by those within Erdos1 and also having authors connected by coauthorship. The search engine <http://citeseerx.ist.psu.edu> was utilized in the search, and the resulting article network was one of ten nodes, which then determined our subgroup of Erdos1 as a network of 11 nodes. An attempt was made to build upon an authorrank algorithm in order to better analyze these networks.

8 Modified Authorrank Algorithm

In *Co-Authorship Networks in the Digital Library Research Community* by Lui, Bollen et al., an algorithm is developed to rank authors within a coauthorship network. In this *authorrank* algorithm, a generalization of google's pagerank algorithm, strength of collaboration between authors is quantified using link weights in a symmetric (bidirectional) coauthorship network. Weighting the links in this way aims to weight the vote (endorsement) contributed by an author to a coauthor. Thus, an endorsement by a prominent author is worth more than an endorsement by a less prominent author. This approach is very promising, but there are other options for what can be used to set the link weight in a coauthorship network. If a network of articles is constructed which cite each other, and each article is authored by a member of the coauthorship network, a measure of work quality can be used to weight links between coauthors. Using the ArticleImport algorithm, weights (magnitude of importance) are assigned to all articles in a relative fashion. Then, for each author in the coauthorship network, the weight of its' outward links can be set as the average of the article weights for the articles that author helped write. Thus, an endorsement (coauthoring) from (with) an author who has produced many important articles will contribute more to the "endorsement receiving" author's rank than an endorsement from an author who has produced lower quality articles, or fewer high quality articles. Deciding when to give an author more endorsement to transfer to others based on the quality of his work seems to be preferable to deciding the level of endorsement based on the strength of the collaboration between two authors. MATLAB was used to create ArticleImport. To create the matrix, MATLAB had to loop through the data in order to create the ordered pairs representing edges between two nodes, then use this to create a matrix representing the system of equations based on the number of edges for each node. Arbitrary values were then assigned to nodes with indegree of zero, the system solved, and the appropriate eigenvector reported as set of ranks of corresponding to each article.

8.1 Implementation

Let $W_j \equiv r_j$ where the r_j is the ranking of the j th article computed using one of the aforementioned algorithms.

Let $A_n \equiv \{ \text{set of articles coauthored by author } n \}$ Let $Z_{mn} \equiv \frac{\sum_{j \in A_n} W_j}{|A_n|}$ where $|A_n|$ refers to the cardinality of A_n or the number of articles author n has coauthored in the article network corresponding to the coauthor network.

Then, the weight given to the link from author n to author m is Z_{mn} . Normalize these weights to 1, still denoting them Z_{mn} for simplicity of notation. Let $R_n \equiv$ the rank of author n .

Now, as in the basic pagerank algorithm, a system of equations can be produced

resulting in the eigensystem:

$$Zr = \begin{pmatrix} 0 & Z_{12} & Z_{13} & \dots & Z_{1n} \\ Z_{21} & 0 & Z_{23} & \dots & Z_{2n} \\ Z_{31} & Z_{32} & 0 & \dots & Z_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Z_{n1} & Z_{n2} & Z_{n3} & \dots & 0 \end{pmatrix} \begin{pmatrix} R_1 \\ R_2 \\ R_3 \\ \vdots \\ R_n \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \\ R_3 \\ \vdots \\ R_n \end{pmatrix}$$

Note that once again, the elements along the diagonal are all zero because coauthorship with oneself does not occur.

The matrix of coefficients Z is column-substochastic, which implies that it is no longer guaranteed to have an eigenvalue of 1. However, a column-substochastic matrix is guaranteed to have a Perron-Frobenius eigenvector corresponding to the dominant eigenvalue. Before stating this very important result, a few terms must be defined:

Definition Dominant Eigenvalue: Let $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n$ be the eigenvalues of an $n \times n$ matrix A . λ_1 is called the dominant eigenvalue of A if

$$|\lambda_1| > |\lambda_i| \quad \forall i = 2, \dots, n$$

The eigenvectors corresponding to λ_1 are called dominant eigenvectors of A .

Definition Irreducible Matrix: A matrix A is a nonnegative, irreducible $n \times n$ matrix if and only if

$$(I_n + A)^{n-1} > 0$$

Definition Simple Root: A root having multiplicity of 1.

Theorem 8.1. (Perron-Frobenius Theorem): *If A is an $n \times n$, nonnegative, irreducible matrix, then the following statements are true:*

- 1) *A is guaranteed to have a positive, dominant eigenvalue λ_1*
- 2) *There is a positive eigenvector corresponding to λ_1*
- 3) *λ_1 is a simple root of the characteristic equation of A*

Appealing to the Perron-Frobenius theorem, first calculate all of the eigenvalues of matrix Z . Find the dominant eigenvalue, i.e. the one with the largest magnitude. In the case of complex eigenvalues use the modulus. Then calculate the eigenvector corresponding to the dominant eigenvalue, the Perron-Frobenius eigenvector. Finally, normalize this vector to 1, and the author rankings are given by its' components.

9 Using Modified Authorrank Algorithm on Erdos1 Subnetwork

9.1 Dangling Nodes, Zero Indegree, and Perron-Frobenius Eigenvector Revisited

9.2 Applying Modified Authorrank Algorithm to Article Network 2 and Erdos1 Subnetwork

10 Utility of Modeling Influence and Impact Within Networks

11 Conclusion

Works Cited

Citeseerx. Web. 10 Feb 2014. <<http://citeseerx.ist.psu.edu>>.

Cytoscape Network Software

Matlab 2012b, The MathWorks, Inc., Natick, Massachusetts, United States.

"THE PERRON-FROBENIUS THEOREM." Prentice-Hall Inc.. Prentice-Hall Inc., n.d. Web. 09 Feb 2014.

<<http://www.prenhall.com/divisions/esm/app/ph-linear/leon/html/perron.html>>.

RStudio

Xiaoming, Liu. "In much the same way, a database of articles was created for a subnetwork of Erdos1. The search was laborious, seeking articles citing other articles authored by those within Erdos1 and also having authors connected by coauthorship.

The search engine <http://citeseerx.ist.psu.edu> was utilized in the search, and the resulting article network was one of ten nodes, which then determined our subgroup of Erdos1 as a network of 11 nodes.." Elsevier Science. (2008): n. page. Print.