“Φασματικά μέτρα κεντρικότητας και η εφαρμογή τους στην επιστημονομετρία”

Γιώργος Σιδέρης, Πανεπιστήμιο Θεσσαλίας, Φεβρουάριος 2018.

Περίληψη

Σε αυτή την διπλωματική εργασία συνέχεια του ειδικού θέματος που υλοποιήθηκε στο Χειμερινό εξάμηνο 2016/17 με την επίβλεψη του Επίκουρου Καθηγητή Δ. Κατσαρού. Η διπλωματική εργασία επιβλέπεται και αυτη από τον κ. Κατσαρο.

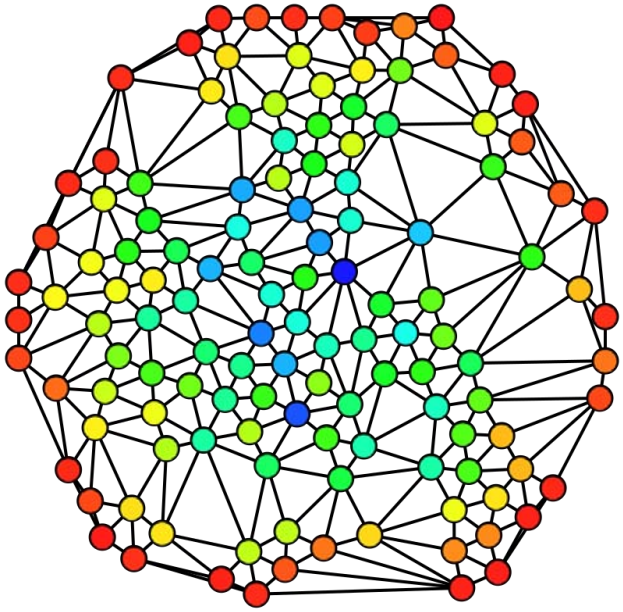
Σε αυτό το έγγραφο, το οποίο είναι γραμμένο στην Αγγλική γλώσσα, παρουσιάζονται τρεις διαφορετικοί αλγόριθμοι κεντρικότητας για κατάταξη επιστημόνων σε ένα ερευνητικό πεδίο, χρησιμοποιώντας τις δημοσιεύσεις τους (άρθρα, papers) και τις αναφορές αυτών. Παρουσιάζεται ο τρόπος που λειτουργούν, ο ορισμός τους και γίνονται για τον καθένα πειράματα, με την ίδια είσοδο. Η υλοποίηση των αλγορίθμων έγινε σε Matlab.

Έπειτα, τα αποτελέσματα αυτά συγκρίνονται μεταξύ τους, οπτικά και με τον δείκτη συσχέτισης και χρήσιμα συμπεράσματα αποκομούνται για τον τρόπο λειτουργίας τους, την αποδοτικότητα και την ορθότητα τους.

Τελικά, προτείνεται και ένας νέος αλγόριθμος για κατάταξη επιστημόνων, που επιχειρεί να λύσει κάποια από τα προβλήματα που σημειώνονται στους υπόλοιπους.

Εν κατακλείδι, τα αποτελέσματα , με την ίδια είσοδο, του αλγορίθμου συγκρίνονται με τα προηγούμενα, και χρήσιμα συμπεράσματα βγαίνουν για την βέλτιστη χρήση των αλγορίθμων και αναλύεται ποια είναι η πιο ορθή μέθοδος σε κάθε περίπτωση.

UNIVERSITY OF THESSALY



Spectral Centrality Measures in Multilayer Networks and their application in Scientometrics

A brief summary, comparison of the existing algorithms and proposal of an alternative solution

Supervised by Dimitrios Katsaros

GIORGOS SIDERIS

Department of Electrical & Computer Engineering
UNIVERSITY OF THESSALY
Volos, Greece 2018

Master's thesis 2018

Examination of centrality algorithms as a mean of evaluation of scientists

A brief summary, comparison of the existing algorithms and proposal of an alternative solution

GIORGOS SIDERIS

Department of Electrical and Computer Engineering
University of Thessaly

Department of Electrical & Computer Engineering
University of Thessaly
Volos, Greece 2018

Examination of centrality algorithms as a mean of evaluation of scientists
A brief summary, comparison of the existing algorithms and proposal of a more accurate solution
GIORGOS SIDERIS

Supervisors: Dimitris Katsaros, Manolis Vavalis

Department of Electrical & Computer Engineering
University of Thessaly
Argonafton & Filellinon 38221 Volos
Telephone +30 24210 74000

Cover: An undirected graph colored based on the betweenness centrality of each vertex from least (red) to greatest (blue).

Typeset in LaTeX
Volos, Greece 2018

Abstract

In this thesis, the use of various centrality measures is examined, some of which are commonly used in the scientific community, in multilayer networks and the application they can have in scientometrics. Scientometrics is the study of measuring and analysing science, technology and innovation. Three algorithms are used in this article and these are by order of testing: Biplex PageRank, H-index and C^3-index. Each of these has its advantages and special uses, however the the the best method of these three is defined and a new algorithm for a more accurate and efficient way of ranking scientists and researchers is proposed.

Keywords: centrality, scientometrics, multilayer, networks, pagerank, ranking, researchers, papers , articles, measurement

IEEE Keywords: Algorithms, Data Processing, Data Analysis, Iterative algorithms, Linked Data, Ranking (statistics), Research and development, Scientific computing

Acknowledgements

This is dedicated to my family, for always supporting me and being there for me.

I would like to thank my supervising professor Dimitrios Katsaros, for his constant help on the production of this project, Antonis Sidiropoulos for providing the datasets which were used.

Special thanks to Fotis Tsokos, student of University of Thessaly, for providing feedback on the project and the current document and Chris Marinos for providing support and feedback

Giorgos Sideris, Volos, February 2018

Contents

List of Figures

List of Tables

1

Introduction

1.1 Significance of this algorithmic research

Long before the era of Internet and the overload of information, a scientist's credibility and prestige played an important role on the importance of his findings and articles.

Nowadays, with the Internet and the globalization of the scientific community, innumerable articles are available for a wide range of topics and it's now more important than ever for the reader to be able to distinguish the most important authors and the most important articles.

A common way to identify the best of the best, is to make a ranking with the most important authors and articles. Reviewing the citations of each paper, we can rank them for their credibility.

Furthermore, it's vital to identify the leading researchers in the world based on their impact and scientific value.

1.2 Background

The precursor of this thesis was implemented by Giorgos Sideris, with the guidance and supervision of Dimitrios Katsaros, and it was titled "*Node ranking in bilayer networks using Biplex PageRank*", where Biplex Pagerank was implemented in MATLAB.

This was a project for the University of Thessaly in the first semester of 16/17 academic season.

Biplex Pagerank was implemented and tested for various networks, but in this thesis it is optimised and tested with coauthorship and citation networks.

For more information on this project, see "*Appendix 1: Previous project & Accuracy of Biplex PageRank compared to single-layer Google PageRank*"

1.3 Contribution

This thesis provides multiple implementations of some of the most promising author-level metrics for scientific publications.

Biplex Pagerank, H-index, C^3-Index were analyzed and implemented in MATLAB and are available for use under the MIT License.

Also, a new algorithm is suggested - the C^4-Index - which combines the benefits of C^3-Index and Biplex Pageranks and intends to provide an alternative solution to the other metrics while examining more factors.

The code is available in Github on: https://github.com/siderisng/multilayer-centrality

1.4 Thesis outline

Throughout this thesis, a incremental way of thought is presented.

1. First of all, in the second chapter, the methods of ranking scientists for the value of their publications are presented
2. In the third chapter, the test inputs (datasets) used for this thesis are presented along with the method of extracting them
3. In the fourth chapter, Biplex Pagerank is introduced, including its definition, calculation method, and the test results
4. In the fifth chapter, H-index is introduced
5. C3-Index is presented
6. In chapter VII, the technical aspect and challenges of this thesis are noted
7. In chapter VII, the technical aspect and challenges of this thesis are noted
8. Afterwards, technical and visual correlation between the test results is examined
9. Observations concerning this correlation and the efficiency of the algorithms are presented
10. In chapter X, a new algorithm is suggested
11. New observations are made basen on the new algorithm results
12. Chapter XII, includes all the final conclusions deducted from this research
13. Future research steps are discussed and suggested
14. Appendices containing important notes, tools
15. References / Bibliography

2
Ranking

The first way to rank them is by the quantity of citations of each paper. The more citations a paper has, the higher it ranks.

Another way is to measure the quality of the citations. Citation quality can be determined by the ranking of the citing paper/author. The bigger ranking a paper has, the more value its citing has. This type of ranking is accomplished using iterative procedures. The procedure starts with every author having the same rankings and as iterations complete the highest are distinguished.

Of course these 2 methods can be combined, to maximize an algorithm's accuracy defining the best authors.

3

Inputs

The main input for the tests applied on all algoriths was a dataset extracted[4] from MAS (Microsoft Academic Search). The selected field was Computer Science and the top 500 authors were extracted according to MAS's ranking (and all of the others who had interacted with these top 500 eg. being cited by them). The actual number of authors was **50601**, who collectively published **13566** articles. Finally, there were **252142** citations in total.

Three files were used of the dataset. The first one contained the coauthorship details of every article published by every scientist in the dataset, the second one contained the citations between any of the aforementioned articles and the third one the names of the first 500 authors, for display purposes.

Using all this information of the dataset, all the needed adjacency matrices for the algorithms were able to be created. Also all the id's of the authors/papers were mapped to achieve faster results (see more in chapter "*VII - THE TECHNICAL ASPECT AND THE CHALLENGES*").

It was really important that a big enough dataset was used, so that the results could be more reliable when comparing between algorithms and eliminate the chances of abnormal values.

Furthermore, Antonis Sidiropoulos[4] needs to be mentioned for his help in providing the datasets.

4

Biplex Pagerank

It is widely know that Google uses an algorithm called "PageRank" to rank websites in its search engine. A paper titled "*A biplex approach to PageRank centrality: From classic to multiplex networks*"[1] written by Francisco Pedroche, Miguel Romance and Regino Criado et al. suggests that PageRank can be extended to be implemented in multilayered networks and distinguishing the most important nodes.

First some important terms used below for the biplex PageRank need to be defined.

4.1 Important note

It has been already proved that Biplex Pagerank is a valid pagerank algorithm for multilayered networks. On tests conducted in the Special Project at University of Thessaly, the algorithm was implemented into Matlab and it was proved that Biplex Pagerank returns the same results as single-layer Pagerank. For more information on that matter, turn to "Appendix 1: Accuracy of Biplex PageRank compared to single-layer Google PageRank"

4.2 Glossary

- **Virtual and Real Nodes**: Each layer contains node from both layers. Nodes that are actually in that layer (they are included in any link) are now called Real and the rest are called Virtual

- **Dangling Nodes**: Nodes that have no outgoing links to other nodes

- **Personalization Vector**: The vector that contains the probability of "jumping/teleporting" from one node to the other (without a link between them)

- **Tolerance**: We define as tolerance the biggest difference of PageRank values between two iterations , before we can say that the iteration converges and stop the calculation

4.3 Calculation

1. Initialization of arrays and parameters
2. Calculation of Pa (adjacency matrix)
3. Calculation of Real and Virtual Nodes
4. Extension to calculate Biplex PageRank in a network containing Dangling Nodes
$$Pa = Pa + du^T \tag{4.1}$$
5. Calculation of v (personalization vector)
6. Initialization of pagerank and tolerance
7. Iterative procedure

The algorithm was used to identify the leading researchers. The method of creating the networks for this calculation was as follows.

Given two source files, with one containing the paper and its authors and the other containing the paper citations:

- There were two layers: Authors and Papers
- Bidirectional links were created between co-authors
- Bidirectional links were created between Authors and their Papers
- Directional links (Paper to Paper) based on paper citations
- Directional links (Author to Author) based on paper citations from every author of the citing paper to every author of the cited paper

Definition: Given a biplex network G containing $n \epsilon \mathbb{N}$ nodes, and adjacency matrices Pa and Pa_2, then:

$$P = \frac{1}{2} \cdot (Pu + Pu_2 + Pd + Pd_2) \epsilon R_n \tag{4.2}$$

Calculation of Pu, Pu_2, Pd, Pd_2 :
With iterative procedure

$$\begin{aligned} 2Pu^T &= Pu^T \cdot a \cdot Pa + Pu_2^T + 2 \cdot a \cdot Pd^T \\ 2Pu_2^T &= Pu^T + Pu_2^T \cdot a \cdot Pa2 + 2 \cdot a \cdot Pd_2^T \\ 2Pd^T &= (1-a) \cdot (Pu_2^T + Pd^T \cdot e \cdot v^T + Pd2 \cdot e \cdot v^T) \\ 2Pd_2^T &= (1-a) \cdot (Pu_2^T + Pd^T \cdot e \cdot v_2^T + Pd_2 \cdot e \cdot v_2^T) \end{aligned} \tag{4.3}$$

The initial values of Pu, Pu_2, Pd, Pd_2 are for every element x of any of these vectors: $x = \frac{1}{2n}$, where n the amount of nodes in the network.

4.4 Denotations

Personalization Vector: A vector representing the possibility of "teleportation" from one node to another (without needing an actual link)
Dangling Node : A node without any outgoing links
a : The algorithm's damping ratio. eg. if it is 0.5 , there is an equal change of a "user" transferring to another node via "teleportation" and transferring via a link. In the test ran, a equals 1. (no teleportation chance)

e : A vector $(1, ...1)^T$ with n length (n nodes in the network)
$\mathbf{K}_{out}$: A vector representing the amount of outgoing links for each node
Pa: A n^2 matrix where $Pa(i, j)$ is defined as follows:
$Pa(i, j) = 0$, for every j if i is dangling node
$Pa(i, j) = \frac{1}{K_{out}(i)}$, else
u : the possibility distribution of the dangling nodes
d : vector with n length where:
$d(i) = 1$, if i is dangling node
0 , else
Pu : Pagerank for the "real" network
Pd : Pagerank for the "teleportation" network

4.5 Tests

The test was ran successfully using the input mentioned before (in chapter "III - TEST INPUTS") from MAS. The elapsed time was 23600.033 seconds. The top 25 authors are as following (papers' ranks are filtered out of the results)

Author unique ID	Final pagerank	Name
2037300	0.0000000363	
2074100	0.0000000358	
1719800	0.0000000356	
1474100	0.0000000354	
131520	0.0000000351	
2209700	0.0000000350	
180290	0.0000000350	
1028700	0.0000000350	
73121	0.0000000346	Yehoshua Sagiv
1142000	0.0000000346	
1886100	0.0000000345	
195120	0.0000000345	
254600	0.0000000344	
539830	0.0000000343	
80996	0.0000000343	Arun Swami
1061000	0.0000000343	
1166300	0.0000000342	
1551300	0.0000000341	
305440	0.0000000341	
939670	0.0000000340	
911160	0.0000000339	
778150	0.0000000339	
1480700	0.0000000338	
74920	0.0000000336	Catriel Beeri
399230	0.0000000336	

Table 4.1: Biplex Pagerank Scores on MAS Dataset

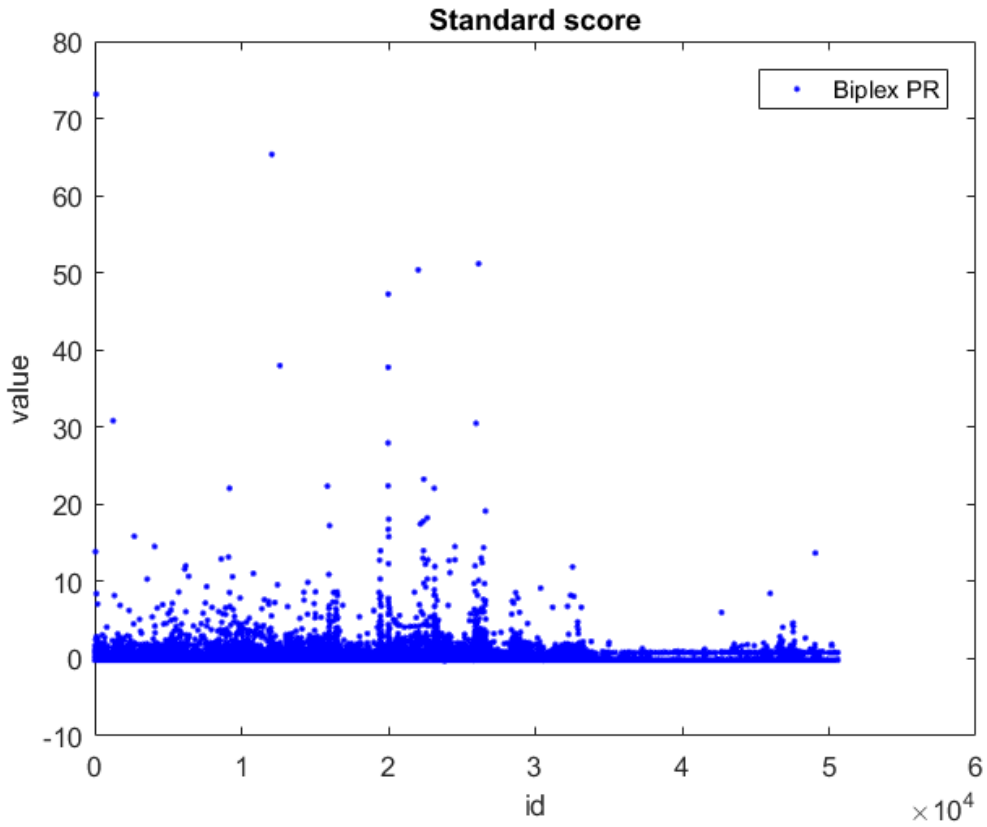


Figure 4.1: The distribution of the final Biplex Pagerank values for each node

5

H-Index

The H-index is an author-level scientific metric that attempts to measure both the productivity and citation impact of the publications of an author. This metric suggests that the quality of a scientist or scholar can be determined by a set of his most cited papers and the citations they have receivd in other publications. The algorithm was suggested in 2005 by Jorge E. Hirsch, a physicist at UCSD, as a tool for determining theoretical physicists' relative quality[2] and is sometimes called the Hirsch index or Hirsch number.

5.1 Calculation

1. Get every authors papers and citations of each paper
2. Sort number of citations of each author in descending order
3. For every author start a loop checking for each paper's citations if their index of paper citations is equal to or higher than the number of citations of this paper, with the last index this is true being their h-index.

   For example a scientist with an ordered set of citations $c = 56, 25, 4, 2, 1$ has an h-index of 3 because $c(3) = 4$ and $c(4) = 2$

5.2 Definition

Given an author's publications number and the number of citations of each publication sorted in descending order (let it be c), their h-index is the number of papers each of which has been cited in other papers at least h times.

$$H\text{-}index(j) = max(min(c(i), i)) \tag{5.1}$$

5.3 Tests

The test was ran successfully using the input mentioned before (in chapter "*III - TEST INPUTS*") from MAS. The elapsed time was 6.4 seconds. The top 25 authors are as following:

Author's unique ID	Final H-Index	Name
2037300	57	
131520	52	
1480700	47	
180290	46	
1545100	44	
767320	43	
2074100	42	
354310	42	
1886100	41	
1474100	41	
1061000	41	
111060	41	
1551300	39	
254600	38	
778150	37	
74920	37	Catriel Beeri
73121	37	Yehoshua Sagiv
2390000	35	
539830	35	
1329	35	Philip Yu
300420	34	
195120	33	
85097	33	Divesh Srivastava
2209700	32	
973480	32	

Table 5.1: H-Index Scores on MAS Dataset

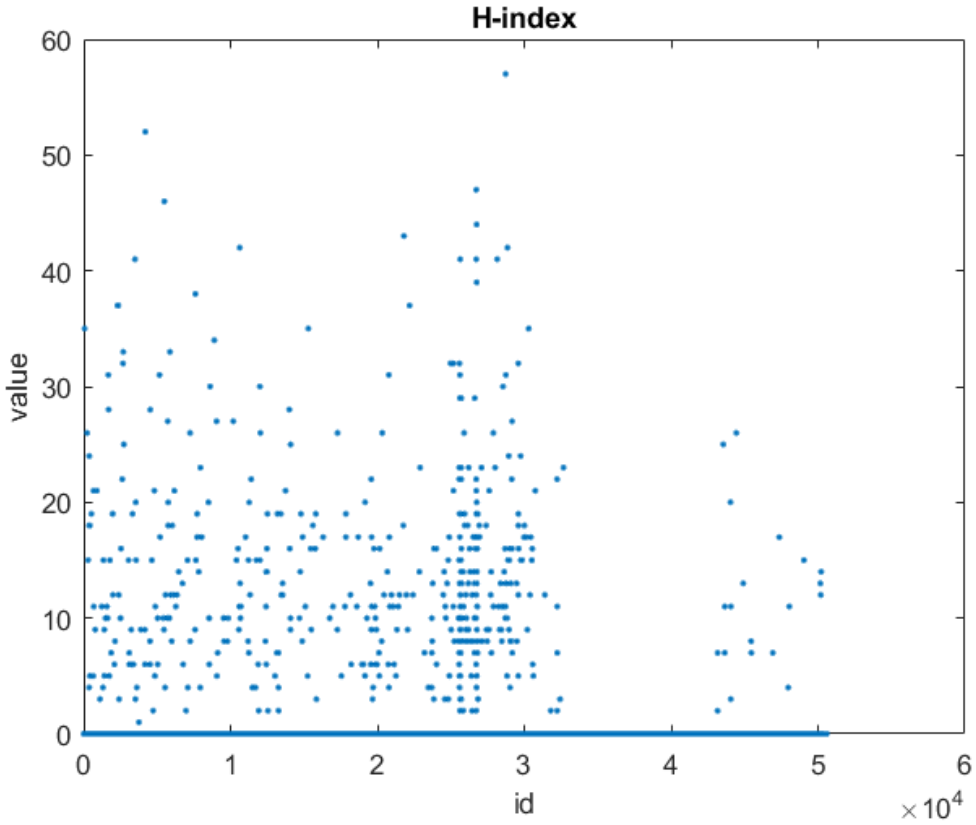


Figure 5.1: Distribution of the final H-index values for each node

6

C^3-Index

C^3-index is another metric proposed for ranking scientists based on their publications. It was presented in "*C^3-index: A PageRank based multi-faceted metric for authors*" performance measurement" by Pradhan et al [3]. The basic idea is that other scientific index metrics like h-index produce great results in highly-cited scientists, but lack in ranking medium and low cited ones. C^3 promises more accurate results for these types and identifying researchers with a promising future ahead of them.

6.1 How it works

C^3-index combines 3 different metrics and produces a ranking containing more information for each author. The three metrics are:

- ACI - Author citation Index
- PCI - Paper citation Index
- AAI - Author coAuthorship Index

ACI correlates strongly with h-index (up to 90%), but PCI and AAI correlate less (40-50%). This shows the added information the C^3-index encapsulates. It is a very promising method for ranking scientists and showing the ones with a great potential in their work.

6.2 The network model

The C^3-index is developed on an underlying multi-layered citation-collaboration network model described in Figure 2, where three layers from left to right correspond respectively to author-author citation network, author-author coauthorship network, and paper-paper citation network

6.3 Creating the networks for testing

Given the paper-paper citation and co-authorship paper information:

1. Create undirected weighted links in Author Coauthorship layer for authors who coauthored a paper together , with the weight being the number of papers these two published together
2. Create directed unweighted links between paper using the Paper Citation information (which paper cites another paper)

3. Create undirected unweighted links between authors in Author Coauthorship Layer and papers in Paper Citation Layer linking every author with every paper they published
4. Create directed weighted links in the Author Citation Layer linking every author of the citing paper to every author of the cited paper (directed to the cited ones)

6.4 Calculation

Using iterative procedure:

$$C_j^3(t) = (1-\theta) + \theta * (ACI_j(t) + AAI_j(t) + PCI_j(t)) \tag{6.1}$$

$$ACI_j(t) = (1-\theta) + \theta \cdot \sum_{A_k \in C(A_j)} \frac{ACI_k(t-1)}{outdeg(A_k)} \tag{6.2}$$

$$AAI_j(t) = \sum_{A_k \in CA(A_j)} \frac{AAI_k(t-1)}{deg(A_k)} \tag{6.3}$$

$$PCI_j(t) = (C_j^3(t-1))^\alpha \cdot \sum_{P_k \in C(P_i)} \frac{PQI_k(t-1)}{\sum_{A_l \in A(P_k)} (C_l^3(t-1))^\alpha} \tag{6.4}$$

$$PQI_i(t) = (1-\theta) + \theta \cdot \sum_{P_k \in C(P_i)} \frac{PQI_k(t-1)}{outdeg(P_k)} \tag{6.5}$$

6.5 Denotations

$C(A_j)$ denote the set of authors who cited at least one paper of author A_j , $CA(A_j)$ denote the set of authors who coauthored with author A_j in at least one paper, $outdeg(A_k)$ denotes the sum of the degrees of the out-going edges from node A_k in the author-author citation layer of the network, $deg(A_k)$ denotes the sum of the degrees of the edges incident on node A_k in the author coauthorship layer, and θ is the damping factor for the PageRank based strategy.
t and $t-1$ represent times. t represents the current iteration's time, $t-1$ the previous one's.

6.6 Definition

C^3-Index is a PageRank based multi-faceted metric for authors' performance measurement that combines the effect of citations and collaborations of an author in a systematic way using a weighted multi-layered network to rank authors.

6.7 Tests

The test was ran successfully using the input mentioned before (in chapter "*III - TEST INPUTS*") from MAS. The elapsed time was 23508.04 seconds. The top 25 authors are as following:

Author's unique ID	Final C3-Index	Name
3313700	186.5400000000	
147670	115.2400000000	
594570	107.5400000000	
1630000	68.1270000000	
195120	65.3410000000	
1453400	62.9800000000	
354310	58.9980000000	
2037300	58.1240000000	
1545100	53.2170000000	
664070	51.3300000000	
1500900	48.9380000000	
7004100	48.9230000000	
131520	47.2670000000	
2209700	45.2950000000	
1418900	42.7800000000	
28553	42.4170000000	W CROFT
511580	41.6590000000	
1480700	40.8480000000	
1122800	40.2580000000	
2041800	39.4880000000	
1552500	39.4480000000	
290490	39.3230000000	
307980	39.1710000000	
56949	37.4880000000	D MCLEOD
300420	37.3720000000	

Table 6.1: C^3-Index Scores on MAS Dataset

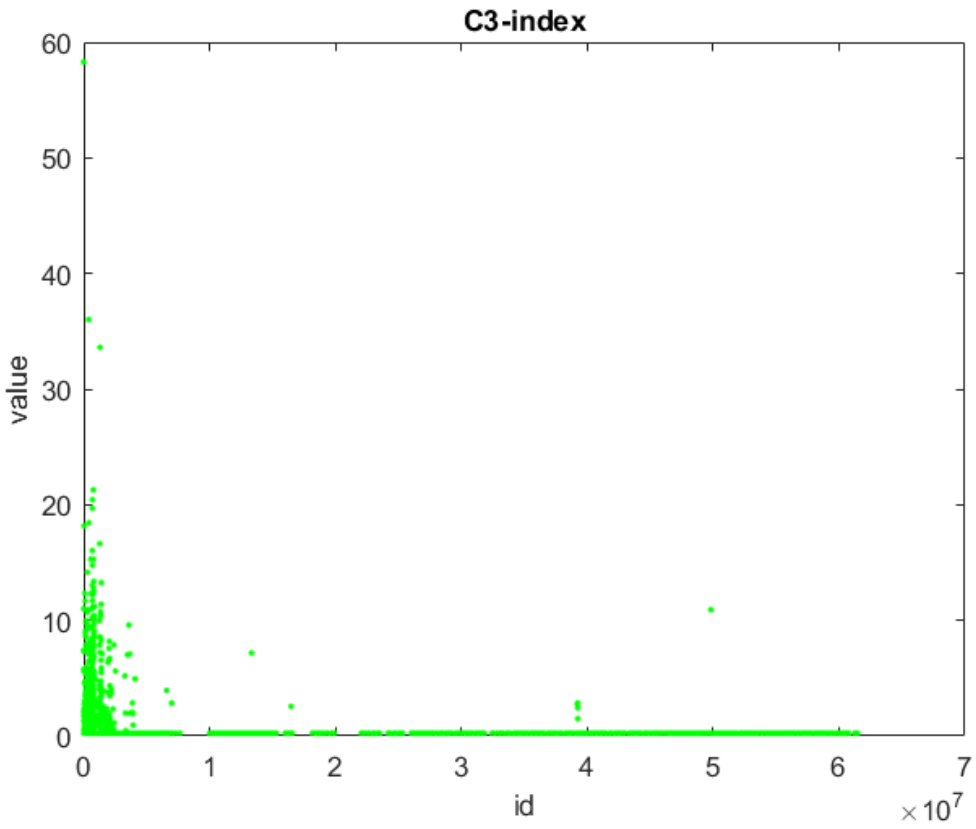


Figure 6.1: Distribution of the final C^3-index values for each node

7

The technical aspects and the challenges

7.1 Coding

The algorithms were implemented in MATLAB. MATLAB provides many useful tools for handling arrays, delimeter seperated files, plotting figures, and performing computations for getting some important measures we used.

Furthermore, MATLAB is efficient when it comes to handling large arrays and provides data structures like sparse matrices that were frequently used in this project.

7.2 Size of required data

The first encountered when running the algorithms with the MAS dataset, was that there were matrices that needed to be created that were too large to fit in the computer's memory (and in any modern personal computer's memory).

For example, Biplex Pagerank required $4N_2$ adjacency matrices, where N is the number of nodes in the test, in the aforementioned case $62 \cdot 10^6$. So the required space in RAM only for these 4 adjacency matrices would be $4 \cdot 62 \cdot 10^6 \cdot 62 \cdot 10^6 \cdot 4$ Bytes (this translates to 4 matrices $\cdot N \cdot N \cdot$sizeof(float) which results to 52 PetaBytes! Taking a closer look though, we realize that there is not a link from every node to every other node, not even a big percentage of links are connected with the average node. So storing adjacency matrices with many "empty" values, even if it was feasible, it would not be efficient.

At this point, MATLAB's sparse matrices were used. Sparse Matrices [7] store only the non-zero values of a matrix, therefore saving storage space.

After this change, the adjacency matrices were able to fit in the computer's memory, with relative ease. The structure of sparse matrices though, made speed a big issue, as explained below.

Example: Converting a full MATLAB matrix to a sparse one

```
% M:   full Matrix
```

SparseM = **sparse** (M)

7.3 Speed

As mentioned above, sparse matrices are not as efficient as "full" (not sparse) matrices, and this is understandable, if one understands the underlying technology in them.

As opposed to "full" matrices, sparse matrices don't store zero values. This means that instead of creating N array elements and saving each value to its corresponding index, a list containing each non-zero element's index and value is saved. This makes indexing exponentially slower. For indexing a single element, the whole list might be searched (one by one element) and storing a new value can be even less efficient.

The strategy followed in this project was as follows: sparse arrays where used only when this was the only option of running the test, without running out of memory.

However, algorithms like C^3-Index , which required various and frequent accesses (read/write) to sparse matrices, required not feasible times to run (days/weeks)

But the sheer amount of the dataset still made tests not feasible, so the next step that was taken was, the filtering of the dataset to a smaller, more manageable one. Having the information of the top 500 authors in MAS, every interaction/link that was not associated with the 500 authors and anyone that had at least one link with them or their paper, was filtered out.
Previous file sizes: 195MB + 78MB
File sizes after filtering: 5.2MB + 3.8MB

Furthermore, one issue still affecting performance were the potential cache misses in the referencing of the matrix elements. Sparsely populated matrices, produce more misses because when an element is referenced, this element and the neighbouring ones are brought to the cache, for faster future referencing. In matrices with many empty values between elements, the use of caching is not optimal, and every reference requires a memory read and transfer to cache. So, the action taken was mapping the ID's to the smallest ID available and recreating the dataset, but this time with less empty space in it. This improved the performance significantly, and was implemented by helper function CountUnique, explained at Appendix 2 below. After completing the tests, the results had to be traslated back to their original ID's.

The last step taken to overstep this issue, was a careful understanding of the code wrote, and in nested loops, where in the inner loop only a row/column of the sparse matrix was referenced. So, before accessing the inner loop, the matrix row/column at need , was converted to a full vector for faster referencing. This made a defining change in the efficiency (100x faster for the MAS dataset) of C3-Index.

Below is an example of this practice:
Previous code:

```
% Pa3 : Sparse Matrix (N*N)
for j = 1:n
    sum = 0;
    for k= 1:n
        if Pa3(k,j)~=0 % find a paper that cites j'th paper
            sum = sum + prevPQI(k)/Kout3(k);
        end
    end
    PQI(j) = (1-theta) + theta* sum;
```

After optimization:

```
for j = 1:n
    sum = 0;
    Pa3k = full(Pa3(:,j));
    for k= 1:n
        if Pa3k(k)~=0 % find a paper that cites j'th paper
            sum = sum + prevPQI(k)/Kout3(k);
        end
    end
    PQI(j) = (1-theta) + theta* sum;
end
```

7.4 Correlation

Another reasonable question after implementing the three algorithms would be how would the correlation between the algorithms be measured.

A common practice is using Pearson Correlation Coefficient[5], which is explained briefly above and it was the method used for these experiments.

A helper function (*getCorrCoeff2D.m*) was implemented in MATLAB, which uses built-in function corr2[8]

7.5 Definition of a good method

There is not a universally approved method of ranking scientists according to their publications at this moment. Many favour some metrics more than others. H-Index, for example, is recognized as a valuable metric for some of the scientific community, but its weaknesses are presented this document, as well as in others.

Having that is mind, it is understood that there's still a lot of research in this

particular field, to better define what is considered an accurate method of ranking authors of scientific publications.

In this thesis, we try to logically prove that some algorithms, have more value than others and there is also a comparison between their results, their efficiency, their strengths and weaknesses.

Also, we suggest a new method for better results and no interest in efficiency. Better results ,in this document, are defined as:

- Attention to consistency of authors
- Attention to outstanding articles
- Ability to rank lower and middle-ranking scientists better, therefore recognizing the researches with a high potential

8

Correlation

8.1 Visual Correlation

For the visual comparison of the algorithm results, the Standard Scores of each author were used.
Standard Score (z) stands as : $\zeta = \frac{\chi - \mu}{\sigma}$, where μ is the mean value of scores, and σ is the standard deviation of scores.

8.1.1 Biplex / H-Index

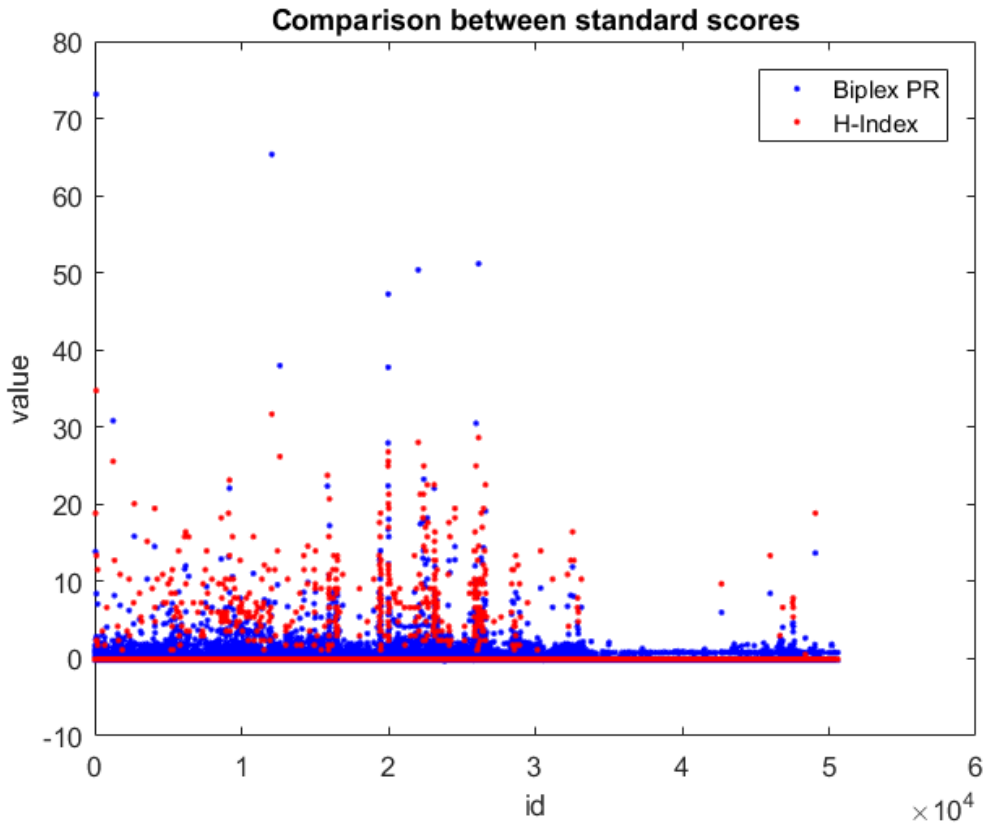


Figure 8.1: Visual comparison of Biplex Pagerank and H-index results

8.1.2 Biplex / C^3-Index

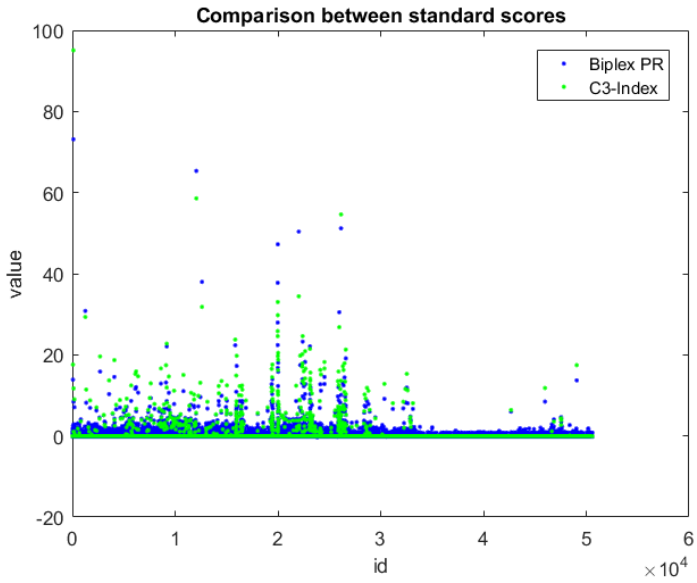


Figure 8.2: Visual comparison of Biplex Pagerank and C^3-index results

8.1.3 H-Index / C^3-index

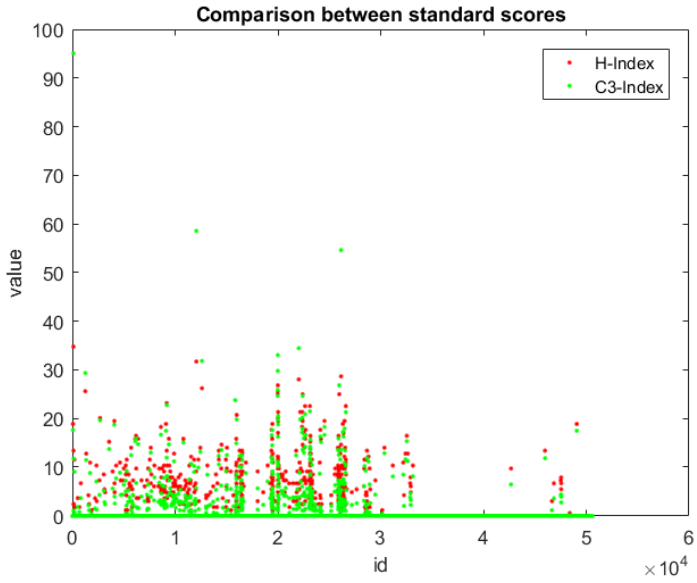


Figure 8.3: Visual comparison of C^3-Index Pagerank and H-index results

8.1.4 Biplex / H-Index / C^3-Index

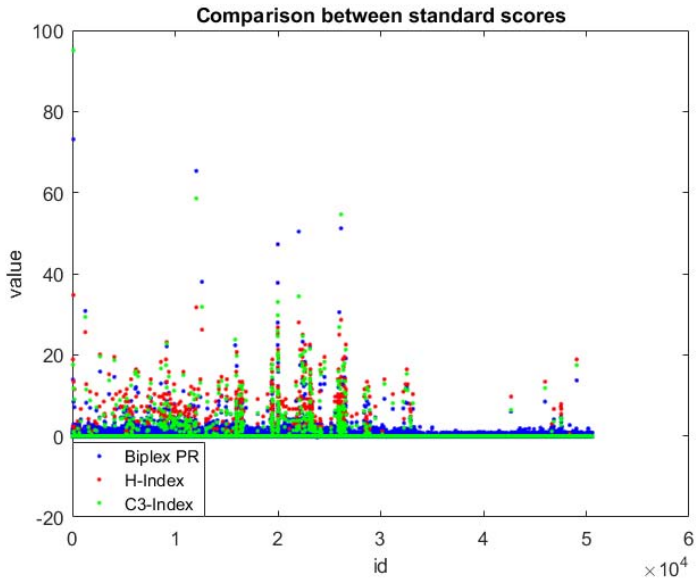


Figure 8.4: Visual comparison of Biplex Pagerank, H-index and C^3-Index Pagerank results

8.2 Correlation Coefficients

As a first measure, **Pearson correlation coefficient** (PCC) was used, which returns the correlation between two vectors.

Pearson correlation coefficient is a measure of the linear correlation between two variables X and Y. It has a value between +1 and -1, where 1 is total positive linear correlation, 0 is no linear correlation, and -1 is total negative linear correlation. It is widely used in the sciences. It was developed by Karl Pearson from a related idea introduced by Francis Galton in the 1880s. [5]

Correlation between Biplex PR and H-index is: 0.132232

Correlation between H-index and C^3-index is: 0.748588

Correlation between Biplex PR and C^3-index is: 0.101006

9

Observations

9.1 Strong Correlation between H-Index and C^3-Index

As noted in Chapter "VI - C^3-INDEX", strong correlation between the two algorithms was expected. And after the correlation coefficients between each algorithm result values were evaluated, it is clear that "Correlation between H-index and C3-index is: 0.748588".

This is because of the strong correlation of between AAI (Author citation Index) which is one the 3 metrics C^3-Index uses (~90%). The overall correlation is ~75%, because of the other 2 metrics: PCI (Paper citation Index) and AAI (Author coAuthorship Index), which return better scores for medium and low ranked researchers - that translates to better rankings for younger scientists and larger possibility to identify scientists with a high potential for the future.

Various authors being in the top 25 for each of these two algorithms can be observed:

Unique ID	**# C^3-index**	**# H-index**	**# Biplex PageRank**
195120	6	23	12
354310	8	9	699
2037300	9	1	1
1545100	10	6	32
2209700	15	25	6
1480700	19	4	23

Table 9.1: Common high ranking of authors on both H-Index and C^3-Index

Having that in mind, the elapsed times for each algorithm were compared, using the same dataset.

C^3-index	**H-Index**
23508.04 s	6.4 s

Table 9.2: Time comparison of C^3-Index and H-Index

It is clear that H-Index was ~3673x faster for this dataset, and the results have a relatively strong correlation.

The first observation that can be made is that:

On a large dataset, if the ranking of the most important and well-established authors is the desirable result and not the potential of young researchers, using the **H-index** algorithm provides much more **efficient** and relatively **accurate** results. Also, H-index could be useful for implementations, where we need fast results for the ranking of scientists, without need for total accuracy eg. a Social Network of scientists. Finally, one circumstance where H-index could be useful is the case where we have only the basic information about the authors (how many publications and how many citations they have), whereas C3-Index needs much more information.

On the other hand, if looking for the most promising scientists is what we look for and performance is not as important as accuracy, C3-Index performs better in that context.

9.2 Biplex PageRank produces significantly different results with attention to outstanding articles

In this subsection, the actual train of thought until the actual conclusion is presented. Outstanding articles are articles that have a significantly larger number of citations than the average, so they can be considered as widely acknowledged and recognized.

Comparing Biplex PageRank results to those of those of C^3-Index and H-Index, using the visualization or the correlation coefficient or by simply taking a look at ranking of particular scientists in these 3 results, what can be noticed is that Biplex Pagerank produces different results.

The next goal is to evaluate how accurate the results are and what makes these results different. Maybe this algorithm is not suited for ranking scientists, it could rank some particular set of scientists higher. The first assumption is that Biplex Pagerank is not as accurate as the others, beacuse it isn't designed with this implementation in mind.

This derives from the fact that H-Index is considered an accurate metric in the scientific community , and C^3-Index has strong correlation with H-Index and because of the larger amount of information it requires and the well-known accuracy for Medium and Low ranking scientists, can be considered even more "accurate".

Also, the correlation coefficients of Biplex PageRank compared to the others are compared and it is observed that with H-index, it is 0.13 and with C^3-Index it is 0.10.
Having in mind C^3-index as the most “accurate” algorithm and Biplex Pagerank the less “accurate”, the correlation coefficients seem to satisfy the initial assumptions.

At this point, some important **notes** need to be recorded:

1. There is no universal approved metric of scientists’ value. The word accurate often comes in “” because it could have many different meanings. In this article, we assume C^3-Index to be the most accurate, considering the strong correlation with H-Index , which is considered in the scientific community as an accurate metric and the fact that it performs better in identifying promising scientists, which is considered as an important factor for its accuracy

2. Biplex PageRank was proposed as a Centrality measure for multilayered networks and it was used for the purposes of this article for evaluating scientists’ significance (more information on how the layers were created on "*Chapter IV - BIPLEX PAGERANK*".

3. Biplex PageRank ranks papers as well as authors, as these are all considered nodes of the network

With these two in mind, it is known that Biplex PageRank creates links between Authors, between Papers, and between Papers/Authors and Authors/Papers. We also consider that in the context of Biplex Pagerank, these links have the same importance among them, where as the other two algorithms are designed specifically for scientists’ evaluation. So the deduction to be made, is that **Biplex PR pays more attention to the importance of links and not at their consistency**, compared to the other two.

To further explain that, H-Index and C^3-Index check for consistency in a scientist’s work, eg. a scientist that had a good amount of citations for each of each paper, would rank higher that a scientist that has composed some of the most cited articles but the rest of his articles are not so well received/cited.

On the other hand, Biplex PageRank is built on the foundation of centrality algorithms, like PageRank, so a "link" to a really important node can skyrocket a scientist’s score.

Of course, this needs to be proven, so it’s important a closer look is taken.

A first glance is taken at how the top 10 scientists in Biplex PR, rank in the other two algorithms, and what is the common factor between these scientists.

ID	#Biplex - Score	#H-Index - Score	#C^3 - Score	Author name
2037300	1 - 0.0000000363	1 - 57	8 - 58.124	
2074100	2 - 0.0000000358	7 - 42	42 - 31.9900	
1719800	3 - 0.0000000356	63 - 23	188 - 11.333	
1474100	4 - 0.0000000354	10 - 41	29 - 35.232	
131520	5 - 0.0000000351	2 - 52	13 - 47.267	
2209700	6 - 0.0000000350	24 - 32	14 - 45.295	
180290	7 - 0.0000000350	4 - 46	26 - 36.451	
1028700	8 - 0.0000000350	39 - 29	41 - 14.985	
73121	9 - 0.0000000346	17 - 37	132 - 16.028	Yehoshua Sagiv
1142000	10 - 0.0000000346	66 - 23	95 - 21.4610	

Table 9.3: Ranking of the top 10 Authors in Biplex Pagerank, compared to the others

It is clear that there are differences between Biplex Pagerank and the rest of the algorithms, and an even closer look needs to be taken to understand.

By seeing each of the above authors' citations to their publications, it is understood that Biplex Pagerank - as mentioned above - values a scientist's involvement in outstanding articles more than their consistency throught the range of their publications.

For example, for two sample authors with an descending order of the citation made on the articles they authored/co-authored:
#1 Author's citations: 660, 600, 800, 3, 2, 2, 2
#2 Author's citations: 5, 5, 4, 4

For these authors:
$H(1) = 3$, $H(2) = 4$, therefore H(1) < H(2)
but: Biplex(1) » Biplex(2), there's no need to actually compute this, as the nature of the algorithm is guaranteed to give these results (as long as there is a similar quality of citations for each scientist).

In conclusion, Biplex PageRank can be a useful tool for extracting the leaders of a scientific field, while H-index can return the most consistent and valuable scientists. Furthermore, C^3-Index, can return similar results to H-index, but more accurate (which is natural, as C^3-Index requires more information) and most importantly, can identify scientists with a bright future ahead of them.

10

A new Algorithm proposed

10.1 The algorithm definition

As mentioned above, we identified important tools for extracting the most outstanding scientists (Biplex Pagerank) , the most consistent ones with really good efficiency in computation time (H-index), the most consistent ones with a concern for quality too (C3-Index), the ones with the most potential (C3-Index).

At this point, it is valuable to mention **G-Index** (or g-Index). The g-index can be seen as the h-index for an averaged citations count. [6]

$$g \leqslant \frac{1}{g} \sum_{i \leqslant g} c_i \tag{10.1}$$

It is clear that, G-index can provide better results that H-index for the example we used in the previous section, and it would still be very efficient, but not ideal.
What if there was an algorithm with a concern for the scientists with the most outstanding articles, while paying attention to consistency and still being able to identify the authors with the biggest potential? Of course, almost no algorithm imaginable could achieve that with some trade-off.

C^4 - Index

In this article, we propose the C^4 - Index. C^4-Index is an improved implementation of C^3-Index that differentiates itself by adding a fourth metric.
C^4-Index uses 4 metrics (the first 3 are identical to C^3-Index's metrics):

- **ACI** - Author Citation Index
- **PCI** - Paper Citation Index
- **AAI** - Author CoAuthorship Index
- **NBP** - Author Centrality Index (Normalised Biplex Pagerank)

The basic idea is to acquire all the important information from C^3-Index, while using a factor of centrality by adding the fourth factor, the Biplex Pagerank, which was implemented in a University project by Giorgos Sideris with the guidance of Dimitrios Katsaros, as a predecessor of this article.

It is clear, that this method, is much more time-consuming and one can argue about its efficiency, but it was already mentioned that if efficiency is the most important

factor, H-index and G-index can be more suitable.

Also, every algorithm mentioned can be better in the context of what the user is looking for. C^4-Index is implemented as an all-around algorithm that takes into account as many factors as possible, for a more concerned view of a field's scientists and their value.

The steps to calculate the C^4-Index are:

- Using C^3-Index's definition, we calculate ACI, PCI , AAI for every author in the dataset
- Computation of Biplex PageRank
- Normalization of Biplex PageRank results
- $C_j^4(t) = (1-\theta) + \theta \cdot (ACI_j(t) + AAI_j(t) + PCI_j(t) + NBP_j(t))$

10.2 Tests

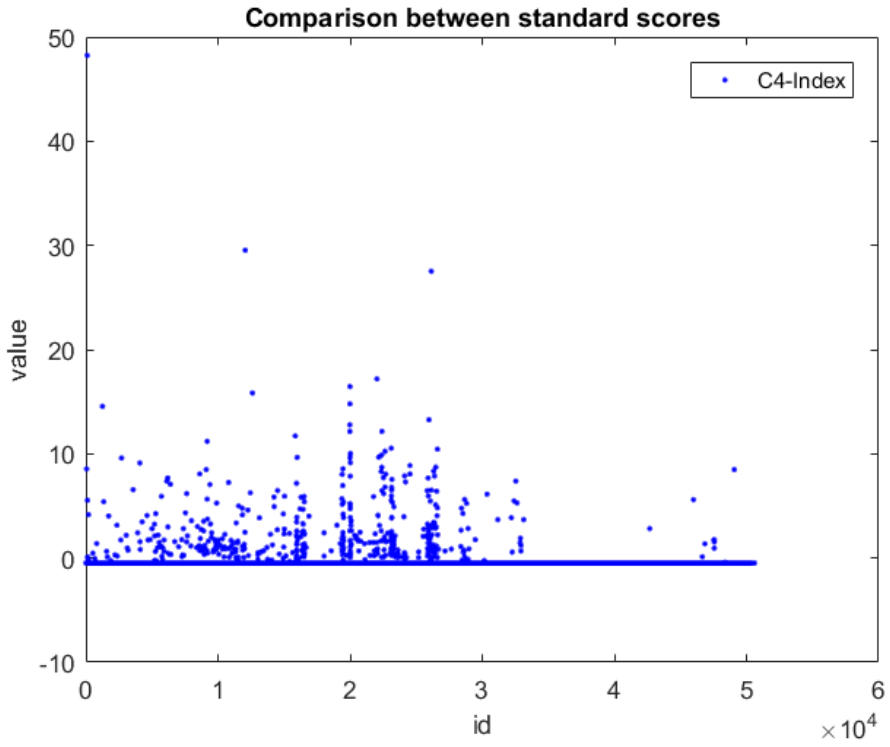


Figure 10.1: C^4-Index results

10.3 Visual Comparison with other algorithms

C^3-Index / C^4-Index

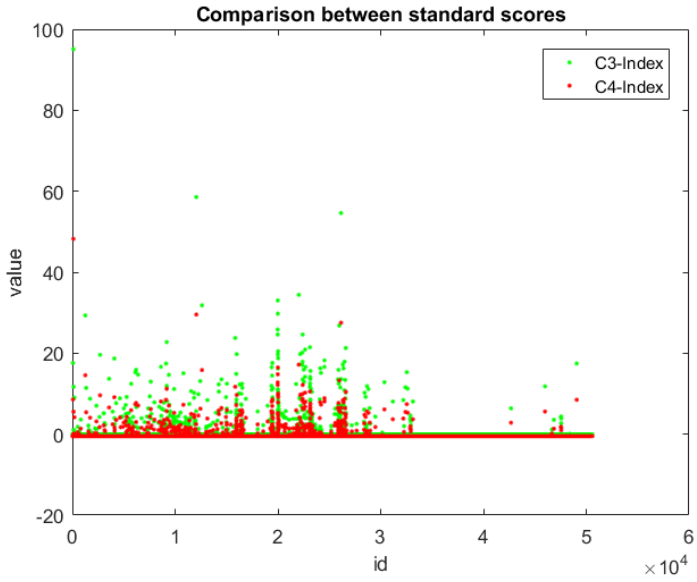


Figure 10.2: C^3-Index / C^4-Index visual comparison

Biplex Pagerank / H-Index / C^3-Index / C^4-Index

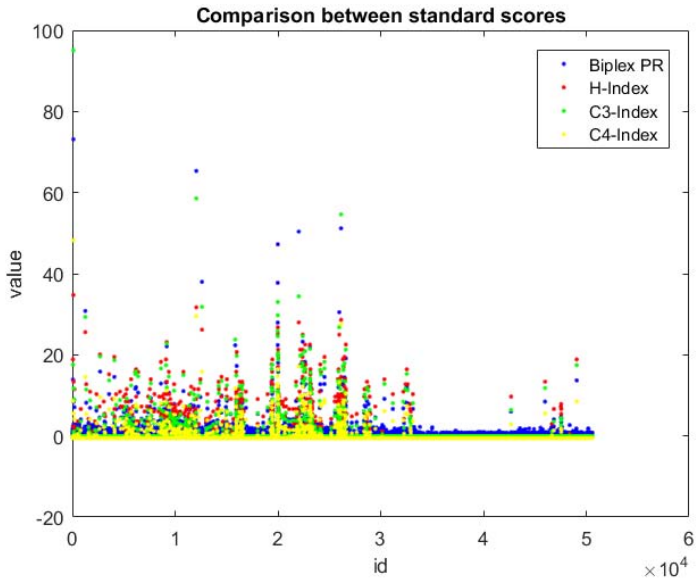


Figure 10.3: Biplex Pagerank / H-Index / C^3-Index / C^4-Index visual comparison

10.4 Correlation with other methods

Using Pearson Correlation Coefficient:

Correlation between Biplex PR and C^4-Index is: 0.133342

Correlation between H-Index and C^4-Index is: 0.772306

Correlation between C^3-Index and C^4-Index is: 0.976218

11

Observations after C^4-index implementation

Strong correlation with C^3-Index can be observed, as expected, having in mind that C^4-Index was based on it.

The run time for the MAS Dataset, is equal to the sum of run times of Biplex Pagerank and C^3-Index

This metric takes into account all the aspects of the algorithms we examined:

1. Consistency of authors
2. "Exceptional" articles
3. Better ranking for low and medium-ranked authors (better results for authors with a high potential)

Therefore, in a test scenario, where there run time is not an issue, and taking into account what already been noted, C^4-Index is a realistic and accurate option for an author-level metric for evaluating researchers according to their publications.

12
Conclusion

Defining a scientist's ranking in a specific field according to their publications and their citations, can be a quite challenging task.

There are many author-level metrics available, with the most widely accepted being the H-Index, that compute an author's ranking between his peers.

However, there is no universally accepted metric for evaluating a scientist. Some value consistency (H-Index), some other value very important articles (Biplex Pagerank), some value both, some have the ability of detecting authors with great potential (C^3-Index).

Also, the algorithms efficiency should be taken into account. For example H-Index is a very efficient algorithm with relatively accurate results.

In this thesis, after 3 algorithms were examined, implemented and tested with the same dataset (Biplex Pagerank, H-Index, C^3-Index) and their correlation was examined, the observations were:

1. Strong Correlation between H-Index and C^3-Index
2. Biplex PR paysmore attention to the importance of links and not at their consistency
3. A need for an all-around solution is visible

A new algorithm is proposed in this thesis, and it is called **C^4-Index**. It is based on the foundations of C^3-Index, with main difference its use of 4 metrics, instead of 3 as C^3-Index. The fourth metric is the normalized vector of Biplex Pagerank for the same dataset.

That way, C^4-Index pays attention to different aspects of scientific value.

The field of evaluation researchers according to their publications still has a long way to go, exploring new methods and improved the ones already known.

It is a really important issue, in the age of overflow of information, where the need to distinguish exceptional scientists is bigger than ever.

13
Future Research

As a future step at the end of the project, further research needs to be done in the rest of important author-level metrics for scientific contributions.

As well as more algorithms, more inputs should be used too, for a more diverse range of results.

Furthermore, a web and mobile application will be developed, where scientists will be able to easily acquire their rankings by different metrics and in comparison with others. This will be a useful tool for scientists who want to find out their scores easily and in many different metrics and also they will be able to showcase their rankings using the widgets provided, in forums and scientific communities in general.

Bibliography

[1] "A biplex approach to PageRank centrality: From classic to multiplex networks" by Francisco Pedroche, Miguel Romance and Regino Criado et al

[2] "An index to quantify an individual's scientific research output" - J. E. Hirsch https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1283832/

[3] "C^3 -Index: A PageRank based multi-faceted metric for authors' performance measurement" by Dinesh Pradhan et al. - https://arxiv.org/pdf/1610.07061.pdf

[4] Antonis Sidiropoulos - https://www.researchgate.net/profile/Antonis_Sidiropoulos

[5] Pearson's correlation - http://www.statstutor.ac.uk/resources/uploaded/pearsons.pdf

[6] "Mathematical Theory of the H-Index and g-Index in Case of Fractional Counting of Authorship" by Leo Egghe

[7] MATLAB sparse function - http://uk.mathworks.com/help/matlab/ref/sparse.html

[8] MATLAB corr2 function - https://uk.mathworks.com/help/images/ref/corr2.html

14

Appendices

14.1 Appendix 1: Previous project & Accuracy of Biplex PageRank compared to single-layer Google PageRank

In the course of the project, two algorithms were implemented, one was a bi-layer approach of classic PageRank, and the other as a way of ranking nodes in multiplexed networks.

The two algorithm process of calculation are similar, with small differences.

14.1.1 Biplex Approach to Classic PageRank (v1)

Definition: Given a network G that consists of $n \in \mathbb{N}$nodes and adjacency matrix Pa, then: $p = Pu + Pd \in \mathbb{N}$

Calculation of Pu , Pd: Using iterative procedure
$Pu^T = Pu^T \cdot a \cdot Pa + Pd^T \cdot a$
$Pd^T = (1 - a) \cdot Pu^T + (1 - a) \cdot Pd^T \cdot e \cdot v^T$

In the first stage, this algorithm which is an alternative solution for measuring classic PageRank, in two layers. The first network is defined as the network examined by PageRank with its links, and as second, the "teleportation" network where there exist links between all nodes and their weights are given by personalization vector v, and the adjacency matrix is $e \cdot v^T$.

Comparison of results with PageRank

We run biplex.m with the input of lesmix2.txt (lesmis.txt without dangling nodes), a network of 77 nodes and 284 links. After 67 iterations, the ten nodes with the higher ranking, are as follows:

Value:	Index:
0.3319	77
0.3294	66
0.1653	67
0.0096	56
0.0072	76
0.0067	40
0.0064	39
0.0052	55
0.0050	65
0.0047	24

Afterwards, on CentiBin, we run classic PageRank for the same network and these are the results:

Centrality:	**PageRank**
Vertex	Centrality
77	0,3973688
66	0,3972592
67	0,1986529
56	0,0004355
76	0,0003072
40	0,0002945
39	0,0002776
55	0,0002305
65	0,0002191
54	0,0001971
24	0,0001887
53	0,0001722
64	0,0001719

Comparing the results, it is clear the the node ranking is identical, proving the accuracy of Biplex PageRank

14.1.2 Multilayer Approach for PageRank of Multiplex Networks (v2)

Definition: Given a network G that consists of $n \in \mathbb{N}$ nodes, and adjacency matrix Pa, then:
$p = \frac{1}{2} \cdot (Pu + Pu_2 + Pd + Pd_2) \in \mathbb{R}$

Calculation of u , Pu_2, Pd, Pd_2 : Using iterative procedure
$2Pu^T = Pu^T \cdot a \cdot Pa + Pu_2^T + 2 \cdot a \cdot Pd^T$
$2Pu_2^T = Pu^T + Pu_2^T \cdot a \cdot Pa_2 + 2 \cdot a \cdot Pd_2^T$
$2 \cdot Pd^T = (1-a) \cdot (Pu_2^T + Pd^T \cdot e \cdot v^T + Pd_2 \cdot e \cdot v^T)$
$2 \cdot Pd_2^T = (1-a) \cdot (Pu_2^T + Pd^T \cdot e \cdot v_2^T + Pd_2 \cdot e \cdot v_2^T)$

The algorithm described above was implemented for biplex networks, and can be implemented for more complex networks, with small adjustments.

We run biplex_v2.m with lesmis.txt as first and second network, and we expect the result to be similar with the previous algorithm results, for the same network.

Biplex v2	Biplex v1	Index
0.0372	0.0370	77.0000
0.0351	0.0348	76.0000
0.0346	0.0343	39.0000
0.0336	0.0334	32.0000
0.0327	0.0325	75.0000
0.0326	0.0324	53.0000
0.0325	0.0323	57.0000
0.0325	0.0323	45.0000
0.0325	0.0323	46.0000
0.0325	0.0323	43.0000

Indeed, we see that the two results are identical, somethings that proves the claim we made before.

In addition, we can make a quick evaluation of the correctness of the algorithm, using as first network the one we used before, and as second, the one below:
77 1
76 1
39 1

Its links are directed from the most important nodes to the node with ID 1. We expect node 1 to be substantially more significant in this run. And indeed:

Value:	Index:
0.0301	1.0000
0.0258	77.0000
0.0215	32.0000

The node with the ID of 1, is the highest ranking node in the biplex network, somethings that partly validates, the correctness of the algorithm.

The conclusion to be made is that the second algorithm (Biplex PageRank) can easily be extend for more complex network, having many uses for ranking nodes. For example, in order to find the most important stations in subway network, with every line being a layer or for the most valuable scientists according to their publications, with layers the Paper to Paper Citation layer, and the Author Citation Layer.

14.2 Appendix 2: Use of helper functions

A set of helper, simple, reusable functions were developed in MATLAB, to make common tasks easier and faster.

Filter500.m (coauthorship file, citations file, top500 authors file) : Given a complete dataset and a file containing the filtering factors, it removes every link and information that is not associated with the top 500 authors and everyone that interacted with them, and save the filtered dataset in 2 files (coauthorships, citations)

function filter500(file1,file2,file3,delimeter)

```
tic; % start timer
top500=dlmread(file3 ,delimeter);
A= dlmread(file1 ,delimeter);
A2= dlmread(file2 ,delimeter);
unA2 = unique(A2);

n=size(A,1);
w=size(A,2);

toDelete=[];

for i=1:n
    i/n*100
    flag=1;
    for j=2:w
```

```
        if (A(i,j)==0)
            continue;
        end
        if (flag==0)
            break;
        end
        if ((ismember(A(i,j), top500(:)))==0)
            flag=0;
            toDelete(end+1)=i;
        end
    end
end

A(toDelete,:)=[];
unA = unique(A);

n=size(A2,1);
w=size(A2,2);

toDelete=[];
for i=1:n
    i/n*100
    flag=1;
    for j=2:w
        if (A2(i,j)==0)
            continue;
        end
        if (flag==0)
            break;
        end
        if (~ismember(A2(i,j), unA(:)))
            flag=0;
            toDelete(end+1)=i;
        end
    end
end

A2(toDelete,:)=[];

dlmwrite('paper_authors_filtered',A,' ')
dlmwrite('citations_filtered',A2,' ')

toc;
end
```

CountUnique.m (coauthorship file, citations file, delimiter) : Counts the unique elements in 2 files, creates mapping arrays, and savesfiltered and mapped Datasets and the Mapping Dictionary

```
function count_unique(file1,file2,delimeter)

tic; % start timer
A= dlmread(file1,delimeter);

A = unique(A);
A2= dlmread(file2,delimeter);

A2 = unique(A2);
A= unique(vertcat(A,A2));
size(A)

n=size(A,1) % number of links


for i=1:n
    B(i,1) = i;
    B(i,2) = A(i);
end

A= dlmread(file1,delimeter);
A2= dlmread(file2,delimeter);

n=size(A,1); % number of links
n2=size(A2,1);
w2=size(A2,2); % Width of A
w=size(A,2);
n3=size(B,1);



for i=1:n
    i/n*100
    for j=1:w
        if (A(i,j)~=0)
            for k=1:n3
                if (A(i,j)==B(k,2))
                    A(i,j)=B(k,1);
                end
            end
        end
```

```
    end
end

for i=1:n2
    i/n2*100
    for j=1:w2
        if (A2(i,j)~=0)
            for k=1:n3
                if (A2(i,j)==B(k,2))
                    A2(i,j)=B(k,1);
                end
            end
        end
    end
end

dlmwrite('paper_authors_filtered_mapped',A,' ')
dlmwrite('citations_filtered_mapped',A2,' ')
dlmwrite('mapping',B,' ')

toc;

end
```

GetSum.m (file,delimiter): Given a vector , it returns the sum of its elements. This was used to evaluate the accuracy of the single-layer version of Biplex Pagerank.
translate_mapped_results.m (results, mapping, names, delimiter): Given the results file, the mapping dictionary file, the names dictionary file and the delimeter, it saves sorted results, translated to their original ID's with the information of the author's name when it's available
function translate_mapped_results(file,mapping,names,delimeter)

```
tic;
A= dlmread(file,delimeter);
A2= dlmread(mapping,' ');
T = readtable(names,'Delimiter','        ',
'Format','%d    %s       %s       %s');
T = table2cell(T);
A = num2cell(A);
A2 = num2cell(A2);

n=size(A,1); % length of A
n2=size(A2,1); % length of A2
n3=size(names,1); % length of A2
w=size(A,2); % width of A
w2=2; % width of A2;
```

```
fid = fopen('outputs/biplex_data_final','w')

for i=1:n
    i/n*100
    q = A{i,1};
    for j=1:n2
        if A2{j,1}==q
            A{i,1} = A2{j,2};
        end
    end
    flag = 0;
    if (A{1,2}~=0)
        for k=1:91751
            if T{k,1}==A{i,1}
                firstname = T{k,3};
                lastname = T{k,2};
                A{i,3} = strcat(firstname,{' '},lastname);
                flag = 1;

            end
        end
    end
    if flag==0
        A{i,3} = '';
    end
    fprintf(fid,'%d      %2.10f  %s\n',A{i,1},A{i,2},char(A{i,3}));
end

fclose(fid)
toc; % end timer
beep;

end
```

plotmatrices.m (Results1 file, Results2 file, Results3 file, delimiter): Given 3 files containing the results of three tests, plotmatrices converts the values of each one to their normalised values and then plots them together in the same figure, to visually compare them.

```
function plotmatrices(file1,file2,file3,delimeter)
    A= dlmread(file1,delimeter);
    A2= dlmread(file2,delimeter);
    A3= dlmread(file3,delimeter);

    maximum = max(max(max(A),max(max(A2),max(A3))));
    x = A(:,1);
```

```
    y = A(:,2);
    x2 = A2(:,1);
    y2 = A2(:,2);
    y3 = A3(:,2);
    mean1 = mean(y);
    mean2 = mean(y2);
    mean3 = mean(y3);
    dev = std(y);
    dev2 = std(y2);
    dev3 = std(y3);
    f = (y-mean1)./dev;
    y = arrayfun(@(x) (x-mean1)/dev,y);
    y2 = arrayfun(@(x) (x-mean2)/dev2,y2);
    y3 = arrayfun(@(x) (x-mean3)/dev3,y3);
    plot(x,y,'b.',x,y2,'r.',x,y3,'g.')
    xlabel('id'); ylabel('value');
    title('Comparison between standard scores');
    legend('Biplex PR','H-Index','C3-Index')
end
```

getCorrCoeff2D.m (results1, results2, results3, delimiter) Given the 3 test results, the correlation coefficient between each other is computed and printed.

```
function getCorrCoeff2D(file1,file2,file3,delimeter)
    A= dlmread(file1,delimeter); %biplex
    A2= dlmread(file2,delimeter); %h-index
    A3= dlmread(file3,delimeter); %c3-index

    n=size(A,1); % length of A
    n2=size(A2,1); % length of A2
    n3=size(A3,1); % length of A2
    for i=1:n
        if (A(i,1)~=0)
            v(A(i,1))=A(i,2);
        end
    end

    for i=1:n2
        if (A2(i,1)~=0)
            v2(A2(i,1))=A2(i,2);
        end
    end

    for i=1:n3
        if (A3(i,1)~=0)
            v3(A3(i,1))=A3(i,2);
```

```
        end
    end


    fprintf('Correlation between Biplex PR and H-index is:
    %f\n\n',corr2(v,v2))
    fprintf('Correlation between H-index and C3-index is:
    %f\n\n',corr2(v2,v3))
    fprintf('Correlation between Biplex PR and C3-index is:
    %f\n\n',corr2(v,v3))
end
```

keepOnlyAuthors.m (results file, delimiter) : Given a file with test results, it saves a file with only the rows associated with an author, not papers (this was used in Biplex Pagerank results for better understanding of the ranking)

```
function keepOnlyAuthors(file,citations)

tic; % start timer
A= dlmread(file,' ');
citations = dlmread(citations,' ');

n = size(A,1);
fid = fopen('outputs/biplex_only_authors_filtered_mapped','w');
for i=1:n
    if ismember(A(i,1),citations)==0
        fprintf(fid,'%d %2.10f\n',A(i,1),A(i,2));
    end
end
fclose(fid);
toc;
beep;

end
```