Microsoft OAG Network Analysis

Marco Cardia
m.cardia@studenti.unipi.it
Student ID: 530567

Francesco Sabiu
f.sabiu@studenti.unipi.it
Student ID: 533231

ABSTRACT

The co-authorship of articles in learned journals provides several patterns of collaboration within the academic community. In this study, we collect the data related to the 2016 academic co-authorships and we build a network from them. We use these data along with several measures and algorithms in order to answer a broad variety of questions about the patterns they reveal. After characterizing the network, we analyse it by means of different community discovery algorithms. Thus, we apply some Opinion Dynamics models in order to simulate the spreading of plausible innovations. Moreover, we find some correlations between network properties and authors' features. The achieved results described in our paper are available at the team GitHub repository [1].

KEYWORDS

Social Network Analysis, Scale-Free Networks, Co-authorship

ACM Reference Format:
Marco Cardia and Francesco Sabiu. 2019. Microsoft OAG Network Analysis. In *Social Network Analysis '20*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Coauthorship of papers can be seen as a fact assessing the collaboration between two or more authors. From a more collective point of view, such data form a global network of collaboration, known in literature as co-authorship network. Several studies of scientific networks have been published in bibliometrics and social network analysis: they investigate properties of different types of networks in which the nodes are usually papers [23], authors [21] or organizations [17]. In our study, we analyse a network whose nodes are authors linked by their collaborations extracted from papers information, making use of Microsoft Open Academic Graph data.

Our contribution is twofold. Firstly, we construct the graph described by the adopted dataset and we analyse its topological properties in order to provide a structural understanding of the academic community. Secondly, we dig into a deeper analysis of the network by applying several state-of-the-art algorithms related to community discovery, link prediction and opinion dynamics. Finally, we investigate for interesting correlations among individual and collective properties of the network.

The rest of the paper is organized as follows. Section 2 provides a detailed implementation of the techniques and algorithms used for the collection of data from the Open Academic Graph provided by Microsoft [35][32], together with the description of the available data. In Section 3 we describe both the extracted network and the larger of its connected components by means of the typical measures and statistics that characterize networks, providing also a comparison with respect to the most known network models. Next, in Section 4 we apply, evaluate and compare four state-of-the-art algorithms with the purpose of individuating the communities composing the co-authorship network. In Section 5 we describe the algorithms adopted for the link prediction task, their goal and the achieved results, while in Section 6 we do the same for five state-of-the-art algorithms adopted for the analysis of the opinion spreading within the network. Furthermore, in Section 7 we exploit the weights of the collected network by providing an implementation of an overlapping community discovery algorithm for weighted networks. In the same section, its description together with the evaluation of this application on the network are provided. Finally, we answer several questions related to the authors. In particular, we try to discover pattern involving both the collective properties (communities, overall network characteristics) and the individual ones, such as the performance index of the authors or the number of citations and publications. The results of the investigation are presented in Section 8 and 9. In the latter, we also highlight the main criticisms addressed in our work, as well as the reason of the described choices.

[1] **Project Repositories**
Data Collection: https://github.com/sna-unipi/data-collection-2020_sabiu
Analytical Tasks: https://github.com/sna-unipi/network-analysis-analytical-tasks-2020_sabiu
Report: https://github.com/sna-unipi/project-report-2020_sabiu

SNA '20, 2019/20, University of Pisa, Italy

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$0.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 DATA COLLECTION

In this section, the source data, as well as the steps involved in the extraction of relevant information from them, are described. After providing a brief characterization of the available data, we dig into the details of the methodology adopted for their manipulation, illustrating the most meaningful steps along with the raised criticisms.

Selected Data Sources

Open Academic Graph (OAG)[35][32] is a knowledge graph provided by Microsoft as a service to the research community. It unifies two billion-scale academic graphs: Microsoft Academic Graph (MAG) and AMiner, which are constantly evolving according to the exponential growth of research bibliography. In order to perform our analysis we selected the January 2019 snapshot of the OAG v2.
The data that we considered were included in the following files:

- *AMiner papers*: a table containing 172,209,563 papers with their information.
- *AMiner authors*: a 113,171,945 entries table containing information regarding the authors.

For our purpose, we decided to consider only a subset of the columns of such tables. For the first, we took into account only the fields *year* and *authors*, necessary to extract only the IDs of authors whose collaboration took place in 2016. Concerning the second, we considered *name*, *h-index*, number of both *publications* and *citations*, the related list of *organizations* and the *research fields*.
A challenging aspect of the collection phase has been represented by the size of the analysed files: those containing papers' information occupy 177 GB, while the authors related ones have size slightly greater than 35 GB. For this reason, the algorithms performing the data collection tasks have followed a streaming approach.

Extraction Methodology and Assumptions

Due to the size of the files mentioned in section 2, the extraction of authors' IDs and authors' information has been performed in an online fashion. In particular, files have been read in chunks of 10000 entries in order to overcome to the gap between RAM capacity and data size. Such technique has been feasible because of the following property on the papers data P: no dependency with records $\neq i$ exists for each $i \in P$.
Algorithms 1 and 2 illustrate the developed algorithms. While the former obtains the list of relevant links from the AMiner papers table, the latter retrieves authors' information from AMiner authors table.

An observation is in order at this point. The goal of Algorithm 2 is to retrieve information related to m over n total authors, with n » m. In a usual batch algorithm we would implement this task by performing m lookups in the n entries (both sets are unsorted). In our case, because of the online paradigm, we have to revert such approach: for each author whose information is stored, we look whether it is contained in the stored links. If yes, its information is fetched.

Algorithm 1 Link extraction

Require: $paper_i.txt$ to be in $papers/$ folder
Initialize $chunk\_size = 10000$, $links = \{\}$
$P = papers/*.txt$
repeat
  $paper = P.pop()$
  repeat
    $C = readChunk(paper, chunk\_size)$
    for each: $p \in C$ **do**
      if $p.year == 2016$ **then**
        for each: pair $auths \in p.authors$ **do**
          $links = links \cup pair$
        end for
      end if
    end for
  until $C \neq \{\}$
until $P \neq \{\}$

Algorithm 2 Authors information extraction

Require: $links$ to be a set of pairs of valid authors IDs
$authors_i.txt$ to be in $authors/$ folder
Initialize $chunk\_size = 10000$, $authors = \{\}$
$A = authors/*.txt$
repeat
  $author = A.pop()$
  repeat
    $C = readChunk(author, chunk\_size)$
    for each: $a \in C$ **do**
      if $\exists pair \in links : a \in pair$ **then**
        $authors = authors \cup a$
      end if
    end for
  until $C \neq \{\}$
until $A \neq \{\}$

After the execution of Algorithms 1 and 2 we have obtained the worldwide 2016 co-authorship network, made up of 373263 nodes and 4511734 links.
In the next sections, such network will be object of some analyses and comparisons. Notice that one or more sub-samples of it may be used to estimate some measures that otherwise would be computationally too expansive with respect to the

adopted hardware [2]. In particular, the majority of the tasks described in the following made use of the biggest connected component of the original graph. When used, this assumption will be specified in the related section.

3 NETWORK CHARACTERIZATION

Let B be a set of bibliographic entities such as scientific papers, journals, publications or books published in 2016. Let A be the set of authors appearing in B, and $coauthors(b, a_1, a_2)$ a predicate that is true if and only if both a_1 and a_2 appear in B as authors. The co-authorship network corresponding to B is the undirected weighted graph $G = (V, E)$ in which:

- The set of nodes V corresponds to the set of authors A.
- Two nodes v_1 and v_2 are connected by an undirected edge (link) $e \in E$ iff $\exists b \in B : coauthors(b, v_1, v_2) = true$, i.e. if and only if b is jointly co-authored by v_1 and v_2.
- Each edge $e = \langle v_1, v_2, w \rangle$ is a tuple in which the weight $w = \#\{b \in B : coauthors(b, v_1, v_2) = true\}$

Furthermore, network nodes are characterized by the following attributes:

- *Name*: a non unique string containing the name of the author;
- *h-index*: a number representing the *de facto standard* author-level metric that measures both author's productivity and citation impact;
- *Number of publications* of the author related to its publications up to 2016;
- *Number of citations* of the author related to all the publications up to 2016;
- *Organizations*: a list of the author's affiliated organizations up to according to OAG data source.

Since some of the following measures involve expansive computation, such as the one of the diameter, we perform them on the biggest connected component individuate in the connectedness analysis.

On the degree distribution

Once we have obtained the network, some consideration can be made about its structure. Firstly, as expected by the knowledge of the context, we have that $L_{max} = \frac{N(N-1)}{2} >> L$ and the average degree is 24, meaning that the network is sparse. Secondly, as shown in Figure 1, the degree distribution follows the typical pattern of the Scale-Free networks, from which we can get the power law.

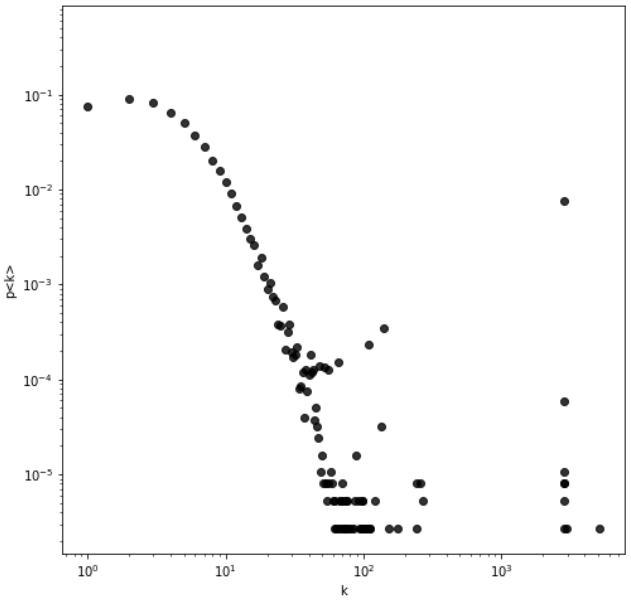


Figure 1: Co-authorship network degree distribution

Connectedness

Given the context that we are analysing, one could expect the network to made up by several connected components: since researchers tend to interact with other researcher of the same field such components could depend on the research fields, or on geographical attributes for example. In order to fact-check this intuition we have performed a connected components analysis, whose quantitative results are illustrated in Figure 2. In the log-log plot, the distribution of the size of the connected components is provided: expectedly, it follows the same pattern of the degree distribution, confirming the scale-free nature of the network.

Figure 2 highlights the presence of a particular connected component: its size is between 2 and 3 orders of magnitude greater than both mean and median of the other cluster sizes. In the following subsection, we perform some analyses on such cluster, since it represent the only connected component made up of more than some hundreds nodes. Other components of approximately mean size will be analyzed, as well.

Path analysis

Since many connected components compose the graph, its diameter and average path length are infinite. Hence, we

[2]Data Collection has been performed by means of Google Colaboratory Reserch, providing n1-highmem-2 instance with 2vCPU @ 2.2GHz, 25GB of RAM and 110 GB of secondary memory.
Several algorithms for Link Prediction, Opinion Dynamics and Community Discovery have been executed on the AI Platform provided by Google Cloud Platform (n1-highmem-8 with 8 vCPUs, 52 GB RAM) and on Amazon AWS machines (4 CPUs, 32 GB RAM).

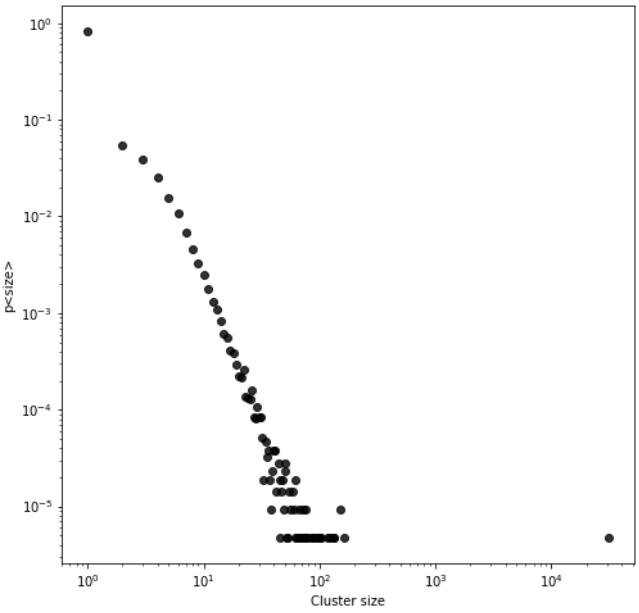


Figure 2: Co-authorship network connected components size distribution

Table 1: Connected components degrees with respect to the network

Component	Avg degree	Max degree
Full network	24	5134
Biggest component	50	5136*
All others (max)	29	66
All others (avg)	0.48	0.57

* We consider this result incongruent with respect to the maximum degree detected in the whole network, that is lower.

have performed the path analysis on the sub-graph identified in the previous section, that was expected to represent an upper bound (for such path-related measures) with respect to the other clusters. We discovered that its diameter and path length are actually greater than those of the others clusters, confirming that the network structure follows the same pattern within each connected component.

Clustering coefficient and density analysis

By definition, co-authorship networks are made up of groups of authors related each other by one or more article. From a topological point-of-view, their nature leads the network to be composed by a chain of cliques, ideally representing research groups. Such property is reflected in the value of the global clustering coefficient of the network, obtained as follows:

$$C_{network} = \frac{\sum_{i=1}^{n} \frac{2 \times L_i}{k_i(k_i-1)}}{n}$$

where:

- L_i is the maximum number of existing links among the neighbors of the node i.
- k_i is the number of neighbors of the node i, thus the term $k_i(k_i - 1)$ represents the maximum number of possible links (fully connected network) among them.

We calculated the global clustering coefficient for both the biggest connected component and the whole graph, obtaining almost equivalent values: 0.998. This information confirms the expected network structure, in which we have cliques-like networks composing each cluster.

Concerning the graph density, we also evaluated it separately for the whole graph and for its biggest connected component. They differed for several orders of magnitude: the whole network density is indeed 6×10^{-5}, while the one of the biggest connected component is 8×10^{-3}. We can conclude that the considered subgraph is quite denser than the rest of the network.

Table 2: Connected components paths comparison among full network, biggest component and all other components

Component	Diameter	Avg path length
Full network	∞	∞
Biggest component	50	11.32
All others (max)	–	6.99
All others (avg)	–	1.08

* The computations required too much time.

Centrality analysis

We performed the network centrality analysis by applying the classic network centrality measures used in literature, namely betweenness centrality, closeness centrality, degree centrality and eigenvector centrality. All of the following centrality measures have been computed by means of the Python *igraph* library [10].

Betweenness centrality. The betweenness centrality of a node is a measure of its centrality based on shortest paths. Indeed, it is informally defined as the number of shortest paths between every possible pair of nodes passing for it. More formally, it is defined as follows.

Definition 3.1. (Betweenness centrality)
Given a network $G = (V, E)$ with $|V| = n$ and $|E| = m$, its

average betweenness centrality is defined as

$$C_b(i) = \sum_{j \neq k} \frac{\sigma_{jk}(i)}{\sigma_{jk}}$$

where

- $\sigma_{jk}(i)$ is the number of geodesic paths from j to k via i;
- σ_{jk} is the number of geodesic paths from j to k.

Degree centrality. This measure is defined on the base of the number of links incident upon a node, also known as grade of the node itself. It is the most basic measure since it relies on a local property. Formally, we define it in the following way.

Definition 3.2. (Degree centrality)
Given a network $G = (V, E)$ with $|V| = n$ and $|E| = m$, the degree centrality of a node i is defined as

$$k_i(i) = \sum_{j=1}^{n} A_{ij}$$

where A is the adjacency matrix of G.

Closeness centrality. Intuitively, closeness centrality measures how close a vertex is to all other vertices in the graph. Its value is obtained calculating the reciprocal of the sum of the length of the shortest paths between the node and all other nodes in the graph [4]. In formulas, we define it as follows.

Definition 3.3. (Closeness centrality)
Given a network $G = (V, E)$ with $|V| = n$ and $|E| = m$, its closeness centrality of a node i is defined as

$$C_{cl}(i) = \frac{n-1}{\sum_{d_{ij}<\infty} d_{ij}} \tag{1}$$

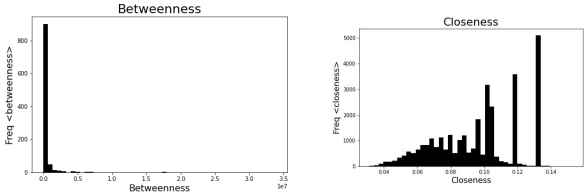
where d_{ij} is the distance between the nodes i and j of G.

Eigenvector centrality. Eigenvector centrality measures the importance of a node in the network. After initializing initial scores, the algorithm give importance to a node if it is linked to other nodes, depending on the importance of its neighbors. More formally, we can define it as follows.

Definition 3.4. (Eigenvector centrality)
Given a node u belonging to a network $G = (V, E)$ with $|V| = n$ and $|E| = m$, the centrality score of a node i is defined as follows:

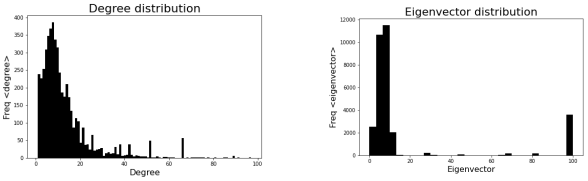
$$x_i^{(t+1)} = \sum_{j=1}^{n} A_{ij} x_j(t)$$

where x_i is the centrality of the node i.



(a) Betweenness centrality **(b) Closeness centrality**

Figure 3: Centrality measures distribution calculated over the biggest component.



(a) Degree centrality **(b) Eigenvector centrality**

Figure 4: Centrality measures distribution calculated over the biggest component.

From the application of the above defined centrality measures, we find that almost all of their distributions follow an exponential pattern. This is a direct consequence of the network structure. Figure **??** graphically illustrates such results. Notice that the eigenvector centrality values have been logarithmically scaled.

Comparison with network models

As a matter of fact, we provide a brief comparison among our network and the most known models existing in literature, namely Erdős–Rényi (ER)[12], Barabási–Albert (BA)[2] and Watts–Strogatz (WS)[36]. Intuitively, such models generalize networks whose nodes respectively form (i) a random network (ii) a configuration model (iii) a (random) scale-free network and (iv) a small-world network. Nodes collective behaviour (i.e. their emergent properties in terms of link to other nodes) can be expressed more accurately by means of a set of structural network measures such as degree distribution, path length and clustering coefficient. With this purpose, we firstly provide an informal description summarizing such properties for the three models (Table 3). Next, we formalize them in Table 4 by means of formulas which are function of the number of nodes and links.

For the comparison of our network with the cited models, we proceeded as follows. Firstly, in order to get the distribution we exploit NetworkX Python library [15] to create the graphs according to the network models of Table 4. Then, we calculated the statistics reported in the previous subsection for each of these models. In particular, we focused on degree, average path length and clustering coefficient. The

Table 3: Order of magnitude of topological measures for ER, BA and WS models

Network	Degree distribution	Path length	Clustering coefficient
Erdős–Rényi (Random Network)	Poissonian	Short	Small
Configuration model	Custom, can be broad	Short	Small
Watts–Strogatz (WS) (small world)	Poissonian	Short	Large
Barabási–Albert (Scale-free)	Power-law	Short	Rather small

Table 4: Models topological measures with parameters

Network	Degree distribution	Path length	Clustering coefficient
Erdős–Rényi (Random Networks)	$p_k = \frac{\langle k \rangle^k}{k!} e^{-\langle k \rangle}$	$O(log(N))$	$C_i = \frac{\langle k \rangle}{n-1} = p$
Configuration model	$p_{nei,k} = \frac{k p_k}{\langle k \rangle}$	$O(log(N))$	$C = \frac{1}{n} \frac{[\langle k \rangle^2 - \langle k \rangle]^2}{\langle k \rangle^3}$
Watts–Strogatz (WS) (small world)	$P(k) = e^{-Kp} \frac{(Kp)^{k-K}}{(k-K)!}$	$\frac{ln(nKp)}{K^2 p}$	$\frac{3(K-2)}{4(K-1)+8Kp+4Kp^2}$
Barabási–Albert (Scale-free)	$P(k) \sim Ck^{-\gamma}$	$\frac{lnN}{lnlnN}$	$\frac{m}{4} \frac{(lnN)^3}{N}$

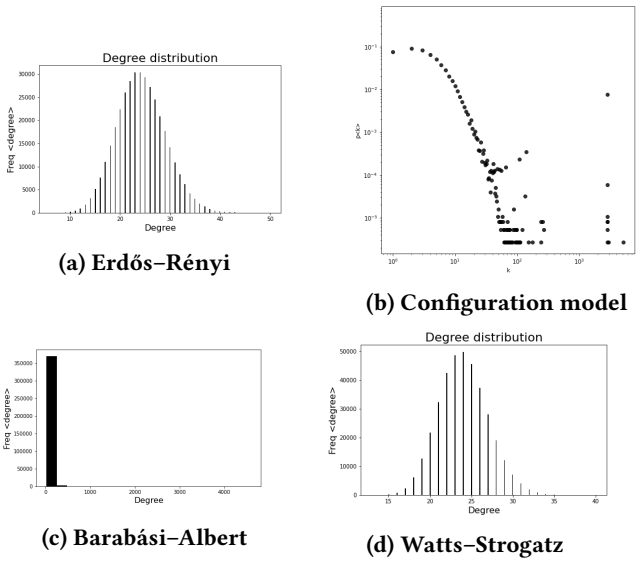
Table 5: Comparison among network models and co-authorship network

Network	Degree distribution	Path length	Clustering coefficient
Erdős–Rényi model	Fig. 5a	10.35	6.48×10^{-5}
Configuration model	Fig. 5b	10.35	1×10^{-4}
Watts–Strogatz (WS)	Fig. 5d	1	0.08
Barabási–Albert	Fig. 5c	4.43	1×10^{-3}
Co-authorship network (subgraph)	Fig. 1.	24	0.998

results of this analysis are reported in Table 5, in which we compare our network statistics with the one of the studied models. Where values are infinity, we substitute them with the same measures applied to graphs having the same size of the biggest subgraph of ours.

4 COMMUNITY DISCOVERY

In this section, several Community Discovery (CD) approach are described, and applied. A description of their result is also provided. Each applied algorithm corresponds to a different technique having as goal the detection of similar authors of the network. Depending on the algorithm, similarity can refer either to the network structure or to the node features. We applied some of the algorithms provided by the Python library CDlib [28], that allows all extraction, comparison, and evaluation of communities from complex networks. The algorithms we run are Label Propagation [26], Louvain [5], K-clique ($k = 3$) [22] and Demon ($\epsilon = 0.25$) [9]. Concerning the last, we had to choose between it and Angel [27] because they were the two most computational expansive. We evaluated the discovered communities by means of four evaluation functions among those provided by CDlib, namely Internal Edge Density [24], Average Internal Degree [25], Modularity



(a) Erdős–Rényi

(b) Configuration model

(c) Barabási–Albert

(d) Watts–Strogatz

Figure 5: Correlation between community properties and authors' features.

Density [18] and Conductance [16]. Table 6 shows the resulting scores. Notice that the k-Clique algorithm returns the same results in terms of evaluation for each of the following values of the parameter k (representing the size of smallest clique): 3, 5 and 7. This implies that each of the detected communities has size greater or equal to 7.

Table 6: Community Discovery algorithms evaluation

Evaluation	Label Propagation	Louvain	k-Clique	Demon
Internal Edge Density	0.19	0.08	**0.23**	0.17
Average Internal Degree	5.38	**78.16**	5.14	10.02
Modularity Density	**1106.91**	10.01	-59631.28	-17937.91
Conductance	0.25	**0.06**	0.46	0.38

Some considerations are in order at this point. At first glance, we observe significant differences among the modularity values applied to the communities discovered by the three algorithms. In particular, the k-Clique algorithm produces (together with Demon) low-modularity communities. Instead, modularity is maximized by Label Propagation, resulting the best approach for a such dense network.
Also the conductance measure is importantly affected by the applied algorithm, that impact on this measures. Results range in one order of magnitude, from 10^{-2} to 10^{-1}, having best values for the communities discovered by means of *Label Propagation* and *Louvain*. Similarly, the average node degree of the communities is optimized (i.e. maximized) by *Louvain*, while k-Cliques optimizes the Internal Edge Density: this result is reasonable and expected, since k-Cliques creates communities starting from adjacent k-cliques.

5 LINK PREDICTION

Link prediction algorithms are applied to networks for several purposes: for example, they can provide missing information from incomplete networks, identify their evolving mechanisms or find future or spurious interactions. Depending on their approach, they can be supervised or unsupervised. In our analysis, we applied the following four unsupervised link prediction algorithms to the biggest connected component extracted from the starting network.

- *Common neighbors* [19], in which the correlation between the size of the common neighborhood between two nodes and the probability of a their future connection is exploited.
- *Jaccard* [31], that is based on the Jaccard similarity, i.e. a set similarity measure widely adopted in Information Retrieval since 1980s. In this link prediction algorithm, it is applied on the features of two nodes, relying on the assumption that the more portion of features two nodes share, the more they are likely to create a relationship.
- *Academic-Adar* [1]: conceptually it is equivalent to the Jaccard-based one, since it makes use of nodes features to compare them. Its novelty relies on the computation of such similarity: rarer features (i.e. low frequency ones) are considered more important than the more frequent ones, contributing with a stronger weight in the calculation.
- *Preferential attachment*: this category of algorithms is based on the fact that the probability of a new link for a node is proportional to the size of its neighborhood. This observation comes from [3], in which authors shown the positive correlation between the link probability of two nodes and the product of their neighborhoods sizes.

We chose such algorithms because they reflect quite realistically the behavioural model that a co-authorship network could adopt. Indeed, if two researcher share a significant number of colleagues or research topics (common neighbors and features similarity) this fact increases the probability that they write a paper together. Contrarily, if they don't share neither a research group nor a research interest, it is reasonable to considerate such probability near to zero. The same reasoning applies for the preferential attachment -based predictions.
We availed ourselves of both linkpred and and NetowrkX [15] Python libraries for all the prediction algorithm described in this section. For the sake of presentation, we applied them to the 9^{th} and 10^{th} biggest communities discovered by means of Louvain algorithm, since the application to the whole graph would be computational unsustainable for our resources. Nonetheless, only the results of the application of the Jaccard model is available. The rest of the algorithm are not available due to their execution time. Indeed, co-authorship networks are very dense networks, in which applying such kind of algorithms results time expensive. Concerning the Jaccard model, it returned a precision of 0.97 and recall value of 0.20 for the prediction. The graphical output of its evaluation is reported in Figure 6.

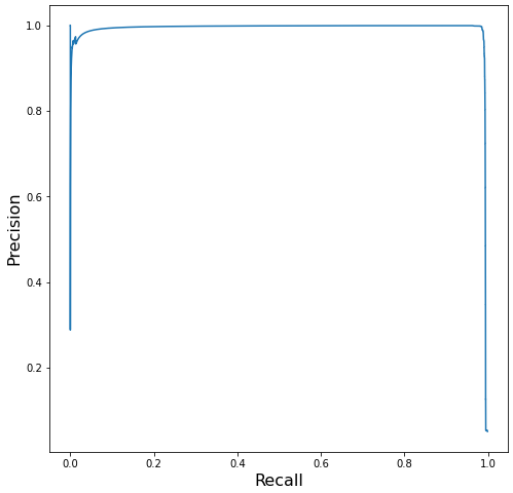


Figure 6: Evaluation of link prediction based on Jaccard model.

For the explained reasons, in order to achieve the purposes of the present work, we applied to our network some Network Diffusion algorithms that are described in the next section. Contrarily to the link prediction algorithms, they don't rely on the analysis of each possible link of the network. Instead, they visit nodes just a constant number of time.

6 OPINION DYNAMICS

In this section, several applications of Opinion Dynamics algorithm are described. For the sake of example, one could imagine an opinion in a co-authorship network as a tool, idea or methodological innovation involving a research field. In the following presentation of diffusion algorithms, we abstract from the specific innovation and we apply algorithms taking into account the more possible generic meaning of opinion.
In the next paragraphs we compare several algorithms executed with several initial configurations and parameters. They take into account both discrete and continuous opinions. The considered models are Voter, Q-Voter, Majority Rule, Sznajd and Deffuant. In order to understand the obtained results and the differences among them, a concise description of them is reported in the following.

- *Voter* [8] is a discrete Opinion Dynamics model in which an agent opinion changes if it is randomly selected. The new opinion of it will be the one of one of its neighbors.
- *Sznajd* [34] introduces the concept of group, considering the opinion of a collectivity more influential than the ones of single individuals. Indeed, two random-selected neighbors spread their opinion to their neighbors if they share the same opinion. Otherwise, the opposite effect is produced.
- *Q-Voter* [6] combine the ideas of the previous two models: q neighbors are randomly selected, and if they agree they influence another random neighbor opinion.
- *Majority rule* [14]: based on the opinion dynamics that characterizes public debates, it assumes that in a random group individuals take the majority opinion within the group.
- *Deffuant* [11] is a continuous model that takes into account the interaction between pairs of individuals. Depending on their open-mindness, two individuals can change their opinion. Their new opinion at the new iteration result the average of their opinion at the previous one.

For all the described models we availed ourseleves of the functions provided by the Python library NDlib (Network Diffusion) [29]. Concerning the last of the described models, we implemented it by means of the Algorithmic Bias algorithm [33], whose implementation is provided by such library (released in NDlib 4.0.1). Indeed, the results of the application of the Deffuant model correspond to the execution of Algorithmic Bias algorithms with parameter $gamma = 0$.

Experiments

The algorithms implementing all the mentioned models have been executed with several parameters, varying both the initial condition and the algorithms iterations. For each of the algorithm, the initial population characterized by the new opinion (i.e. *fraction_infected* parameter) has been 0.1, 0.3, 0.4 and 0.5. Concerning Majority Rule and Q-Voter, we also varied the q parameter, from 3 to 7, with an increment of 2. Table 7 illustrates the algorithms executions performed on the network. For each execution we got both the trend and the prevalence graphics, by which we could evaluate and discuss the diffusion of the opinions.

In addition to the configuration of Table 7, we executed Deffuant algorithm varying the *fraction_infected* (0.15 and 0.25) and *epsilon* (0.30 and 0.40) parameters. Figure 7 shows the most relevant outcomes of such executions.

For all the executions reported in Figure 7 we executed 1500 iterations. Only 300 have been executed, instead, for Deffuant algorithm executions, by which we didn't get significant results. As we can observe from subfigures 7a, 7c, 7e and 7g, low initial values (lower than 0.5) do not lead to an important spreading within the given number of iteration. Contrarily, subfigures 7b, 7d, 7f do reveal more interesting patterns: in two of the three cases, the infected individuals (i.e. authors reached by the innovation, in our study case) prevail within the first 1500 iterations. Finally, with Sznajd a different pattern is followed, since the number of susceptible and infected tends to the stability after a few dozens of iterations.

Table 7: Opinion Diffusion algorithms configurations

Algorithm	Initial opinion spreading	q	iterations
Voter	0.1, 0.3, 0.4, 0.5	-	1500
Q-Voter	0.1, 0.3, 0.4, 0.5	3, 5, 7	1500
Sznajd	0.1, 0.3, 0.4, 0.5	-	1500
Majority rule	0.1, 0.3, 0.4, 0.5	3, 5, 7	1500

7 COMMUNITY DISCOVERY IN WEIGHTED NETWORKS

Canonical Community Discovery algorithms such as those applied in Section 4 do not exploit network weights. Nonetheless, they can reveal additional information about communities. Thus, we analysed and implemented a local, efficient algorithm for detecting communities in weighted networks [7]. By means of it, we are also able to find overlapping communities, that are quite frequent in co-authorship networks, since an author can belong to different research groups during its career.

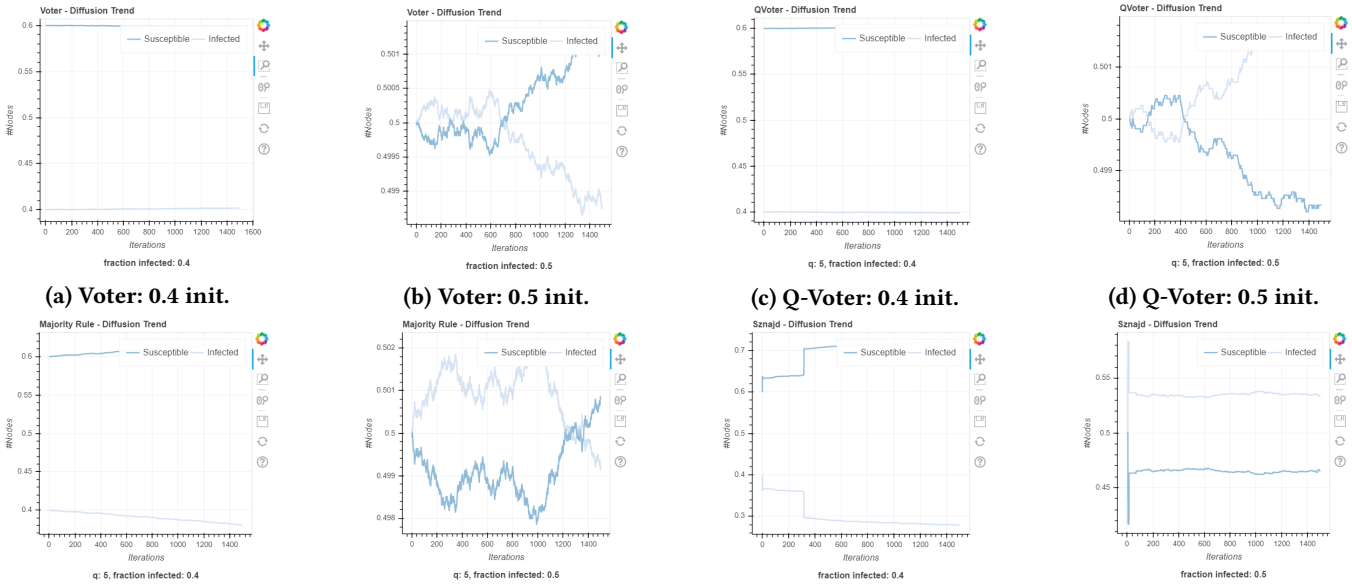
Adopted measures

The community detection technique adopted by the chosen algorithm [7] mainly exploits three measures regarding both nodes and community, namely *Node Strength*, *Belonging degree* and Q_0 *Modularity*. In order to keep our paper self-contained, we provide their definitions together with a brief description of the authors method. While node strength is a local property, belonging degree and Q_0 *Modularity* take into account respectively the community of the node and the whole graph.

Definition 7.1. (Node strength)
Given a node u belonging to a network $G = (V, E)$ with $|V| = n$ and $|E| = m$, its node strength is defined as

$$k_u = \sum_{v \in V} w_{uv}$$

where w_{uv} is the weight of the edge e_{uv}, that is 0 if the edge does not exist. Intuitively, it represent its weighted degree, since it is obtained as the sum of the weights of its links.



(a) Voter: 0.4 init. **(b) Voter: 0.5 init.** **(c) Q-Voter: 0.4 init.** **(d) Q-Voter: 0.5 init.**

(e) Majority Rule: 0.4 init. **(f) Majority Rule: 0.5 init.** **(g) Sznajd: 0.4 init.** **(h) Sznajd: 0.5 init.**

Figure 7: Results of the opinion dynamics algorithms.

Definition 7.2. Belonging degree
Given a node u belonging to a community C, the belonging degree $B(u, c)$ is defined as

$$B(u, C) = \frac{\sum_{v \in C} w_{uv}}{k_u}$$

where w_{uv} is the weight of the edge e_{uv}, that is 0 if the edge does not exist. Less formally, the belonging degree of a node with respect to a community takes into account both the absolute node strength and the portion of it deriving from nodes of the same community. It is easy to demonstrate that this value is always between $[0, 1]$.

Finally, the algorithm proposed in [7] needs a measure that quantifies the strength of division of the network into communities. Authors adopt the following measure derived by the one proposed in [20]. In such a way, the discovery algorithm can be also adopted for overlapping networks, since adopting Q_0 Modularity the algorithm doesn't suffer the resolution limit [13].

Definition 7.3. Q_0 Modularity
Given a network $G = (V, E)$, the adjacency matrix A and its set C of communities, the Q_0 modularity is defined as follows:

$$Q_0 = \frac{1}{2m} \sum_{c \in C} \sum_{u,v \in V} \alpha_{cu} \alpha_{cv} \left(A_{uv} - \frac{k_u k_v}{2m} \right)$$

where a_{cu} is a belonging coefficient defined as follows

$$\alpha_{cu} = \frac{k_{cu}}{\sum_{c \in C} k_{cu}}$$

and satisfying the condition $0 \leq \alpha_{cu} \leq 1, \forall c \in C, u \in V$. Furthermore, we have that $k_{cu} = \sum_{v \in C} w_{uv}$.
In this way, the belonging coefficient includes the possibility of a node to be assigned to more communities and its definition is consistent with the one of modularity for a non-overlapping community [20].

The algorithm

The implemented strategy for the detection of overlapping communities in weighted networks is composed by iterations of two main steps, namely *detection of the initial community* and *expansion of the community*. Although all the implementation details are available at our mentioned GitHub repository, we describe the key steps of both phases. The following steps are iterated until every node belongs at list to a community. For this, an initialization procedure in which all nodes are labeled is performed.

Detection of the initial community. The initial community is detected among the non-labeled nodes. It starts from the strongest node (w.r.t. above defined node strength measure) and it spreads over its neighbors respecting some defined conditions related to their *belonging degree*. Such process is iterated until the constraints on such measure are satisfied by all the nodes in the community. These constraints are based on their belonging degree with respect to the community.

Community expansion. The expansion is performed considering the neighbors of the initial community. New nodes are classified depending on their *belonging degree*. Those having

belonging degree in a defined range are either removed from or added to the community depending on their influence on the Q_0 Modularity: if they increase it, they remain and are labeled. Otherwise, they are discarded.

Complexity

In order to calculate the computational complexity of the algorithm, some observations are fundamental. In particular, the most expansive step is the computation of the $Q_0 Modularity$ (Def. 7.3). According to the *Community expansion* step, $Q_0 Modularity$ is computed every time a neighbor of the initial community has a *belonging degree* ranging from *min_bel_degree* and *max_bel_degree*. Such computation is quadratic with respect to the size of the community.
Thus, the overall worst case time complexity of the algorithm is $O(n^3)$. We overcame this problem by setting the two mentioned parameter to the same value: in this way, the set for which the $Q0Modularity$ needs to be calculated is empty at each iteration, although the overlapping of the communities cannot be exploited.

Evaluation

To the best of our knowledge, no universally shared definition of community exists in literature. Furthermore, we don't know the ground truth underlying our network with respect to the composition of the communities. For these reason, evaluate and compare the result of the execution of our algorithm is a challenging task. We address this problem in two ways. On the one hand, we extended the table including the evaluation of the standard community detection algorithm (Table 6). Next, we propose a comparison of the algorithms performance based on the results of Label Propagation, since it was the best according to the evaluations of Section 4.
The updated table is reported in Table 8.

Table 8: Community Discovery algorithms evaluation

Evaluation	Label Propagation	Louvain	k-Clique	Demon	Custom
Internal Edge Density	0.19	0.08	**0.23**	0.17	0.06
Average Internal Degree	5.38	**78.16**	5.14	10.02	2.23
Modularity Density	**1106.91**	10.01	-59631.28	-17937.91	832.07
Conductance	0.25	**0.06**	0.46	0.38	**0.06**

Although the overlapping structure of the community has not been exploited due to our computational limitations, the implemented algorithm achieved the best result in terms of conductance, that is the same as the one obtained by *Louvain*. Finally, the results of the comparison among the algorithms are reported in Table 9.

For the comparison of Table 9 we adopted the normalized F1 score, also adopted in [30].

Table 9: Community Discovery algorithms comparison

Measure	Louvain	k-Clique	Demon	Custom
Normalized F1	0.00024	0.492	0.352	0.17

8 FINDING CORRELATIONS

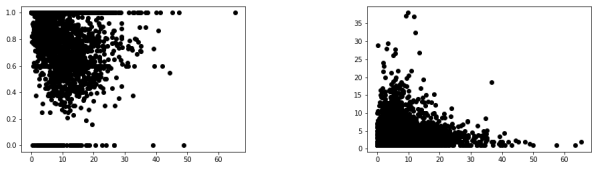
In this section, we avail of both the network and the performed analyses in order to answer several questions concerning the correlation between the authors' features and some local and global network and community properties. The graph adopted for this analysis is a subgraph of the initial one, corresponding to its biggest connected component. We considered the following key questions: Are centrality measures correlated with authors' citations and/or performance-index? Are node strength and belonging degree (w.r.t. a community) representative for other features? Are community measures good predictors of individual features?

Correlations

In addition to the obvious (positive) correlations among *number of publications*, *h-index* and *number of citations*, we found some pattern underlying authors' data. Although the low quality of data (see Sec. 9) in which important information miss, we discovered that authors having a number of publications over the average (and over the median) have a lower betweenness centrality, and vice-versa ($p \leq 0.03$). The same result, although less evident, is valid for *h-index*.
Contrarily to the results shown in [37], we didn't find positive correlations between centrality measures and citation counts in the analysed subgraph. In our opinion, this is due to the fact that we considered only a subgraph of the co-authorship graph, namely the 22^{nd} biggest connected component of the original graph. Moreover, our initial graph is a subgraph of the global co-authorship graph, since it is obtained only from papers published in 2016.

Communities

We also asked ourselves whether communities are good predictors of authors' performance. We conducted separated analyses of the collected authors' measures grouped by community and we observed the relevant trends. In particular, we wanted to assess whether higher clustering coefficient (i. e. denser networks) correspond to better performance, since they indicate stronger collaborations among researchers. For it, we obtained the communities from Label Propagation algorithm, since it resulted one of the best community discovers according to the evaluations presented in Section 4.



(a) Average h-index (x axis) and global clustering coefficient (y axis)

(b) Average h-index (x axis) and average community degree (y axis)

Figure 8: Correlation between community properties and authors' features.

The results of the correlation are illustrated in the two scatterplots of Figure 8. From them, we can observe that communities with high average h-index show medium-high clustering coefficient (Subfigure 8a) and low average degree (8b).

9 DISCUSSION

Some of the collected authors' features presented a significant number of missing values, so we could not exploit all of the attributes such as research fields and list of organizations. Concerning the number of citations, we had about two thirds of the total values, thus we performed a t-test for the observations in Sec 8. As a future investigation, it would be interesting to enrich the analysed data to expand our knowledge about the global research community, and to group them by location, studied topics or communities. We think that a high number of intuitive and counter-intuitive correlations among them could be revealed.

A link prediction approach in which we also consider research topics and paper titles could also be designed. For it, we would need more data.

Another relevant property of networks are flows. We could exploit directness of a similar network and consider this concept in order to predict the network flows described by citations. Concerning the achieved results, in Section 8 we found interesting and counter-intuitive correlations between h-index and clustering coefficient.

All the presented tasks highlighted the importance of computational efficiency in graph algorithms. All network analysis, link prediction and community discovery tasks have presented limits in this sense with respect to our computational resources. Indeed, the calculation of the diameter of a high-dimensional network, the task of link prediction as well as the computation of the modularity have constituted the main criticism of our work. While for the first the execution time have been considered as acceptable (see Sec. 3), we skip the second (see Sec. 5) and we avoided the calculation of the third by means of the apposite parameters (see Sec. 7). Concerning the adopted and implemented community discovery algorithms, our effort has been focused on the evaluation, since no ground truth was available about the network communities. Finally, more interesting results could be achieved in the opinion dynamics analysis of the network, especially by means of more executions of the Algorithmic Bias model (the most computational expensive one).

10 CONCLUSIONS

In our paper we collected and analysed the co-authorship network obtained by all the publications of the year 2016 available in the Microsoft Open Academic Graph. During our work, two significant limitations raised: data quality and computational limits. Nonetheless, we analysed the network structure and we applied several state-of-the art algorithms concerning Community Discovery, Link Prediction, and Opinion Dynamics. Furthermore, we implemented an interesting (w.r.t. our network) version of CD algorithms considering both network weights and communities overlapping, but we could not exploit the proposed definition of modularity due to computational limits. Finally, some correlations have been found between network properties and authors' ones.

We consider that our results are useful in order to lay the groundwork for further investigations in Community Discovery and Network Diffusion applied to collaboration networks.

REFERENCES

[1] Lada Adamic and Eytan Adar. 2003. Friends and Neighbors on the Web. *Social Networks* 25 (07 2003), 211–230. https://doi.org/10.1016/S0378-8733(03)00009-1

[2] Albert-László Barabási and Réka Albert. 1999. Emergence of Scaling in Random Networks. *Science* 286, 5439 (1999), 509–512. https://doi.org/10.1126/science.286.5439.509 arXiv:https://science.sciencemag.org/content/286/5439/509.full.pdf

[3] A.L Barabási, H Jeong, Z Néda, E Ravasz, A Schubert, and T Vicsek. 2002. Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications* 311, 3 (2002), 590 – 614. https://doi.org/10.1016/S0378-4371(02)00736-7

[4] Alex Bavelas. 1950. Communication Patterns in Task-Oriented Groups. *Acoustical Society of America Journal* 22, 6 (Jan. 1950), 725. https://doi.org/10.1121/1.1906679

[5] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (oct 2008), P10008. https://doi.org/10.1088/1742-5468/2008/10/p10008

[6] Claudio Castellano, Miguel Angel Muñoz, and Romualdo Pastor-Satorras. 2009. Nonlinear q-voter model. *Physical review. E, Statistical, nonlinear, and soft matter physics* 80 4 Pt 1 (2009), 041129.

[7] Duanbing Chen, Mingsheng Shang, Zehua Lv, and Yan Fu. 2010. Detecting overlapping communities of weighted networks via a local algorithm. *Physica A Statistical Mechanics and its Applications* 389, 19 (Oct. 2010), 4177–4187. https://doi.org/10.1016/j.physa.2010.05.046

[8] PETER CLIFFORD and AIDAN SUDBURY. 1973. A model for spatial conflict. *Biometrika* 60, 3 (12 1973), 581–588. https://doi.org/10.

1093/biomet/60.3.581 arXiv:https://academic.oup.com/biomet/article-pdf/60/3/581/576759/60-3-581.pdf

[9] Michele Coscia, Giulio Rossetti, Fosca Giannotti, and Dino Pedreschi. 2012. DEMON: a local-first discovery method for overlapping communities. In *KDD*.

[10] Gabor Csardi, Tamas Nepusz, et al. 2006. The igraph software package for complex network research. *InterJournal, complex systems* 1695, 5 (2006), 1–9.

[11] Guillaume Deffuant, David Neau, Frédéric Amblard, and Gérard Weisbuch. 2000. Mixing Beliefs Among Interacting Agents. *Advances in Complex Systems* 3 (01 2000), 87–98. https://doi.org/10.1142/S0219525900000078

[12] P Erdös and A Rényi. 1959. On Random Graphs I. *Publicationes Mathematicae Debrecen* 6 (1959), 290–297.

[13] Santo Fortunato and Marc Barthelemy. 2007. Resolution limit in community detection. *Proceedings of the National Academy of Sciences* 104 (2007), 36 – 41.

[14] Serge Galam. 2002. Minority Opinion Spreading in Random Geometry. *Physics of Condensed Matter* 25 (03 2002). https://doi.org/10.1140/epjb/e20020045

[15] Aric Hagberg, Pieter Swart, and Daniel Chult. 2008. Exploring Network Structure, Dynamics, and Function Using NetworkX. *Proceedings of the 7th Python in Science Conference*.

[16] Jianbo Shi and J. Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 8 (2000), 888–905.

[17] Stefan Krätke and Arno Brandt. 2009. Knowledge Networks as a Regional Development Resource: A Network Analysis of the Interlinks between Scientific Institutions and Regional Firms in the Metropolitan region of Hanover, Germany. *European Planning Studies* 17, 1 (2009), 43–63. https://doi.org/10.1080/09654310802513930 arXiv:https://doi.org/10.1080/09654310802513930

[18] Zhenping Li, Shihua Zhang, Rui-Sheng Wang, Xiang Zhang, and Luonan Chen. 2008. Quantitative function for community detection. *Physical review. E, Statistical, nonlinear, and soft matter physics* 77 (04 2008), 036109. https://doi.org/10.1103/PhysRevE.77.036109

[19] M. Newman. 2001. Newman, M.E.J.: Clustering and preferential attachment in growing networks. Phys. Rev. E 64, 025102. *Physical review. E, Statistical, nonlinear, and soft matter physics* 64 (09 2001), 025102. https://doi.org/10.1103/PhysRevE.64.025102

[20] Mark Newman and Michelle Girvan. 2004. Finding and Evaluating Community Structure in Networks. *Physical review. E, Statistical, nonlinear, and soft matter physics* 69 (03 2004), 026113. https://doi.org/10.1103/PhysRevE.69.026113

[21] M. E. J. Newman. 2004. Coauthorship networks and patterns of scientific collaboration. *Proceedings of the National Academy of Sciences* 101, suppl 1 (2004), 5200–5205. https://doi.org/10.1073/pnas.0307545100 arXiv:https://www.pnas.org/content/101/suppl$_1$/5200.$full.pdf$

[22] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* 435 (07 2005), 814–818.

[23] Derek J. De Solla Price. 1965. Networks of Scientific Papers. *Science* 149, 3683 (1965), 510–515. http://www.jstor.org/stable/1716232

[24] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. 2004. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America* 101 (04 2004), 2658–63. https://doi.org/10.1073/pnas.0400054101

[25] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. 2004. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America* 101 (04 2004), 2658–63. https://doi.org/10.1073/pnas.0400054101

[26] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* 76 (Sep 2007), 036106. Issue 3. https://doi.org/10.1103/PhysRevE.76.036106

[27] Giulio Rossetti. 2020. Exorcising the Demon: Angel, Efficient Node-Centric Community Discovery. In *Complex Networks and Their Applications VIII*, Hocine Cherifi, Sabrina Gaito, José Fernendo Mendes, Esteban Moro, and Luis Mateus Rocha (Eds.). Springer International Publishing, Cham, 152–163.

[28] Giulio Rossetti, Letizia Milli, and Rémy Cazabet. 2019. CDLIB: a python library to extract, compare and evaluate communities from complex networks. *Applied Network Science* 4 (Dec. 2019), 52. https://doi.org/10.1007/s41109-019-0165-9

[29] Giulio Rossetti, Letizia Milli, and Salvatore Rinzivillo. 2018. NDlib: A Python Library to Model and Analyze Diffusion Processes over Complex Networks. 183–186. https://doi.org/10.1145/3184558.3186974

[30] Giulio Rossetti, Luca Pappalardo, and Salvatore Rinzivillo. 2016. A novel approach to evaluate community detection algorithms on ground truth.

[31] Gerard Salton and Michael J. McGill. 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., USA.

[32] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. 2015. An Overview of Microsoft Academic Service (MAS) and Applications. 243–246. https://doi.org/10.1145/2740908.2742839

[33] Alina Sîrbu, Dino Pedreschi, Fosca Giannotti, and János Kertész. 2019. Algorithmic bias amplifies opinion fragmentation and polarization: A bounded confidence model. *PLoS ONE* 14 (2019).

[34] Katarzyna Sznajd-Weron and J. Sznajd. 2000. Opinion evolution in closed community. *HSC Research Reports* (2000).

[35] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, li Zhang, and Zhong Su. 2008. ArnetMiner: Extraction and Mining of Academic Social Networks. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 990–998. https://doi.org/10.1145/1401890.1402008

[36] D.J. Watts and S.H. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature* 393 (1998), 440–442.

[37] Erjia Yan and Ying Ding. 2009. Applying centrality measures to impact analysis: A coauthorship network analysis. *Journal of the American Society for Information Science and Technology* 60, 10 (2009), 2107–2118. https://doi.org/10.1002/asi.21128 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.21128