GRAPH MINING AND COMMUNITY EVALUATION WITH DEGENERACY

Christos Giatsidis

GRAPH MINING AND COMMUNITY EVALUATION WITH DEGENERACY

DISSERTATION

submitted in partial fulfillment

of the requirements for the degree of

DOCTOR OF ÉCOLE POLYTECHNIQUE

by

GIATSIDIS CHRISTOS

FINAL DRAFT

If the only tool you have is a hammer,
every problem will look as a nail.

Acknowledgments

First and foremost my appreciation and gratitude to my supervisor **Michalis Vazirgiannis** (LIX,École Polytechnique) for showing his trust and support throughout my Phd studies.

To Prof. **Dimitrios M. Thilikos** (LIRMM, CNRS & Univ. Athens) I express my gratitude for helping me with his vast expertise on Graph Theoretic subjects.

Many thanks to Professor **Stefano Leonardi** (Sapienza University of Rome), Professor **Jordi Herrera-Joancomartí** (Universitat Autònoma de Barcelona) and Professor **Jie Tang** (Tsinghua University) for their thorough reviews and comments of this PhD thesis

Moreover I would like to thank Professor **Nicolas Vayatis** (CMLA,Ecole Normale Supérieure de Cachan) and Professor **Marc Barthelemy** (Institut de Physique Theorique, CEA, CNRS-UR) for their participation in the Jury for my defense and for their valid and interesting comments.

I would also like to thank Prof. **Christos Faloutsos** (Carnegie Mellon University) for accepting to be in the Jury for my defense and his assistance in preparing a concise presentation.

Finally, many thanks to Prof. **Jean-Marc Steyaert** (LIX,École Polytechnique) for his advice and also for serving as the President of the Jury.

Abstract

The study and analysis of social networks attract attention from a variety of Sciences (psychology, statistics, sociology). Among them, the field of Data Mining offers tools to automatically extract useful information on properties of those networks. More specifically, Graph Mining serves the need to model and investigate social networks especially in the case of large communities – usually found in online media – where social networks are prohibitively large for non-automated methodologies.

The general modeling of a social network is based on graph structures. Nodes of the graph represent individuals and edges signify different actions or types of social connections between them. A community is defined as a subgraph (of a social network) and is characterized by dense connections. Various measures have been proposed to evaluate different quality aspects of such communities – in most cases ignoring various properties of the connections (e.g. directionality).

In the work presented here, the k-core concept is used as a means to evaluate communities and extract information. The k-core structure essentially measures the robustness of an undirected network through degeneracy. Further more extensions of degeneracy are introduced to networks that their edges offer more information than the undirected type.

Starting point is the exploration of properties that can be extracted from undirected graphs (of social networks). On this, degeneracy is used to evaluate collaboration features – a property not captured by the single node metrics or by the established community evaluation metrics – of both individuals and the entire community. Next, this process is extended for weighted, directed and signed graphs offering a plethora of novel evaluation metrics for social networks. These new features offer measurement tools for collaboration in social networks where we can assign a weight or a direction to a connection and provide alternative ways to signify the importance of individuals within a community. For signed graphs the extension of degeneracy offers additional metrics that can be used for trust management.

Moreover, a clustering approach is introduced which capitalizes on processing the graph in a hierarchical manner provided by its core expansion sequence, an ordered partition of the graph into different levels according to the k-core decomposition

The graph theoretical models are then applied in real world graphs to investigate trends and behaviors. The datasets explored include scientific collaboration and citation graphs (DBLP and ARXIV), a snapshot of Wikipedia's inner graph and trust networks (e.g. Epinions and Slashdot). The findings on these datasets are interesting and the proposed models offer intuitive results.

Résumé

L'étude et l'analyse des réseaux sociaux attirent l'attention d'une variété de sciences (psychologie, statistiques, sociologie). Parmi elles, le domaine de la fouille de données offre des outils pour extraire automatiquement des informations utiles sur les propriétés de ces réseaux. Plus précisément, la fouille de graphes répond au besoin de modéliser et d'étudier les réseaux sociaux en particulier dans le cas des grandes communautés que l'on trouve habituellement dans les médias en ligne oú la taille des réseaux sociaux est trop grande pour les méthodes manuelles.

La modélisation générale d'un réseau social est basée sur des structures de graphes. Les sommets du graphe représentent les individus et les arêtes des actions différentes ou des types de liens sociaux entre les individus. Une communauté est définie comme un sous-graphe (d'un réseau social) et se caractérise par des liens denses. Plusieurs mesures ont été précédemment proposées pour l'évaluation des divers aspects de la qualité de ces communautés mais la plupart d'entre elles ignorent diverses propriétés des interactions entre individus (par exemple l'orientation de ces liens).

Dans la recherche présentée ici, le concept de "k-core" est utilisé comme un moyen d'évaluer les communautés et d'en extraire des informations. La structure de "k-core" mesure la robustesse d'un réseau non orienté en utilisant la dégénérescence du graphe. En outre, des extensions du principe de dégénérescence sont introduites pour des réseaux dont les arêtes possèdent plus d'informations que celles non orientées.

Le point de départ est l'exploration des attributs qui peuvent être extraits des graphes non orientés (réseaux sociaux). Sur ce point, la dégénérescence est utilisée pour évaluer les caractéristiques d'une collaboration entre individus et sur l'ensemble de la communauté - une propriété non capturée par les métriques sur les sommets individuels ou par les métriques d'évaluation communautaires traditionnelles. Ensuite, cette méthode est étendue aux graphes pondérés, orientés et signés afin d'offrir de nouvelles mesures d'évaluation pour les réseaux sociaux. Ces nouvelles fonctionnalités apportent des outils de mesure de la collaboration dans les réseaux sociaux oú l'on peut attribuer un poids ou un orientation à une interaction et fournir des moyens alternatifs pour capturer l'importance des individus au sein d'une communauté. Pour les graphes signés, l'extension de la dégénérescence permet de proposer des métriques supplémentaires qui peuvent être utilisées pour modéliser la confiance.

De plus, nous introduisons une approche de partitionnement basée sur le traitement du graphe de manière hiérarchique, hiérarchie fournie par le principe de "core expansion sequence" qui partitionne le graphe en différents niveaux ordonnés conformément à la décomposition "k-core".

Les modèles théoriques de graphes sont ensuite appliqués sur des graphes du monde réel pour examiner les tendances et les comportements. Les jeux de données explorés incluent des graphes de collaborations scientifiques et des graphes de citations (DBLP et ARXIV), une instance de graphe interne de Wikipédia et des réseaux basés sur la confiance entre les individus (par exemple Epinions et Slashdot). Les conclusions sur ces ensembles de données sont significatives et les modèles proposés offrent des résultats intuitifs.

Contents

1

Introduction

1.1 Introduction

Large and evolving graphs constitute an important element in current large-scale information systems. Common cases of such graphs are the Web, social networks, citation graphs, CDRs (Call Data Records) where nodes – featured with, in some cases, many attributes – are connected to each other with directed edges, representing a relation such as endorsement, recommendation or friendship. The Web, social networks, and citation graphs form a context where the detection and evaluation of communities constitutes an important and challenging task. In all cases, due to the economic significance of these networks, the ranking of individual nodes is also a necessity.

The research methods in this area have mainly capitalized on the Hub/Authority concepts (see [75, 87]), evaluating communities based on the centrality of nodes in terms of incoming/outgoing links. Graphs of real-world data with community structure have vertex degree with a wide range. As pointed out in [38], nodes of low degree coexist with nodes of high degree making the graph inhomogeneous both globally and locally which usually indicates particularities in its structure, for instance, communities.

But the inherent mechanisms of community creation and evolution are not solely based on the Hub/Authority concepts. An important constituent of such a mechanism, generally neglected, is the community cohesion in terms of a dense distribution of in/outlinks within the community – as opposed to sparse connections across them. One of the main interests of this work is in quantifying the degree of cohesion of a community sub-graph as a measure of collaboration among its members.

1.2 Network Communities

Graphs representing real systems have a unique structure that displays a form of order. The degree distribution of the nodes can vary greatly and it usually follows a power law [37]. Specifically, a lot of low degree nodes coexist with a few nodes of high degree. The same inhomogeneous pattern can be seen locally as well; there are groups of nodes displaying high concentration of connections while the

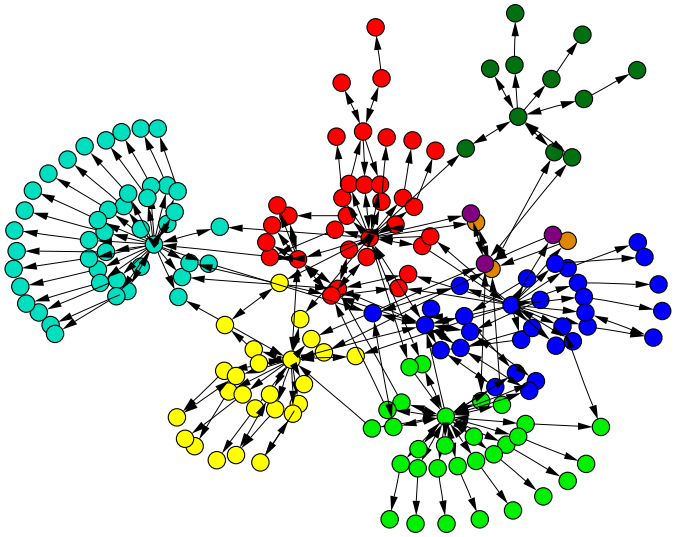


Figure 1.1: Example of a network with different communities (marked by color)[38].

groups themselves are spatially connected. These attributes are all characteristics of social networks (among other types of real networks) and are attributed as a "community structure" [41].

Community detection and evaluation is an important task in graph mining. The general idea of community detection is to identify groupings within the graph structure and possibly hierarchies of groups within them. Community evaluation aims at quantifying the "importance" of nodes (or groups of them that belong to the same community) using criteria that depend on the definition of importance (e.g. importance of influence). Community evaluation metrics can also be used in community detection as either a way to define the similarity function between nodes or as a feature to help in selecting points of interest.

But the idea of communities in graph networks has ambiguous interpretations. There is not a universal definition [38]. Here, a community is considered as a sub-graph with much denser connections (or interactions) among its nodes than the rest of the graph. Although this is not very specific for a community detection task, the purpose of this definition is only to provide a basic and intuitive reference to the structures that will be regarded as communities.

An example can been seen in Figure 1.1 where different colors represent community sub-graphs of the entire network. There, the properties of "community structure" become apparent. As the term "community" may refer to an entire network (e.g. the community of physicists) or a part of it (e.g. a group of scientist that collaborate often), for the rest of the document the use of this term is explicitly defined to which case it refers when needed.

1.2.1 *Applications*

Community detection and evaluation serve interesting real world applications. One prominent example can be seen in the area of recommendation systems. People in a social network sharing "friendship" connections create groups. The activities of some individuals in one group (e.g. purchase of a product, liking a movie) can be used as a way to predict activities that others of the same community would seek to participate.

Another example is that of data segmentation (or sharding). Social networks have increased in sizes (in terms of users) too big for single location data management. It is quite often that the need arises to partition the database containing the social network into "shards" (by database rows while trying to maintain the information of the schema within the same part) that are quite often stored in different locations. Community detection can be used to create an efficient partitioning.

Community evaluation can be viewed as an evaluation of single entities and their "role" within their community or as an evaluation of a sum of entities and their collective actions. Both situations can be used as part of the community detection procedure but can also serve directly application oriented requirements. A simple example would be identifying people with influence within their "environment". While at first this appears to have a trivial solution (e.g. node of high degree) it becomes more complex when restrictions from real world applications are applied (e.g. high diversity in the groups someone belongs to).

As social networks are sometimes integrated along with reviewing systems (e.g. Epinions) or collaborative systems (e.g. Wiki platforms of information) the issue of discovering good reviewers can be seen as a community evaluation issue as well. In such scenarios one may need to evaluate either the trustworthiness of individuals or even the credibility of their collective "work".

A need for expert consulting is needed in many small businesses (usually for a short period of time). For this need, there are on-line services (e.g. Expert Exchange) that provide the means to contact professional with very specific technical skills. Quite often, these on-line platforms of expertise are accompanied with a social network of the experts. While issues of specific nature are easy to address with such services, the need of ad doc team formation has also become frequent. Requiring a team to assist in broader issues can be also seen as a community detection problem. The benefit of knowing the underlying social network is that the choice in the list of professionals can also be assisted by their connections (through community detection and evaluation).

Community detection and evaluation methodologies are not limited to social networks. The same principles can be (and are) applied to networks of a different context but display the same community like structure. Most networks that are associated in some way with human (inter)actions display properties of

the "community structure". Whether it is pages of the World Wide Web, articles in Wikipedia, personal WebBlogs or just comments under a YouTube video, all of these create underlining networks that follow similar properties. In essence the same algorithmic techniques can be used to find structure that are similar throughout the different networks but with altered (depending on the situation) semantics.

1.2.2 *Collaboration*

An interesting aspect of social networks, highlighted by the some of the examples, is that of collaboration. Of course, collaboration is a concept not applicable to all social networks. But that is only semantics; collaboration is the process of separate entities (people, organizations etc.) working together to accomplish a task. While this is directly applicable to some social networks (e.g. social network of professionals) it can also make sense in other types of networks as well. Taking for example a group of densely connected web pages with articles one could interpret their "collaboration" as their cumulative ability to provide information on a subject.

When looking some of the examples of applications, it is obviously a necessity to be able to quantify collaboration. One may need to decide the collaborative value of an individual entity in relevance to a specific team in order to decide whether it is a "good fit". Even the evaluation of entire communities show interest when trying to inspect whether they offer a good environment for collaboration. For example one may try to compare communities drawn from various conferences to see which ones have strong collaborations and therefore might offer a welcoming environment.

1.2.3 *Degeneracy*

The degeneracy of an undirected graph G is also known as its "k-core number". The k-core of a graph G is the largest subgraph of G for which every node has a degree of at least k within the sub-graph. The degeneracy of a graph is the maximal value of k such that the k-core is not empty. In simple terms, in a k-core of a network an individual is connected to at least other k members (of the network) and those are also connected to (at least) k "neighbors" as well. The process of computing all k-cores of a graph G, for $k = 1, 2, 3, \ldots, k_{max}$ (where k_{max} is the degeneracy of the graph), is the called k-core decomposition.

The graph theoretic study of degeneracy and k-cores dates back to the 60's [36, 66, 81, 85]. It has been used to understand various properties of random graphs [61, 76]. Moreover, it has been extensively used, in an experimental level, for eval-

uating and detecting strongly cohesive communities in real-word graphs [2, 3, 8, 11, 20, 90]. The k-core decomposition process has also been used to provide an approximation algorithm for the dense subgraph problem [5, 50].

One of the main benefits of the k-core structure is the efficiency of the algorithms for computing a k-core. For graph G, all that is needed is to recursively remove the vertices (which represent nodes of the modeled network) that have a degree lower than k. Once a vertex is removed all of its connections are removed as well. The running time of this process can be proportional to the number of edges (the connections of the network) of the graph [13].

1.2.4 *Beyond Simple Graphs*

Degeneracy has been defined and explored on simple undirected graphs. For real networks this simple representation bears the semantic of an equal and symmetric relationship (e.g. two people being friends). This representation is not sufficient in the study of a more complex context of connections that may have:

- **Weight:** Weight can represent the strength of the relationship. For example the weight of a professionals' social network could indicate the frequency two of them work together.

- **Direction:** Directed networks have become most prominent, but not limited, in the online environment of the world wide web. Some examples are web sites linking to one another, social networks of users following posts in microbloggs, networks formed by on line voting/liking and citation networks. This directionality is obviously necessary in the corresponding graphical model. For instance, one could compare social networks of friendship with microblogging social networks. On one, friendship is in both directions and therefore unidirectional in the corresponding graph model. On the other, not including the direction would make the statement "A follows microblogg of B" equivalent to the exact opposite.

- **Labels:** Labeling a connection creates entirely new contexts (depending on the label). One example is that of the " trust" concept in social networks. Practically, there are two ways to treat labels, one is to assume them as separate entities and essentially interpret the entire modeled network as an overlap of many networks with the same nodes. The other is to map the labels into weights (e.g. in the case of trust integer values: 0 & 1 or -1 & 1). Choosing either way depends on the reason for modeling the network.

1.2.5 *Extending Degeneracy*

The work presented here starts with community evaluation, based on the k-core concept, as a means of evaluating collaborative nature of individuals in networks modeled by simple graphs. The vertices of a graph G represent a set of entities and its edges represent collaboration links between them, thus a core of high value for k (or simply a high index core) in a graph G can be seen and treated as a community of entities that demonstrates a strong collaboration between them.

Building upon the findings of the previous, extensions of the k-core concept are introduced. These extensions aim at introducing degeneracy to the more complex graph structure cases mentioned above. Along with the extensions, interesting metrics and ways to visualize graph structure arise. The extensions are utilized for the evaluation of collaboration (and trust) under weighted, directed and labeled graphs. Specifically:

- Fractional k-cores are defined for edge-weighted graphs. This new concept displays interest in both theory and practice and the necessity of this extension is displayed by evaluating the same network under the two structures (k-cores and the fractional extension) and comparing the resulting subgraphs.

- The theoretical framework of cores is vastly extended to the case of directed graphs. Such graphs emerge naturally from social/citation networks and the Web. D-cores constitute dense directed sub-graphs of the original one involving intensive and mutual collaboration in terms of directed links. Interestingly, all these notions induce a 2-dimensional setting indicating qualitative differences for the directed case and are later employed and visualized during experimentation.

- Based on the D-core extension, new structures and metrics are defined for the evaluation of the collaborative nature of directed graphs. Namely, such are the D-core matrix for a graph, its frontier, and a series of novel metrics to evaluate:

  a. the robustness of the directed graph under degeneracy, as a metric of cohesiveness and hence the collaboration among the members of the graph under study and

  b. the dominant patterns of the graph with respect to the inlink/outlink trade off indicating macroscopic graph patterns related to whether the graph is extrovert or "selfish".

- Finally, cores on signed graphs (that model trust networks) are defined for the evaluation of trust (S-cores). As this extension can not be compared with

existing metrics, existing metrics are also redefined (extended) in parallel for trust networks. The signs on the edges of the graph are treated as labels and this creates a much richer setting for experiments and the interpretation of their results. Additional metrics -based on the S-core- are defined here as well for both individuals and entire network communities.

Moreover, as the graphs that are explored are in the size of millions (of members), the core decompositions (k, fractional, D and S) are used as a structure to build interesting visualizations and management tools for such large graphs.

The core concept, all of its extensions and the relevant structures and metrics, which are defined throughout this work, constitute a framework of tools for efficient and valid evaluation of cohesiveness and collaboration in directed networks and of trust in signed.

1.2.6 *Degeneracy and Clustering*

Graph clustering or community detection constitutes an important task for investigating the internal structure of graphs, with a plethora of applications in several diverse domains. While the main focus of this work was a direct use of degeneracy for evaluating collaborative behavior, the application of degeneracy in graph clustering is also investigated. Traditional tools for graph clustering, such as spectral methods, typically suffer from high time and space complexity. The last part, of the work presented here, is *CoreCluster*, an efficient graph clustering framework based on the concept of graph degeneracy, that could be used along with any known graph clustering algorithm.

The approach capitalizes on processing the graph in a hierarchical manner provided by its core expansion sequence, an ordered partition of the graph into different levels according to the k-core decomposition. Such a partition provides a way to process the graph in an incremental manner that preserves its clustering structure, while making the execution of the chosen clustering algorithm much faster due to the smaller size of the graph's partitions onto which the algorithm operates. It is proven experimentally on a multitude of real and synthetic graphs that this approach accelerates systematically the clustering process by orders of magnitude, especially as the graph's size increases, while the quality of the clustering results is not compromised or even is improving.

1.2.7 *Explored Data*

Three types of graph data are explored with degeneracy: **a.** undirected social structures, **b.** directed social and web structures and **c.** trust networks. A brief description of these follows next.

1.2.7.1 *Undirected Graph Data*

Experiments are mainly dealing with two undirected graph datasets: the *DBLP* bibliographic dataset and the *ArXiv* on High Energy Physics - Theory (*ArXiv.hep-th*). Both networks can be seen as bipartite graphs of author entities being connected with paper entities. A transformation of the bipartite graphs into collaboration networks (between authors) is the final model; upon which an extended experimental evaluation studying in depth the core subgraphs is performed, both integer and fractional, of their edge-weighted co-authorship connections.

1.2.7.2 *Directed Graph Data*

Large scale experiments are also conducted in the following directed graph data:

- A snapshot from the (English) *Wikipedia* in 2004. There the underlying graph is that of the network *Wikipedia* articles create by linking to one another.
- Citation graphs from *DBLP* and *ArXiv*. These graphs are created from the aforementioned data. In this instance thought, citations from one paper to another create directed connections between their authors.
- Scale-free/preferential attachment synthetic graphs, generated by well established procedures/algorithms, are used to compare the behavior under degeneracy of real world datasets vs generative models.

The different type of networks (first two) explored here are used to display the diversity of applications the methods established in this work can be applied to.

1.2.7.3 *Signed Graph Data*

As signed graph data, trust networks were used where one expresses his judgment of trust/distrust towards the actions of another. Experiments were performed on the explicit signed graphs (*Epinions* and *Slashdot*). The term "explicit" is used to signify that the relations from members of these networks were direct actions of trust/distrust through mechanisms offered by each platform (voting). In the opposite side, inferred networks from *Wikipedia* were used as well. These networks were inferred from interactions of users upon the editing platform *Wikipedia* offers (i.e. deleting content, creating new, correcting etc.).

1.2.8 *Dissertation Organization*

The rest of the document is organized in three parts:

1. **Theoretical Models**. This section will begin with presenting all the related work on the subjects that will be addressed through degeneracy. While degeneracy is the main focus, other aspects of graph mining will be mentioned

in this part for a complete picture. Continuing, the footing will be set - through the basic definitions of the k-core structure and decomposition- for the proper presentation of the theoretical aspects for the degeneracy extensions introduced in this work. Moreover, additional metrics and structures will be defined along with an intuitive explanation of their interpretation. Such (metrics and structures), will be the collaboration index, the decomposition forest, the D-core matrix and others with main focus targeted at the evaluation of collaboration.

2. **Data Exploration**. The aforementioned data will be explored in this section with the appropriate degeneracy methodologies. Structures defined in the previous section will be presented on large scale real graphs. Through this study the new concepts will display interesting results thus establishing not only the theoretical but also the practical potential of the proposed models. Moreover, a demonstration will be shown for potential applications of this work.

3. **Degeneracy and Clustering**. This section inspects degeneracy's potential use in graph clustering. While many indications about the implication of degeneracy in graph clustering will be also made at the previous section, in this one degeneracy is directly used as a heuristic to improve the computational cost (in running time) of a highly complex algorithm (specifically spectral clustering). This is done to demonstrate the well diverse use of degeneracy. The experiments contacted here are in both synthetic and real datasets and display very good results.

Part I

THEORETICAL MODELS

2

Community Evaluation

2.1 Introduction

Before presenting the graph theoretic extensions of k-cores, the related work on community evaluation measures must be introduced. This section presents related work on such metrics for individuals and entire (sub)networks and foundational work of degeneracy. Moreover, work on signed networks and theoretic models of trust is also presented.

2.2 Related Work

Related work on community detection is reserved for the last part of this document. For reference, a thorough review on community detection in graphs is offered by Fortunato in [38]. In that work techniques, methods, and datasets are presented for detecting communities in sociology, biology and computer science, disciplines where systems are typically represented by graphs. Most existing relevant methods are presented, with a special focus on statistical physics, including discussion of crucial issues like the significance of clustering and how methods should be tested and compared against each other.

2.2.1 *Graph Theoretic Metrics*

Studying the general behavior and properties of real graphs, both edge-weighted and unweighted, is the subject of [68] where a pattern on the behavior of connected components over time is observed and, upon that, a generative model is build.

In recent literature, various metrics are proposed relevant to the graph structure of a social network. Such are “Betweenness” [87], “Centrality" [75], and “Clustering coefficient” [88] (a measure of the likelihood that two associates of a node are associates themselves).

Clustering coefficient can been seen both as a metric to evaluate single nodes and the entire network:

- The clustering coefficient of a vertex (that represents a network node) is the ratio of links the vertex has with other vertices to the total number of links that could exist between between them. For simplicity, this is referred as "local clustering coefficient".
- The clustering coefficient of an entire network is the average of the local one over all vertices.

A higher clustering coefficient indicates a greater "cliquishness", i.e. cohesion degree or density. Centrality is a more general term (which can include betweenness):

- Degree centrality is a more accurate term when referring to centrality as an evaluation metric. This is essentially the degree of a vertex.
- Closeness centrality is a notion that is connected to the "farness"/"closeness" of a node to other nodes. The measurement of those two concept is a function over the sum of distances between a node in the network and the rest (of the nodes). This function has many definitions from a simplistic one (only the sum) to more complex for specific applications [26]
- Betweenness centrality (or just betweenness) measures the number of times a node of a network is on the shortest path between two other ones [33].
- Of special interest here is the eigenvector centrality – a measure of the importance of a node in a network. It assigns relative scores to all nodes in the network based on the principle that connections to nodes having a high score contribute more to the score of the node in question. PageRank can be considered a form of eigenvector centrality [74].

Other interesting measures include "path length" (i.e. distances between pairs of nodes in the network), and "Structural cohesion" - the minimum number of members who, if removed from a group, would disconnect the group [71].

In [52] an alternative "core notion" is considered for the case of directed graphs where a core is seen as a complete bipartite graph where all edges are directed from the one part to the other. In [52], such cores are detected and are then fed to a generalized HITS algorithm used to expand the communities within them.

Within the work presented here, a direct comparison of degeneracy and reciprocity will be made for signed networks. The basic definition of reciprocity is a local property based on mutuality in pairs of nodes in directed graphs [69, 82, 87]:

$$r \equiv \frac{L^{\leftrightarrow}}{L} \tag{2.1}$$

where $L^{\leftrightarrow}$ is the number of links pointing in both directions and L is the total number of links. Thus, the highest value of r is 1, when the network is fully bi-directional, and the lowest is 0 when the network is completely unidirectional.

Reciprocity is used to examine directed networks of various kinds [40, 69, 82], an extension for weighted networks is in the recent work of [1]. In [40], reciprocity is extended in order to take into account the density of the network.

2.2.1.1 *Signed Networks*

In the area of signed graphs, a machine learning-based approach for inferring negative or positive links in Epinions was published in [60], whose techniques rely on an existing signed network complemented by user interactions. In [64], a signed network over the editors of the *Wikipedia*, denoted *Wiki-Signed*, is inferred exclusively from interactions; it is evaluated, at both local and global level, in relation with social theories and existing signed networks on the Web. We rely in this paper on networks built as in [64]. Another approach for detecting positive and negative interactions in *Wikipedia* was presented in [18], showing the emergence of polarization in *Wikipedia* articles.

Several papers have also studied the prediction of links and link signs, when only the signed network is known, a problem also known as *trust propagation*. The first rigorous treatment of this problem is given in [43], where the authors define four atomic operators to predict link signs (direct propagation, co-citation, transpose trust and trust coupling). This approach was extended in [56, 57], where trust propagation was studied through the lens of social theories such as balance and status, and a prediction model based on the number of triangles involving each candidate link was proposed.

For undirected signed graphs, the theory of Social Balance [6] is a model for the dynamics of friendship and enmity through time. The weakness of this model is that it assumes that all relationships are reciprocal. A more advanced model called Status Model is introduced in [43] and elaborated in [56]. The advantage of this model is that it takes into account the direction of the relationships but it is built upon the structure of triangles within the graph. The main point of this model is that a directed signed edge signifies someone of either higher or lower status and thus predicts that the flipping of a direction should flip the sign as well. But this would not account for the relationships of trust (that we attempt to study). In principle (and shown by the experimental results), it is counter intuitive to assume that showing trust or distrust to others would lead to the opposite assumptions of others to us.

Recently, the problem of *ranking* vertices in signed networks has been studied in [27, 70]. This problem is challenging, since power iteration methods used for regular networks do not apply to networks in which links can also have negative

scores. The PageTrust algorithm, extending the ubiquitous PageRank algorithm to handle also negative links, has been proposed in [27]. An algorithm that uses a signed network to derive two scores for each vertex, called prestige and bias, has been presented in [70].

2.2.1.2 *Cores*

The k-cores are fundamental structures in graph theory and their study dates back to the 60's [36, 66, 81, 85]. A k-*core* of a graph G is the maximum subgraph H of G where each vertex in H has at least k neighbors in H. The *degeneracy* of a graph is defined as the biggest k for which a graph contains a non-empty k-core [59]. The same notion has appeared with several names such as *width* [67], *linkage* [39, 49], or *coloring number* [28] and has been proven to be equal to the smallest k for which we can find a linear ordering of the vertices of the graph such that for each vertex v, the number of its neighbors that appear before v in the ordering is at most k (see [39, 59, 66]).

The existence of k-cores of large size in sufficiently dense graphs has been theoretically studied by [76] for random graphs generated by the Erdős-Rényi model [35]. As shown in [76], a k-core whose size is proportional to the size of G (i.e. a "giant" k-core) "suddenly" appears in a random graph with n vertices and m edges when m reaches a threshold $c_k \cdot n$, for some constant c_k that depends exclusively on k. Also, it was proved in [15, 61] that, in the Erdős-Rényi model, almost all k-cores are k-connected (see [46] for more recent results on this topic).

An efficient algorithm for the computation of the k-core of a graph was given in [13] and its running time is proportional to the number of edges of the input graph. Actually, the algorithm in [13] can compute the *core decomposition* of a graph consisting of the sequence of all the non-empty i-cells of G where each i-cell is defined as the vertices contained in the i-core but not in the $(i+1)$-core. Core decompositions provides useful information on the way subgraphs of a graph are clustered according to their degrees and has been used extensively in several topics such as the study of internet topology [3, 20], large scale network visualization [2, 3, 11], networks of protein interaction [8, 90], and complex network modeling and organization [14, 30]. A more general notion of k-cores was introduced in [12] where, instead of vertex degrees, more general functions where considered.

In [21], greedy approximation algorithms are proposed for finding the dense components of a graph. Both undirected and directed graphs are examined. In the case of directed graphs the vertices are divided into hubs (S) and authorities (T). Then, based on a value of $|S|/|T|$, a greedy algorithm removes the vertex of minimum degree from either S or T until both sets are empty. Also, in [84], the

subject of finding dense subgraphs, based on query nodes, is studied, where the issue is to find a community that contains certain given nodes.

2.2.2 *Citation Graphs*

The experiments, that will be presented in the next part of this document, focus partially on applying our evaluation techniques on citations graphs (DBLP, ArXiv). Recent work on citation graphs can be found in [4] where a study is carried out on the citation graph of Computer Science Literature and [47]. In [4], an attempt is made to extract a descriptive summary of the graph through a study of fundamental and well established properties (degree distribution, giant component size etc.). In contrast, our work focuses on novel techniques for evaluating community graphs and expands on a wider scope of study. In [47] the focus is on community detection and the evolution through time. The community detection is performed on the authors through the papers they have co-cited and the evaluation of the citation graph is based on the detected clusters.

2.3 THEORY ON DEGENERACY

2.3.1 *Preliminaries*

As degeneracy for undirected graphs will only be explored on bibliographic datasets, some of the following definitions refer specifically on network structures formed by authors collaborating (or not) on the work of a published paper.

For the basic definition of degeneracy, graphs are considered undirected and simple (i.e. they do not have multiple edges or loops). The vertex and the edge set of a graph are denoted by $V(G)$ and $E(G)$ respectively. The cardinality of $V(G)$ will be referred as the *size* of G. Moreover, *edge-weighted* graphs (or, simply *weighted* graphs) are denoted by pairs $(G, \mathbf{w})$ where $\mathbf{w}$ is a weighting function assigning rational numbers to the edges of G.

A graph H is a subgraph of a graph G if H occurs from G after removing vertices or edges (the removal of an edge implies the removal of all edges that are incident to it). A graph is *connected* if for every pair of its vertices there is a path connecting them. A *connected component* of a graph is a maximal connected subgraph of it. Given a graph G, the size of the largest connected component of G is denoted by $g(G)$ and it is called *giant* component.

Definition 1. *Given a vertex* $v \in V(G)$*, the* degree *of* v *in* G *is the number of edges that are incident to it. Also,* $\delta(G)$ *denotes the minimum degree of a vertex in* G*. The* degeneracy *of a graph* G *is defined as follows:*

$$\delta^*(G) = \max\{\delta(H) \mid H \text{ is a non-empty subgraph of } G\}. \tag{2.2}$$

Definition 2. *Given a graph* G *and a non-negative integer* k*, the* k-core *of* G *is defined as the maximum size subgraph* H *of* G *where* $\delta(H) \geqslant k$*. It is easy to see that such a subgraph is unique. Given a* k*-core,* k *is referred as its* core index *or simply* index*.*

Assume that for $i = 0, \ldots, \delta^*$, G_i is the i-core of G. Then the sequence

$$V(G_0), V(G_1), \ldots, V(G_{\delta^*(G)})$$

is called *core sequence* of G.
Observe that $V(G_i) \subseteq V(G_{i+1})$ for $i \in \{0, \ldots, \delta^*(G) - 1\}$.

Additionally, the sequence:

$$V(G_1) - V(G_1), \ldots, V(G_{\delta^*(G)}) - V(G_{\delta^*-1})$$

is called *cell sequence* of G and its elements form a partition of $V(G)$.

Definition 3. *For every graph* G *where* $\delta^*(G) = k$*, its core expansion sequence is defined as the sequence of vertex sets* $\{V_k, V_{k-1}, \ldots, V_0\}$ *that is recursively defined as follows:*

$$\begin{aligned} &V_k = V(\mathbf{core}_k(G)) \text{ and} \\ &V_i = V(\mathbf{core}_i(G)) \setminus V_{i+1}, \quad i = k-1, \ldots, 0. \end{aligned} \tag{2.3}$$

The sets of a core expansion sequence are also refered as layers *agreeing that the set* V_i *is its* i*-th layer.*

2.3.1.1 *The* Trim *Procedure*

Notice that, for each $i \leqslant j$, the j-core of a graph is a subgraph of its i-core. Furthermore, the degeneracy of a graph is the maximum k for which G contains a non-empty k-core. Given a graph G where $\delta^*(G) = d$ and an integer i where $0 \leqslant i \leqslant d$, the i-core of G is denoted by G_i and the *core sequence* of G is defined as $\mathcal{G}(G) = G_0, G_1, \ldots, G_d$, where $G_0 = G$ and G_d is the *densest* core of G. For every $i \geqslant 0$, the graph G_{i+1} can be computed by the following simple procedure.

Procedure *Trim*(G, k)
Input: An undirected graph G and a positive integer k
Output: the $(k+1)$-core of G
1. let $F := G$
2. **while** there is a node x in F such that $\mathbf{deg}_F(x) \leqslant k$
3. **delete** node x from F
4. **return** F

The *Trim*(G, k) procedure runs in $O(kn)$ steps, thus computations are feasible even in large scale graphs [13]. Applying successively *Trim*(G, i), for $i = 0, \ldots, \delta^*(G) - 1$,

gives a fast way to compute the core sequence of G. In fact an optimal implementation of the above pruning procedure that is able to produce the core sequence of a graph in $O(\delta^*(G) \cdot n)$ steps and been given in [13]. The procedure in [13] works for much more general variants of the core notion, including the fractional core notion that will be defined later in this section.

Definition 4. *The* core index *of a vertex v of G is the maximum k for which v belongs in the k-core of G.*

Notice that one may also define the *core index of a set S* of vertices in G as the maximum k for which all vertices of S belong in the k-core of G [12]. It is easy to see that this number is the minimum core index of all the vertices in S.

2.3.1.2 *Core Decomposition Forest*

In this section, the *Core Decomposition Forest* is defined for the case of the undirected cores (weighted or not). For the directed case, an almost identical definition will be in the sections that follow.

Definition 5. *Let $\mathcal{G} = G_0, G_1, \ldots, G_d$ be a sequence of graphs such that for each i, j where $i \leqslant j$, G_i is a subgraph of G_j (such a sequence is called* monotone*). The* Decomposition Forest *of a monotone graph sequence $\mathcal{G}$ is the graph $\mathbf{DF}(\mathcal{G})$ that is defined as follows. For each $i = 0, \ldots, d$, the connected components of G_i are denoted by $G_i^1, \ldots, G_i^{m_i}$ and each such connected component is a vertex of $\mathbf{DF}(\mathcal{G})$ (the isomorphic graphs are treated here as different graphs). The pair $(G_i^j, G_{i'}^{j'})$ is a directed edge of $\mathbf{DF}(\mathcal{G})$ if $j' = j + 1$ and G_i^j contains $G_{i'}^{j'}$ as a subgraph.*

It is easy to verify that the directed graph defined above is a rooted forest. In fact, each of its components is a rooted tree where all its edges are directed away from the root and each root is a connected component of G_0. Given that the core sequence of G is monotone, the Core Decomposition Forest of a graph (edge-weighted or not) is defined as the decomposition forest corresponding to its core sequence. The notion of the core decomposition forest appeared for the first time in [44] under the name *hierarchical degree core tree* and was used in order to visualize the connected components of several real-word graphs including the graph extracted by the common-author relation of the papers of the DBLP citation graph. As the graphs that are extracted from DBLP and ARXIV are expressing relations between authors, the core decomposition forests that are described and presented in the second part are of radically different nature than the one extracted in [44].

2.3.2 *Cores for bipartite graphs*

A great part of the datasets (the bibliographic ones e.g. *DBLP*), that are studied here, are represented by bipartite graphs where edges denote relations between

papers and authors. Such a graph is denoted by $G = (A, P, E)$ where A is the set of authors, P is the set of papers, and E is a set of edges. Each edge $\{x, y\}$ (where $x \in A$ and $y \in P$) expresses the fact that x is one of the authors of paper y. As the aim is to evaluate the collaboration between authors, a restriction is applied to the study of papers that are written by at least two authors, i.e., there is an assumption that all the vertices in P have degree of at least two.

Definition 6. *The* co-authorship graph *corresponding to* G *is defined as follows:*

$$H_G = (A, \{\{x, x'\} \mid \exists y \in P : \{x, y\}, \{x', y\} \in E), \tag{2.4}$$

i.e., two authors are adjacent if they appear as co-authors in at least one paper. Notice that the above definition of H_G is radically different from the one used in [44], where they study graphs whose vertices correspond to authors and edges indicate joint publications between two authors. In fact, the construction in [44] can be seen as being the dual of the one used here for creating H_G in the sense of vertex-edge duality of hyper-graphs.

For each dataset (represented by a bipartite graph G), the $\delta^*(H_G)$ is computed along with the core index of each vertex/set of vertices in H_G in order to evaluate the collaboration behavior in the bipartite graph G and the dataset that it represents. The idea of this criterion is to locate communities of authors with a high collaboration between them in the sense that the demand is not just so that they have authored many papers but also that they have all authored them with authors in the same community.

However, this is not an entirely satisfactory evaluation, since the number of authors on a paper has no impact in this measure. For this reason, below is introduced a more refined way to define cores based on the notion of a fractional core.

2.3.3 *Fractional* k*-cores for edge-weighted graphs*

Continuing with the papers-authors paradigm, let $G = (A, P, E)$ be a bipartite graph where all vertices in P have minimum degree of 2. Given an author vertex $x \in A$, the neighborhood $N_G(x)$ of x is defined as the set containing each paper $y \in P$ for which $\{x, y\} \in E$, i.e., $N_G(x)$ is the set of papers co-authored by x. The neighborhood $N_G(y)$ of a paper $y \in P$ can be defined in a symmetrical manner, i.e., the set of the authors of paper y. Also, given an author x, the set of all edges that are incident to x in G is denoted by $E_G(x)$. In what follows, $\mathbb{Q}^+$ denotes the set of all non-negative rational numbers.

Definition 7. *Given a bipartite graph* $G = (A, P, E)$*, the definition of the* edge-weighted co-authorship graph*, denoted by* $(H_G, \mathbf{w})$*, by taking* H_G*, as defined in* (2.4)*, and setting*

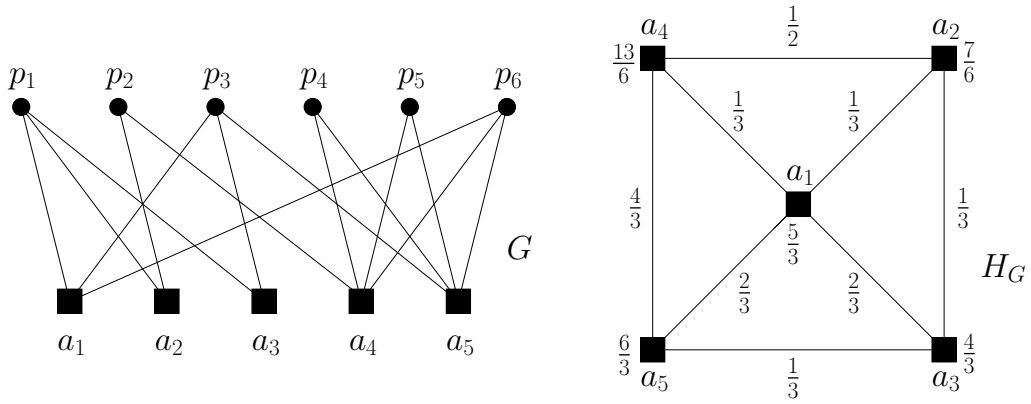


Figure 2.1: An example of a bipartite graph G and its edge-weighted co-authorship graph, $(H_G, \mathbf{w})$.

up a rational weight function $\mathbf{w} : E \to \mathbb{Q}^+$ *on the edges of* H_G *is as follows: For every edge* $e = \{x, x'\}$ *we set*

$$\mathbf{w}(e) = \sum_{y \in N_G(x) \cap N_G(x')} \frac{1}{N_G(y)}. \tag{2.5}$$

Notice that, $\sum_{e \in H_G} \mathbf{w}(e) = |V(P)|$, i.e. the sum of all the weights on the edges is the size of the graph, i.e., the number of its vertices. For example, in Figure 2.1, in order to compute the weight of the edge $e = \{a_1, a_3\}$, one should observe that the authors a_1 and a_3 are co-authors of the papers p_1 and p_3. As p_1 and p_3 have 3 authors each, they contribute $1/3$ to the weight of e, that is $\mathbf{w}(e) = 2/3$. This weighting of e expresses the fact that the collective effort of author a_1 to the papers he/she co-authored with p_3 is of $2/3$ papers, and vice versa.

As agreed before, the notation $(G, \mathbf{w})$ for the graph G is used to denote that it is edge-weighted by $\mathbf{w}$.

Definition 8. *Given an edge-weighted graph* $(G, \mathbf{w})$ *and a vertex* $x \in V(G)$*, the* fractional-degree *of* x *in (G,w) is defined as*

$$\mathbf{deg}_{G,\mathbf{w}}(x) = \sum_{e \in E_G(x)} \mathbf{w}(e). \tag{2.6}$$

In the co-authorship context, the degree $\mathbf{deg}_{G,\mathbf{w}}(x)$ *of an author* x *is the collective effort of author* x *for all the papers she/he wrote. For instance, in Figure 2.1, the degree author* a_4 *is the sum of all the weights of the edges that are incident to it, i.e.,* $1/3 + 2/3 + 4/3 = 13/6$.

A graph $(H, \mathbf{w}_H)$ *is* an edge-weighted subgraph *of* $(G, \mathbf{w})$ *if* H *is a subgraph of* G *and* $\mathbf{w}_H$ *is the restriction of* $\mathbf{w}$ *on* $E(H)$. *Given any such subgraph* $(H, \mathbf{w}_H)$ *of* $(G, \mathbf{w})$, *we define*

$$\delta(H, \mathbf{w}_H) = \min\{\mathbf{deg}_{H,\mathbf{w}_H}(x) \mid x \in V(H)\}. \tag{2.7}$$

For example, if $(G, \mathbf{w})$ is the edge-weighted graph in Figure 2.1, then $\delta(G, \mathbf{w}) = \mathbf{deg}_{G,\mathbf{w}}(a_2) = 7/6$. If H is the subgraph of G containing all edges that are incident to the vertices a_1, a_2, and a_4, then $\delta(H, \mathbf{w}_H) = \mathbf{deg}_{H,\mathbf{w}_H}(a_1) = 2/3$.

Definition 9. *Let* $(G, \mathbf{w})$ *be an edge-weighted graph. The* fractional-degeneracy *of* $(G, \mathbf{w})$ *is defined as follows:*

$$\begin{aligned}\delta^*(G, \mathbf{w}) = \max\{\delta(H, \mathbf{w}_H) \mid (H, \mathbf{w}_H) \text{ is a non-empty} \\ \text{edge-weighted subgraph of } (G, \mathbf{w})\}.\end{aligned} \tag{2.8}$$

Let $k \in \mathbb{Q}^+$. Then the k-core of $(G, \mathbf{w})$ is the maximun-size edge-weighted subgraph $(H, \mathbf{w}_H)$ of $(G, \mathbf{w})$ where $\delta(H, \mathbf{w}_H) \geqslant k$.

The *Trim* procedure can also compute k-cores where k is a rational number. The only modification, in the *Trim* algorithm presented in subsection 2.3.1.1, is that $\mathbf{deg}_F(x) \leqslant k$ should be replaced by $\mathbf{deg}_{F,\mathbf{w}_F}(x) \leqslant k$, i.e., the *Trim* procedure for fractionally weighted graphs would check the fractional degree of x in the edge-weighted graph $(F, \mathbf{w}_F)$, where $\mathbf{w}_F$ is the restriction of $\mathbf{w}$ to the edges of F.

In fact, the definition of the fractional analogue of the core sequence requires more attention, as it now should be indexed by rational numbers. For this, consider the infinite sequence $\mathcal{G} = G_{h_0}, G_{h_1}, \ldots$, recursively defined as follows:

$$G_{h_0} = G, h_0 = 0, \text{ and for } i > 0, G_{h_i} = \mathtt{Trim}(G_{h_{i-1}}, h_{i-1}) \text{ where } h_i = \delta(G_i, \mathbf{w}_{G_i}).$$

Then, the *fractional core sequence* of an edge-weighted graph $(G, \mathbf{w})$ is the prefix of $\mathcal{G}$ that contains all non-empty graphs of $\mathcal{G}$ and is denoted by $\mathcal{G}(G, \mathbf{w})$. The *size* of a *fractional core sequence* is the number of its terms minus one. Notice that the size l of the fractional core sequence of an edge-weighted graph $(G, \mathbf{w})$ can never exceed the size of G. Finally the sequence $h_1, \ldots, h_l$ is called *fractional index sequence* of $(G, \mathbf{w})$.

The *fractional core index* of a vertex of an edge-weighted graph $(G, \mathbf{w})$ is the maximum rational number k for which v belongs in the k-core of G. As in the unweighted case, the *fractional core index* definition can be naturally extended to sets instead of vertices. Again the fractional core index of a set of vertices is the minimum fractional core index of its members.

As an example of the above definitions, the edge-weighted graph $(H_G, \mathbf{w})$ depicted in Figure 2.1, has fractional degeneracy $\frac{7}{6}$, i.e. $\delta(H_G, \mathbf{w}) = \delta^*(H_G, \mathbf{w})$. Indeed if one applies $\mathit{Trim}(H_G, \frac{7}{3})$ then the first vertex to be removed is a_2. This

removal drops the fractional degrees of a_1, a_3, and a_4 below $\frac{7}{3}$. Therefore, they are also removed and, for the same reason, the remaining vertex a_5 is removed as well. Therefore, G_1 is the empty graph, the fractional core sequence contains only graph $G_0 = G$, and the length of the fractional index sequence of $(H_G, \mathbf{w})$ is 0. A mention should be made that a less trivial example would be too complicated to present in a figure and even more complicated to be processed by the reader.

At this point it is important to stress that, as a graph-theoretic notion, fractional cores are defined on bipartite graphs, encoding relations between two sets representing different entities (in this case, papers and authors). Equivalently, fractional cores can be defined in hypergraphs by considering the fractional cores of their (bipartite) incident graphs. In this case, the hypergraph corresponding to the graph G would contain the authors as vertices and the papers as hyperedges. In the work presented here it was chosen to avoid hypergraph notation and, for simplicity, the definition that uses bipartite graphs was adopted.

2.4 D-CORES

Let $D = (V, E)$ be a digraph that is a set V of vertices and a set E of directed edges between them. Each edge $e \in E$ can be seen as a pair $e = (v, u)$ where v is called the *tail* of e while u is the *head* of e. The set of vertices of a digraph D is denoted by $V(D)$. Given a vertex $x \in V$, its *in-degree*, which is denoted by $\mathbf{deg}^{in}_D(x)$, is the number of *in-links* of x, i.e. the edges in D with x as a head. Similarly, the *out-degree* of x, denoted by $\mathbf{deg}^{out}_D(x)$, is the number of *out-links* of x, i.e. edges in D with x as a tail. The *min-in-degree* and the *min-out-degree* of a digraph D are defined as

$$\begin{aligned} \delta^{in}(D) &= \min\{x \mid \mathbf{deg}^{in}_D(x) \mid x \in V(D)\} \text{ and} \\ \delta^{out}(D) &= \min\{x \mid \mathbf{deg}^{out}_D(x) \mid x \in V(D)\} \end{aligned} \tag{2.9}$$

respectively. Given two positive integers k, l and a digraph $D = (V, E)$, a (k, l)-D-*core* of D is a maximal size sub-digraph F of D where $\delta^{out}(F) \geqslant k$ and $\delta^{in}(F) \geqslant l$; if no such digraph exists then the (k, l)-D-core of D is the empty digraph. It is easy to see that when such a sub-digraph exists, it is unique.

Given a digraph D, the (k, l)-D-core of D is denoted by $\mathbf{DC}_{k,l}(D)$. Additionally, $\mathbf{dc}_{k,l}(D)$ denotes the size of $\mathbf{DC}_{k,l}(D)$, i.e. the number of its vertices. As D will always be the network under study, the simpler notations $\mathbf{DC}_{k,l}$ and $\mathbf{dc}_{k,l}$ will be used instead.

The intuition behind (k, l)-D-cores is to find a sub-digraph where all nodes have enough out-links and in-links to the rest of it. Clearly, it is not enough for a node to have big in-degree and/or out-degree in order to be a member of such a core. What counts, on the top of this, is that the node forms part of a community where

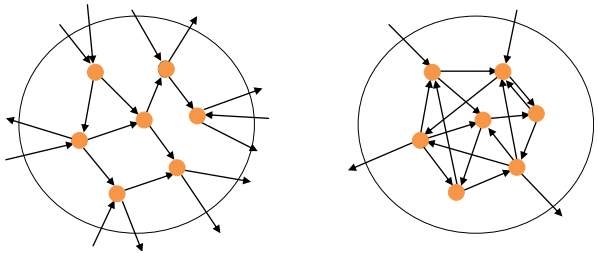


Figure 2.2: Two portions of a digraph. The one in the left does not contain any non-trivial (k, l)-core and the one in the right is a $(2, 2)$-core.

each of its members satisfy the same in-degree and/or out-degree requirements with respect to all the other community members (see Figure 2.2 for an example). This indicates that nodes in a D-core exhibit a strong collaboration behavior among them.

The detection of $\mathbf{DC}_{k,l}$ is computationally easy and can be done by altering the original *Trim* procedure (subsection 2.3.1.1) thus having the following:

Procedure $Trim_{k,l}(D)$
Input: A digraph D and positive integers k, l
Output: $\mathbf{DC}_{k,l}(D)$
1. let $F \leftarrow D$.
2. while *there is a node* x *in* F *such that* $\mathbf{deg}^{out}_F(x) < k$ *or* $\mathbf{deg}^{in}_F(x) < l$,
3. **delete** *node* x *from* F.
4. return F.

Let $L = (v_1, \ldots, v_m)$ be a layout of the vertices of D. For every $i = 1, \ldots, n$, D_i denotes the digraph induced by the vertices in $\{v_1, \ldots, v_i\}$. The layout L is (k, l)-*eliminable* if for every $i \in \{0, \ldots, n\}$, either $\mathbf{deg}^{out}_{D_i}(v_i) < k$ or $\mathbf{deg}^{in}_{D_i}(v_i) < l$.

The following Lemma on (k, l)-D-cores generalizes the classic min-max result of [67] (see also [39, 49]).

Lemma 1. *Given a digraph* D *and two positive integers* k *and* l, *the* (k, l)-D-*core is empty if and only if there exists a* (k, l)-*eliminable layout of* $V(D)$.

Lemma 1 essentially indicates that the elimination procedure of the algorithm $Trim_{k,l}(D)$ works correctly and (optimally) runs in $O(m)$ steps, where $m = |E(G)|$. The proof is easy and follows the arguments of [39] for the undirected case (see also [12]).

For an optimal implementation of the $Trim_{k,l}(D)$ procedure, see the general algorithm of [12] that is based on the same ideas for the undirected case. In the implementation of this procedure, $\mathbf{DC}_{k,l}(D)$ is incrementally computed for all pairs of k and l.

2.4.1 *Degeneracy of digraphs*

The degeneracy of a directed digraph differs radically from its undirected counterpart. Actually, it has a two-dimensional nature since different choices of the lower bounds to the number of incoming/outgoing edges result to different D-cores.

Definition 10. *The* degeneracy *of a digraph* D *is defined as follows.*

$$\delta^*(D) = \frac{1}{2}\max\{\delta^{out}(H) + \delta^{in}(H) \mid H \subseteq D\}. \tag{2.10}$$

The intuition behind the definition of $\delta^*(D)$ is to return the maximum r (for some pair k, l where $k + l \geqslant 2r$) such that D contains a non-empty (k, l)-D-core (δ^* takes semi-integer values). Also the value of $\delta^*(D)$ may correspond to multiple (k, l)-D-cores for different choices of k and l (those where $k + l = 2 \cdot \delta^*(D)$).

Notice that if each edge of a graph is replaced by two opposite direction edges, the degeneracy of the resulting digraph is equal to the degeneracy of G. Thus δ^* is indeed a valid generalization of undirected degeneracy to directed graphs.

It is important to stress that δ^* is the first density parameter on digraphs that takes into account Hub/Authority trade offs as it differs radically (and is not comparable) with previous digraph density measures such as the ones defined in [21] and [52]. A powerful extension of the classic notion of a k-core was given in [12] where the k-core is defined as a set of vertices where some general vertex property function is bounded. While the results in [12] can also provide a natural concept of k-core for directed graphs, they are not able to capture the "two-dimensional" nature of our (k, l)-core concept where degree bounds are applied *simultaneously* on both the in-degrees an the out-degrees.

Definition 11. *Let* τ *be a real number in the interval* $[0, \pi/2]$ *representing an angle. The* τ-degeneracy *of a digraph* D *is defined as follows:*

$$\begin{aligned}\delta^*_\tau(D) = \max\{\frac{\lceil k \rceil + \lceil l \rceil}{2} \mid\ & G \textit{ contains a non-empty } (k,l)\text{-D-core where} \\ & k = r \cdot \cos(\tau) \textit{ and } l = r \cdot \sin(\tau) \textit{ for some } r \textit{ where } r^2 = l^2 + k^2\}\end{aligned} \tag{2.11}$$

In the above definition one may see each pair (k, l) as a point of a Cartesian system of coordinates, corresponding to the D-core $\mathbf{DC}_{k,l}(D)$. To compute $\delta^*_\tau(D)$, we essentially follow the τ-slope segment starting from $(0, 0)$ until $\mathbf{DC}_{k,l}(D)$ becomes empty along this line. The last such non-empty D-core is the one determining the degeneracy of D with respect to the angle τ. The value of τ reflects the Hub/Authority trade-off in the considered D-cores and we refer to it as H/A-*angle*.

Again it is easy to observe that $\delta^*_{\pi/4}$ deteriorates to classic degeneracy when we replace each edge of an undirected graph by two (opposite) directed edges.

Observe that δ_τ can also provide an another definition of δ^*, equivalent to the one in (2.10), as $\delta^*(D) = \max\{\delta^*_\tau(D) \mid \tau \in [0, \pi/2]\}$.

2.4.2 *D-core matrix*

The objective of this work is to define a series of digraph-based metrics, based on directed degeneracy, in order to evaluate the dense collaboration of nodes in networks whose links have directional nature. The whole network is represented by a digraph D and there is a unique $\mathbf{DC}_{k,l}$ for each $k, l \geqslant 0$.

Definition 12. *The sizes* $\mathbf{dc}_{k,l}$*, (for* $k, l \geqslant 0$*) define an (infinite) matrix* $A_D = (\mathbf{dc}_{k,l})_{k,l\in\mathbb{N}}$ *that is called* D-core matrix of D. *The notion of* $A_D(k, l)$ *is the two-dimensional digraph analogue of the notion of core sequence defined in Subsection 2.3.1 for the undirected case.*

For each $k, l \geqslant 0$ the following is defined:

$$\begin{aligned} \mathbf{DCL}^{out}_{k,l} &= V(\mathbf{DC}_{k,l}) - V(\mathbf{DC}_{k+1,l}) \text{ and} \\ \mathbf{DCL}^{in}_{k,l} &= V(\mathbf{DC}_{k,l}) - V(\mathbf{DC}_{1,l+1}). \end{aligned} \tag{2.12}$$

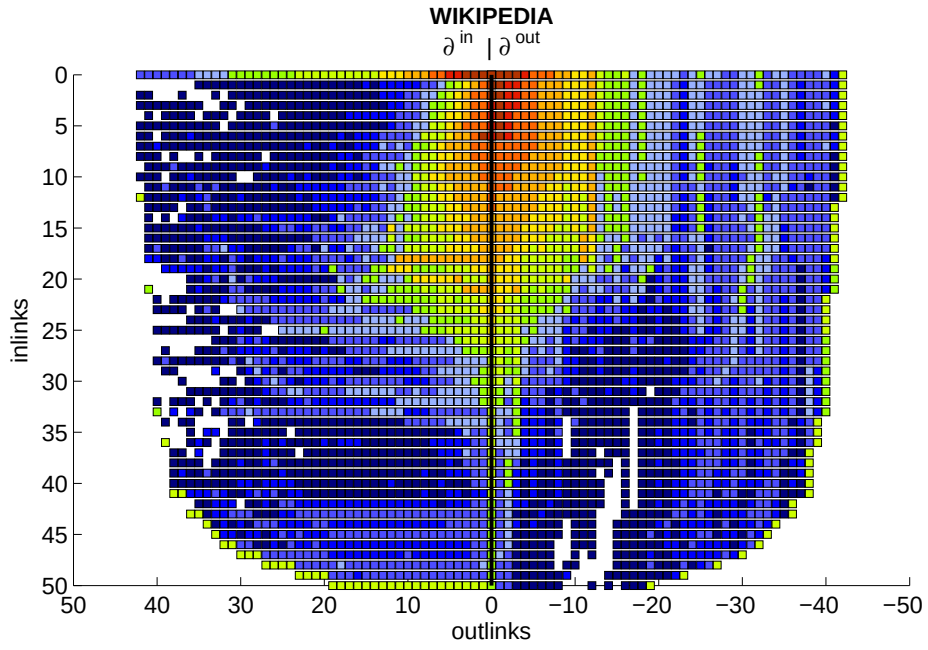
Also, set :

$$\begin{aligned} \mathbf{dcl}^{out}_{k,l} &= |\mathbf{DCL}^{out}_{k,l}| \text{ and} \\ \mathbf{dcl}^{in}_{k,l} &= |\mathbf{DCL}^{in}_{k,l}|. \end{aligned} \tag{2.13}$$

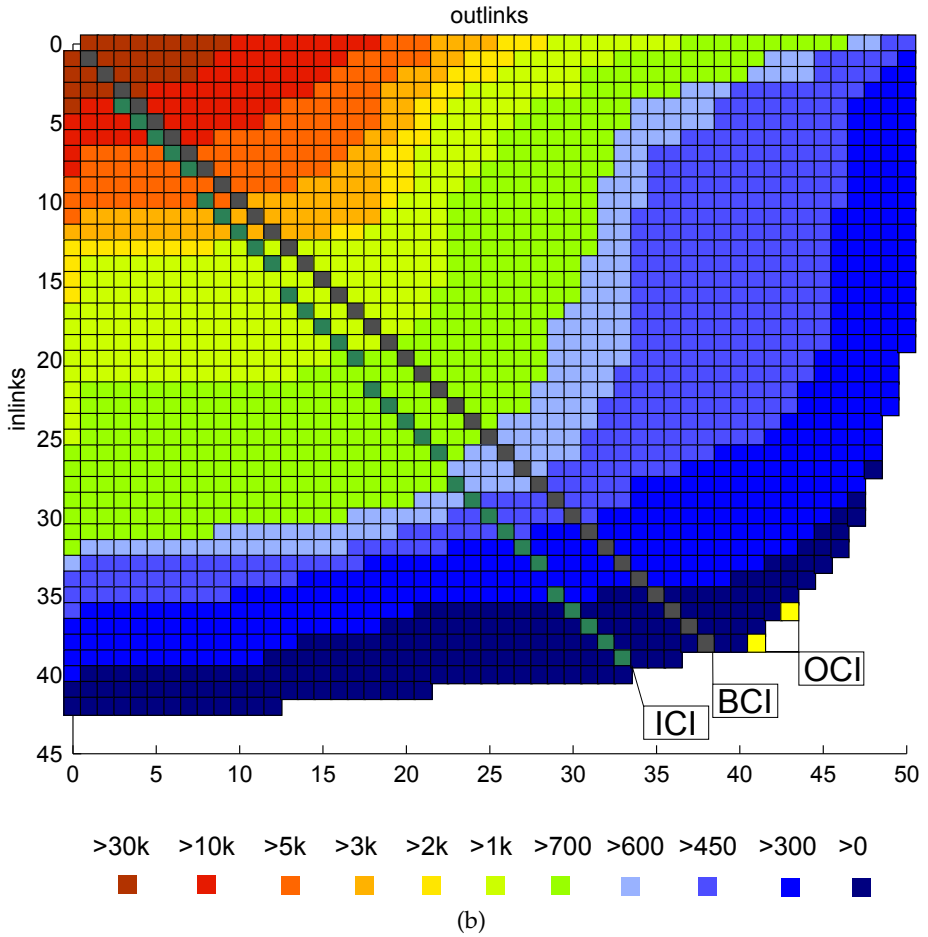
In other words, the values of $\mathbf{DCL}^{out}_{k,l}$ and $\mathbf{DCL}^{in}_{k,l}$ represent the "differential" of the of the matrix A_D taken in both horizontal and vertical direction. For this reason, the matrices $\partial^{out}A_D = (\mathbf{dcl}^{out}_{k,l})_{k,l\in\mathbb{N}}$ and $\partial^{in}A_D = (\mathbf{dcl}^{in}_{k,l})_{k,l\in\mathbb{N}}$ are defined. To visualize them, one may see the values of A_D as being assigned to the squares of an infinite two-dimensional grid centered to the esquire $(0, 0)$ and the values of $\partial^{out}A_D$ and $\partial^{in}A_D$ as assigned to the vertical and horizontal edges of this grid.

2.4.3 *An Example*

As the structures these definitions describe might become a little hard to comprehend, there is a need for an example to demonstrate them. Here, results from *Wikipedia* are used as such an example. The full details on *Wikipedia* will be presented along with the other datasets on the "**Data Exploration**" part. Additionally, these examples will be used for reference to any of the following definitions where needed. Figure 2.3a displays the differentials $\partial^{out}A_D$ and $\partial^{in}A_D$ for the digraph formed by *Wikipedia*. Figure 2.3b displays the D-core of the same dataset along with some metrics and concepts defined in the following subsections.



(a)



(b)

Figure 2.3: a.The differentials $\partial^{\text{out}}A_{\text{D}}$ and $\partial^{\text{in}}A_{\text{D}}$ for the digraph formed by Wikipedia.White squares indicate a value of zero.
b. The D-core matrix of the Wikipedia 2004 digraph

In the example of Figure 2.3 the matrix A_D and its differentials $\partial^{out}A_D$ and $\partial^{in}A_D$ for the digraph formed by the *Wikipedia* (2004, English edition). The nodes correspond to *Wikipedia* pages and each directed edge $e = (x, y)$ is a link from page x to page y. Cell (k, l) in the matrix A_D stores the size $(\mathbf{dc}_{k,l})_{k,l\in\mathbb{N}}$ of the respective d-core $\mathbf{DC}_{k,l}$. As agreed before, the coordinates (k, l) ar5e seen as squares of an infinite two-dimensional grid, the values $\mathbf{dcl}^{out}_{k,l}$ and $\mathbf{dcl}^{in}_{k,l}$ are assigned to its edges.

The result for the case of A_D is depicted in Figure 2.3b. As there is no *Wikipedia* entry with more than 51 out-links or more than 43 in-links this matrix is restricted to its lower 51×43 portion. For each digraph D examined, this matrix is called D-*core matrix* of D; its cells are visualized as squares of an infinite two-dimensional grid Γ_D and the size of its (k, l)-cores is depicted by coloring the corresponding squares with different colors. According to Figure 2.3b, the value of $\delta^*(D_{Wiki})$ for the *Wikipedia* digraph D_{Wiki} is obtained in cell $(38, 41)$ and is equal to $\frac{38+41}{2} = 39.5$. In other worlds, 39.5 is the half of the Manhattan distance between a cell of the D-core matrix of D_{Wiki} and the cell $(0, 0)$; in our case this cell is $(38, 41)$ and this justifies the value of $\delta^*(D_{Wiki})$.

For the cases of $\partial^{out}A_D$ and $\partial^{in}A_D$ the visualization of Figure 2.3a is adopted and makes it possible to depict together differential values in both directions:
Example: Consider the grid Γ_D depicting A_D in Figure 2.3b. For each square in this grid, a new vertex is added in its center, an edge is drawn connecting it to its 4 corners, and then the square is removed. Notice that the resulting graph is a new infinite grid, denoted here by $\partial\Gamma_D$, whose squares are corresponding either to horizontal or to vertical edges of Γ_D. That way one can assign the values of $\partial^{out}A_D$ to "vertical" squares of $\partial\Gamma_D$ and and the values of $\partial^{in}A_D$ to "horizontal" squares of $\partial\Gamma_D$. The colors of the squares of $\partial\Gamma_D$ correspond to the different sizes of $\mathbf{DCL}^{out}_{k,l}$ and $\mathbf{DCL}^{in}_{k,l}$. That way the visualization of Figure 2.3a can be seen as a visualization of the discrete differential values of the matrix A_D depicted in Figure 2.3b.

The sequence of squares in Γ_D is called *incremental* if for each two consecutive squares (x, y), (x', y'), it holds that either $x' = x + 1$ and $y' = y$ or that $x' = x$ and $y' = y + 1$. Each incremental sequence that starts from $(0, 0)$ corresponds to a possible scenario of considering consecutive D-cores of D by gradually incrementing either the demand on the minimum out-degree or the demand on the minimum in-degree.

2.4.4 *Digraph Degeneracy Frontiers*

The following observation follows directly from the definitions:

Observation 1. *For every* k, k', l, l' *where* $k \geqslant k'$ *and* $l \geqslant l'$ *it holds that* $\mathbf{DC}_{k,l}$ *is a sub-digraph of* $\mathbf{DC}_{k',l'}$ *and therefore,* $\mathbf{dc}_{k,l} \leqslant \mathbf{dc}_{k',l'}$.

Here, a cell (k, l) is called *frontier cell* for a digraph D if $\mathbf{dc}_{k,l} > 0$ and $\mathbf{dc}_{k+1,l+1} = 0$ – thus the frontier consists of the cells corresponding to the last non-empty D-cores as k or l increase. The set of frontier cells of a digraph D is denoted as $F(D)$. Formally:

$$F(D) = \{(k, l) : \mathbf{dc}_{k,l} > 0 \ \& \ \mathbf{dc}_{k+1,l+1} = 0\}.$$

See Figure 2.3b where the frontier appears as the squares that have some common point with 0-valued squares (i.e. the white area).

The (k, l)-D-cores corresponding to the frontier cells are the *frontier* D*-cores* of D and all of them together constitute the D*-core frontier of* D. Intuitively, these D-cores exhibit the highest collaboration behavior in the network for different Hub/Authority trade-offs (i.e. H/A-angles).

Let k_{max} be the maximum k for which $(k, 0) \in F(D)$ and l_{max} be the maximum l for which $(0, l) \in F(D)$. We call $(k_{max}, 0), (0, l_{max})$ *extreme cells* of $F(D)$. Observe that number of frontier cells is always equal to $k_{max} + l_{max} - 1$. Thus the extreme $\mathbf{DC}_{0,l_{max}}$ represents the D-core with no in-links and a maximum number of out-links. In the *W*ikipedia graph the $\mathbf{DC}_{0,50}$ represents the sub-digraph bearing to a maximum the Hub-property (i.e. many out-links thus a very "extrovert" D-core). On the contrary, the extreme $\mathbf{DC}_{k_{max},0}$ represents the D-core with no out-links and a maximum number of in-links. In case of the *W*ikipedia digraph, this graph is $\mathbf{DC}_{42,0}$.

2.4.4.1 *Core Sequence in D-Cores*

Consider a core sequence $\mathcal{L}$ in Γ_D that starts from square (k, l) and finishes in square (k', l'). Let $e_1, \ldots, e_r$ be the sequence of edges that belong in consecutive squares of $\mathcal{L}$. Notice that each e_i corresponds to some square of $\partial\Gamma_D$ that, in turn, corresponds to some vertex set that is either $\mathbf{DCL}^{out}_{x,y}$ (in case e_i is a vertical edge) or $\mathbf{DCL}^{in}_{x,y}$ (in case e_i is an horizontal edge) for some value of x and y. We conclude that each monotone sequence $\mathcal{L}$ corresponds to a sequence of vertex sets that form a partition $\mathcal{P}$ of the vertex set $V(\mathbf{DC}_{k,l}) - V(\mathbf{DC}_{k',l'})$. That way, the size of $V(\mathbf{DC}_{k,l}) - V(\mathbf{DC}_{k',l'})$ (or, equivalently, the value $\mathbf{dc}_{k,l} - \mathbf{dc}_{k',l'}$) is the number of vertices that are discarded in order to transform $\mathbf{DC}_{k,l}$ to $\mathbf{DC}_{k',l'}$, following the core sequence $\mathcal{L}$. Notice that this number is always the same no matter the choice of the elimination sequence $\mathcal{L}$ (while certainly the partition $\mathcal{P}$ may vary a lot). Therefore, it can be said that the edge weighting of Γ_D defined by $\partial^{out}A_D$ and $\partial^{in}A_D$ is *adiabatic* in the sense that all paths between two vertices have the same total weight.

It is now possible to define the mono-dimensional analogue of core sequence and cell sequence in directed graphs. A *core sequence* of a directed graph D is an incremental sequence of squares in Γ_D that starts from $(0,0)$ and finishes in some square of the D-core frontier of D.

In conclusion, each core sequence $\mathcal{L}$ corresponds to a sequence of vertex sets that form a partition of the vertex set of D. This sequence is called *cell sequence* of D and is denoted by $\mathcal{P}(\mathcal{L})$. As there exist an exponential number of core sequences, the same holds also to the number of different partitions one may consider. This sharply contrasts the mono-dimensional undirected case where there the corresponding cell partition is uniquely defined.

2.4.5 *Digraph Collaboration indices*

This section treats the issue of choosing the optimal D-core on the frontier, as the most representative of the specific graph D-cores, with regard to the collaborative features as implemented via dense in/out links connectivity. To this end, different properties of digraph degeneracy are taken into account, especially with regard to the frontier. Intuitively, one would be interested in the dominant trend in the frontier D-cores i.e. whether they contain more in-links or out-links. Following this line, we define a series of metrics quantifying distinct measures of robustness.

BALANCED COLLABORATION INDEX(BCI)

One possibility is to choose a D-core with a balanced rate of in/out links. Thus is defined the *balanced collaboration index* of D as the unique integer r for which $\mathbf{DC}_{r,r}$ is a frontier (r,r)-D-core. In other words, these are the coordinates of the cell where the diagonal intersects the D-core-frontier of D. Formally, the *balanced collaboration index* of D, $\mathrm{BCI}(D)$, is equal to $\delta^*_{\pi/4}(D)$ (i.e. the H/A-angle is of 45°). The choice of the diagonal focuses on the D-cores with a balanced Hub/Authority trade-off - thus containing vertices that are connected to others, on average, with equal lower bounds on their in and out links.

OPTIMAL COLLABORATION INDEX (OCI)

In this case the frontier D-cores $\mathbf{DC}_{k,l}$, for which $(k+l)/2$ is maximized, is chosen . In terms of the D-core diagram, the position of such D-core has the maximum (among other frontier D-cores) Manhattan distance from the origin $(0,0)$. Formally the *optimal collaboration index*, $\mathrm{OCI}(D)$, is equal to $\delta^*(G)$. Notice that the frontier (k,l)-D-cores where $\frac{k+l}{2}$ is maximized can be multiple and may correspond to *several* H/A-angles.

INHERENT COLLABORATION INDEX (ICI)

This index aims to represent the inherent hubs/authority trade-off in the graph and is based on the average ratio of out-links to in-links of the vertices in the digraph. Based on this we define the average H/A-angle of a digraph D as follows:

$$\rho_{av} = \tan^{-1}\left(\frac{1}{|V(\mathbf{DC}_{1,1}(D))|} \cdot \sum_{v \in V(\mathbf{DC}_{1,1}(D))} \frac{\mathbf{deg}^{in}_{D}(v)}{\mathbf{deg}^{out}_{D}(v)}\right). \tag{2.14}$$

To make the above formula feasible, vertices with zero in or out links are excluded, i.e. the averaging is applied inside the D-core $\mathbf{DC}_{1,1}(D)$. The *inherent collaboration index*, $ICI(D)$, of the digraph D is equal to $\delta^*_{\rho_{av}}(D)$ where ρ_{av} is defined as above.

Thus the terms: BCI/OCI/ICI - optimal D-core(s) are used respectively for the D-cores corresponding to each particular optimization. See Figure 2.3b for a depiction of the above indices on the *W*ikipedia D-cores matrix frontier.

AVERAGE COLLABORATION INDEX (ACI)

This index is the average of the τ-degeneracies over all possible H/A-angles corresponding to the cells of the D-core frontier of D. Thus, the *average collaboration index*, $ACI(D)$, of the digraph D is defined as:

$$ACI(D) = \frac{1}{|F(G)|} \sum_{(k,l) \in F(D)} \delta^*_{\tan^{-1}(\frac{l}{k})}(D). \tag{2.15}$$

In other words, $ACI(D)$ is the half of the average Manhattan distance of the frontier cells of D. Alternatively defined:

$$ACI(D) = \frac{\sum_{(k,l) \in F(D)} (k+l)}{2 \cdot |F(D)|}. \tag{2.16}$$

ROBUSTNESS

Notice that the maximum value of the average collaboration index of a digraph D with extreme positions $(k_{max}, 0)$ and $(0, l_{max})$ is obtained in the case where

$$F(D) = \{(k_{max}, 0), (k_{max}, 1), \ldots, (k_{max}, l_{max}), \quad (k_{max} - 1, l_{max}), \ldots, (0, l_{max}))\}.$$

In this extreme and, in a sense, ideal case, the digraph D has the maximum possible robustness under degeneracy with respect to its extreme positions and the Average Collaboration Index of such a graph is equal to

$$\frac{2k_{max}l_{max} - k_{max} - l_{max} + \binom{k_{max}+1}{2} + \binom{l_{max}+1}{2}}{2 \cdot |F(D)|}.$$

The above quantity is denoted by $\mu(k_{max}, l_{max})$. That way, the *robustness* of a digraph D, with extreme positions (k_{max} and l_{max}), id defined as the ratio:

$$\frac{\sum_{(k,l)\in F(D)}(k+l)}{\mu(k_{max}, l_{max})}$$

and it always results in a real value in $[0, 1]$.

The above definition implies that the robustness is essentially the surface enclosed between the $F(D)$ frontier and the $(0,0), \ldots, (k_{max}, 0), (0,0), \ldots, (0, l_{max})$ coordinates divided by $\mu(k_{max}, l_{max})$. This represents the endurance of the D-core graph to degeneracy, i.e. the degree of cohesion among the graph nodes – in terms of globally distributed in/out links.

2.4.6 *Set frontiers and indices*

Let X be a subset of nodes in a digraph D. In a similar manner as above, the D-core matrix of X, $\mathbf{DC}^{X}_{k,l}(D)$, is defined as the cells (k, l) where X is a subset of $\mathbf{DC}_{k,l}$ and $\mathbf{dc}_{k,l} > 0$. Similarly, the **D**-core frontier of X is defined as the set of the extreme non-empty D-cores corresponding to the cells (k, l) where $\mathbf{dc}_{k,l} > 0$ and $\mathbf{dc}_{k+1,l+1} = 0$. Thus:

$$F_D(X) = \{(k, l) : X \subseteq D \;\&\; \mathbf{dc}_{k,l} > 0 \;\&\; \mathbf{dc}_{k+1,l+1} = 0\}. \tag{2.17}$$

The D-core matrix of a node set $X \subseteq V(D)$, is defined in an analogous way as in subsection 2.4.6. It represents the capacity of the nodes of X to be part, *all-together*, in subgraphs with strong mutual linking and thus presenting a noteworthy collaboration behavior.

The five collaboration indices for a set $X \subseteq V(D)$ as well as its robustness are defined analogously as in previous sections:

- The *Balanced Collaboration Index of* X, $BCI_D(X)$, is the maximum r for which $X \subseteq V(\mathbf{DC}_{r,r})$.
- The *Optimal Collaboration Index of* X, $OCI_D(X)$, is the maximum value of $\frac{k+l}{2}$ for which $X \subseteq V(\mathbf{DC}_{k,l})$.
- The *Inherent Collaboration Index of* X, $ICI_D(X)$, is the maximum $(\lceil k \rceil + \lceil l \rceil)/2$ for which $X \subseteq V(\mathbf{DC}_{k,l})$, where $k = r \cdot \cos \rho_{av}$ and $l = r \cdot \sin \rho_{av}$, for some r where $r^2 = k^2 + l^2$ (ρ_{av} is the average H/A-angle, defined as in the previous subsection).

- The *robustness of a set* X with extreme positions (k_{max} and l_{max}) is defined as the ratio: $\frac{\sum_{(k,l)\in F_D(X)}(k+l)}{\mu(k_{max},l_{max})}$ where the function μ is defined as in the previous section.

- The *Average collaboration H/A-angle of a set* with extreme positions (k_{max} and l_{max}) is defined as:

$$\sigma_D(X) = \frac{\sum_{(k,l)\in F_D(X)}(k+l)\cdot \tan^{-1}(\frac{l}{k})}{\sum_{(k,l)\in F_D(X)}(k+l)}$$

  As before, this angle conveys the Hub/Authority trade-off for the D-cores in which X is a subgraph.

These indices can be applied also to every individual node $x \in V(D)$ by setting $X = \{x\}$. In this case, all above notations and concepts can also be used for nodes instead of sets of nodes. Notice that all indices defined in this subsection are anti-monotone. In particular:

Observation 2. *Let X_1 and X_2 are subsets of the vertex set of some digraph D. If $X_1 \subseteq X_2$, then the balanced/optimal/inherent/diagonal collaboration index of X_1 will be at least the balanced/optimal/inherent/diagonal collaboration index of X_2.*

2.4.7 *Core Decomposition Forests*

In this section the concept of a core decomposition forest is defined for D-cores in order to examine the structure of a digraph and the connected components of its cores. But first some definitions are required.

A digraph D is *strongly connected* when every pair x and y of vertices in D is been met by some directed cycle, i.e. there is a directed path from x to y and a directed path from y to x.

A *strongly connected* component (in short: SCC) of a digraph D is any maximum sub-digraph of D that is strongly connected. Finding the strongly connected components of a digraph graph can be done in time linear to the sum of its edges and vertices.

Definition 13. *Let $\mathcal{D} = D_0, D_1, \ldots, D_d$ be a sequence of digraphs such that for each i, j where $i \leqslant j$, D_i is a subgraph of D_j (we call such a sequence* monotone*). The* Decomposition Forest *of a monotone digraph sequence $\mathcal{D}$ is the digraph $\mathbf{DF}(\mathcal{D})$ defined in the following way: For each $i = 0, \ldots, d$ we denote the strongly connected components of D_i by $D_i^1, \ldots, D_i^{m_i}$ and each such strongly connected component is a vertex of $\mathbf{DF}(\mathcal{G})$. An edge $(G_i^j, G_{i'}^{j'})$ is added in $\mathbf{DF}(\mathcal{G})$ if $j' = j+1$ and G_i^j contains $G_i^{j'}$ as a sub-digraph.*

The above definition implies directly that $\mathbf{DF}(\mathcal{D})$ is a union of trees, each rooted to some of the strongly connected components of D_0. Given now a directed graph

D and one, say $\mathcal{L}$ of its cell sequences, it is easy to verify that the directed graph defined above is a rooted forest.In fact, each of its components is a rooted tree where all its edges are directed away from the root and each root is a connected component of D_0. Given that, by its definition, each core sequence $\mathcal{L}$ of D is monotone, the *Core Decomposition Forest (CDF) of* D *with respect to* $\mathcal{L}$ is defined as the decomposition forest corresponding to $\mathcal{L}$.

This notion of the *Decomposition Forest* is an extension of the undirected one (2.3.1.2) to digraphs and in the experimental study is used to visualize the core decomposition forests for both Wikipedia 2004 and DBLP, where the sequence $\mathcal{L}$ corresponds to the cells in the diagonal of each D-core matrix (see Figure 2.3b).

2.5 S-CORES AND RECIPROCITY

2.5.1 *Introduction*

In this section the fundamental concept of degeneracy is defined for signed graphs. This will be exploited towards the evaluation of trust and the definition of reciprocity in such graphs. First will be the definition of the notion of S-core – an extension of D-core – that represents degeneracy in signed graphs. Moreover additional concepts are defined to quantify the robustness of the graph under degree based degeneracy.

The concept of reciprocity as defined in existing works – as a measure of local mutuality among pairs of nodes – does not offer an adequate descriptive capability (especially for signed networks) for measuring reciprocity at the graph level. For that reason a comparison will be made between reciprocity in signed graphs and metrics derived from the S-core structure.

The comparison, of reciprocity with the degeneracy based metrics, will be done under the signed graph context not only to display a better concept for reciprocity but also to signify the need of the metrics in an area of graph mining that is in its' "dawning" phase.

Reciprocity, in unsigned directed networks, quantifies the predisposition that the members of a network display in creating mutual connections. In signed trust networks, reciprocity would have different interpretations based on the pairs of signs one examines. The in^+/out^+ pairs, provide an indication on the level of trust. In contrast, in^-/out^- pairs would indicate distrust or vindictiveness. Moreover, the in^+/out^- and in^-/out^+ pairs may reveal interesting aspects as well. The reciprocity of the former would describe impartiality under positive votes (trust), while the latter would describe impartiality under negative votes (distrust). A more strict version of reciprocity could be viewed by the account of only the number of bidirectional links baring the same sign in both ends. This would de-

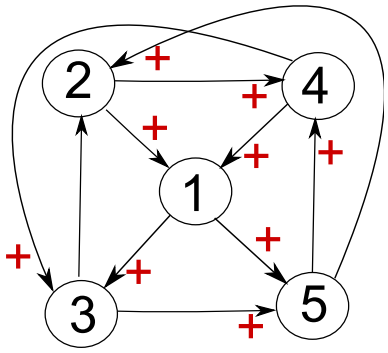


Figure 2.4: Example digraph of signed networks.

scribe only the most basic nature of reciprocity without taking into account any further context.

Disregarding for now what would be the best definition of reciprocity for signed networks, in Figure 2.4 a sample toy signed directed graph can be seen (to keep it simple the assumption of only positive signs on the links is made) representing trust relations. As one observe there is no pairwise mutuality in terms of mutual plus links among pairs of nodes – therefore the local reciprocity as defined in directed and signed network is zero. On the other hand it is clear from the graph that there is a global reciprocity as for each node we observe a balanced in / out positive trust. For example, in this case each node offers two outgoing positive (+) trust links to the community and at the same time receives two positive (+) incoming trust links – although they do not emanate from the same node as those it gives trust to. Thus it is evident that there is a challenge in representing and dealing with reciprocity at a more global level. This issue becomes even greater when the added complexity of signs is taken into consideration; since possible combinations of local reciprocities would lessen their importance when looking at the graph at a node level.

2.5.2 *S-cores*

A *signed digraph* is a triple $G = (V, E, \mathbf{w})$ such that (V, E) is a simple directed graph and $\mathbf{w} : E \to \{+, -\}$ is a labeling of E, assigning either a positive or a negative sign on the edges of G. The existence of a positive signed edge $e = (x, y)$ from a vertex x to a vertex y represents the fact that "x trusts y" or "x likes y", while the existence of the same edge with negative sign means that "x distrusts y" or "x dislikes y".

Given a vertex v of G, the *positive in-degree* (resp. *positive out-degree*) of v in G is denoted by $\mathbf{deg}^+_{in}(v, G)$ (resp. $\mathbf{deg}^+_{out}(v, G)$), i.e. the number of positive-signed

edges tailing (resp. heading) on v. The *negative in-* and *out-degrees* of the vertices of G are defined analogously and are denoted by $\mathbf{deg}^-_{in}(v, G)$ and $\mathbf{deg}^-_{out}(v, G)$.

Definition 14. *Let* $G = (V, E, \mathbf{w})$ *be a signed graph. Let also* $s, t \in \{+, -\}$ *and* $k, l \in \mathbb{N}$. *The* (l^t, k^s)-dicore of G *is defined as the maximum size subgraph* H *of* G *where, for each vertex* v *of* H, *it holds that* $\mathbf{deg}^s_{in}(v, H) \geqslant k$ *and* $\mathbf{deg}^t_{out}(v, H) \geqslant l$.

For the rest of this document the generic term S-*core* is used when there is no need to make explicit the values of the pair (l^t, k^s).

Notice that the (l^t, k^s)-dicore of G can be computed by a greedy procedure similar to Trim and $\text{Trim}_{k,l}$ (sections 2.3.1.1 and 2.4 respectively):
In short, remove from G *a vertex* v *where* $\mathbf{deg}^s_{in}(v, H) < k$ *or* $\mathbf{deg}^t_{out}(v, H) < l$, *until this is not possible anymore.*

It is straightforward that the resulting sub-graph is well-defined – i.e., it is the same regardless of the order of elimination of vertices – and it is indeed the (k^s, l^t)-dicore of G.

Definition 15. (s, t)***-degeneracy:*** *Given a pair* $(s, t) \in \{+, -\}^2$, *the* (s, t)-degeneracy of G *is defined as follows:*

$$\delta^{s,t}(G) = \max\{\tfrac{k+l}{2} \mid G \text{ contains a non-empty } (l^t, k^s)\text{-dicore }\}. \tag{2.18}$$

For convenience, the sign function $\mathbf{s} : \mathbb{Z} \to \{+, -\}$ is defined so that, given an integer i, it outputs $-$ or $+$ depending whether i is negative or not.

Thus the (s, t)-degeneracy of the graph represents its robustness under degeneracy in the four different combinations of edge-direction and sign. In the case of trust networks the (s, t)-degeneracy represents the degeneracy of the graph for each of the combinations of incoming/outgoing and positive/negative trust. For instance refer to Figure 2.5 where the four cases of degeneracy are depicted as $\delta^{++}_{max}(G)$ etc.

Definition 16. ***Signed dicore diagram:*** *The* signed dicore diagram, *of a signed digraph* G, *is defined as a matrix* $A = (\alpha_{i,j})_{(i,j)\in\mathbb{Z}^2}$ *where for each* $(i, j) \in \mathbb{Z}^2$, $\alpha_{a,b}$ *is the size (i.e., the number of vertices) of the* $(i^{s(i)}, j^{s(j)})$*-dicore of* G.

As it becomes obvious many of the definitions of the S-core will describe concepts similar to the ones of the D-core . For example the previous definition is equivalent to the one of the D-core Matrix (section 2.4.2). As the signs of the graph create a more complicated context, the same notions need to be defined with greater intricacy.

SIGNED GRAPH EXTENSION

As the above definition of $A = (\alpha_{i,j})_{(i,j)\in\mathbb{Z}^2}$ produces an infinite matrix, it is sufficient to consider its finite portion, which contains all its non empty dicores. For this, i and j are restricted to belong in the *frame of* G that is the set $F_G = \{-(b+1), \ldots, 0, \ldots, b+1\}$ where

$$b = \max\{l, k \mid G \text{ has a non-empty } (l^t, k^s)\text{-dicore for some } (s,t) \in \{+,-\}^2\}. \quad (2.19)$$

The value b will be called the *extension* of the signed graph G and denoted by $\mathbf{b}(G)$.

Definition 17. S-*core region.* *Given a signed graph* G, *its* core region *is defined as the set* $R_G = \{(i,j) \in F_G^2 \mid \alpha_{i,j} > 0\}$, *that is all the pairs that correspond to a non-empty* S-*core.*

2.5.2.1 *S-core Frontier*

Definition 18. *The* core-frontier *of* G, *denoted by* B_G, *is the set of all entries* $(i,j) \in F_G^2$ *with the property that* $\alpha_{i,j} > 0$ *and* $\alpha_{i+1,j+1} = 0$. *These are the extreme non-empty* S-*cores, in the sense that any further shift of their coordinate results in an empty* S-*core.*

Definition 19. *Similarly to* B_G, *given a vertex* $x \in V(G)$, *its* core region of x *is defined as the set* $R_G(x) = \{(i,j) \in F_G^2 \mid x \text{ belongs to the } (i^{s(i)}, j^{s(j)})\text{-dicore}\}$.

The *core-frontier* of x, denoted by $B_G(x)$, is the set of all entries $(i,j) \in F_G^2$ with the property that $(i,j) \in R_G(x)$ and $(i+1, j+1) \notin R_G(x)$. The semantics of the core frontiers vary and they will become clear in the "**Data Exploration**" part.

Here, an attempt is made for an intuitive presentation of the above definitions based on Figure 2.5. There the reader may see the extension of the degeneracy of the four cases in the areas (e.g. $R^{++}(G)$ for the positive in and out trusts edges degeneracy) enclosed by the respective frontiers.

To evaluate the trust/distrust tendencies in the signed digraphs, a series of parameters need to be defined. For this, it is useful to model the dicore diagram as a graph Γ_G, defined as follows:

Definition 20. *The vertices of* Γ_G *are the pairs in the set* R_G *and an edge* $\{\{i,j\},\{i',j'\}\}$ *exists in* Γ_G *iff* $\{i,j\},\{i',j'\} \in R_G$ *and* $|i-i'|+|j-j'| = 1$. *Therefore,* Γ_G *can be seen as the subgraph of a grid whose vertices are the pairs corresponding to non-empty* S-*cores.*

This graph-theoretic representation of G makes it possible to assess the trust/distrust tendencies in a signed graph, by studying the geometry of Γ under the regular metric of graph distance. Many of the above definitions, (and some that follow in latter sections) can be seen in a visual representation in Figure 2.5.

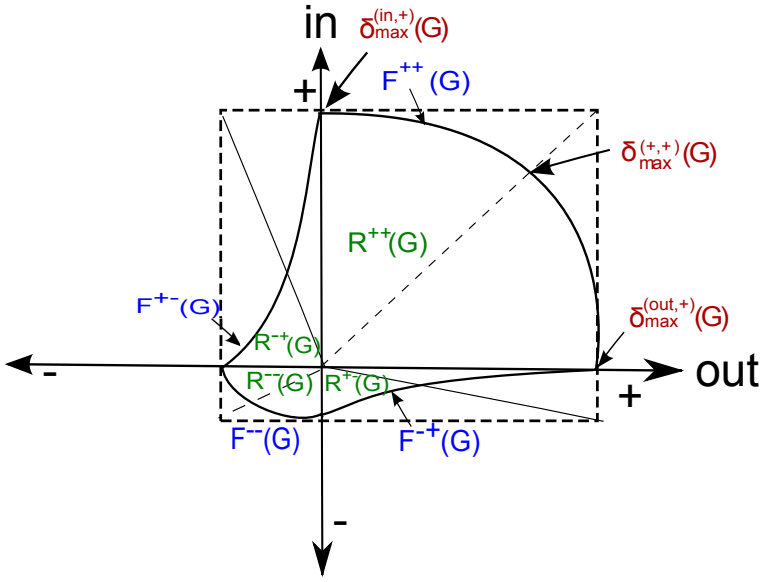


Figure 2.5: Metrics on the S-core graph. The square in the doted line is the extension and the irregular shape is the frontier.

2.5.3 *Reciprocity in Signed Graphs*

In this section, different notions of reciprocity in signed graphs are defined. The definitions are build upon the existing ones of reciprocity for directed graphs based on local criteria and extend them towards signed ones. Moreover novel notions of reciprocity are defined that do not depend only on local binary reciprocity but represent this concept in an aggregate manner at the graph level.

2.5.3.1 *Signed graph reciprocity – local definition*

First, existing definitions [69, 82, 87] have to be adapted to signed digraphs. Trust networks are conspired as a prominent example of signed digraphs where a node can either trust or distrust another. Additionally, since self-trust is trivial, self loops are excluded.

The intuition of reciprocity in signed networks must also be examined. Two different options are explored:

i. *Contextual local reciprocity*, where we examine all four possible sign permutations between two reciprocal edges where each sign permutation defines a context of trust and

ii. *Simple local reciprocity*, where we consider only the mutuality under trust and distrust – i.e. we consider only the pairs of nodes with the same sign that represents the coarse level of trust reciprocity.

CONTEXTUAL LOCAL RECIPROCITY

Following are the definitions of reciprocity emanating from all the possible signs permutations on reciprocal links on a pair on nodes:

$$\begin{aligned} r^{++} &\equiv \frac{L^{+\leftrightarrow+}}{L^{+}} \text{ for the } in^{+}/out^{+} \\ r^{+-} &\equiv \frac{L^{+\leftrightarrow-}}{L^{+}} \text{ for the } in^{+}/out^{-} \\ r^{--} &\equiv \frac{L^{-\leftrightarrow-}}{L^{-}} \text{ for the } in^{-}/out^{-} \\ r^{-+} &\equiv \frac{L^{-\leftrightarrow+}}{L^{-}} \text{ for the } in^{-}/out^{+}. \end{aligned} \tag{2.20}$$

where $L^{+/-}$ is the count of positive/negative edges and the signs on the double arrow of $L^{\leftrightarrow}$ *(links pointing both ways i.e. reciprocations)* indicate the sign of in and out edges respectively. Notice that the denominator is not the same for all the definitions. It could have been L instead of $L^{+/-}$ but, in the trust model explored here, the second and fourth reciprocity would have had identical values. The identical values would not be an issue for a more relaxed model that would allow two edges of different sign to have the same source and target.

The rationale for this definition is that, since reciprocity quantifies mutuality, only the type of actions that are being mutual are of interest. For each type of reciprocity above, a different set of actions is selected which are of interest to see if they are being reciprocated. For example, in the study of reciprocation of trust by trust (in^{+}/out^{+}) it is more intuitive (and more expressive) to compute reciprocity as only the portion of the positive edges and not the total number of them. More over, with this, in the assumptions about the network, it is possible to have distinguishable values between the in^{+}/out^{-} and in^{-}/out^{+} cases of reciprocity.

SIMPLE LOCAL RECIPROCITY

Moving on to the second definition, only the same sign reciprocations are considered and thus the following reciprocity is defined :

$$r^{s} \equiv \frac{L^{+\leftrightarrow+} + L^{-\leftrightarrow-}}{L}. \tag{2.21}$$

For the rest of the document r^{s} will be referred as *simple reciprocity* and the former set of four signed reciprocities as *contextual reciprocity*. Additionally, the average of the local reciprocities over the individual nodes is considered (e.g. $r_{a}^{+\leftrightarrow+}$ is the average of ratios of reciprocal positive edges in individuals over all vertices). These *average local reciprocities* are utilized only for comparison and not any further trust evaluation. All references to local reciprocities in the text correspond to the original five definitions unless specified otherwise.

Observation 1. ***Invariance under sign flipping:*** *A crucial difference between the two definitions is that the first one is not invariant to sign flipping while the second one is. With contextual reciprocity the objective is to quantify different behaviors under a particular context. For example, trust and distrust are two opposite concepts and their measurement should change if there is a different count of reciprocal signs. On the other hand, simple reciprocity remains the same since only one type of behavior is counted. Therefore, flipping the signs should not (and does not) change the measurement of simple reciprocity.*

2.5.3.2 *Signed graph reciprocity – global definition*

As seen in the introduction, the concept of reciprocity as defined in existing works capitalizes on the local property of mutuality among pairs of nodes and does not offer an adequate descriptive capability for measuring reciprocity at the graph level.

Following here is the definition of metrics that represent signed graph reciprocity at graph level. Figure 2.5 is a visual aid to those definitions (the S-core frontier here is the irregular shape outlined with the thick line). In this diagram the trust axes $(\mathtt{in}, \mathtt{out})$ and signs $(+, -)$ define respective quadrants $Q_{\mathtt{out\_sign},\mathtt{in\_sign}}$, where $\mathtt{out\_sign}, \mathtt{in\_sign} \in \{+, -\}$. Each of the quadrants bears specific semantics regarding the in/out trust. For instance $Q_{+,+}$ represents degeneracy in graphs where the criterion is the mutual incoming and outgoing trust. On the other hand $Q_{+,-}$ represents degeneracy under outgoing trust but incoming distrust. The graphs in the S-Core frontier (in $Q_{+,-}$) represent situations where users maximally trust others in the graph but they receive distrust from others. The interpretations are analogous for the remaining two quadrants $Q_{-,-}$, $Q_{-,+}$.

MAXIMUM DEGENERACY ON THE TRUST AXES

Here is discussed the extreme degeneracy on each of the four trust axes – representing the robustness of the graph for each type of trust. For instance the $\delta_{max}^{(out,+)}(G)$ represents the extreme graph with regards to outgoing positive trust degeneracy, i.e. the last non empty graph when we increase the threshold for the outgoing positive trust. Similarly are defined the rest of extreme degeneracies $\delta_{max}^{(in,+)}(G)$, $\delta_{max}^{(out,-)}(G)$, $\delta_{max}^{(in,-)}(G)$ on the other trust axes.

QUADRANT BOUNDING BOX

For each of the four aforementioned quadrant there is the previously defined frontier that has a respective bounding box which is defined by the maximal degeneracies on the relevant axes. For instance the bounding box for $Q_{+,+}$ is defined by the dotted rectangle define by the axes the points: $0, 0$, $(\delta_{max}^{(out,+)}(G)$, $\delta_{max}^{(in,+)}(G))$. This bounding box would be the S-core frontier of G if all its vertices

(or at least a subset of the vertices in G) had degrees of at least $\delta_{max}^{(out,+)}(G)$ and $\delta_{max}^{(in,+)}(G)$ and moreover their out/in edges connected them with vertices having the same property.

QUADRANT MAXIMAL DEGENERACY

By utilizing the bounding box, it is possible to define the *max degeneracy* of each quadrant as the *intersection point between the diagonal of the bounding box*. For instance for the $Q_{+,+}$ quadrant the maximal degeneracy $\delta_{max}^{(+,+)}(G)$ is defined by the intersection of the diagonal $(0,0,(\delta_{max}^{(out,+)}(G),\delta_{max}^{(in,+)}(G)))$ and the respective frontier $F^{++}(G)$. The max degeneracy of each quadrant corresponds to the most extreme core in relevance to the natural ratio of the maximum degeneracies that characterize the quadrant. This metric signifies the overall activity of the signed network (i.e. evaluation of how much the users interact for each type of relationship) while, at the same time, taking into account the over all outward or inward tendencies (e.g. are the users more prone to giving or receiving trust).

CONTEXTUAL RECIPROCITY

Here the definition of reciprocity at a graph level is defined in accordance to the local reciprocities defined in the previous subsections. As it can be observed in Figure 2.5, the S-core frontier covers the bounding boxes reaching different levels of degeneracy in each. This is utilized to measure graph level reciprocity. Thus *contextual reciprocity* at the graph level is defined (per quadrant) as the ratio of the area under the respective quadrant frontier over the corresponding bounding box surface:

$$\begin{aligned} GR^{++} &= R^{++}(G)/(\delta_{max}^{(out,+)}(G) * \delta_{max}^{(in,+)}(G)) \\ GR^{+-} &= R^{+-}(G)/(\delta_{max}^{(out,+)}(G) * \delta_{max}^{(in,-)}(G)) \\ GR^{--} &= R^{--}(G)/(\delta_{max}^{(out,-)}(G) * \delta_{max}^{(in,-)}(G)) \\ GR^{-+} &- R^{-+}(G)/(\delta_{max}^{(out,-)}(G) * \delta_{max}^{(in,+)}(G)). \end{aligned} \tag{2.22}$$

These *Global Reciprocities* have the same contextual meaning as their node level equivalents. For example, GR^{++} (the graph level equivalent of r^{++}) measures global trust reciprocity and would reach a maximum value of one if everybody gave as much trust as they received (but not necessarily to the same people).

GLOBAL RECIPROCITY

Much like the node level reciprocities there is a need to define a more strict version of reciprocity for the mutual interchange of the same types of actions. For this purpose, the quadrants of same sign are considered (i.e. $Q_{+,+}$, $Q_{-,-}$) as those

capturing this type of reciprocity, and the quadrants having different in/out signs as ones capturing inverse reciprocity. Then *Graph Reciprocity* can defined as:

$$GR = \frac{GR^{++} + GR^{--}}{GR^{++} + GR^{+-} + GR^{--} + GR^{-+}}. \tag{2.23}$$

Ideally, the value of this metric would reach a maximum of one only when the reciprocation is in the same sign quadrants (i.e. GR^{+-} and GR^{-+} are equal to zero) thus keeping the same range of values as the rest of reciprocations. By taking into account the inversely reciprocal quadrants, there is a difference between cases where the signed graph is highly reciprocal only in the same sign quadrants and cases where the same applies while simultaneously there is also high reciprocation in the other quadrants as well.

2.5.4 *Conclusion*

Cohesion and collaboration in graphs are cornerstone features for their evaluation, especially with the advent of large scale applications such as the Web, social networks, citations graphs etc. The traditional way to look at graphs is through the authority/hub notion based on *per node* in/out links patterns. Other group evaluation measures do not take into account the directed nature of the aforementioned graphs. On the contrary, with the definitions presented in this part of the document, the importance of cohesion and collaboration among groups of nodes is being stressed. The intuition is that sub-graphs with many links among their nodes convey a high degree of collaboration (adapted to the local application semantics). Thus, extensions of the k-core structure are defined, as means of representing their collaborative features based on their robustness under degeneracy.

Part II

DATA EXPLORATION

3

Experiments

3.1 Introduction

The metrics and structures defined in the previous sections will be used here to evaluate large scale graphs of real world networks. The aim is to detect, in each dataset, the sets of authors that correspond to the most coherent communities in terms of co-authorship collaboration. In the following sections, for each of the aforementioned variations of core structure, all the relative to each case datasets will be described in full detail along with information related to the processes involved in modeling them to their respective graphs. Following the experiments conducted will be presented along with interpretations on the results.

3.2 Undirected and weighted

3.2.1 *Dataset Description*

The application of the k-core and the fractional core framework is done on the bipartite graphs corresponding to the DBLP dataset, concerning publications in computer science, and the ARXIV on High Energy Physics - Theory (ARXIV.hep-th) dataset, concerning publications on High Energy Physics. From now on, for notational convenience, the abbreviation ARXIV will be used instead of ARXIV.hep-th.

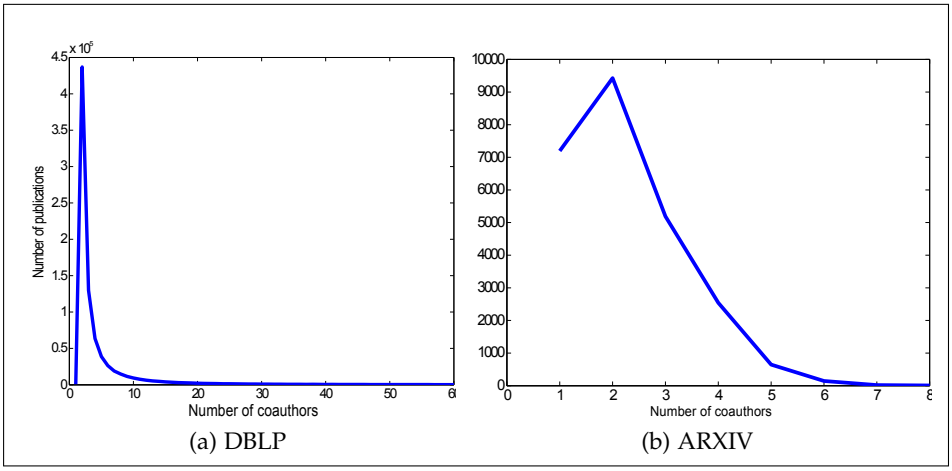
The DBLP dataset is freely available in XML format at

http://dblp.uni-trier.de/xml/

and the ARXIV dataset on High Energy Physics Theory is available in simple text format at:

http://snap.stanford.edu/data/ca-HepTh.html

The bipartite graphs DBLP and ARXIV were extracted from these datasets. In the current snapshot, DBLP has 2208512 papers while ARXIV has 25170 papers. Among them, 817 of the papers in DBLP have only one author, while the same holds for 7196 of the papers of ARXIV. Also, DBLP has 825761 authors and ARXIV



(a) DBLP (b) ARXIV

Figure 3.1: Distribution of number of publications versus cardinality of co-author set for a.DBLP and b.ARXIV.

has approximately 8862 authors. In total, DBLP has 4446765 edges and ARXIV has 56065 edges.

In Figure 3.1, one can see the distribution of the number of co-authors per publication in the DBLP graph and the ARXIV graph. It is clear that the vast majority of the papers are authored by few authors. However, there are some extremities where one specific paper in DBLP has 114 authors! On the other side all papers in ARXIV have at most 8 co-authors.

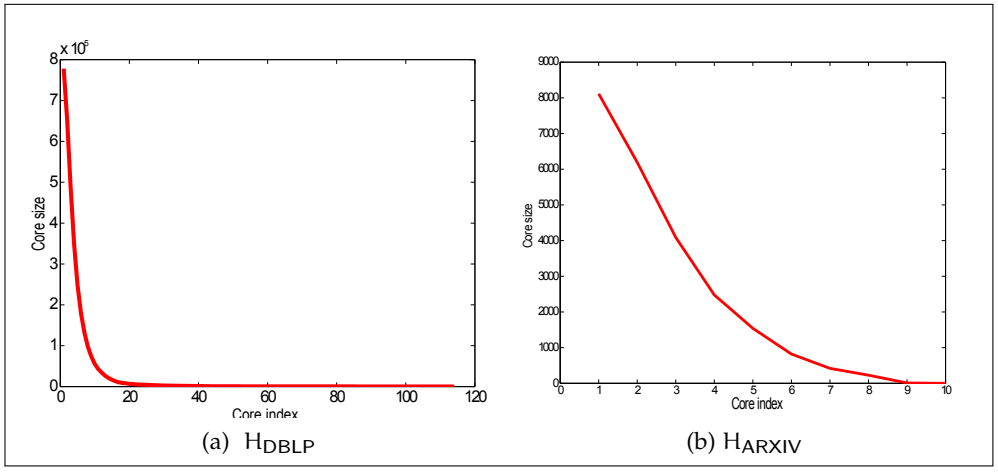
The unweighted graphs H_{DBLP} and H_{ARXIV} and their edge-weighted versions $(H_{DBLP}, \mathbf{w})$ and $(H_{ARXIV}, \mathbf{w})$ were computed, as described in Subsections 2.3.2 and 2.3.3.

Clearly, single-author papers will not create any edge between authors and all isolated vertices in H_{DBLP} and H_{ARXIV} correspond to authors that have written only single-author papers.

3.2.2 k-*cores in co-authorship graphs*

We applied the *Trim* procedure to find the core sequences of the graphs H_{DBLP} and H_{ARXIV}. In this computation, we took into account all the papers regardless of the number of the authors each may have. In Figures 3.2a and 3.2b, we can see the distribution of cores sizes for each graph.

In Table 3.1, a ranking of a few selected authors is presented for both datasets. As mentioned before, one paper with a large number of co-authors can "push" authors with otherwise low co-authorship to the densest k-core. For example, in DBLP, at $k = 113$ there are 114 authors all of which have participated in the same publication and some of them do not appear anywhere else on the dataset.



(a) H_{DBLP} (b) H_{ARXIV}

Figure 3.2: Distribution of the core sizes vs core indices in H_{DBLP} and H_{ARXIV}.

H_{DBLP}	**Name of author**	**Index**
	Serge Abiteboul	28
	Christos Faloutsos	28
	Gerhard Weikum	22
	Christos H. Papadimitriou	17
	Paul Erdös	16
	Andrew Tanenbaum	48

H_{ARXIV}	**Name of author**	**Index**
	Mirjam Cvetic	9
	Riccardo D'Auria	8
	Christoph Schweigert	7
	John Ellis	6
	Jürgen Fuchs	6
	Dimitris Nanopoulos	6

Table 3.1: Ranks of selected authors in H_{DBLP} and H_{ARXIV}.

Actually, the results on the H_{DBLP} graphs are apparently quite biased, i.e. a maximum-index 113-core exists in H_{DBLP} because of the existence of a single paper regardless of their other publication activity. In graph theoretic terms this H_{DBLP} core is a clique of 114 vertices that is created because of the existence of a vertex in DBLP of degree 113. However, this does not hold for the case of the – smaller in size – graph H_{ARXIV} where the maximum number of authors in a paper is 8. The densest core in H_{ARXIV} is the 9-core and is a clique on 10 vertices. The members of this core are presented in the lower part of Table 3.2. It is interesting to note that the edges of this clique are formed by many different papers. In fact there are at least 118 papers in ARXIV that have been co-authored by at least two of the members of the 9-core of H_{ARXIV}.

The biased situation that was detected in H_{DBLP} was the motivation to consider filtering out papers with excessively high number of co-authors. In this case, a filtered version of H_{DBLP} was computed, by taking into account only the papers whose number of co-authors is within the 99% of the corresponding distribution shown in Figure 3.1. This excludes from DBLP papers with more than 15 co-authors. This version of the graph H_{DBLP} is called *filtered* and is denoted it by H^*_{DBLP}.

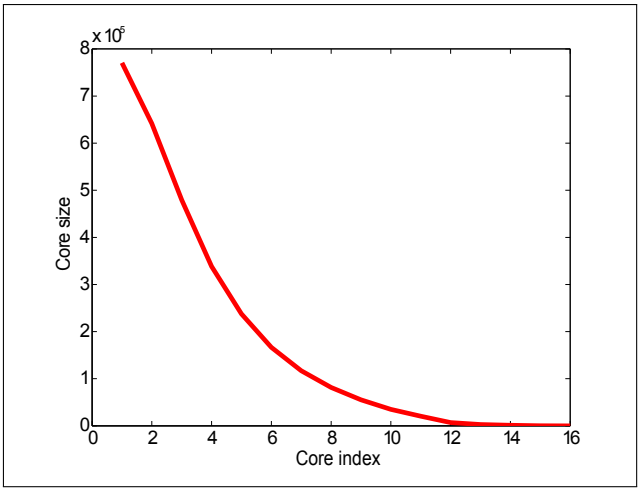


Figure 3.3: Distribution of the core sizes vs core indices in H^*_{DBLP}.

H^*_{DBLP}

Pankaj K. Agarwal	Hee-Kap Ahn	Oswin Aichholzer	Greg Aloupis
Helmut Alt	Esther M. Arkin	Boris Aronov	Tetsuo Asano
Mark de Berg	Therese C. Biedl	Prosenjit Bose	David Bremner
Hervé Bronnimann	Sergio Cabello	Timothy M. Chan	Bernard Chazelle
Otfried Cheong	Sébastien Collette	Mirela Damian	Erik D. Demaine
Martin L. Demaine	Olivier Devillers	Vida Dujmovic	Herbert Edelsbrunner
Alon Efrat	David Eppstein	Jeff Erickson	Hazel Everett
Sàndor P. Fekete	Joachim Gudmundsson	Leonidas J. Guibas	Dan Halperin
Sariel Har-Peled	John Hershberger	Ferran Hurtado	John Iacono
Christian Knauer	Danny Krizanc	Stefan Langerman	Sylvain Lazard
Giuseppe Liotta	Anna Lubiw	Rolf Klein Mark	Jirì Matousek
Kurt Mehlhorn	Henk Meijer	Joseph S. B. Mitchell	Pat Morin
Joseph O'Rourke	Mark H. Overmars	Belén Palop	Richard Pollack
Suneeta Ramaswami	David Rappaport	Gunter Rote	Vera Sacristan
Otfried Schwarzkopf	Raimund Seidel	Micha Sharir	Thomas C. Shermer
Michiel H. M. Smid	Jack Snoeyink	Michael A. Soss	Diane L. Souvaine
Bettina Speckmann	Ileana Streinu	Subhash Suri	Perouz Taslakian
Godfried T. Toussaint	Marc J. van Kreveld	Jorge Urrutia	Sue Whitesides
David R. Wood	Stefanie Wuhrer	Chee-Keng Yap	Emo Welzl

H_{ARXIV}

Mirjam Cvetič	Michael J. Duf	P Hoxha	R Martinez-Acosta
James T. Liu	Hong Lu	Jian-Xin Lu	Christopher N. Pope
Hisham Sati	Tuan A. Tran		

Table 3.2: Authors of the 15-core of H^*_{DBLP} (top) and the 9-core of H_{ARXIV} (down).

Name of author	Index
Serge Abiteboul	14
Paul Erdös	14
Christos Faloutsos	14
Christos H. Papadimitriou	14
Gerhard Weikum	14
Andrew Tanenbaum	12

Table 3.3: Ranks of selected authors in H^*_{DBLP}.

The *Trim* procedure is applied to find the core sequence of the graph H^*_{DBLP}. The distribution of the resulting core sizes appears in Figure 3.3. In the filtered case, the densest core of H^*_{DBLP} has index 15 and has a size of 76 authors. These authors appear in the upper part of Table 3.2.

As expected, in the filtered graph H_{DBLP}, several of the authors "move down" in cores of smaller index. The new indices for the selected sets of authors of Table 3.1 for DBLP are now depicted in Table 3.3. As seen there, in the case of H_{DBLP}, the authors of Table 3.3 get now accumulated in the second densest core, i.e the 14-core.

It is interesting that for some authors of DBLP, such as Andrew Tanenbaum, the core index in the filtered case is much lower (12) that in the unfiltered one (48). Apparently, this happens due to his participation in multi-author papers that were filtered out.

3.2.3 *Fractional cores on the weighted graph*

Here the need is articulated for assigning weights to the edges of the previously defined co-authorship graphs. Assume that two authors x, y have co-authored several papers and therefore they are connected by an edge $e = \{x, y\}$. This co-authorship relation represents a strong collaboration among the two that escapes the unweighted setting of the previous section.

This collaborative effort is apparently larger as the number of co-authored papers increases. On the other hand, the effort to author a paper is naturally divided among all the co-authors (we assume in equal parts). This justifies the definition in Section 2.3.3 of an edge-weighted co-authorship graph where the contribution of each author is now fractional.

In the fractional case, there is no need to apply any filtering of papers with a huge number of authors, as they are now filtered indirectly because the weight they contribute to their authors is tiny. Recall that the weight $\mathbf{w}(e)$ assigned to each edge is proportional to the number of papers they have co-authored and

$DBLP_a$							
34.0	**34.3**	35.2	36.4	37	37.3	38.8	42.7
42	**39**	35	31	29	25	22	20
8	7	7	4	4	3	3	3
$DBLP_b$							
44.2	47.8	48.4	53.8	55.3	64.6	77.8	149.2
18	16	13	11	8	6	4	2
3	3	3	3	2	2	2	2
$ARXIV_a$							
10.4	10.5	10.6	10.7	11.0	11.4	11.5	**12.0**
51	50	36	35	33	26	23	**21**
37	36	20	19	19	16	16	**14**
$ARXIV_b$							
13.1	**13.4**	13.7	14.9	16.0	21.7	24.5	34.9
16	**14**	11	9	6	5	4	2
6	6	6	6	6	5	2	2

Table 3.4: Data of the last 16 graphs of the fractional core sequence of $(H_{DBLP}, \mathbf{w})$ (top) and $(H_{ARXIV}, \mathbf{w})$ (bottom). For each dataset, the first line depicts h_i, the second line contains the size of the h_i-core and the third one contains the size of the biggest connected component of the h_i-core. The data have been split for each dataset into subsets *a* and *b* for better presentation.

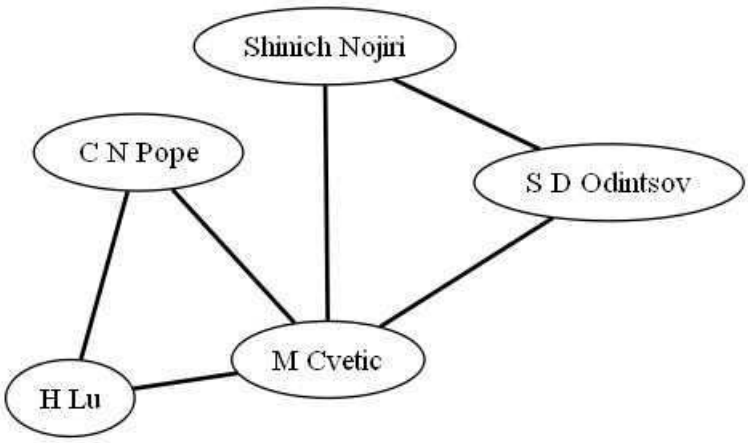


Figure 3.4: The 27.1-core of $(H_{ARXIV}, \mathbf{w})$.

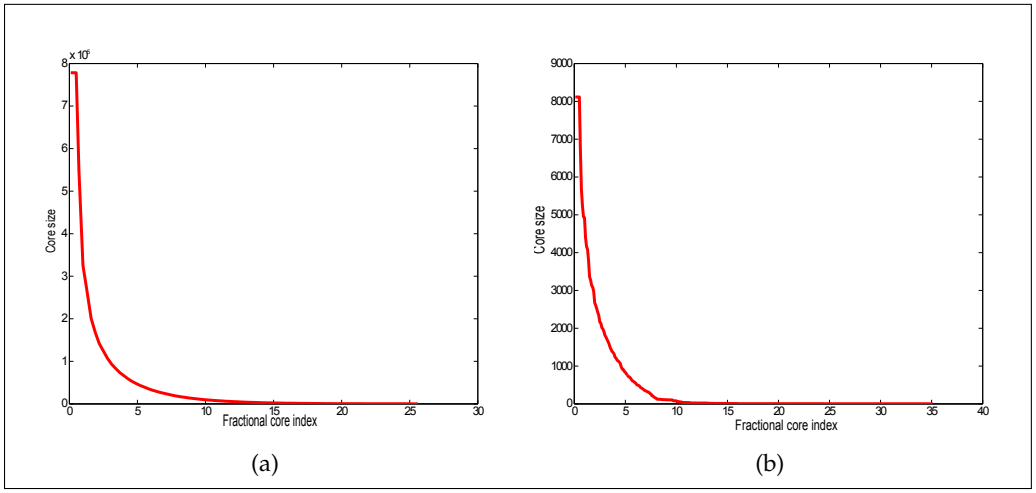


Figure 3.5: Distribution of the fractional core sizes vs core indices in the edge-weighted co-authorship graph of a. DBLP and b. ARXIV

inversely proportional to the number of co-authors per co-authored paper. Thus $\mathbf{w}(e)$ represents the "essential amount" of collaboration among authors x, y in terms of the effort committed for common publications (which is normalized in each case by the number of contributing co-authors). This implies that the best fractional k-core communities contain authors that are intensively co-authoring with others and, while the number of co-authors is not high, it follows that the share of collaborative effort is high.

In Figure 3.5 the reader can see the size distribution of the graphs in the fractional core sequence of $(H_{DBLP}, \mathbf{w})$ and $(H_{ARXIV}, \mathbf{w})$, i.e. the edge-weighted co-authorship graph of DBLP and ARXIV respectively.

For both $(H_{DBLP}, \mathbf{w})$ and $(H_{ARXIV}, \mathbf{w})$, the behavior is of similar flavor in terms of the relation of the h_i-core size and h_i. The fractional index sequence of $(H_{DBLP}, \mathbf{w})$ contains a big number of rational numbers that becomes "sparsest" as it increases, i.e., the differences between two consecutive elements is increasing, especially in the end. The 16 last terms of the fractional index sequence of $(H_{DBLP}, \mathbf{w})$ and $(H_{ARXIV}, \mathbf{w})$ are depicted in Table 3.4.

3.2.4 *Rank vs size*

For $(H_{DBLP}, \mathbf{w})$, the densest fractional core has index 149.2 and contains only two authors (Sudhakar M. Reddy, Irith Pomeranz) whose publication record indeed verifies the claims as they have co-authored 373 papers, 256 of which as the only authors! The second densest core of $(H_{DBLP}, \mathbf{w})$ is the 77.8-core that includes the additional authors: Henri Prade, Didier Dubois whose intense collaboration is verified by the number of co-authored papers (223 according to DBLP). In other words, the 77.8-core of H_{DBLP} consists of just two isolated edges. This trend con-

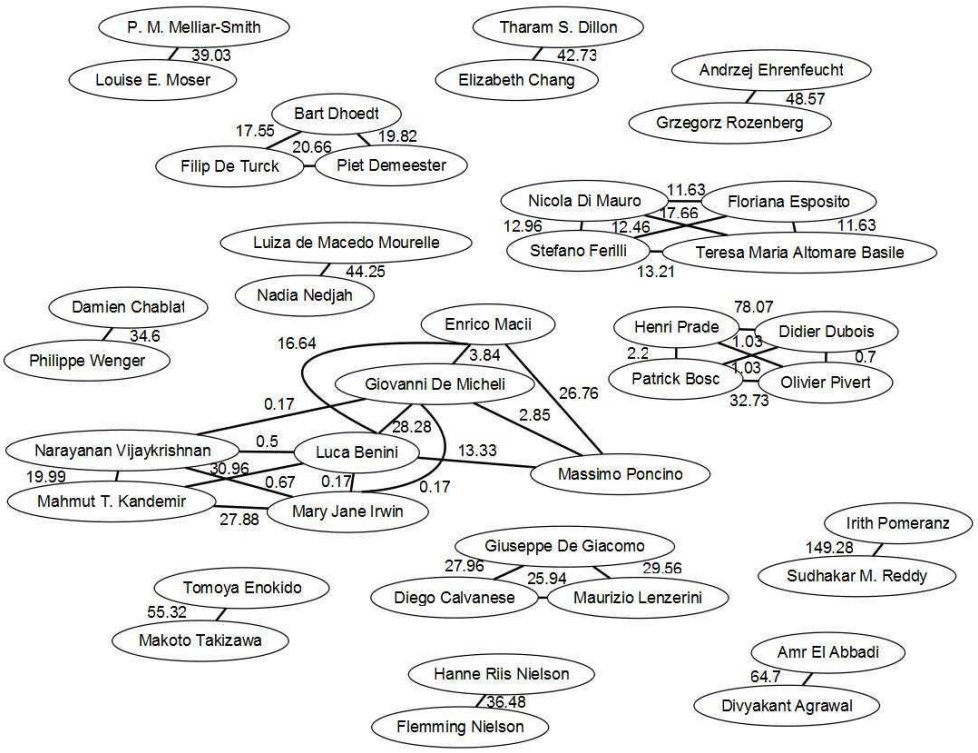


Figure 3.6: The 34.30-core of $(H_{DBLP}, \mathbf{w})$

tinues for some of the next members of the fractional core sequence until the cores become greater and thus more complex.

In the case of $(H_{ARXIV}, \mathbf{w})$, similar behavior is observed for the densest cores. However now the cores swiftly develop large connected components. The densest $(H_{ARXIV}, \mathbf{w})$ core, the 34.9-core, contains only two authors: H. Lu and C. N. Pope that have co-authored 114 papers. The second densest 24.5-core contains two more authors: Shinich Nojiri and Sergey D. Odintsov who co-authored 76 papers. Interestingly, this set of authors becomes connected in the next 27.1-core because of the insertion of Mirjam Cvetic in it who has published papers with all aforementioned authors. The 21.7-core of $(H_{ARXIV}, \mathbf{w})$ is depicted in Figure 3.4.

To amortize the effect of having tiny dense cores or dense cores of small connected components, two criteria are introduced to focus on dense cores:

- SVR (Size Versus Rank) Criterion: discarding from the core sequence of H_G all G_{h_i} for which $h_i > |V(G_{h_i})|$, i.e., the cores whose size is less than their index are not considered.

- GCVR (Giant Component Versus Rank) Criterion: discarding from the core sequence of H_G all G_{h_i} for which $h_i > g(G_{h_i})$, i.e., the cores for which the size of their giant component is less than their index are not considered.

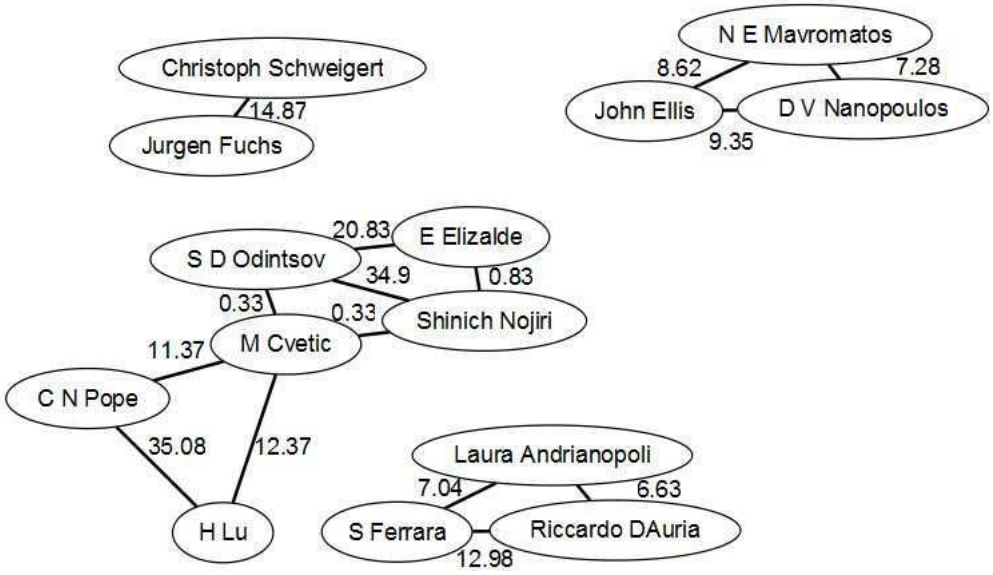


Figure 3.7: The 13.40-core of $(H_{ARXIV}, \mathbf{w})$

Both above criteria are balancing the high index with some quantity criterion on the number of authors that generate it. SVR asks that the essential degree of effort of each author (i.e. the fractional degree of each vertex) is bigger than the total number of authors in the core with whom this effort has been shared. Clearly, GCVR is at least as strict as SVR and reflects the fact that, as cores grow in size, most of their authors are accumulated on the giant component (see Section 2.3.1.2). The application of GCVR on $(H_{DBLP}, \mathbf{w})$ considers the 34.3-core (depicted in Figure 3.6): it has 39 authors while the next 35.2-core has 35 authors. The same criterion applied to $(H_{ARXIV}, \mathbf{w})$ considers the 13.4-core that has 14 authors (depicted in Figure 3.7). Notice that in both Figures 3.6 and 3.7, the graphs are still quite fragmented and, at the same moment, already big enough to reveal several collaboration communities.

The next step is to apply the GCVR criterion on H_{DBLP}. In this case, the biggest k in the fractional index sequence of H_{DBLP} for which the giant component of the k-core is bigger than k is 27.7. Indeed, the 27.7-core has 132 authors and its giant component has 42 authors, while the next index is 28.0 and the 28.0-core has size 122 and its giant component has 23 authors. The 27.7-core is depicted in Figure 3.8 (as it has 122 vertices, the names of the authors are not included).

The application of the GCVR criterion on H_{ARXIV} implies that the 12-core, that has 21 vertices, is the last one whose giant component has more vertices, that is 12 than its index. Indeed, the next index is 13.1 and the 13.1-core has 16 authors and, among them, 6 are in its giant component. The 12-core of H_{ARXIV} is depicted in Figure 3.9.

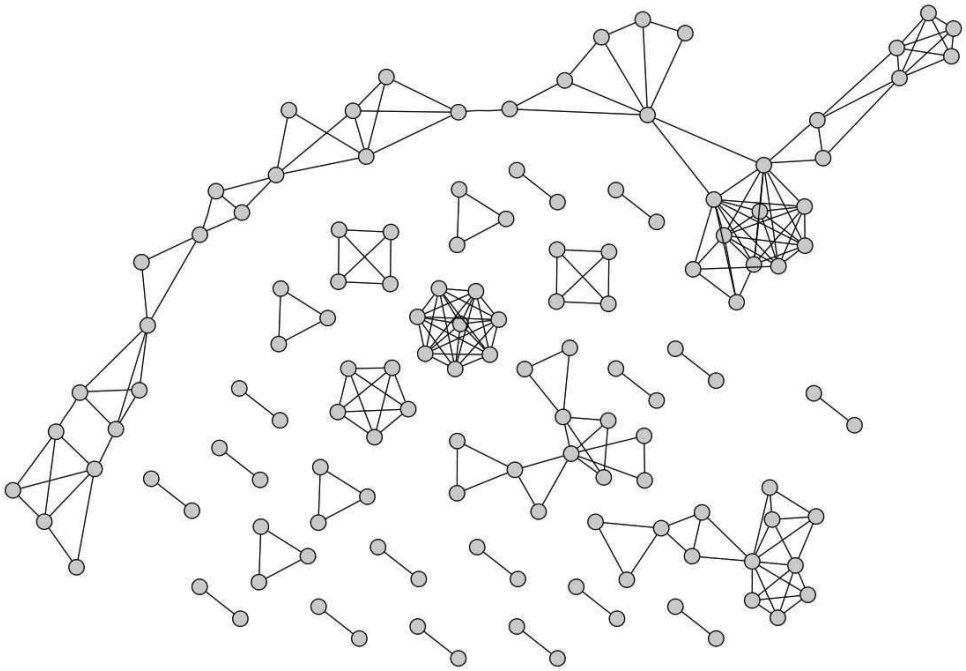


Figure 3.8: The 13.40-core of $(H_{\mathsf{DBLP}}, \mathbf{w})$

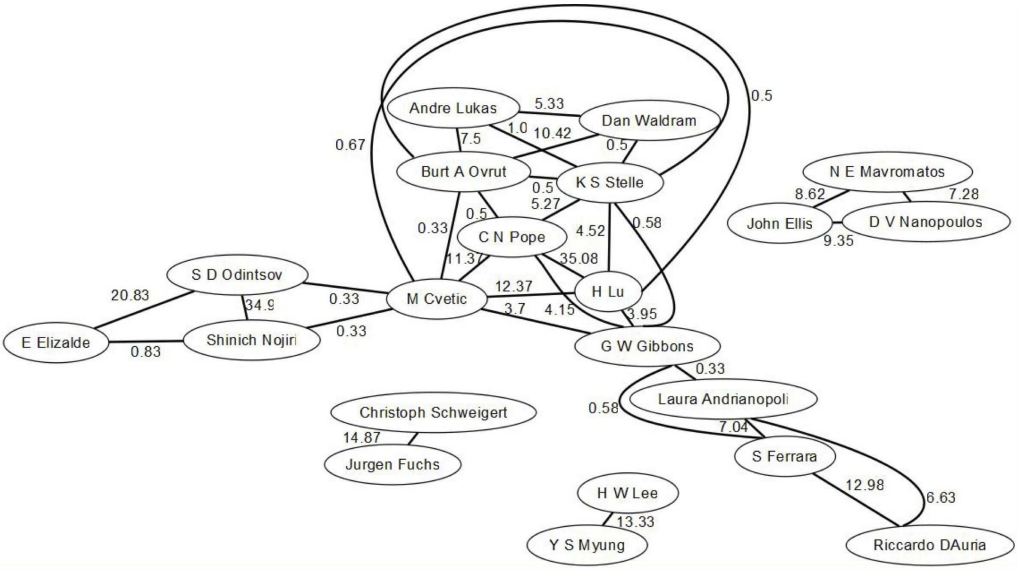


Figure 3.9: The 12-core of $(H_{\mathsf{ARXIV}}, \mathbf{w})$

	Name of author	Index
$(H_{DBLP}, \mathbf{w})$		
	C. H. Papadimitriou	20.8
	Serge Abiteboul	20.5
	Christos Faloutsos	18.7
	Gerhard Weikum	16.3
	Paul Erdós	13.9
	A. Tanenbaum	13.0

	Name of author	Index
$(H_{ARXIV}, \mathbf{w})$		
	Mirjam Cvetic	21.7
	John Ellis	14.9
	D. Nanopoulos	14.9
	C. Schweigert	13.7
	Riccardo D'Auria	13.1

Table 3.5: Ranks of selected authors in $(H_{DBLP}, \mathbf{w})$ and $(H_{ARXIV}, \mathbf{w})$.

3.2.5 *Hop-1 lists*

In Table 3.5 the index of the previous sample is depicted for selected authors of both DBLP and ARXIV, based on the fractional cores computation. It is interesting that the indices are different in this case due to the weighting scheme that favors not just a big number of publications but also repetitive co-authorship with limited number of co-authors. In this case, intensive collaboration with certain co-authors over a long series of publications increases the mutual edge weights and thus the indices in the fractional k-cores.

Assuming an author x in H_{DBLP}, it should be stressed that his/her best hop-1 co-authorship k-core (i.e. immediate co-authors) are those that have at least k co-authors in the same core.

In Table 3.6, the relevant data for fractional cores are presented for a selection of well known and seminal authors from DBLP representing their degree of collaboration with their co-authors. C. H. Papadimitriou has a top score in this measure (20.8) while having a very small but cohesive community of co-authors, with the prominent example of Michalis Yannakakis contributing an awesome weight (19.62) to the vertex fractional degree of Papadimitriou. This implies that they have co-authored many papers together (46) out of which more than 30 are co-authored by the two of them only! On the other hand, G. Weikum has a much more distributed collaboration circle in terms of co-authors that almost uniformly (except the case of Scheck, that is 7.43) contribute to his vertex fractional degree. Finally, Andrew Tanenbaum with a vertex fractional degree 13.0 has a rather small collaboration community with main collaborators Maarten van Steen (contributing a weight 4.68) and Robbert van Renesse (5.4) while the rest is uniformly distributed to the others.

In Table 3.7, the respective data for selected authors from ARXIV can be seen. There can also be found very well known names in the scientific area together with their closet collaborators. Actually in this case all authors indicated in Figure 3.7

Author	Fractional Rank	Size
C.H. Papadimirtiou	20.80	417
Michalis Yannakakis (19.62)	Erik D. Demaine (0.14)	Georg Gottlob (1.00)
Gerhard Weikum	16.30	1506
Hans-Jörg Schek (7.43)	Surajit Chaudhuri (5.05)	Yuri Breitbart (1.49)
Gautam Das (0.70)	Jeffrey F. Naughton (0.57)	Divesh Srivastava (0.53)
DanSuciu (0.50)	Rakesh Agrawal (0.48)	Gustavo Alonso (0.43)
Raghu Ramakrishnan (0.41)	Catriel Beeri (0.33)	Michael Backes (0.33)
Serge Abiteboul (0.33)	Divyakant Agrawal (0.29)	Amr El Abbadi (0.29)
Stefano Ceri (0.275)	Yannis E. Ioannidis (0.23)	Henry F. Korth (0.23)
S. Sudarshan (0.20)	Jennifer Widom (0.19)	David J. DeWitt (0.19)
Abraham Silberschatz (0.17)	David Maier (0.16)	Krithi Ramamritham (0.15)
Hector Garcia-Molina (0.14)	Christos Faloutsos (0.13)	Victor Vianu (0.13)
Edward A. Fox (0.09)	Beng Chin Ooi (0.08)	Richard Snodgrass (0.07)
Jeffrey D. Ullman (0.07)	Timos K. Sellis (0.07)	Umeshwar Dayal (0.17)
Michael J. Carey (0.14)		
Andrew Tanenbaum	13.0	4016
M. Frans Kaashoek (7.00)	Robbert van Renesse (5.40)	Maarten van Steen (4.68)
Frances M. T. Brazier (0.98)	Anne-Marie Kermarre (0.25)	Howard Jay Siegel (0.13)
Michael S. Lew (0.02)		
Paul Erdös	13.9	2678
János Pach (2.53)	Boris Aronov (0.28)	Leonard J. Schulman (0.28)
Ronald L. Graham (1.83)	Fan R. K. Chung (1.74)	Zoltán Furedi (1.58)
Noga Alon (0.50)	Endre Szemerédi (1.40)	Vojtech R′odl (1.33)
Nathan Linial (1.0)	Miklós Ajtai (0.25)	János Komlós (0.25)
László Lovász (0.33)	Shlomo Moran (0.53)	Andreas Blass (0.33)
Michael E. Saks (0.33)	Richard Pollack (0.25)	Shmuel Zaks (0.20)

Table 3.6: Fractional indices and hop-1 list for selected authors from DBLP.

Author	Fractional Rank	Size
Mirjam Cvetic H Lu (12.36) S D Odintsov (0.33)	21.7 C N Pope (11.36)	5 Shinich Nojiri (0.33)
John Ellis N E Mavromatos (8.61)	14.9 D V Nanopoulos 9.35	9
Christoph Schweigert Jurgen Fuchs (14.86)	13.7	11
Riccardo D'Auria S Ferrara (12.98)	13.1 Laura Andrianopoli (6.62)	16

Table 3.7: Fractional indices and hop-1 list for selected authors from ARXIV.

are present in both the 13.40-core and the 12-core of $(H_{ARXIV}, \mathbf{w})$. Especially in the 13.40 they appear in different connected components. Observe that, in the 12-core, Mirjam Cvetic and Riccardo D'Auria appear in the same component while this is not the case in the higher rank 13.40 core. However, the "clusters" of John Ellis, and Christoph Schweigert are already becoming disconnected in the 12-core. Observe also that S D Odintsov enters the hop-1 list of Mirjam Cvetic because of a link of relatively low weight, i.e., 0.33. However, S D Odintsov enters in the hop-1 list of Mirjam Cvetic because of his strong collaboration with Shinich Nojiri and E Elizaide (that, however is not in the hop-1 list of Mirjam Cvetic).

3.2.6 *Community-focused rankings*

In the final experiments (on undirected and undirected weighted graphs), the focus is turned on authors belonging to specific scientific communities and compare their rankings according to our fractional cores method against rankings determined using simpler measures of collaborativeness. More precisely, the names of program committee members of SIGMOD, SIGIR, and SIGKDD for the years 2009, 2010, and 2011 were extracted to obtain subsets of the database, information retrieval, and data mining community, respectively. Most of the authors could be mapped automatically to their entries in DBLP using string matching; for some a best-effort manual mapping (e.g., because of missing middle initials or nicknames in the programme committee lists) had to be performed; about a handful of authors could not be mapped with confidence (author name disambiguation issues e.g.from abbreviations) and are thus missing from the rankings. For each community, authors are ranked therein according to the following measures:

(a) *fractional index*

(b) *number of co-authors*

(c) *number of publications*

(d) *average number of co-authors per publication*

(e) *years active.*

The resulting top-10 rankings are given in Table 3.8, Table 3.9, and Table 3.10. Note that for the fractional cores method, as before, an author's fractional index is determined on the entire DBLP co-authorship graph and not only based on collaborations with authors within the same scientific community. When looking at the top-10 rankings presented, its observed that across all communities rankings according to **(b)**, **(c)**, and **(e)** are biased in favor of senior authors (e.g., Michael Stonebraker, W. Bruce Croft, and Jiawei Han) and overlap sometimes significantly. This is natural, given that authors who have been active longer, tend to have more publications, co-authored with different people at different points in time.

The rankings according to **d)**, the average number of co-authors per publication, contain for all three communities relatively junior alongside senior authors. However, it can also be seen that this is not a robust measure, bringing up authors who have published and collaborated modestly, but happen to have one publication with a large number of co-authors. Finally, the rankings according to **(a)**, the fractional cores method, seem less biased toward senior authors, bringing up a mix of prolific authors with long-lasting intensive collaborations between them (e.g., Amr El Abbadi and Divyakant Agrawal, Ophir Frieder and Abdur Chowdhury, Annalisa Appice and Donato Malerba).

(a)	(b)	(c)
Amr El Abbadi	Wei Wang	Wei Wang
Divyakant Agrawal	Hans-Peter Kriegel	Christos Faloutsos
Christian S. Jensen	Christos Faloutsos	Michael Stonebraker
Richard T. Snodgrass	Divyakant Agrawal	Michael J. Carey
Sourav S. Bhowmick	Elke A. Rundensteiner	Wolfgang Nejdl
Beng Chin Ooi	Kian-Lee Tan	Stefano Ceri
Kian-Lee Tan	Amr El Abbadi	Christian S. Jensen
Pierangela Samarati	Christian S. Jensen	Raghu Ramakrishnan
Sabrina De Capitani di Vimercati	Ming-Syan Chen	Jian Pei
Mong-Li Lee	Richard T. Snodgrass	Beng Chin Ooi

(d)	(e)
Michael Stonebraker	Nesime Tatbul
David B. Lomet	Anastasia Ailamaki
Theo Häsrder	Laura M. Haas
Philip A. Bernstein	Mitch Cherniack
Hans-Peter Kriegel	John McPherson
Michael Hatzopoulos	Brian Cooper
Carlo Zaniolo	Daniel J. Abadi
Umeshwar Dayal	Jayavel Shanmugasundaram
Stefano Ceri	Tim Kraska
Meral Ozsoyoglu	Fatma Ozcan

Table 3.8: Database community ranking. Labels *a*, *b*, *c*, *d* and *e* indicate the rankings defined in 3.2.6.

(a)	(b)	(c)
Ee-Peng Lim	Lei Zhang	Lei Zhang
Paolo Boldi	Jun Wang	Jun Wang
Jie Lu	Gerhard Weikum	Yi Zhang
Steven M. Beitzel	Hsinchun Chen	Tao Li
Abdur Chowdhury	Tao Li	Qiang Yang
Ophir Frieder	Wei-Ying Ma	Wei-Ying Ma
Juan M. Fernández-Luna	Qiang Yang	Jun Xu
Juan F. Huete	C. Lee Giles	Gerhard Weikum
Wei-Ying Ma	Lee Giles	Hsinchun Chen
Yong Yu	Ricardo A. Baeza-Yates	Yong Yu

(d)	(e)
Michael Lesk	Michael Taylor
Erich J. Neuhold	Gerald Benoit
Jun-ichi Tsujii	Yifen Huang
W. Bruce Croft	Claus-Peter Klas
Fredric C. Gey	Mark Greenwood
Donald H. Kraft	Yantao Zheng
Jaime G. Carbonell	Maria M. Nikolaidou
David Lewis	Jinhui Tang
William R. Hersh	Jayavel Shanmugasundaram
Nicholas J. Belkin	David Smith

Table 3.9: Information retrieval community ranking. Labels *a*, *b*, *c*, *d* and *e* indicate the rankings defined in 3.2.6.

(a)	(b)	(c)
Floriana Esposito	Jiawei Han	Jiawei Han
Ee-Peng Lim	Christos Faloutsos	Christos Faloutsos
Annalisa Appice	Alok N. Choudhary	Gang Wang
Donato Malerba	Alberto Del Bimbo	Alok N. Choudhary
Charu C. Aggarwal	C. Lee Giles	C. Lee Giles
Alok N. Choudhary	Gonzalo Navarro	Jian Pei
Diane J. Cook	Ee-Peng Lim	Bing Liu
Alberto Del Bimbo	Jeffrey Xu Yu	Jeffrey Xu Yu
Jeffrey Xu Yu	Floriana Esposito	Aoying Zhou
Carlo Zaniolo	Carlo Zaniolo	Ee-Peng Lim

(d)	(e)
Andrzej Skowron	Jonathan Chang
Carlo Zaniolo	Jeffrey Yu
Christos Faloutsos	Byron J. Gao
Heikki Mannila	Jennifer Dy
Daniel Barbara	Edwin V. Bonilla
Dennis Shasha	Gui-Rong Xue
Alberto Del Bimbo	Ashok Savasere
Foto N. Afrati	Benoit Huet
David Poole	Jiangtao Ren
C. Lee Giles	Dou Shen

Table 3.10: Data mining community ranking. Labels *a*, *b*, *c*, *d* and *e* indicate the rankings defined in 3.2.6.

3.2.7 *Core Decomposition Forest on* DBLP *and* ARXIV

In this section, the relation between the core structure of a graph and the connected components of its cores is examined with Core Decomposition Forest as defined in section 2.3.1.2 The following Core Decomposition Forests were computed:

- $\mathbf{DF}(\mathcal{G}(H^*_{\mathsf{DBLP}}))$,
- $\mathbf{DF}(\mathcal{G}(H_{\mathsf{ARXIV}}))$,
- $\mathbf{DF}(\mathcal{G}(H_{\mathsf{DBLP}}), \mathbf{w})$, and
- $\mathbf{DF}(\mathcal{G}(H_{\mathsf{ARXIV}}), \mathbf{w})$.

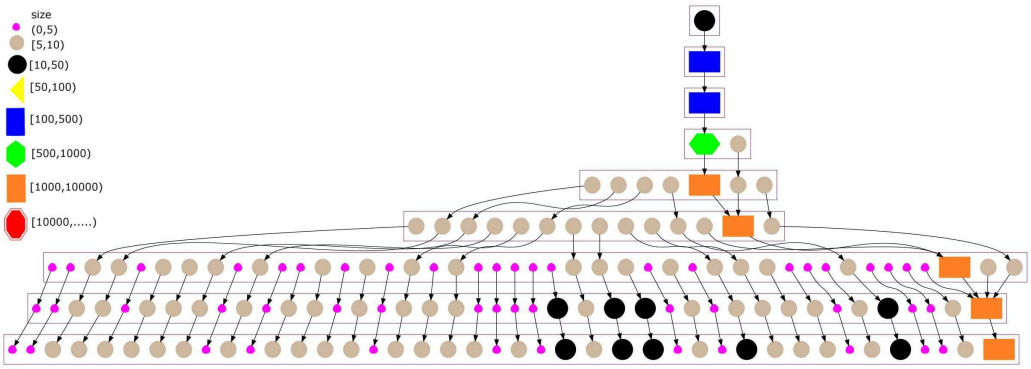


Figure 3.11: The Core Decomposition Forest of the core sequence of H_{ARXIV}

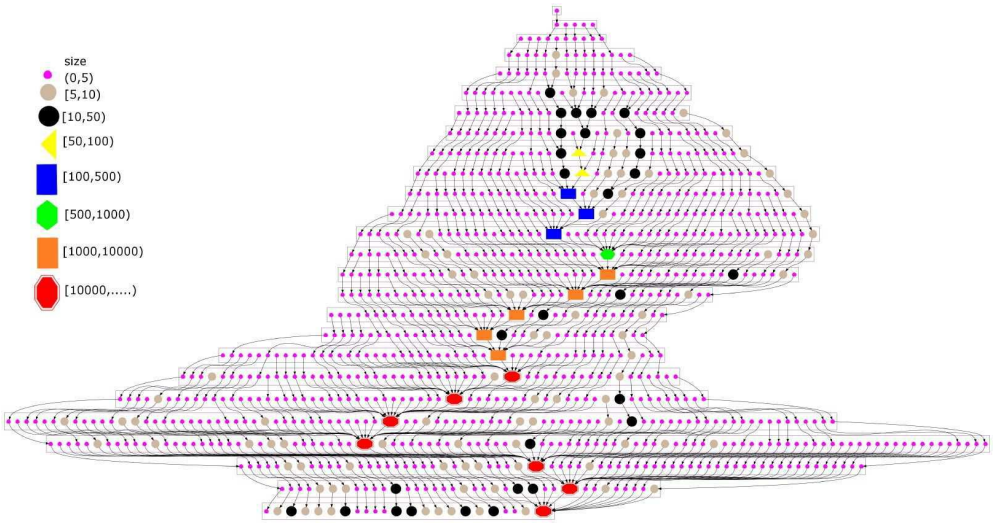


Figure 3.12: The Core Decomposition Forest of the core sequence of $(H_{DBLP}, \mathbf{w})$

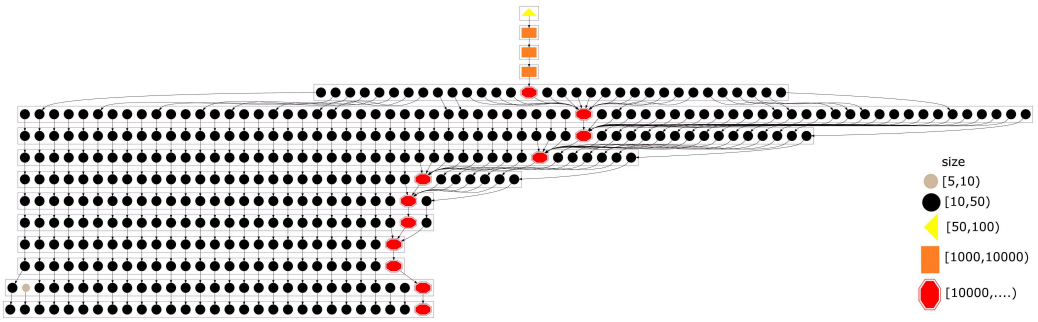


Figure 3.10: The Core Decomposition Forest of the core sequence of H^*_{DBLP}

The results for the case of H^*_{DBLP} and H_{ARXIV} are depicted in Figures 3.10 and 3.11 respectively, while the results for the cases of $(H_{DBLP}, \mathbf{w})$ and $(H_{ARXIV}, \mathbf{w})$ are depicted in Figures 3.12 and 3.13 respectively. It should be pointed out that these figures depict only an approximation of these trees as their sizes are too

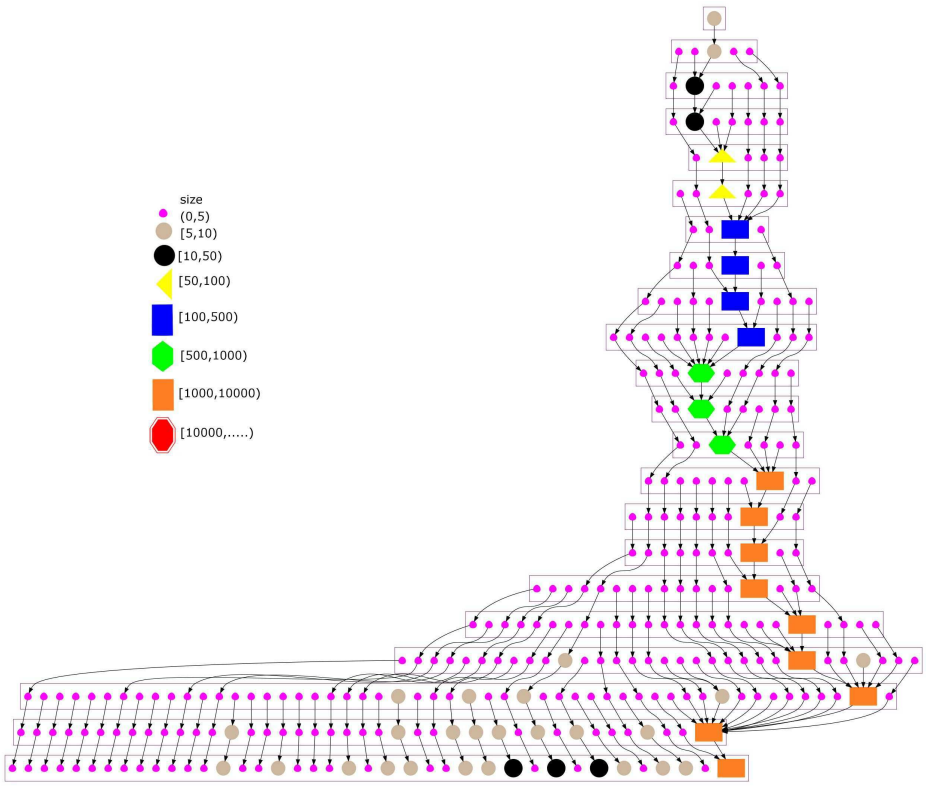


Figure 3.13: The Core Decomposition Forest of the core sequence of $(H_{ARXIV}, \mathbf{w})$.

big to fit in a visible way. To facilitate the visualization of the core decomposition forests we applied the following relaxations parameterized by α and n:

(1) suppress in $\mathcal{G}$ consecutive terms that are the same (i.e. if two consecutive core have the same corresponding components -most probable one component in the higher valued cores- then one of them is omitted),

(2) consider only the members of the resulting sequence that are indexed by multiples of α, and

(3) in the core decomposition forest of the (fractional) core sequence remaining after relaxations (1) and (2), exclude all subtrees that do not have ancestors after the n-th core of this sequence.

For the visualization of the core decomposition forest for H^*_{DBLP} and H_{ARXIV} we applied steps (1)–(3) for $\alpha = 1, n = 8$ and $\alpha = 1, n = 1$ respectively. For the visualization of the core decomposition forests for $(H_{DBLP}, \mathbf{w})$ and $(H_{ARXIV}, \mathbf{w})$, we only applied the relaxation steps (2) and (3) (relaxation step (1) is unnecessary on fractional sequences) for $\alpha = 5$, $n = 10$ and $\alpha = 5$ and $n = 8$ respectively. In each case the values of the parameter n and α have been chosen as to optimize the visualization of the corresponding datasets.

As we see in Figure 3.10, the H^*_{DBLP} dataset presents the following behavior in terms of connected components: There is clearly a giant component that evolves as k increases and survives until the last 15-core. It is interesting that many connected components survive until core index 11, thus the H^*_{DBLP} dataset is rather robust under degeneracy.

In Figure 3.11 we see the robustness of the cores of the ARXIV co-authorship graph under degeneracy. Again there is a giant component that evolves as k increases and survives until the last 9-core. It is interesting that many connected components survive until core index 5.

As for the edge-weighted graphs there is a remarkable behavior. In Figure 3.12 we see the evolution of the connected components of the $(H_{DBLP}, \mathbf{w})$ graph. In this case the graph is much more robust as the steps of degeneracy are fractional while we see again a giant component that splits into other components that merge before they shrink again.

In Figure 3.13 the evolution of the connected components of the $(H_{ARXIV}, \mathbf{w})$ graph is depicted. In this case the graph is much less robust as the number of connected components is swiftly shrinking and only a few - together with the giant component survive until the highest index fractional core.

3.3 REAL WORLD APPLICATION

In this section, a demo application is presented which was developed under the context of evaluating communities with the fractional core framework. The demonstrated system leverages the notion of fractional cores to rank and filter vertices in the network and in the hop-1 coauthoring community to create connections between them. It is available at the following URL:

`http://www.graphdegeneracy.org/fcores/dblp/`

It is stressed that the visualization of the hop-1 coauthoring community that is proposed here conveys much more meaning and information than the simple co-author graphs presented in other approaches such as the *Co-author Graph* at `http://academic.research.microsoft.com/` or the respective in Arnet Miner at `http://arnetminer.org/`). More specifically, in the case presented here the visualization depicts the neighbors of authors in the DBLP co-authorship network that are *inside* the highest index core they belong to.

3.3.1 *System Architecture*

Figure 3.14 depicts the overall architecture of the system, which is the subject of this section. For our demonstration, we use a co-authorship network derived from the DBLP bibliographic dataset[1]

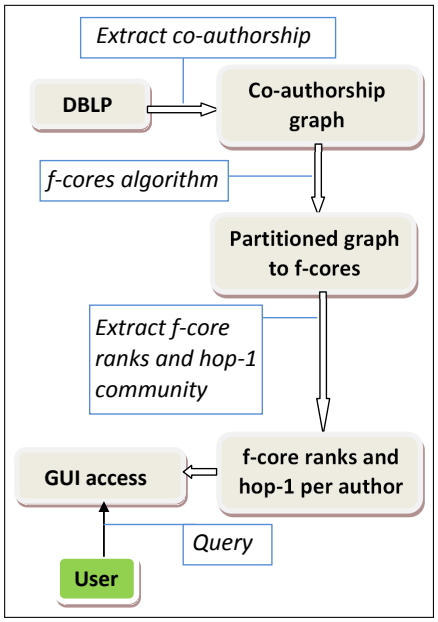


Figure 3.14: System Architecture

Going from top to bottom in Figure 3.14, first the bibliographic dataset is converted into an undirected edge-weighted co-authorship graph as described in Section 2.3.2.

Next, as a one-time pre-computation, the fractional core sequence of this graph is iteratively computed. That is, the *Trim* procedure is repeatedly invoked removing more and more vertices from the graph and thus implicitly partition it. When a vertex is removed from the graph, its fractional core index and its hop-1 neighborhood (consisting of immediate neighbors in the remaining graph) are recorded. This information is stored in a relational database, so that it can be retrieved efficiently by our system at runtime. Note that, despite of the large scale of the DBLP co-authorship network, we were able to run the entire computation on a commodity notebook equipped with a 2-core CPU and 2 GB of main memory – an indication that fractional cores are computationally lightweight and can be applied to large-scale collaboration networks.

Users interact with the system through a web-based GUI that be accessed at the aforementioned URL. Its functionality, described in more detail in the following Section 3.3.2, includes the display of rankings for authors by their fractional core

1 Freely available at http://dblp.uni-trier.de.

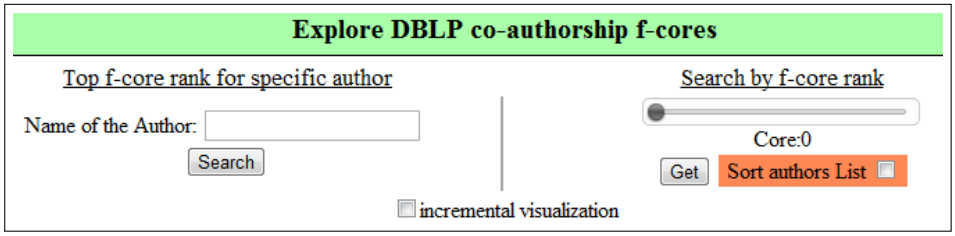


Figure 3.15: Main input interface.

index but also an interactive visualization of an author's neighborhood, displaying only those co-authors with an equal or higher fractional core index to provide a clutter-free view on the collaboration network.

3.3.2 *Demonstration*

The main user interface of the system is shown in Figure 3.15, offering users two options to interact. On the right, a slider allows users to select a threshold on the fractional core index and browse through qualifying authors. On the left, an input box allows users to search for a specific author by name – both exact match and fuzzy match are supported here. Either way, once an author has been selected, the system shows the fractional core index of the author together with a visualization of the surrounding hop-1 neighborhood, i.e., the author's closest co-authors who have at least an equally high fractional core index. Figure 3.16 shows an example hop-1 neighborhood (in this case for Michalis Vazirgiannis). From the visualization, users can see author's fractional core index (here 10.6) and his "tightest" collaborators each linked with a weighted edge indicating the "strength" of their partnership.

On this initial star-like graph the user can explore the surrounding authors by clicking on the "Find" function that appears when the mouse hovers over the author's vertex. If the "incremental visualization" option is not activated the result of the "Find" function is a fresh star-like graph centered around the newly selected author. Otherwise, if the option is activated, the user gets to explore the intersection of the two authors' (the original and the newly selected one) strongest collaborators. This "incremental" visualization can continue for multiple steps to reveal an increasingly broader and possibly highly interconnected community around an author. Figure 3.17 demonstrates this functionality building on the earlier example. Here, the star-like graph from Figure 3.16 was expanded by selecting Timos K. Sellis from the surrounding hop-1 neighborhood of collaborators. Interestingly, as can be observed from the figure, there is an overlap between the "tightest" collaborators of the two authors, revealing a much richer view on the community that they belong to.

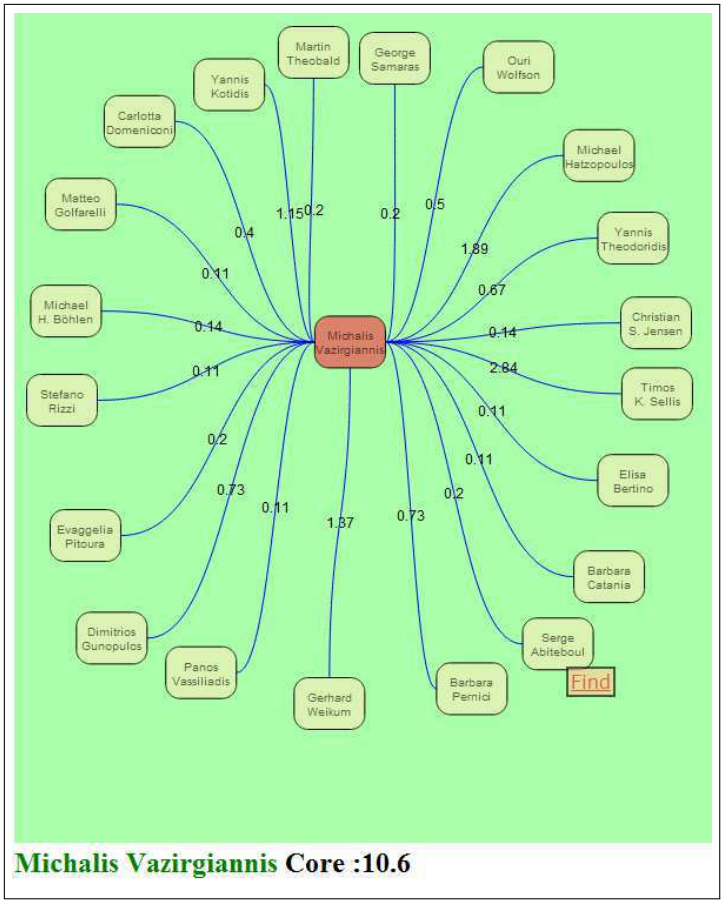


Figure 3.16: Output for the query "Vazirgiannis".

When the user selects a threshold for the fractional core index with the slider shown in Figure 3.15, using the "incremental", function more than one author can be selected and the user can see how different communities are formed within the same fractional core from the interconnection of the authors within that core. Figure 3.18 shows a small example based on authors from the 19.6-core – it is interesting to see the selected authors belong to two disconnected communities.

3.3.3 *Application Scenarios*

What are other application scenarios where a visual exploration of collaboration networks, as provided by the developed system, can provide useful insights?

Bibliographic data and measures derived therefrom (e.g., the H-Index [45] and G-Index [34]) nowadays play a big role in *academic hiring*. Measures like the aforementioned ones have focused on citations and sought to capture a candidate's scientific impact.

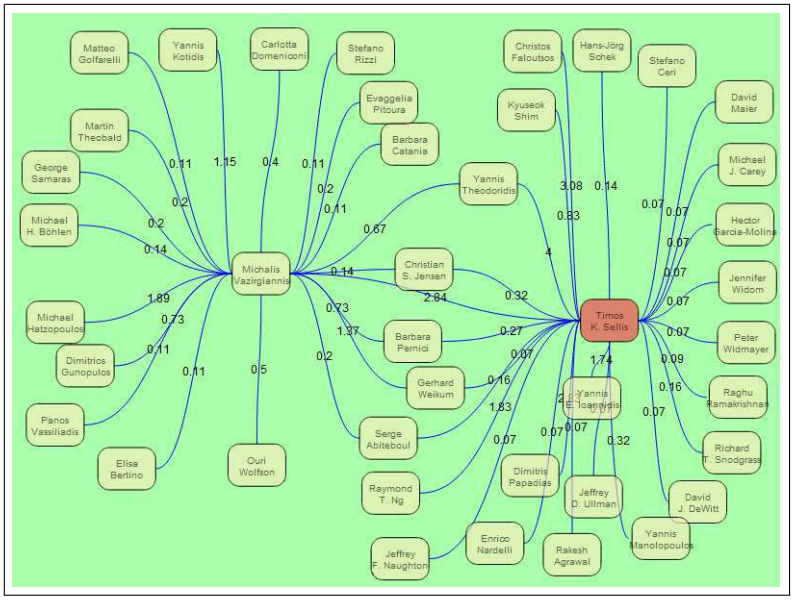


Figure 3.17: Example of browsing the hop-1 neighborhood with the "incremental" function activated.

As networking skills and the ability to work in teams become more important, even in academia, our system provides the means to inspect a candidate in these regards. Relying on the notion of fractional cores, our system automatically zooms in on the connections to peers and more senior scientists and, as a consequence, provides a clutter-free view on the candidate's collaborations. Thus, collaborations that are ephemeral or with less prolific individuals (e.g., students who left academia after graduating) are automatically filtered out thanks to the use of fractional cores.

Similar applications for *human resources* and *team building* could be found in industry, where collaboration networks can be derived from other datasets, e.g.,

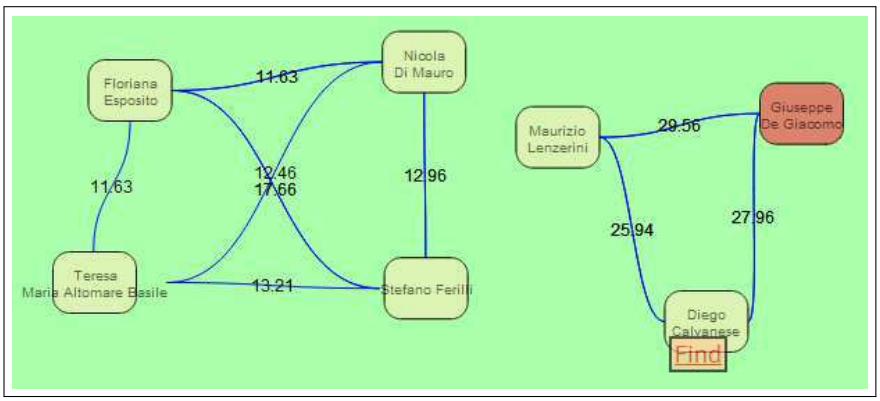


Figure 3.18: Example of browsing the 19.6-core with the "incremental" function activated.

patents (based on the listed inventor names), e-mails (based on recipient lists) or records about who worked together on which projects in the past. A visual exploration of collaboration networks can also turn out insightful for other collaboration networks, for example, those of actors who played in the same movie (e.g., derived from a movie database such as IMDB[2]) or politicians who co-signed a petition or participated in other joint activities.

The ideas behind this system are not limited to collaboration networks but can be applied to any dataset from which undirected weighted graphs can be derived in a sensible manner. This includes various facets of social networks with ties that signify, for instance, joint interests (based on group memberships) or reciprocal activities. Also in other contexts, the ability to visually explore large-scale datasets becomes more important in face of the ongoing data deluge. Some of the now available data has natural interpretations as graphs, for example, RDF datasets like those connected in the linked data cloud[3]. It is expected that a visual exploration of such datasets can greatly profit from the use of fractional cores that, as in the applications above, help to get a clutter-free view on the data that focuses on essential highly-connected data items.

3.4 D-CORES

This section is dedicated for presenting the experiments performed by applying the D-core structures and algorithms on real-world and artificial data sets. As the artificial data are produced by an algorithmic process, a description on that process will be made before presenting the real-world data.

3.4.1 *Directed Graph Degeneracy for Scale-Free Graphs*

Real world web graphs have been found to display scale-free characteristics[9, 10, 51] evident by the power law degree distribution. Here are also explored author citation graphs which share the same properties (as it can been seen in their degree distributions). Scale-free graphs are frequently modeled by the combination of growth with preferential attachment. There have been many variations in this modeling both for directed and undirected cases but the main idea is that the graph grows one vertex at the time and edges are added (between vertices that may be new or old). The key idea in the preferential attachment scheme is that the probability of taking an edge is proportional to the respective degrees of its endpoints. This intuitively matches with the mechanism of the evolution of both web graphs and citations graphs of authors (i.e. a "popular" page is more likely

2 http://www.imdb.com
3 http://linkeddata.org

to get in-links and a "famous" author is more likely to get a citation from a new page/paper following the "rich get richer" of the preferential attachment process).

As the scale-free model seems to approximate the graphs examined, it has been chosen for evaluation with the D-core computation procedure to see if the results are similar for both various parameters and as well parameters that produce graphs with approximately similar degree distributions with the real world graphs.

3.4.1.1 *Preliminaries for preferential attachments*

Barabasi and Albert in [9] were the first to introduce a scale-free model for undirected graphs. In that model, the graph is generated with a small number of initial vertices m_0 and grows by adding each time a new vertex with $m(\leqslant m_0)$ edges from the new vertex to the old ones. Preferential attachment is introduced in the selection of the old nodes; the probability a vertex i depends on the degree of that vertex, so that $\Pi(k_i) = k_i / \sum_j k_j$ where k_i the degree of the vertex. The Barabasi-Albert model was examined in more detail by Bollobás et. al in [17] and in[78] where a detailed model called *Linearized Chord Diagram* (LCD) was designed. This applies to directed and undirected graphs as well; a parameter m is used and if $m = 1$ then at each step t a new vertex v_t is added to a given graph $G_1^{(t-1)}$ with a single edge between v_t and v_i where i is chosen randomly with

$$\mathbb{P}(i = s) = \begin{cases} \frac{deg_{G_1^{(t-1)}}(v_s)}{2t-1} & 1 \leqslant s \leqslant t-1, \\ \frac{1}{2t-1} & s = t \end{cases} \tag{3.1}$$

For $m > 1$, m edges are added from v_t to v_i one at a time, each time counting the previous edges in the total degree of each v_i.

In [31] and [29] a variation on the Barabasi-Albert model is introduced, where a constant parameter α represets the "initial attractiveness" of a node. Here the old vertices are chosen based on a probability proportional to their degree plus the "initial attractiveness". Thus the selection probability, defined in detail in [19], is:

$$\mathbb{P}(i = s) = \begin{cases} \frac{deg_{G_1^{(t-1)}}(v_s)+\alpha}{2t-1} & 1 \leqslant s \leqslant t-1, \\ \frac{\alpha}{(\alpha+1)t-1} & s = t \end{cases} \tag{3.2}$$

The constant parameter here is important as it introduces a mixture of uniform and preferential attachment behavior (where if $\alpha = 1$ we have only preferential attachment). This model is also important as it resembles the directed one we utilized for our experiments. Another model that also introduced a mixture of uniform and preferential attachment was in the work of Cooper and Frieze [25].

Here instead, the uniformity was defined explicitly by defining additional parameters that would determine the probability of selecting a uniform or preferential attachment model. Furthermore, they define two different steps : a) one of growth and b) one that chooses to connect two old vertices together with a new edge. This model is also important as it gives the opportunity to control the density of a graph by controlling the probability between the two steps.

As these models seemed to be better suited for models of undirected graphs, the model introduced by Bollobás, Borgs, Chayes, and Riordan in [16] was chosen. This model, as seen bellow in the description, has an initial preference parameter for both the in- and out-degrees, while also following the general idea between different steps as in the Cooper-Frieze model. Following the description of that model is offered (taken from [16]):

> We consider a graph which grows by adding single edges at discrete time steps. At each such step, a vertex may or may not also be added. For simplicity we allow multiple edges and loops. More precisely, let α, β, γ, δ_{in}, and δ_{out} be non-negative real numbers, with $\alpha+\beta+\gamma=1$. Let G_0 be any fixed initial graph, for example a single vertex without edges, and let t_0 be the number of edges of G_0. (Depending on the parameters, we may have to assume $t_0 \geqslant 1$ for the first few steps of our process to make sense.) We set $G(t_0) = G_0$, so that at time t the graph $G(t)$ has exactly t edges, and a random number $n(t)$ of vertices. In what follows, to choose a vertex v of $G(t)$ according to $d_{out}+\delta_{out}$, means to choose v so that $Pr(v=v_i)$ is proportional to $d_{out}(v_i)+\delta_{out}$, i.e., so that $Pr(v=v_i) = (d_{out}(v_i)+\delta_{out})/(t+\delta_{out}n(t))$. To choose v according to $d_{in}+\delta_{in}$, means to choose v so that $Pr(v=v_i) = (d_{in}(v_i)+\delta_{in})/(t+\delta_{in}n(t))$, where all degrees are measured in $G(t)$.
>
> For $t \geqslant t_0$ we form $G(t+1)$ from $G(t)$ according the following rules:
>
> (A) With probability α, add a new vertex v together with an edge from v to an existing vertex w, where w is chosen according to $d_{in}+\delta_{in}$.
>
> (B) With probability β, add an edge from an existing vertex v to an existing vertex w, where v and w are chosen independently, v according to $d_{out}+\delta_{out}$ and w according to $d_{in}+\delta_{in}$.
>
> (C) With probability γ, add a new vertex w and an edge from an existing vertex v to w, where v is chosen according to $d_{out}+\delta_{out}$.
>
> The probabilities α, β, and γ clearly should add up to one. To avoid trivialities, we will also assume that $\alpha+\gamma>0$. When considering the web graph, we take $\delta_{out}=0$; the motivation is that vertices added under step (C) correspond to web pages which purely provide content

– such pages never change, are born without out-links and remain without out-links. Vertices added under step (A) correspond to usual pages, to which links may be later added. While mathematically it seems natural to take $\delta_{in} = 0$ in addition to $\delta_{out} = 0$, this gives a model in which every page not in G_0 has either no in-links or no out-links, which is rather unrealistic and uninteresting! A non-zero value of δ_{in} corresponds to insisting that a page is not considered part of the web until something points to it, typically one of the big search engines. It is natural to consider these edges from search engines separately from the rest of the graph, as they are of a rather different nature; for the same reason, it is natural not to insist that δ_{in} is an integer. We include the parameter δ_{out} to make the model symmetric with respect to reversing the directions of edges (swapping α with γ and δ_{in} with δ_{out}), and because we expect the model to be applicable in contexts other than that of the web graph.

The choice for this model was based both on the sophistication it displayed and the ability to produce graphs with behavior, in the degree distribution, very similar to the real datasets explored (see next sub-section).

3.4.1.2 *Generating preferential attachment graphs*

A set of graphs was created by adopting the preferential attachment model according to [16] (see the previous sub-section) for various parameters. Here are discussed the findings on this model for a set of 4 different parameters:

1. $\alpha = 0.018, \beta = 0.102, \gamma = 0.88, \delta_{in} = 1, \delta_{out} = 2$
2. $\alpha = 0.018, \beta = 0.102, \gamma = 0.88, \delta_{in} = 5, \delta_{out} = 1$
3. $\alpha = 0.102, \beta = 0.238, \gamma = 0.66, \delta_{in} = 1, \delta_{out} = 3$
4. $\alpha = 0.001, \beta = 0.009, \gamma = 0.99, \delta_{in} = 1, \delta_{out} = 1$

The size off the graph is 16500 nodes so that it will approximate the number of nodes that have in/out-degree of at least 1 (in relevance to the DBLP citation graph described in section 3.4.2). The reader can see the distributions of resulting graphs in Figure 3.19a in the same order from left to right and top to bottom. It is clear that all these graphs are scale free. We ran the defined algorithms and metrics and, in what follows, we report on their expressive power and features.

3.4.1.3 *D-core matrices for the synthetic data*

Following the same sequence of parameters as before, the findings on the created datasets are discussed. Firstly, the meaning of the parameters starting with the

γ is explained, the parameter that controls the density of the network. Parameters α and β control the out- and in-degree behavior respectively while δ_{in} and δ_{out} represent the aforementioned"initial preference" for the respective in and out degrees.

For the first two datasets the same values were chosen for α,β, and γ so that it can be compared on how the other two affect the results. The value of γ was chosen, experimentally, to produce an "average" density. Given the fact that the α parameter is lower than β a more extrovert behavior is expected but we this is expected to change for the second dataset as the δ_{in} parameter is a lot larger than the δ_{out}. These expectations are confirmed by the D-core matrix behavior as seen in 3.19b. It is clearly visible that the ICI angle changes when the δ_{out} increases and the ICI line (in green) moves closer to the diagonal (in dark gray).

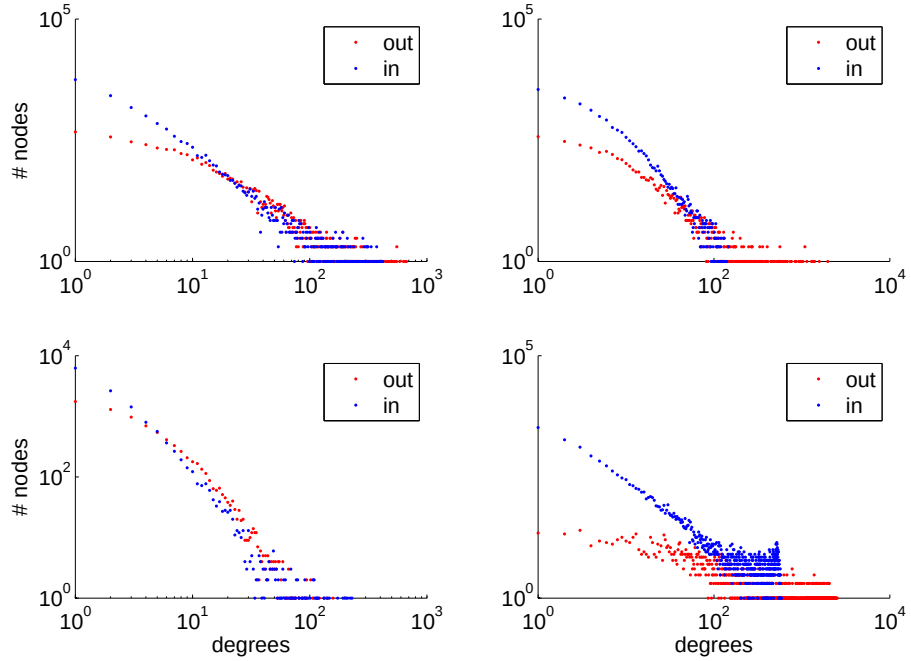
The next two datasets demonstrate how the γ parameter affects the "extend" of the D-cores. Since it is closely correlated to the density of a graph, it is expected that the degeneracy would be affected accordingly. This would mean that, for a low value of γ, one would get graphs that would produce only low-degeneracy D-cores and for a high value the opposite. This is also confirmed by the results. As the reader can see in the two D-core matrices in the bottom part of Figure 3.19b, the graph degrades really fast for a γ value of 0.66. On the other hand, when a value of 0.99, is chosen it can be easily seen that the resulting graph is much more robust. This is evident by the high numbers for in- and out-degrees that the graph survives in the D-core matrix.

3.4.1.4 *Comparison to real world Data*

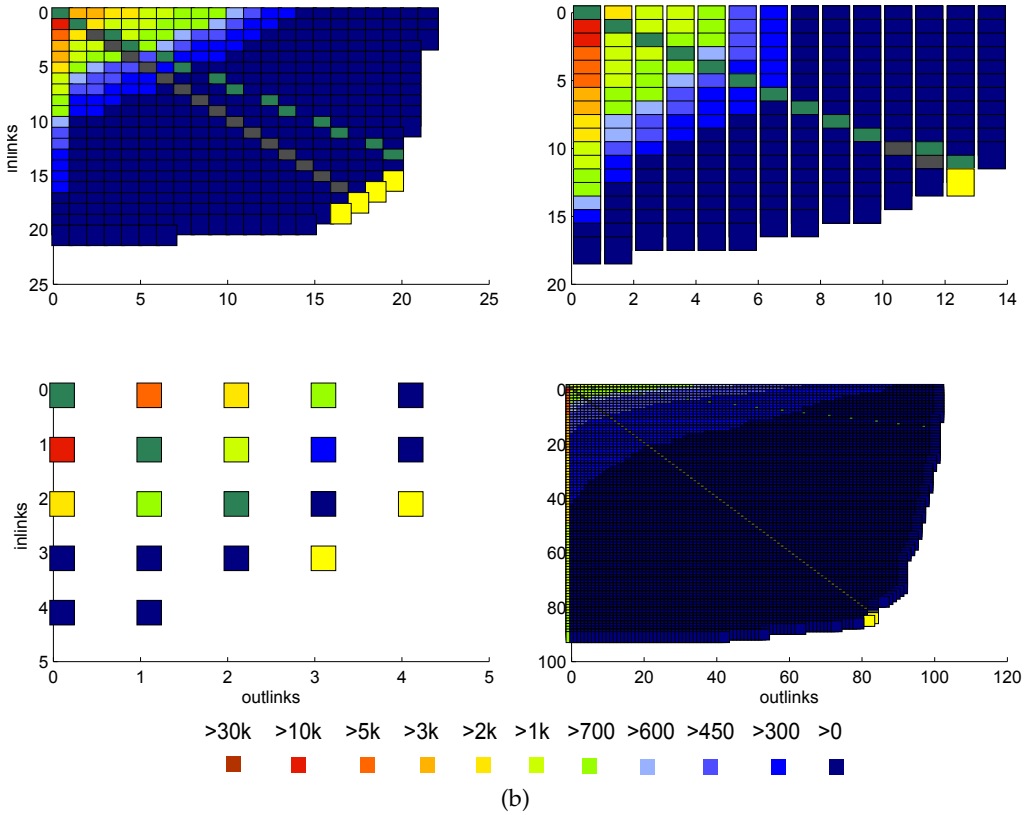
In this section, parameters were chosen so that to produce a graph with degree distributions similar to a real world dataset and verify this via a comparison to the DBLP data. For this reason the following parameters were chosen experimentally in order to approximate the DBLP digraph : $\alpha = 0.011$, $\beta = 0.031$, $\gamma = 0.958$, $\delta_{in} = 2$, and $\delta_{out} = 5$. Evidence of the approximation can be seen from the comparison of the in/out degree distributions in Figure 3.20.

In Figure 3.21 one can see that the behavior is quite similar to the previous one. The single interesting difference is how the size of the D-cores drops. On the synthetic graph case we see a dramatic drop indicating that the inner structure is less connected.

Looking at the CDF forest comparison in Figure 3.22 - excluding the small SCCs in the initial cores of the DBLP digraph - the two figures look similar as in both cases there is a giant component that survives robust until the end. Again there are some insignificant differences mostly on the rate at which the size of giant SCC drops.



(a)



(b)

Figure 3.19: a. Distributions for 4 different parameter sets on the adopted model.**A.Top left**: $\alpha = 0.018$,$\beta = 0.102$, $\gamma = 0.88$, $\delta_{in} = 1$, $\delta_{out} = 2$ **B.Top right**: $\alpha = 0.018$, $\beta = 0.102$, $\gamma = 0.88$, $\delta_{in} = 5$, $\delta_{out} = 1$ **C.Bottom left**: $\alpha = 0.102$, $\beta = 0.238$, $\gamma = 0.66$, $\delta_{in} = 1$, $\delta_{out} = 3$ **D.Bottom right**: $\alpha = 0.001$, $\beta = 0.009$, $\gamma = 0.99$, $\delta_{in} = 1$, $\delta_{out} = 1$.
b. D-core matrices for 4 different parameter sets on the adopted model. **A.Top left**: $\alpha = 0.018$,$\beta = 0.102$, $\gamma = 0.88$, $\delta_{in} = 1$, $\delta_{out} = 2$. **B.Top right**: $\alpha = 0.018$,$\beta = 0.102$, $\gamma = 0.88$, $\delta_{in} = 5$, $\delta_{out} = 1$. **C.Bottom left**: $\alpha = 0.102$, $\beta = 0.238$, $\gamma = 0.66$, $\delta_{in} = 1$, $\delta_{out} = 3$. **D.Bottom right**: $\alpha = 0.001$, $\beta = 0.009$, $\gamma = 0.99$, $\delta_{in} = 1$, $\delta_{out} = 1$.

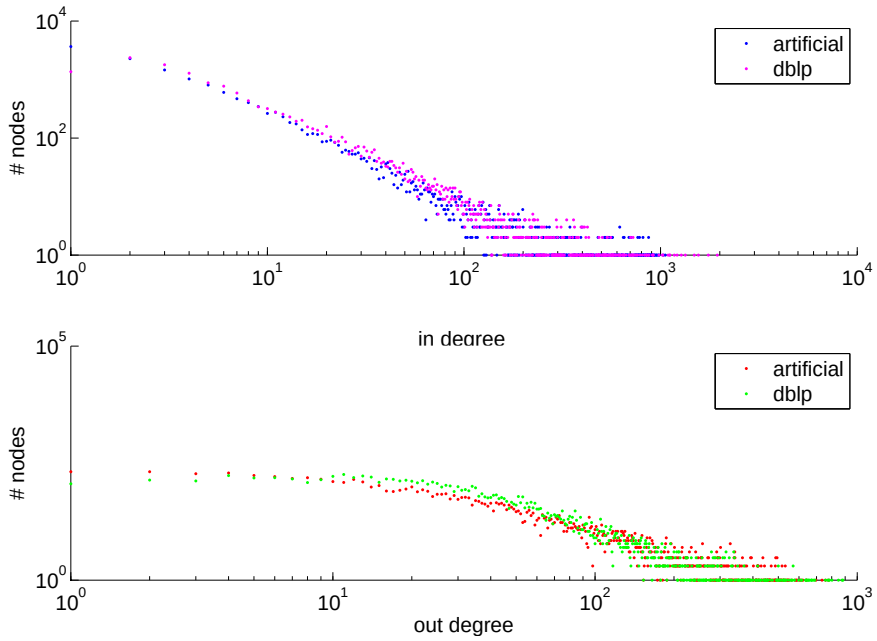


Figure 3.20: Comparison of the distributions for the in/out degrees between the chosen parameters ($\alpha = 0.011, \beta = 0.031, \gamma = 0.958, \delta_{in} = 2, \delta_{out} = 5$) and the DBLP graph.

In conclusion the synthetic digraph seems to approximate quite well the DBLP graph with regards to the D-core behavior. This is important as it could be possible to predict the D-core metrics of a real world graph of immense scale simply by producing a down-scaled 'miniature' of it by its parameters.

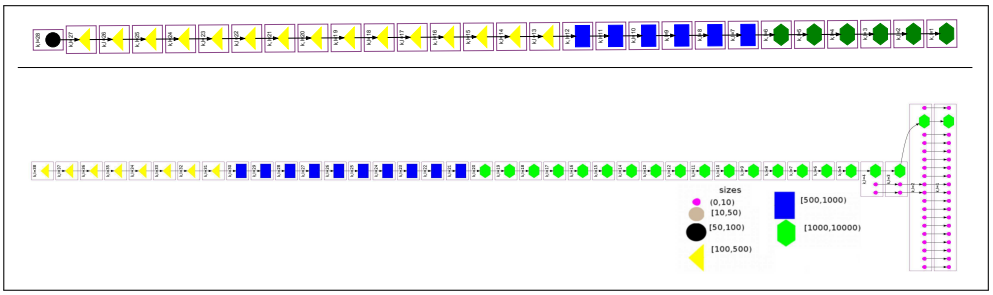


Figure 3.22: The CDF corresponding the diagonal D-cores(i, i) for the synthetic/artificial (upper), DBLP (bottom). SCC's are depicted with different colors depending on their sizes.

3.4.2 *Data sets description*

Starting with the Wikipedia dataset, a snapshot of the English version of Wikipedia was utilized, the digraph consists of about 1.2M nodes and 3.662M links. The snap-

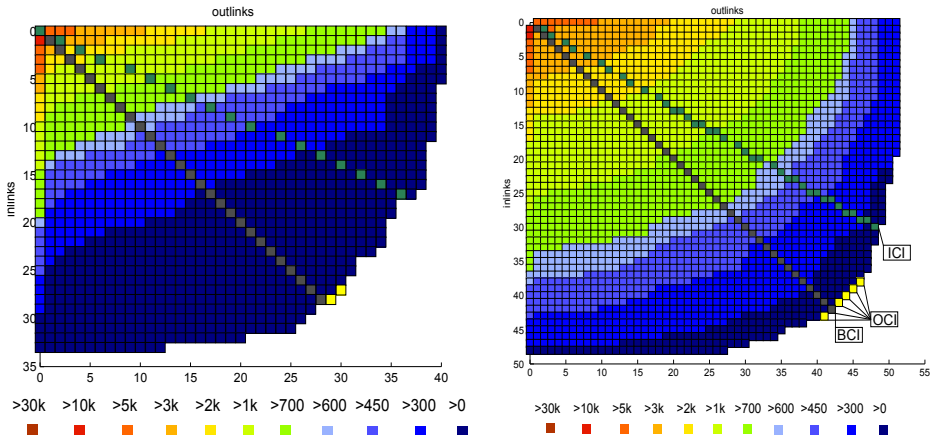


Figure 3.21: The D-core matrices of the synthetic digraph (left) and the DBLP digraph (right)

shot depicts Wikipedia as it was in the January of 2004 and was extracted from a database dump containing the entire history of the encyclopedia; available at:

http://download.wikipedia.org/.

In the experiments, was also used a popular bibliographic dataset derived from the available snapshot of DBLP, which is freely available in XML format at:

http://dblp.uni-trier.de/xml/.

The digraph structure was obtained from the dataset as follows: authors correspond to the nodes of the digraph and each directed edge $e = (x, y)$, express the fact that author x cited in his/her papers a paper of author y. That way, was obtained a digraph containing about 825K author nodes and 351K edges. The vast majority of them have no in-/out- links (about 800K) thus we remain with the rest 25K authors that are minimally connected.

Additionally, experiments were run on the ARXIV HEP-TH (high energy physics theory) citation graph. This is a paper citation graph is originally from the e-print arXiv with 27.700 papers and is freely available at

http://snap.stanford.edu/data/cit-HepTh.html.

From the paper citation graph was extracted the author citation digraph similar to the DBLP one, containing 8821 authors and 391K edges/citations.

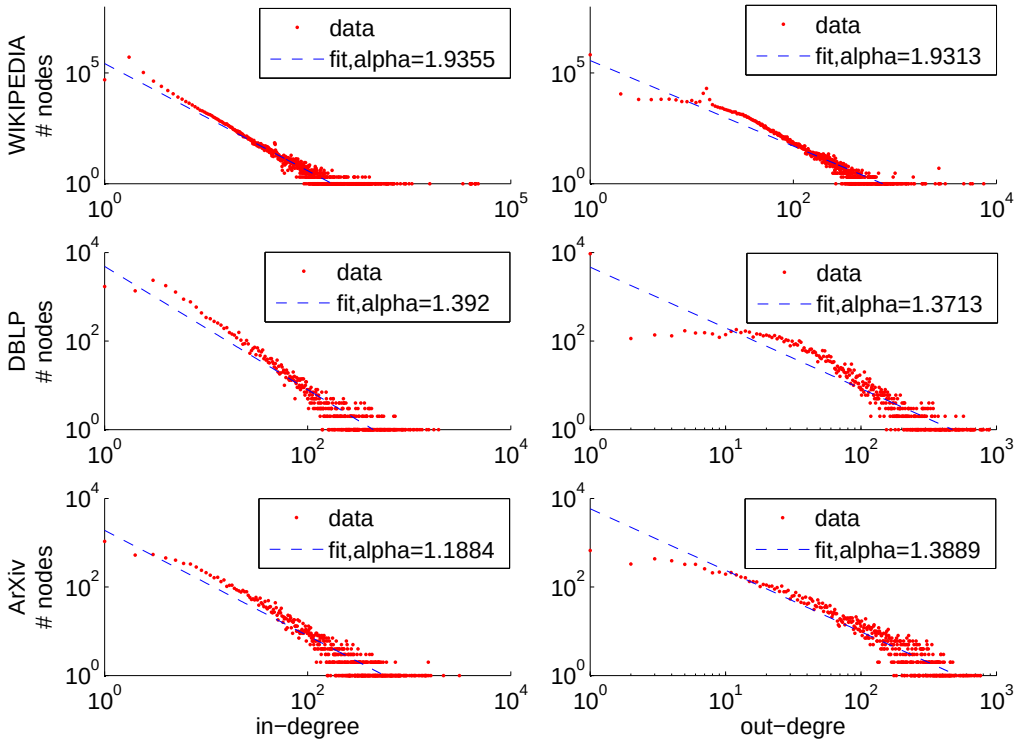


Figure 3.23: Distributions of the in and out degree for the real world datasets as noted above in log-log scale with power-law fitting.The exponent of the power law is also displayed.

In Figure 3.23, the reader can see the degree distribution of both in- and out-degree for the three datasets. There, all of them display a scale-free behavior governed by a power law; a parameter fitting was carried out to identify approximately that behavior. In more detail, all three of them display a clear preferential attachment behavior with regards to the in-degree, probably with no "initial attractiveness" (see the described models above). Instead, in the out-degree, even though there is a general scale free behavior, there is also evidence of the "initial attractiveness" parameter being larger than the absolute minimum. This is evident by the somewhat uniform behavior for the "smaller" degrees (not including the degree of 1). Intuitively papers with more than zero citations to other papers will cite a few papers, meaning more than one. On the other end, a paper can not have too many citations, i.e. out links. The previous applies naturally to authors as well. This in a way resembles the δ_{out} parameter of our adopted model. Thus the δ_{out} (for the model we adopt) has to be larger than 1 for the citations networks. As it can be seen later, the parameters that fitted the closest to the DBLP dataset adapt to this intuition.

3.4.3 *Algorithms complexity*

The proposed D-core algorithm is of low complexity thus D-core computations are feasible even in large scale digraphs. As shown in procedure $Trim_{k,l}(D)$ in section 2.4, the computation of each D-core is linear to the number of its edges and thus optimal. Moreover as the digraphs we examine are sparse, the identification of the D-cores is very fast.

The D-core matrix computation, starts from the original digraph and reduces it until the degeneracy leads to an empty one. This procedure involves about $(40 \times 50) \sim 2000$ repeated executions, in the case of the Wikipedia digraph, of the basic $Trim_{k,l}(D)$ procedure. Depending on the implementation, each execution can be done on commodity desktops in the scale of minutes even in million scale sized graphs, as it is also noted in [12] for the case of non directed graphs.

3.4.4 *Experimental methodology*

The experimental method for processing the previously mentioned digraphs involved the following phases:

1. D-*core matrix computation:* this involves computing the D-core $\mathbf{DC}_{k,l}$ subgraph, where $(k,l) \in \{0,\ldots,k_{max}\} \times \{0,\ldots,l_{max}\}$ where $(k_{max},0),(0,l_{max})$ are the extreme cells of $F(D)$. According to Observation 2, a D-core $\mathbf{DC}_{i,j}$ is a subgraph of every D-core $\mathbf{DC}_{i',j'}$ where $i' \leqslant i$ and $j' \leqslant j$. Based on this property, it is to compute e.g. the D-core $\mathbf{DC}_{0,2}$ having computed and stored in memory the D-core $\mathbf{DC}_{0,1}$. Therefore, in order to compute the entire D-core diagram, the process has to be started by computing only the D-cores in row 0 and column 0 and used those two sets of D-cores to "fill in" the rest of the matrix (note that the D-cores $\mathbf{DC}_{0,1}$ and $\mathbf{DC}_{1,0}$ are not correlated so we need to compute both but we only need *one of them* to fill the rest of the matrix). Each D-core occupies moderate storage space, such that the whole D-cores matrix occupies less than 4GB of disk space, so storing them for subsequent use was an obvious choice.

2. *Collaboration indices computation*: The values that optimize the criteria set along with the sizes of the corresponding D-cores are computed. Namely, those are the corresponding BCI/ICI/OCI/ACI, indices and the Robustness.

3. *Strongly Connected Components (SCCs) and Core Decomposition Forests (CDF's)*: Let D be the digraph corresponding to Wikipedia 2004 or DBLP. For each D-core $\mathbf{DC}_{i,i}$ – i.e. on the D-core matrix diagonal – the strongly connected components were computed. The core elimination sequence $\mathcal{L} = \mathbf{DC}_{0,0},\ldots,\mathbf{DC}_{r,r}$

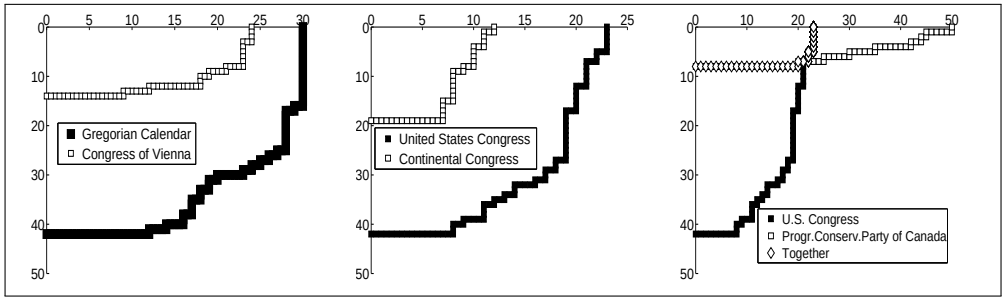


Figure 3.24: Selected term-pages and sets of term-pages frontiers from Wikipedia.

was considered; where r is the BCI of D and the corresponding Core Decomposition Forests was computed for both graphs. SCCs indicate groups of strong cohesiveness in the D-core. See Figure 3.25 for a detailed view on the SCCs size evolution and sub-digraph relationships as i, running along the D-core matrix diagonal, increases for both datasets considered.

4. *Frontiers for sets of entries*: moreover were computed the frontiers for single terms/authors for the Wikipedia and DBLP digraphs respectively. This is also extended to sets of terms/authors. These indicate the robustness (represented by the values of the indices) for the D-cores containing them.

3.4.5 *Experimental results on Wikipedia*

D-CORE MATRIX AND INDICES VALUES.

The Wikipedia digraph was processed and for each (k, l) cell of the D-core matrix the sizes of the resulting D-cores were computed (see Figure 3.26) as well as the sizes of the SCC's in each of the D-core(i, i), i.e. on the diagonal of the matrix as mentioned before.

The indices, defined in section 2.4.5, were computed for the global Wikipedia digraph as well as for selected representative terms and sets of terms (see Figure 3.24). For Wikipedia 2004 the balanced collaboration index(BCI) value is 38 while the respective D-core $\mathbf{DC}_{38,38}$ contains 237 nodes. For the same digraph, the inherent collaboration index ICI is 36 and is obtained for the D-cores $\mathbf{DC}_{39,33}$ that contains 206 nodes. For the OCI index we obtain two OCI-optimal frontier cells corresponding to the $\mathbf{DC}_{38,41}$ and $\mathbf{DC}_{36,43}$ D-cores containing 228 and 233 nodes respectively. The *robustness* of the global Wikipedia digraph is remarkably high at 0.963, while the maximum value is 1, indicating a very robust digraph.

D-CORES FRONTIERS FOR TERMS AND SETS OF TERMS.

Afterwards the cohesion and in/outlinks trade-off was investigated for D-cores containing specific term-pages. These metrics are perceived as indication of the

collaborativeness and authority/hubness of the digraphs containing these term-pages. Further we present representative terms-pages D-core matrices evaluating them.

As defined in 2.4.6, the D-core diagram of a vertice containing term X corresponds to the D-cores of the D-core diagram of D whose vertices sets contain X. In Figure 3.24 the reader can see the D-cores matrix frontiers for the digraphs containing the terms: Congress of Vienna, Continental Congress, Gregorial Calendar, Progressive Conservative party of Canada, and United States Congress. In each sub-figure, the frontier of the respective digraphs degeneracy can be seen, each presenting different features and trends. The frontier for the term Continental Congress for example is presenting a low BCI index with regard to the global digraph (the BCI index is 38), as the page is participating in D-cores with low degeneracy. Its respective ICI index is (19.7) much lower than the global ICI value 36. This is a rather "selfish" page as it participates in D-cores dominated by in-links.

Contrary to the previous, the Gregorian Calendar page participates in much more robust D-cores as its BCI index reaches a high 26, while its OCI is a very high – occurring at cell (42,12) – indicating a very "selfish behavior" dominated by inlinks and thus having an authority digraph behavior. On the other hand, the Congress of Vienna page is presenting a rather extrovert behavior as its OCI index occurring at cell (8,23), an indication of outlinks domination in the optimal subgraphs. The robustness of the digraph is rather low with a BCI index at 11, a low value as compared to the global BCI 38.

In Figure 3.24 (right) we present the joint D-core matrix and frontier of two term pages (Progressive Conservative Party of Canada and United States Congress). The "together" frontier represents the frontier of the D-core digraphs containing both terms. The joint D-core frontier can exhibit much worse robustness under degeneracy (i.e. removing in/out links) that the individual ones. This can be the case when the D-core frontiers of term pages with contradictory trends are put together; as it is in our example, where the joint frontier is at $\mathbf{DC}_{8,22}$. Thus we obtain a much weaker digraph than the ones of the individual terms.

THEMATIC FOCUS OF WIKIPEDIA SCCS.

The SCCs of the Wikipedia D-cores $\mathbf{DC}_{i,i}$ were computed on the balanced diagonal direction (BCI direction). The intuition is that the SCCs are considered as digraph areas with high cohesion. In Figure 3.25 the reader can see the cardinality of the SCCs in each Wikipedia D-core $\mathbf{DC}_{i,i}$, the size of the SCCs and their hierarchical containment relation as i increases along the BCI axis. As it can be noticed there, starting in D-core $\mathbf{DC}_{1,1}$, there are several SCCs moderately sized (<100 nodes) – excluding one significantly larger sized SCC (>100K nodes in D-core $\mathbf{DC}_{1,1}$. Many of the SCCs survive until the D-core $\mathbf{DC}_{32,32}$, after this only the initial giant component survives until the extreme BCI D-core $\mathbf{DC}_{38,38}$.

(k, k)	# SCCs	Top-k SCCs size	Thematic Focus
1	1024	24	Wisconsin
		10	Cynodonts Species
		10	Iowa
		10	Eurovision
		5	History of the British penny
		5	Submarines
		10	Wyoming
2	23	30	Music albums
		10	Eurovision
		6	Cynodonts Species
		6	Metal Deficiencies
		5	History of the British penny
		3	Helladic
3	13	23	Extinct species
		10	Eurovision Young Dancers
		6	Metal Deficiencies
		6	Books
		5	Cynodonts Species
		5	History of the British penny
4	12	26	poker jargon
		10	Eurovision
		6	Metal Deficiencies
		5	History of the British penny
		5	films by decade
		4	Fayette
5	8	26	poker jargon
		17	Sibley-Monroe checklist
		10	Eurovision
		7	North Carolina
...			...
38	1		Dates

Table 3.11: The thematic focus of the Wikipedia SCCs for increasing degeneracy along the BCI axis.

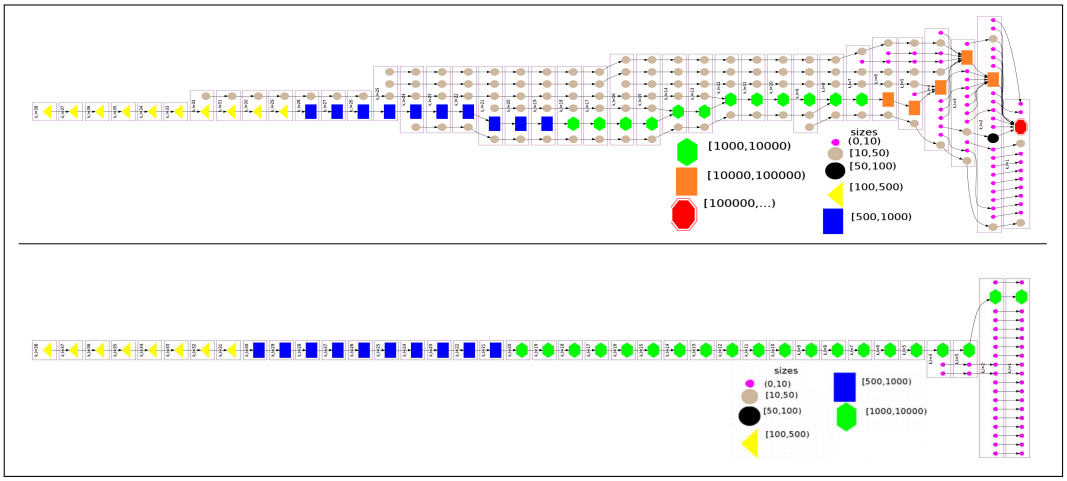


Figure 3.25: The CDF corresponding to the diagonal D-cores(i, i) for Wikipedia 2004 (upper), DBLP (bottom). SCC's are depicted with different colors depending on their sizes.

Further, the thematic focus of the SCCs is investigated by studying the D-cores along the BCI optimal axis, see Table 3.11. A giant component is observed that dominates and almost all the pages within it contain the terms "time". The digraph was pruned, removing those pages and that were noticed to have a similar behavior, this time with the term Grammy awards dominating the single giant SCC remaining. It is interesting to stress that in D-core $\mathbf{DC}_{1,1}$ there are 1034 SCCs (apart from the giant one). The size of the top-5 SCCs ranges between 5 and 24 nodes while for each one there is a remarkably narrow focus in their thematic area. For instance, see Table 3.11, the top sized SCC is about Wisconsin. The rest of the SCCs are thematically focused in: Cynodonts species, Iowa, Eurovision, History of the British penny, Submarines, Wyoming. In D-core $\mathbf{DC}_{2,2}$ there are only 23 SCCs (apart from the giant one). The size of the top-5 SCCs ranges between 3 and 30 nodes while the thematic focus of the top sized SCCs is to a large degree identical to the top SCCs in D-core $\mathbf{DC}_{1,1}$ A similar trend continues as i increases along the diagonal $\mathbf{DC}_{i,i}$.

The DBLP digraph was processed and the size of the resulting D-cores was found for each cell (k, l) (see Figure 3.26 bottom). Additionally, the number of strongly connected components (SCC's) in each of the D-cores $\mathbf{DC}_{i,i}$ – i.e. on the diagonal (see Figure 3.25 bottom) – was computed. All the indices, defined in section 2.4.5, were computed for the global DBLP digraph and for selected representative authors and sets of authors.

For the case of the DBLP digraph, the value of BCI is 42 (see Table 3.12 a summary of all indices values) while the respective D-core $\mathbf{DC}_{42,42}$ contains 188 nodes (see the lower part of Figure 3.26). For the same digraph, the inherent collaboration index ICI is 39 and is obtained for the D-core $\mathbf{DC}_{30,48}$ that contains 220 nodes.

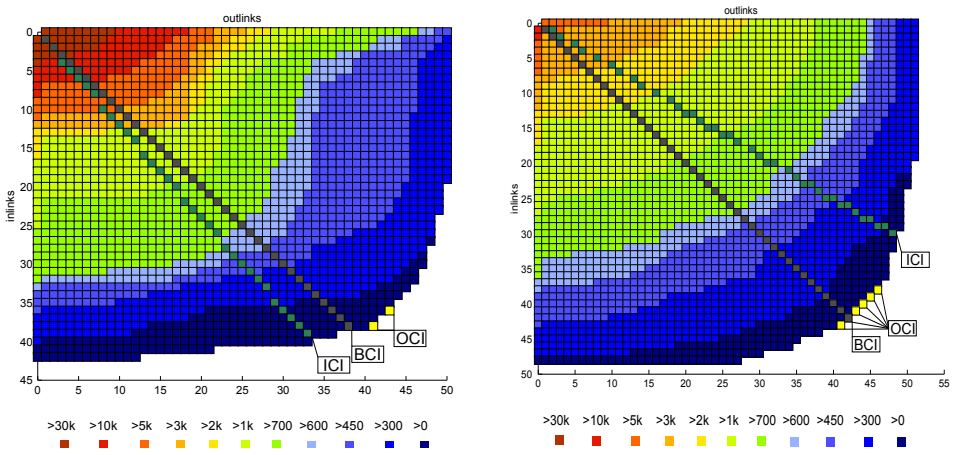


Figure 3.26: The D-core matrices of the Wikipedia 2004 digraph (left) and the DBLP digraph (right)

For the OCI index we get a value 42, which occurs in six D-cores located at the positions: $(38,46), (39,45), (40,44), (41,43), (42,42), (43,41)$ on the D-core matrix frontier. The *robustness* of the global DBLP digraph is remarkably high at 0.966 indicating a very robust to degeneracy digraph. It is evident that the DBLP digraph has significant extrovert features (i.e. more out than in citations, an expected result)

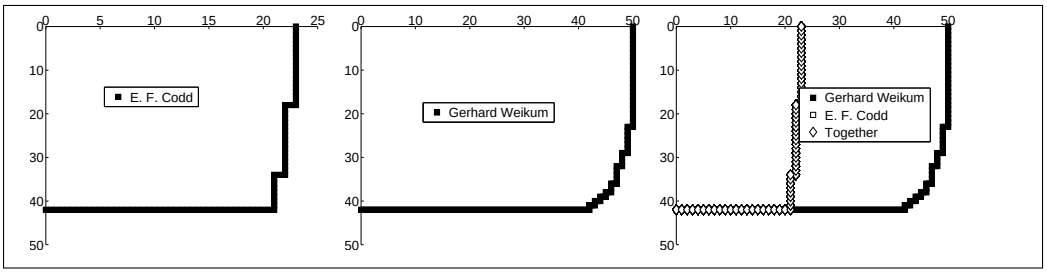


Figure 3.27: Representative authors D-core frontier from the DBLP digraph

3.4.6 *Experimental results on DBLP*

Furthermore, the SCCs of the DBLP D-cores $\mathbf{DC}_{i,i}$ were computed on the balanced diagonal direction (BCI direction). In Figure 3.25, bottom, one can see the cardinality of the SCCs in each DBLP D-core $\mathbf{DC}_{i,i}$, the size of the SCCs, and their containment relation as i increases. As it can be noticed, starting in D-core $\mathbf{DC}_{1,1}$, there are few SCCs poor sized (<10 nodes) – excluding one significantly larger sized SCC (>1000 nodes in $\mathbf{DC}_{1,1}$ – that survive until $\mathbf{DC}_{4,4}$. After this, only the initial giant component continues until the extreme BCI D-core $\mathbf{DC}_{42,42}$. This SCC apparently contains the nodes/authors with a large number of mutual citations.

	DBLP	E.F. Codd	G. Weikum
BCI(k,l)/ Size of optimal DC	42/188	22/913	41/221
ICI/(k,l)/angle/ size of optimal DC	39/(30,48)/ 32.01/220	19/(15,23)	38/(29,47)
OCI/⟨(k,l)/angle/ size of optimal DC⟩	42/⟨(43,41)... (38,46)/43.63 ,...,50.44/165, 188,217,187, 185,188⟩	31.5/(42,21)	41.5/(38,45)
Robustness,Local	-	0.457	0.966
Robustness, Global	0.966	0.952	0.928
ACI	35.17	23.083	33.66
AC H/A-angle (deg)	43.90	55.66	41.91

Table 3.12: Collaboration indices values for the DBLP digraph

The giant SCC contains 188 authors (Table 3.13) presenting both top publication activity, thus many outgoing citations, as well as high rate of incoming citations. This group of authors indeed contains well known and reputable scientists' names and looks pretty reasonable. Of course, it has to be stressed that the partial coverage of the DBLP data set as its citation bulk is before 2004. Also in the first years of its function the emphasis is on database related papers.

Further on, the D-cores corresponding to specific authors were studied and the respective D-core matrices and frontiers were computed. We selected two characteristic cases of seminal authors. In Figure 3.27 (left), the D-core matrix and frontier for "E.F Codd" can be seen, founder of the relational database area. His BCI extreme is $\mathbf{DC}_{42,23}$ indicating an intensive inlinks (incoming citations) trend. This is natural as he was authoring in the early years of computer science with few previous works to cite. On the contrary his works enjoy a very high number of citations, thus a high number of inlinks in the citations digraph.

On the other hand a more modern seminal author G. Weikum presents a very robust to degeneracy D-core structure for both in/out links tendency. This is explained by the facts.

i. his works are highly cited during many years and

ii. he is intensively authoring and thus citing other authors.

In Figure 3.27 (right) the joint D-core matrix and frontier for the two aforementioned authors are presented. The "together frontier" represents the frontier of the D-cores that contain both E.F. Codd and G. Weikum author (nodes), thus representing the D-cores (i.e. citation subgraphs) in which the two aforementioned cite in common and they are commonly cited.

José A. Blakeley	Hector Garcia-Molina	Abraham Silberschatz	Umeshwar Dayal
Eric N. Hanson	Jennifer Widom	Klaus R. Dittrich	Nathan Goodman
Won Kim	Alfons Kemper	Guido Moerkotte	Clement T. Yu
M. Tamer Özsu	Amit P. Sheth	Ming-Chien Shan	Richard T. Snodgrass
David Maier	Michael J. Carey	David J. DeWitt	Joel E. Richardson
Eugene J. Shekita	Waqar Hasan	Marie-Anne Neimat	Darrell Woelk
Roger King	Stanley B. Zdonik	Lawrence A. Rowe	Michael Stonebraker
Serge Abiteboul	Richard Hull	Victor Vianu	Jeffrey D. Ullman
Michael Kifer	Philip A. Bernstein	Vassos Hadzilacos	Elisa Bertino
Stefano Ceri	Georges Gardarin	Patrick Valduriez	Ramez Elmasri
Richard R. Muntz	David B. Lomet	Betty Salzberg	Shamkant B. Navathe
Arie Segev	Gio Wiederhold	Witold Litwin	Theo Härder
Francois Bancilhon	Raghu Ramakrishnan	Michael J. Franklin	Yannis E. Ioannidis
Henry F. Korth	S. Sudarshan	Patrick E. O'Neil	Dennis Shasha
Shamim A. Naqvi	Shalom Tsur	Christos H. Papadimitriou	Georg Lausen
Gerhard Weikum	Kotagiri Ramamohanarao	Maurizio Lenzerini	Domenico SaccÃ
Giuseppe Pelagatti	Paris C. Kanellakis	Jeffrey Scott Vitter	Letizia Tanca
Sophie Cluet	Timos K. Sellis	Alberto O. Mendelzon	Dennis McLeod
Calton Pu	C. Mohan	Malcolm P. Atkinson	Doron Rotem
Michel E. Adiba	Kyuseok Shim	Goetz Graefe	Jiawei Han
Edward Sciore	Rakesh Agrawal	Carlo Zaniolo	V. S. Subrahmanian
Claude Delobel	Christophe Lécluse	Michel Scholl	Peter C. Lockemann
Peter M. Schwarz	Laura M. Haas	Arnon Rosenthal	Erich J. Neuhold
Hans-Jörg Schek	Dirk Van Gucht	Hamid Pirahesh	Marc H. Scholl
Peter M. G. Apers	Allen Van Gelder	Tomasz Imielinski	Yehoshua Sagiv
Narain H. Gehani	H. V. Jagadish	Eric Simon	Peter Buneman
Dan Suciu	Christos Faloutsos	Donald D. Chamberlin	Setrag Khoshafian
Toby J. Teorey	Randy H. Katz	Miron Livny	Philip S. Yu
Stanley Y. W. Su	Henk M. Blanken	Peter Pistor	Matthias Jarke
Moshe Y. Vardi	Daniel Barbará	Uwe Deppisch	H.-Bernhard Paul
Don S. Batory	Marco A. Casanova	JÃŒrgen Koch	Joachim W. Schmidt
Guy M. Lohman	Bruce G. Lindsay	Paul F. Wilms	Z. Meral Özsoyoglu
Gultekin Özsoyoglu	Kyu-Young Whang	Shahram Ghandeharizadeh	Tova Milo
Alon Y. Levy	Georg Gottlob	Johann Christoph Freytag	Klaus KÃŒspert
Louiqa Raschid	John Mylopoulos	Alexander Borgida	Anand Rajaraman
Joseph M. Hellerstein	Masaru Kitsuregawa	Sumit Ganguly	Rudolf Bayer
Raymond T. Ng	Daniela Florescu	Per-Ake Larson	Hongjun Lu
Ravi Krishnamurthy	Arthur M. Keller	Catriel Beeri	Inderpal Singh Mumick
Oded Shmueli	George P. Copeland	Peter Dadam	Susan B. Davidson
Donald Kossmann	Christophe de Maindreville	Yannis Papakonstantinou	Kenneth C. Sevcik
Gabriel M. Kuper	Peter J. Haas	Jeffrey F. Naughton	Nick Roussopoulos
Bernhard Seeger	Georg Walch	R. Erbe	Balakrishna R. Iyer
Ashish Gupta	Praveen Seshadri	Walter Chang	Surajit Chaudhuri
Divesh Srivastava	Kenneth A. Ross	Arun N. Swami	Donovan A. Schneider
S. Seshadri	Edward L. Wimmers	Kenneth Salem	Scott L. Vandenberg
Dallan Quass	Michael V. Mannino	John McPherson	Shaul Dar
Sheldon J. Finkelstein	Leonard D. Shapiro	Anant Jhingran	George Lapis

Table 3.13: Authors in the D-core $\mathbf{DC}_{42,42}$ of the DBLP digraph.

3.4.7 *Experimetnal results on ARXIV*

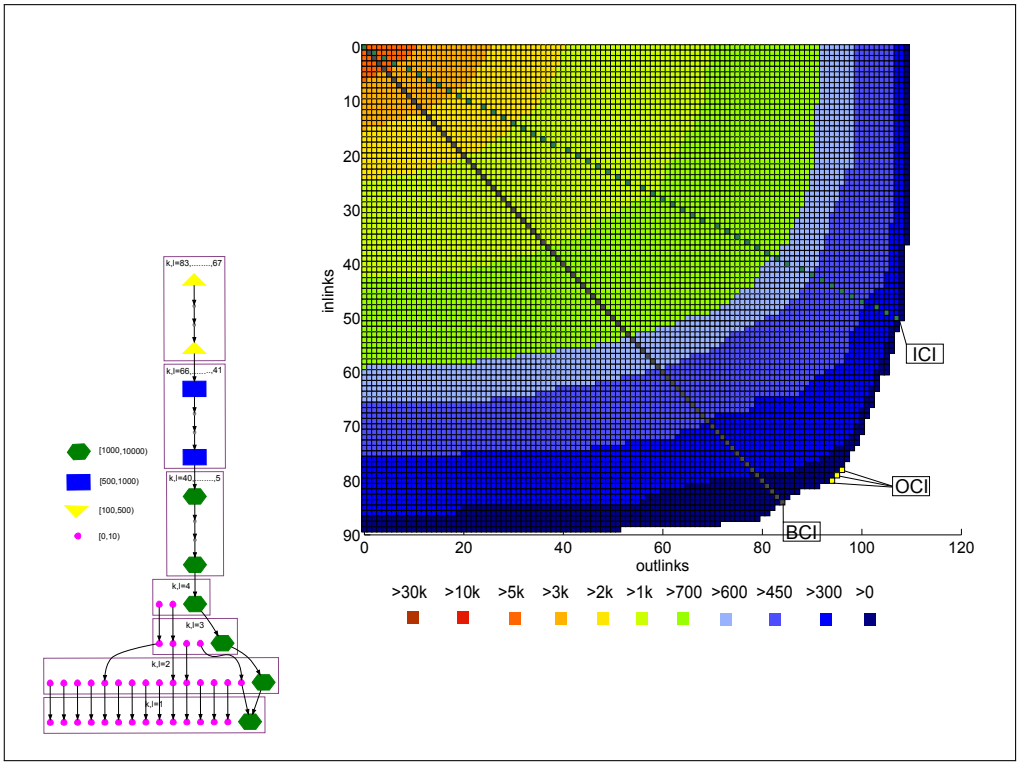


Figure 3.28: Left:The CDF corresponding to the diagonal D-cores(i,i) for ARXIV. SCC's are depicted with different colors depending on their sizes. Right: The D-core matrix of the same data.

Following the same procedure, with the other two datasets, the D-core matrix and the CDF for the ARXIV citation graph were produced. The reader can see the results in Figure 3.28. It is interesting to stress that this graph has a much denser core resulting in much larger metric values as it can be easily seen in the respective D-core matrix. Additionally it can be seen that the CDF is dominated by one SCC in its largest part. Furthermore it can be noticed that the initial giant component survives until the extreme BCI D-core $\mathbf{DC}_{83,83}$. Thus this graph is much more robust to degeneracy than all the others we tested indicating thus a very dense collaboration among the members of the theoretical Physics community. The authors of this core can be seen in Table 3.15. It is evident that all the senior names in this scientific area appear here justifying their close collaboration to the community in terms of in/out citations. It is stressed here that the abbreviated version for most of the author names was used as these was more frequent in the dataset.

As for the other characteristics, the inherent collaboration index ICI is characterized by an angle of 25 degrees at the $\mathbf{DC}_{50,107}$ D-core with size of 306. For the

A. Klemm	S. Theisen	P.S. Aspinwall	B.R. Greene
B.R. Greene	D.R. Morrison	A. Giveon	M. Porrati
M. Porrati	E. Rabinovici	N. Seiberg	E. Witten
E. Witten	M. J. Duff	Andrew Strominger	Shyamoli Chaudhuri
Shyamoli Chaudhuri	Shamit Kachru	Cumrun Vafa	S. Ferrara
S. Ferrara	J. A. Harvey	J. Polchinski	A. Ceresole
A. Ceresole	R. D'Auria	Katrin Becker	Melanie Becker
Melanie Becker	James T. Liu	J. Rahmfeld	W. Lerche
W. Lerche	P. Mayr	M. Bershadsky	Jan Louis
Jan Louis	Sheldon Katz	M. Ronen Plesser	Michael R. Douglas
Michael R. Douglas	Gregory Moore	Micha Berkooz	Robert G. Leigh
Robert G. Leigh	John H. Schwarz	J. Distler	K. Intriligator
K. Intriligator	B. Craps	A. Van Proeyen	Julie D. Blum
Julie D. Blum	Kentaro Hori	Hirosi Ooguri	Ashoke Sen
Ashoke Sen	Ruben Minasian	Moshe Rozali	Mirjam Cvetic
Mirjam Cvetic	Burt A. Ovrut	K.S. Stelle	Daniel Waldram
Daniel Waldram	H. Lu	C.N. Pope	Klaus Behrndt
Klaus Behrndt	A. Zaffaroni	N.P. Warner	A. Kehagias
A. Kehagias	K. Sfetsos	Steven S. Gubser	D.Z. Freedman
D.Z. Freedman	A. Brandhuber	A. Karch	Per Kraus
Per Kraus	J. de Boer	E. Verlinde	H. Verlinde
H. Verlinde	A. Sagnotti	N. Dorey	Matthias Blau
Matthias Blau	T. Banks	W. Fischler	L. Susskind
L. Susskind	A. Fayyazuddin	Juan M. Maldacena	B. Pioline
B. Pioline	Edi Halyo	G.W. Gibbons	I.R. Klebanov
I.R. Klebanov	Mans Henningson	Kostas Skenderis	Cesar Gomez
Cesar Gomez	L. Girardello	Vijay Balasubramanian	G. Papadopoulos
G. Papadopoulos	P.K. Townsend	A.A. Tseytlin	Jerome P. Gauntlett
Jerome P. Gauntlett	E. Kiritsis	T.R. Taylor	Gary T. Horowitz
Gary T. Horowitz	Robert C. Myers	Donam Youm	E. Sezgin
E. Sezgin	Chris M. Hull	Anamaria Font	Yaron Oz
Yaron Oz	Zheng Yin	Ilka Brunner	Albion Lawrence
Albion Lawrence	John McGreevy	Joaquim Gomis	N. Nekrasov
N. Nekrasov	T. Tada	D. Minic	M.B. Green
M.B. Green	M. Gutperle	Petr Horava	Clifford V. Johnson
Clifford V. Johnson	Gabriel Lopes Cardoso	Dieter Lust	O. Bergman
O. Bergman	G. Lifschytz	Atish Dabholkar	Barton Zwiebach
Barton Zwiebach	Nathan Berkovits	R.R. Metsaev	S. Yankielowicz
S. Yankielowicz	Philip C. Argyres	Amihay Hanany	M. Bianchi
M. Bianchi	Duiliu-Emanuel Diaconescu	Ofer Aharony	Hong Liu
Hong Liu	P.S. Howe	P.C. West	Nakwoo Kim
P.C. West	Richard Corrado	Horatiu Nastase	Amanda W. Peet
Amanda W. Peet	M.R. Gaberdiel	Piljin Yi	Rajesh Gopakumar
Rajesh Gopakumar	C. Bachas	Akikazu Hashimoto	Marco Billo
Marco Billo	I. Pesando	Keshav Dasgupta	Sunil Mukhi
Sunil Mukhi	Joseph A. Minahan	D. Kutasov	Juan Maldacena

Table 3.14: Authors in the D-core $\mathbf{DC}_{83,83}$ of the of the ARXIV digraph.

Juan Maldacena	Jeremy Michelson	C. Kounnas	R. Dijkgraaf
R. Dijkgraaf	Nissan Itzhaki	Jacob Sonnenschein	S. Gukov
S. Gukov	David Berenstein	Pei-Ming Ho	Angel M. Uranga
Angel M. Uranga	Sangmin Lee	J.G. Russo	E. Bergshoeff
E. Bergshoeff	M. de Roo	Soo-Jong Rey	Jung-Tay Yee
Jung-Tay Yee	Finn Larsen	Sandip P. Trivedi	I. Antoniadis
I. Antoniadis	E. Gava	K. S. Narain	R. Kallosh
R. Kallosh	J. Kumar	H.J. Boonstra	Kyungho Oh
Kyungho Oh	Radu Tatar	Mina Aganagic	Jaemo Park
Jaemo Park	David A. Lowe	Andrei Linde	Eric G. Gimon
Eric G. Gimon	L. E. Ibanez	Zurab Kakushadze	F. Quevedo
F. Quevedo	Ramzi R. Khuri	J. X. Lu	S. Sethi
S. Sethi	Sanjaye Ramgoolam	Sumit R. Das	Miao Li
Miao LI	Chris Hull	Washington Taylor	Curtis G. Callan
Curtis G. Callan	Samir D. Mathur	E. Martinec	Daniel Kabat
Daniel Kabat	BS Acharya	JM Figueroa-O'Farrill	Bernard de Wit
Bernard de Wit	Chong-Sun Chu	T. Ortin	Michael Dine
Michael Dine	Eva Silverstein	Laura Andrianopoli	Leonardo Rastelli
Leonardo Rastelli	Ulf H. Danielsson	Ori J. Ganor	Anastasia Volovich
Anastasia Volovich	H. Partouche	Barak Kol	Shmuel Elitzur
Shmuel Elitzur	A. Rajaraman	J.L.F. Barbon	Gabriele Ferretti
Gabriele Ferretti	Adel Bilal	S. P. de Alwis	Steven B. Giddings
Steven B. Giddings			

Table 3.15: Authors in the D-core $\mathbf{DC}_{83,83}$ of the of the ARXIV digraph(continued).

OCI index we obtain three cells $\mathbf{DC}_{78,95}$, $\mathbf{DC}_{79,94}$ and $\mathbf{DC}_{80,93}$ with respective sizes of 237, 241, and 244 nodes respectively.

The *robustness* of the ARXIV graph is high as well at 0.9704 indicating, much like the DBLP one, very high robustness to degeneracy digraph. Again, overall are observed some very extrovert features meaning that the graph is featured mostly by outgoing citations. In this case it ca be said that the ARXIV digraph displays higher extroversion than the DBLP. On the other hand this could be attributed to the fact that the DBLP dataset is not very well maintained thus lots of citations missing.

3.4.8 *Conclusions*

In this section the behavior of the new concepts and metrics was investigated for the case of synthetic preferential attachment graphs - dominant in real world cases. The study is extended to various parameters values in an attempt to fit the features of the real-worlds graphs. In order to achieve this a multi-parametric graph generator was developed. Moreover, an extensive experimental evaluation was conducted for scale-free/preferential attachment synthetic graphs as well as real-world large scale directed graphs: the (English) Wikipedia - 2004 edition, the ARXIV and DBLP citation graphs. The D-cores matrices, frontiers and metrics were

computed and explored respectively, thus deriving interesting results and observations both at the macroscopic (graph) and at the microscopic (node) level. The goal of this section was to validate that the D-core concept and the relevant structures and metrics that were defined in this work constitute a framework of tools for efficient and valid evaluation of cohesiveness and collaboration in directed networks.

3.5 S-CORES

In this section, the metrics and structures defined in section 2.5.2 are used on two kinds of signed networks: *explicit* ones, from existing Web applications publishing such networks, and *implicit* ones, *inferred* from interactions that can be interpreted as positive or negative.These datasets are described below. For the latter kind, signed networks from articles in four domains on the English *Wikipedia* are considered.

3.5.1 *Datasets Description and Methodology*

EXPLICIT SIGNED NETWORKS.

Two explicit signed networks are explored, available on the SNAP website[4], *Epinions* and *Slashdot*. The *Epinions* network is extracted from the epinions.com website, an user-driven product review website, in which any user of the site can indicate in their profile if they trust or distrust other users. Similarly, in the *Slashdot* signed network, extracted from the slashdot.org website, users declare friends or foes.The basic properties properties of these explicit signed networks – number of nodes, number of edges and ratio of negative edges – are presented in Table 3.16. For a more in-depth description, the reader is reffed to [56], where these networks are studied through the lens of social theories like status and balance, as well for signed link prediction.

IMPLICIT (INFERRED) SIGNED NETWORKS.

In this part, the network *WikiSigned*[5] was adopted, which is a signed network built with the methodology of [64], over the Wikipedia editors, based on the articles of the English Wikipedia and the revision history thereof. In short, this network tracks the various interactions between contributors, either in text editing, in votes for adminship of pages, or in acknowledgments of contributions (so called barnstars). For the reader's convenience, the methodology of link construction from interactions in the Wikipedia is detailed in section 3.5.2 . From the global

4 http://snap.stanford.edu/data/index.html#signnets
5 http://www.infres.enst.fr/wp/maniu/datasets/

Network	Nodes	Edges	Negative
Epinions	119,217	841,200	15.0%
Slashdot	82,144	549,202	22.6%

Table 3.16: Properties of the explicit signed networks.

Domain	Articles	Nodes	Edges	Negative
History	3,331	141,983	534,693	17.5%
Politics	12,921	453,116	2,428,945	13.5%
Religion	6,459	277,482	1,423,279	12.6%
Mathematics	9,610	158,671	651,450	15.9%

Table 3.17: The signed networks extracted from the four Wikipedia domains.

WikiSigned network, four subsets of articles were selected, from the following domains History, Politics, Religion and Mathematics. Each of these sets gave a corresponding subgraph of WikiSigned.

In Table 3.17 the properties of the four signed networks are given – corresponding to the four domains of articles: number of extracted articles for each domain, number of nodes, number of edges and ratio of negative edges.

3.5.2 *WikiSigned methodology*

For the construction of the four signed networks, the following methodology was used, as in [64]. For each *Wikipedia* article in the corpus, its complete version (or revision) history was processed, in chronological order. For each revision, the following types of interactions were extracted:

1. the number of *words inserted* by the author of the current revision in the vicinity of the text belonging to other authors,
2. the number of *words deleted or replaced* between the author of the current revision and the previous authors of the modified text,
3. if the current revision is a *reversion*, i.e., the current author decided that a previous revision of the article needs to be restored; a *revision restore* interaction was established between the current author and the author of the targeted revision, and one or more *revision revert* interactions were established between the current author and the authors of the revisions discarded in the process.

The first two items are called *text interactions*, while interactions resulting from the last item are called *revision interactions*. Note that in order for these interactions

to be properly tracked, an up-to-date text author list needs to be computed for each revision. When revisions are reverted, so are their corresponding author lists. Furthermore, the following interactions between the authors of the above revisions need to be extracted:

1. votes in administrator elections[6], as either *positive votes* or *negative votes*,
2. *barnstars*, i.e., prizes acknowledging important contributions, which can be put on a user's profile page by other contributors.

For each such interaction, either a *positive* or *negative* interpretation is assigned. Deleted or replaced text, reverted revisions and negative administrator votes were assigned a negative interpretation. The rest were assigned a positive interpretation. Then, each unique contributor pair was aggregated, via summation, resulting in an *interaction vector* between the contributors. This vector summarizes the *directed* interactions, from an interaction *generator* to a *recipient*.

For deciding the link sign from this interaction vector, the following straightforward heuristic was used. For each of the four types of interactions (text, revision, election and barn star), its corresponding interpretation was the one of the more prevalent interaction, taking values in $\{-1, 0, 1\}$ (depending on its negative, undecided or positive outcome).[7] For example, if more words were replaced or deleted than inserted between a pair of contributors, then the text interactions between them were labeled as negative. Finally, these interpretations were aggregated into a link sign from the generator to the recipient, by the sign of the sum of the four per-interaction interpretations. This of course means that the various interactions may cancel each other out, in which case no link is built. In order to avoid creating links between contributor pairs that have too few interactions, two thresholds were imposed in the WikiSigned construction: a threshold on the minimum number of words interacted upon, and a parameter k for the minimum number of unique revisions each contributor pair must have interacted upon. In the experiments shown here, the word threshold is set at 10.

3.5.3 *Experimental Evaluation*

Here are displayed the experiments that were performed on the explicit signed graphs (*Epinions* and *Slashdot*), all the inferred *Wikipedia* networks and semiartificial networks. The algorithm for computing the s-cores of a signed digraph is linear to the number of the graph edges. As the signed graphs examined are sparse, the construction of the s-cores is hence very fast. The computation is quite

6 Each *Wikipedia* contributor can be candidate for election to a page administrator position. Other contributors can either oppose or support a candidate in an election.

7 This is justified by the fact that, for instance, one cannot objectively compare a number of deleted words with a number of negative votes.

straightforward: for a given pair of in- and out-degree thresholds, the vertices having degrees that are below the desired threshold are removed and the degrees of the remaining nodes are updated. This is repeated until there are no more nodes in the graph to remove.

The algorithm follows the same logic as in k/fractional-cores and S-cores. In particular, the efficiency of computing all the S-cores is optimized by utilizing the following property. A $(i^{s(i)}, j^{s(j)})$-dicore is a subgraph of every $(i'^{s(i')}, j'^{s(j')})$-dicore where $i' \leqslant i$ and $j' \leqslant j$ and for the signs: $s(i') = s(i)$ or i'=0 and $s(j') = s(j)$ or j'=0 (i.e. both of the dicores are in the same quadrant). Thus, the (e.g.) $(2^+, 0)$-dicore can be computed having computed and stored in memory the $(1^+, 0)$-dicore. Moreover, the entire S-core diagram can be computed by computing firstly the S-cores on the axes. Note that two S-cores upon different axes are not correlated so we need to compute all of them across the axes but we need on each quadrant only one of the axes to fill in the rest.

The execution time of the processing algorithm is in the order of tens of minutes on commodity hardware (2.5 Ghz dual-core processor with 4GB RAM), even for graphs of millions of nodes. In total, 40 different graphs were processed in a matter of hours (the two explicit networks and the inferred ones for various parameters). All the computed S-cores were stored for further analysis accumulating in total 60GB of disk space.

3.5.4 *Slashdot and Epinion graphs*

The graphs derived from the *Slashdot* and *Epinions* networks are explicitly defined by the users thus providing ground truth examples for the S-cores and their metrics. Figure 3.29a displays comparatively the frontiers of the *Slashdot* and *Epinion* graphs. It can be seen that the *Epinions* network has a larger negative area, which is interpreted as a more distrustful community. For more details the reader ca look at Table 3.18, containing the general trends of the two networks. For example we see the higher activity of mutual trust ($Q_{+,+}$) displayed from *Slashdot* (compared to *Epinions*) evident by the higher valued coordinates of $\delta_{max}^{(+,+)}(G)$ in that quadrant.

Additionally, a pattern arises concerning the equivalent values of reciprocity at node and graph level. Both types of reciprocity agree in the inversely reciprocal cases with low values. The more distinct differences are in the other quadrants and when comparing *simple reciprocity* (r^s) with the graph reciprocity (*GR*). It is quite evident that, concerning the remaining reciprocities, on a node level there is much less reciprocity. On the other hand, the general reciprocities are quite large which is attributed to users reciprocating more at a community level than a local one.

Graph	$Q_{+,+}$	$Q_{+,-}$	$Q_{-,-}$	$Q_{-,+}$	r^s/GR
Epinions					
max degeneracy ($\delta_{\mathbf{max}}$)	(19,18)	(1,-4)	(-5,-5)	(-5,1)	-
local reciprocity (**r**)	0.347	0.003	0.038	0.022	0.302
mean local reciprocity	0.230	0.002	0.046	0.013	0.160
graph reciprocity (**GR**)	0.976	0.109	0.886	0.197	0.859
Slashdot					
max degeneracy ($\delta_{\mathbf{max}}$)	(37,35)	(2,-2)	(-4,-4)	(-3,1)	-
local reciprocity (**r**)	0.197	0.004	0.072	0.016	0.169
mean local reciprocity	0.228	0.004	0.091	0.025	0.133
graph reciprocity (**GR**)	0.978	0.067	0.8	0.108	0.911
History					
max degeneracy ($\delta_{\mathbf{max}}$)	(17,17)	(1,-1)	(-2,-2)	(-1,1)	-
local reciprocity (**r**)	0.065	0.004	0.010	0.020	0.055
mean local reciprocity	0.050	0.004	0.030	0.029	0.010
graph reciprocity (**GR**)	0.938	0.158	1.0	0.205	0.842
Politics					
max degeneracy ($\delta_{\mathbf{max}}$)	(64,65)	(1,-2)	(-2,-2)	(-3,1)	-
local reciprocity (**r**)	0.105	0.006	0.020	0.040	0.094
mean local reciprocity	0.059	0.006	0.067	0.048	0.014
graph reciprocity (**GR**)	0.955	0.535	1.0	0.564	0.640
Religion					
max degeneracy ($\delta_{\mathbf{max}}$)	(42,43)	(1,-2)	(-2,-2)	(-2,1)	-
local reciprocity (**r**)	0.084	0.006	0.022	0.044	0.076
mean local reciprocity	0.058	0.007	0.064	0.050	0.015
graph reciprocity (**GR**)	0.952	0.396	1.0	0.540	0.676
Mathematics					
max degeneracy ($\delta_{\mathbf{max}}$)	(46,47)	(1,-1)	(-2,-2)	(-2,1)	-
local reciprocity (**r**)	0.099	0.006	0.011	0.032	0.085
mean local reciprocity	0.063	0.006	0.037	0.029	0.012
graph reciprocity (**GR**)	0.963	0.327	1.0	0.356	0.742

Table 3.18: The calculated metrics for all graphs. The first four columns (of the calculated values) present the for contextual reciprocities (e.g. **r** in column $Q_{+,+}$ is the contextual local reciprocity in the $+,+$ quadrant r^{++}). The last column is Simple Local Reciprocity r^s or the Global Reciprocity GR(depending on the corresponding row).

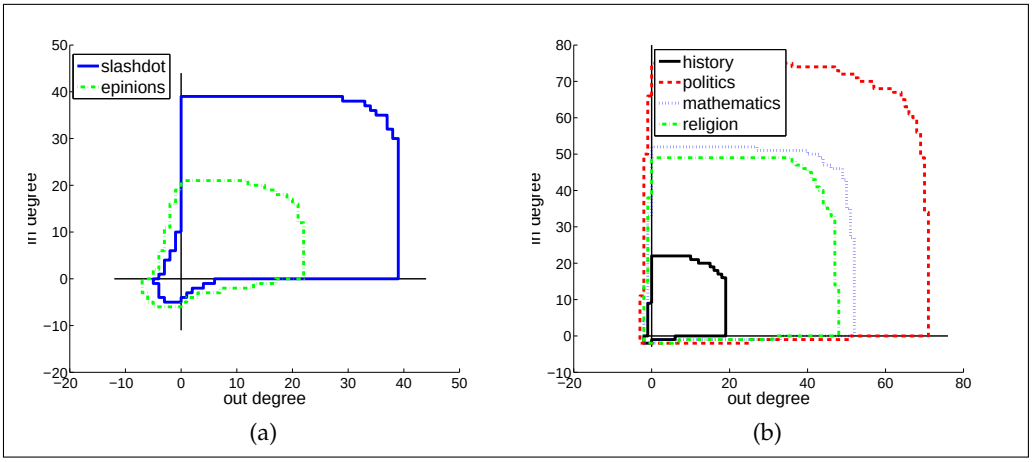


Figure 3.29: a. The frontiers for *Slashdot* and *Epinions* networks. b. The frontiers for the *Wikipedia* topics.

3.5.5 *Wikipedia topics*

The S-cores structure are analyzed for the four Wikipedia topics selected above: *politics*, *history*, *mathematics* and *religion*. In Figure 3.29b are the corresponding S-core frontiers, and in Table 3.18 the values for the defined metrics. In terms of *maximal degeneracy* the topic of *Politics* has by far the largest trust quadrant. This is expected since there is more activity in that topic, in comparison with the others, evident by the larger number of articles, resulting in a larger overall graph. On the other hand, the *History* graph has the smallest value (a direct result from the smaller number of articles). It is quite interesting that, despite the difference in size, all four networks present the same behavior in many of their aspects. The network derived from the topics under *History* seems to display a slightly different behavior. The larger value of *GR* expresses a general tendency for the users to reciprocate edges of only the same sign back to the community, which in turn can be assigned to a larger bias in the actions of a user.

Again, the node level reciprocities cannot describe the bigger picture of the community's collective actions. For example, *Contextual Global Reciprocity* GR^{++} has a high value – indicating a lot of trustworthiness coming and going from the user to the community– while at the same time the equivalent node level reciprocities for all four topics are very low. It should also be pointed out that GR^{--} has reached maximum value for all four topics as well. Meaning that, despite the fact there is less unbiased actions (evident by the higher values of GR^{+-} and GR^{-+} and with the exception of *History*), there is a (small) remaining part of the community where distrust is at its maximum – but not directly as the node level *contextual reciprocities* values are very small.

3.5.6 *Local vs. Global reciprocity*

It is visible from the comparison in Table 3.19, that node level reciprocity, although it captures somewhat the different trends in trust and distrust, it can not evaluate the wider concept of general ("social") trust/distrust. This is more visible in quadrant $Q_{+,+}$ as reciprocity r^{++} is low in most of the cases while the respective GR^{++} is close to maximum. The large value of *quadrant maximal degeneracy* $\delta_{max}^{(+,+)}(G)$ indicates that there is a strong community of individuals that reciprocate their trust with their community.

In Table 3.18 are also compared the *Graph Reciprocities* to the average local reciprocities over all vertices in the presented data graphs. The motivation for this is to establish that *Global Reciprocity* represents the reciprocal behavior better than the local one. The *local reciprocities* are ratios over the edges while the *average local reciprocities* are the mean values over individual behaviors and could be perhaps better at capturing the average behavior over the entire graph. From the comparison in Table 3.18, it is clear that:

a) the values of *average local reciprocities* and of *local reciprocities* display more or less the same trends and

b) *Graph Reciprocity* can capture collective behaviors the other two metrics cannot.

It is also visible that, when the visual information of the frontier is reduced to sets of numbers, the descriptive capability of the s-core frontier is also reduced. In many cases, the relative (to its bounding box) volume of the $Q_{-,-}$ quadrant is equal to 1 (for the *Wikipedia* articles). This can be interpreted by a sub-community of distrustful members; although the total volume is quite small meaning that the members are not that many. This would be quite easy to interpret both by the size and shape of that portion of the frontier. Again at this quadrant, there isn't much of agreement on the magnitude with the equivalent node level reciprocity values. This suggests, again, that the distrust is not directly reciprocated at a individual member but at the community (or part of it) as a whole.

3.5.7 *General Trends of Graph Reciprocity*

When looking at the set of GR^{st} it is noticeable that all of the signed networks seem to display a very high reciprocity in trust $Q_{+,+}$ and distrust $Q_{-,-}$. This is also evident by the respective high values of the *Global Reciprocity* GR. The behavior these high values express can be described as biased. In a real graphs it is expected that edge signs distributions are not uniform as the users tent to be reciprocal in the context of trust signs: a user tends to give and receive same

sign trust to/from the community or another user. It is unlikely that users receive positive trust and return negative or vice versa. Thus in a real world signed graph the graph generation mechanism is highly biased in terms of sign distribution. Is is of interest to study the values and behaviors of the metrics that have been defined in the case of graphs where signs are assigned in an unbiased manner.

This exploration is attempted in artificially created networks with the assumption that in an unbiased behavior the sign of an edge would be equally probable for either positive or negative. For easier comparison, a single structure of a real graph (*Epinions*) is utilized while changing, in a random manner, the assignment of trust signs. Firstly, the signs are assigned while keeping the original ratio of positive to negative signs. Afterwards, different ratios are explored and their following affect on the signed reciprocity aspect of the graph is noted. Random sign assignment should result in a less biased reciprocity of signs. Specifically, it is expected (and verified by the results) that the model for general *Graph Reciprocity* to always result in an equally balanced reciprocity for all quadrants – without being affected by the ratio of signs – and thus always to produce (in an unbiased scenario) for GR a value close to 0.5.

In Figure 3.30a one can see the effect, after the redistribution of signs, upon the S-core frontier. As seen, in the green dotted line, the portion of the inversely reciprocal quadrants has become larger even though we have not changed the structure of the graph and we have kept the same ratio of positives to negative edges. It is quite clear, just from the image, that on a graph level the reciprocity is balanced for the random redistribution.

In fact, the reciprocity at graph level always indicates the unbiased behavior (with $GR \approx 0.5$) as seen in Figure 3.30b – and as it we will be seen with particular values. In Figure 3.30b the visualization represents different scenarios of random sign distribution for different percentages of positive and negative signs. Starting from the lower left part, cases of larger portions of negative signs can be seen (with the extreme lower left frontier having only negative signs) and moving up and to the right to larger portions for positive signs (likewise the extreme "right-up" frontier has only positive signs). With the exceptions of purely positive/negative signs, the global reciprocity will always be $GR \approx 0.5$ as all the quadrants reach approximately maximum "capacity" in their respective bounding boxes. As all of this cases are unbiased, it is to be expected to be characterized by the same reciprocity values. At each case, the maximal degeneracy values differentiate to indicate where the general attitude of trust/distrust is "pointing".

It is essential to point out that in Figure 3.30b the frontier moves upon the diagonal of $Q_{+,+}$ and $Q_{-,-}$ because only the signs are changed and not the structure of the network. Thus the same relationships remain and only their nature changes. In fact the two most extreme cases of full positive/negative signs are essentially a mirror of one another and have the exact same frontier shape.

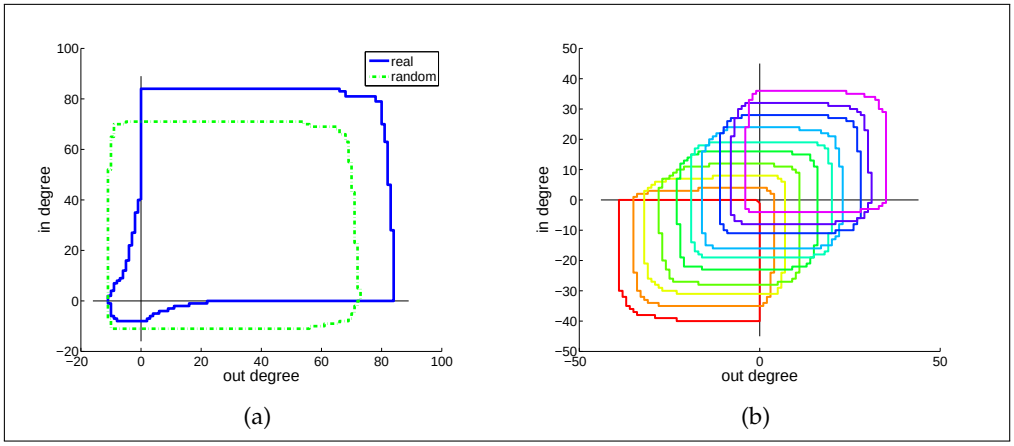


Figure 3.30: a. The frontiers for *Epinions* originally and with random redistribution of the signs (while keeping the same sign ratio). b. The frontiers for *Epinions* with random sign distribution for varying positive/negative ratios.

Graph	Reciprocity	$Q_{+,+}$	$Q_{+,-}$	$Q_{-,-}$	$Q_{-,+}$	r^s/GR
Original	*local*	0.347	0.003	0.038	0.022	0.302
	graph	0.976	0.109	0.886	0.197	0.859
Random¹	*local*	0.263	0.045	0.046	0.261	0.231
	graph	0.961	0.932	0.966	0.962	0.504
Random²	*local*	0.154	0.153	0.154	0.154	0.154
	graph	0.971	0.961	0.985	0.980	0.501

Table 3.19: The calculated metrics for different sign distributions in *Epinions*. $\mathtt{Random}^1$ is the random distribution of signs while keeping the original ratio and $\mathtt{Random}^2$ is the random distributions of signs while + and - have the same probability.

Additionally in more detail are studied two cases of random distribution:

a $\mathtt{Random}^1$ where the sign ration is the same as the original and

b $\mathtt{Random}^2$ where the sign ratio is equal for both signs (i.e. the S-core frontier centered at the axes in Figure 3.30b).

The reciprocities (local and global) are compared to the original network of *Epinions* in Table 3.19. As it can be seen, the local reciprocity does not offer an intuitive result (i.e. a result that would signify that signs have been assigned in an unbiased manner) unless the ratio of signs is also balanced. At the same time, it is validated that the unbiased behavior will be clearly evident by the global reciprocities (GR^{st} and GR) no matter what the ratio of signs may be.

3.5.8 *Author Frontiers*

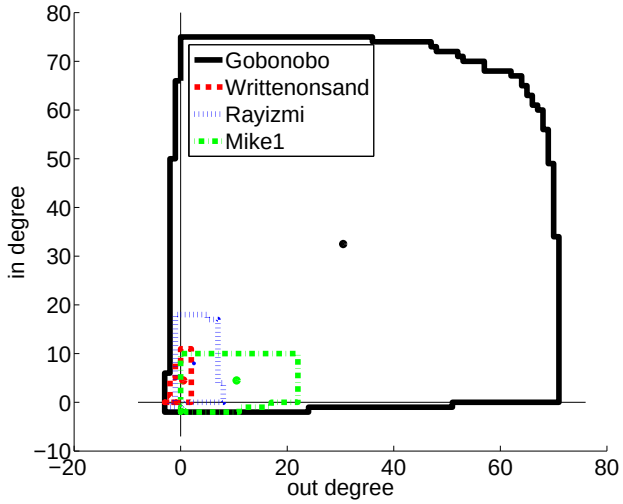


Figure 3.31: Comparison between *Wikipedia* contributor frontiers randomly selected from the politics topic.

As seen in section 2.4.6, a frontier can be defined for individual nodes. This frontier could be used as an evaluation tool for individuals in a trust network. Here, is presented an example of a such an evaluation of *Wikipedia* contributors and we show different trends that might appear.

In Figure 3.31 are presented frontiers for four randomly chosen users from the domain of politics in *Wikipedia*. The frontier of a user is similar to the frontier of graph and it gives a visual representation of the user's trustworthiness. In the example cases appearing here, user *Gobonobo* has his frontier almost as large as the entire politics graph frontier (Figure 3.29b). Thus being an author that has had many interactions and many of them positive establishing him at a high status of trustworthiness. A counter example is the user *Writtenonsand* which displays a very small frontier.

Furthermore, with the frontiers a global "pictures" is given of the users' interaction with the community and the general reciprocation from the community as a whole. For example, for the user *Mike1*, the frontier is extended more on the positive out degree which is interpreted as an author that votes positively more than he is being voted. On the other hand, user *Rayizmi* has a more introvert behavior on the positive side.

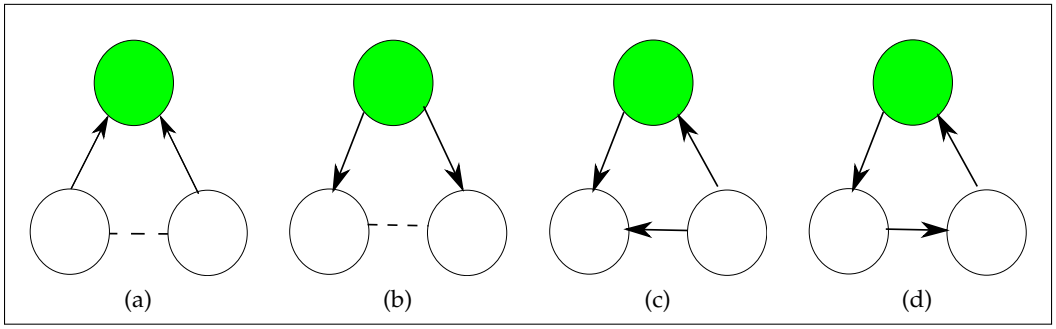


Figure 3.32: The four possible triangle configurations (for a node – the green one) in a directed network. The dashed line indicates that the direction is not important (as the two possible directions create a "mirror" of one another). From left to right:a) In-triangle, b) Out-triangle, c) through triangle, d) cycle triangle.

3.5.9 S-*core reciprocity vs clustering structure*

While reciprocity (global or local) is an intuitive measure and can be used to describe important behavioral aspects of signed networks, it is also important to explore the properties these metrics (that we defined here) might have in relevance to more complex tasks. In this section the correlation between triangles formed by the nodes of signed networks and the properties of reciprocity are explored. The number of triangles a node participates is important to tasks like graph clustering. As graph clustering in signed networks is still quite unexplored, the results presented here could be the seed for further research in graph mining algorithms of signed networks.

In Figure 3.32 there is an example of the four possible combinations of edges in a triangle formation of a directed graph. From left to right:

a the in-triangle 3.32a,

b the out-triangle 3.32b,

c the through triangle 3.32c and

d the cycle triangle 3.32d.

In a signed network the types of triangles (for a given node) is sixteen. In the cases studied here though, the types of triangles can be seen as four sets of the four directed types in Figure 3.32. Each set is the aforementioned four types that is "restricted" by the "properties" imposed from each quadrant. For example, for $Q_{+,+}$ we have the four types displayed in Figure 3.32 where the incoming edges must be positive and the outgoing as well. While for $Q_{+,-}$ only the outgoing must be positive and the incoming negative.

In Table 3.20 is displayed the correlation between a node's triangle count and the respective reciprocities of the nodes for each quadrant. Much like the frontier

for each individual node, the global reciprocity (GR) for that node can be calculated for each of the quadrants. Afterwards we measure the correlation coefficient between a nodes triangle count (four types for each quadrant) and the GR. All the values displayed in Table 3.20 have $p - value < 0.05$ (from comparison to the no correlation null hypothesis) thus indicating that the correlation is significant.

Quadrant	in-triangle	out-triangle	through triangle	cycle triangle
$Q_{+,+}$				
GR^{++}	0,56	0,79	0,96	0,87
r^{++}	0,29	0,09	0,08	0,09
$Q_{-,+}$				
GR^{-+}	0,17	0,26	0,60	0,47
r^{-+}	0,10	0,02	0,01	0,02
$Q_{-,-}$				
GR^{--}	0,05	0,79	0,97	0,87
r^{--}	0,46	0,03	0,04	0,04
$Q_{+,-}$				
GR^{+-}	0,17	0,19	0,42	0,36
r^{+-}	0,10	0,01	0,04	0,03

Table 3.20: Correlation coefficient values between the four types of triangles and reciprocities for each quadrant.

The global reciprocity can be computed by the cores of a node's S-core frontier and a connection has been found between the k-core properties of an undirected graph and its clustering coefficient (e.g. see [42]).

And, Even though the local reciprocity of a node is not connected by intuition to clustering properties, we also presented the correlation of the local reciprocity of a node to the number of triangles for comparison and completeness of the exploration.

It is very interesting that, with the exception of the in-triangles, the global reciprocity presents a consistently higher that the local one correlation with the triangles count. In the in-triangle case the reason for this inconsistency (in the $Q_{-,-}$) could be that newcomers into a social network –since they are new to the network and less knowledgeable and/or more eager to participate– are more likely to receive negative votes due to inexperience (which in turn could have a direct reciprocation with a negative vote for the same reason). Never the less the high correlation of GR with triangle count is an indication that GR could be used in signed network clustering (for perhaps a selection phase of seed nodes). Further-

more this indicates the validity and superiority of the GR measure over the local reciprocity.

3.6 CONCLUSIONS

This section displayed various uses of the extended degeneracy concept. Capitalizing on the core structure and models and metrics derived from that :

- In the case of graphs that do their edges do not display directionality : collaboration was evaluated and, with the fractional core methodologies, a more meaningful framework was established for the evaluation of individuals on their role within a community.
- In the case of directed graphs: the collaboration aspect was extend in concepts of inward /"egotistical" and outward/"unselfish" behaviors. This was displayed in diverse environments (Wikipedia articles and DBLP citation graph) thus establishing the multiple purposes this tool can be utilized for.
- The case of signed digraphs "transformed" the collaboration aspect into a trust management context where the evaluation shows properties of signed networks, both explicit and implicit ones, that are better captured by the new reciprocity measures, suggesting its potential as an objective for optimization algorithms in the context of directed and/or signed graphs, such as graph clustering or link formation models.

All of the above were displayed to be applicable in both individuals and various sets of them or communities.

Part III

DEGENERACY AND CLUSTERING

4

Scaling Graph Clustering with the k-core Expansion Sequence

4.1 Introduction

Detecting clusters or communities in graphs constitutes a cornerstone problem with many applications in several disciplines. Characteristic application domains include social and information network analysis, biological networks, recommender systems and image segmentation. Due to its importance and multidisciplinary nature, the problem of graph clustering has received great attention from the research community and numerous algorithms have been proposed (see [38] for a survey in the area).

Spectral clustering (e.g., [74]) is one of the most sophisticated methods for capturing and analyzing the inherent structure of data and can have highly accurate results on different data types such as data points, images, and graphs. Nevertheless, spectral methods impose a high cost of computing resources both in time and space regardless of the data on which it is going to be applied [38]. Other well-known approaches for community detection are the ones based on modularity optimization [24, 73], stochastic flow simulation [79] and local partitioning methods [38]. In any case, scalability is still a major challenge in the graph clustering task, especially nowadays with the significant increase of the graphs' sizes in various networks.

Typically, there are two main methodologies for scaling up a graph clustering method:

i algorithm-oriented and

ii data-oriented.

The first one considers the algorithm of interest and appropriately optimizes – whenever is possible – the "parts" of the algorithm responsible for scalability issues. Prominent examples here are the fast modularity optimization method [24] and the scalable flow-based Markov clustering algorithm [79]. The second and widely used methodology is to rely on sampling/sparsification techniques. In this case, the size of the graph onto which the algorithm will operate is reduced,

by disregarding nodes/edges (see Section 4.2). However, in this approach possibly useful structural information of the graph (i.e., nodes/edges) is ignored.

This chapter presents the proposal of a graph clustering framework that capitalizes on the notion of graph degeneracy – also known as k-core decomposition [36, 66, 81, 85]. The main idea behind this approach is to combine any known graph clustering algorithm with an *easy-to-compute*, *clustering-preserving* hierarchical representation of the graph – as produced by the k-core decomposition – towards a scalable graph clustering tool. The k-core of a graph is a maximal size subgraph where each node has at least k neighbors in the subgraph (the k will be also referred as the *rank* of such a core). That way, the notion of the *core expansion sequence* (also known as layers) $V_k, \ldots, V_0$ is defined, where V_k contains the vertices of the highest-rank core, V_{k-1} contains the vertices of the $(k-1)$-core that do not belong in V_k, and so on (see section 2.3.1, definition 3 for formal definitions).

Based on the idea of degeneracy, its shown that the densest cores of a graph are roughly maintaining its clustering structure and thus constitute *good starting points* (seed subgraphs) for computing it. Given the fact that the size of the densest core of a graph is orders of magnitude smaller than that of the original graph, a clustering algorithm is applied starting from its densest core and then, on the resulting structure, the rest of the nodes in the lower rank cores are clustered incrementally in decreasing order – following the hierarchy produced by the k-core decomposition. It is shown experimentally that this process is considerably improving the execution time of the clustering process (using as baseline a spectral clustering method), while the quality of the clustering results is maintained or even, in some cases, is improved.

The rest of this section is organized as follows. Firstly are reviews of the related work. Afterwards follows a description of the proposed framework for graph clustering. Finally is the presentation of the experimental results.

4.2 RELATED WORK

In this section there is a review on the related work regarding graph clustering, approaches for scaling-up graph clustering.

GRAPH CLUSTERING

The problem of community detection and graph clustering has been extensively studied from several points of view. Some well-known approaches include spectral clustering (e.g., [74, 83, 89]), modularity optimization (e.g., [24, 72]), multilevel graph partitioning (e.g., Metis [48]), flow-based methods [79], hierarchical methods [73] and many more. A very informative and comprehensive review over the different approaches can be found in [38]. Also Fortunato et al. [54] has conducted

a comparative analysis on the performance of some of the most recent algorithms, in artificial data produced by their parametrized generator of benchmark graphs. In the work presented here, the same graph generator as in [54] is used to evaluate the proposed framework. Another recent empirical comparison of community detection algorithms has been performed by Leskovec et al. [58]. There, due to lack of ground-truth data, the evaluation of the produced clusters is achieved applying quality measures, such as conductance. As we will be see in Section 4.4, a similar practice is used to evaluate the framework in real-world graph data.

SCALING-UP GRAPH CLUSTERING

The efficiency of graph clustering can be improved in various ways. Two well-known approaches are the ones of *sampling* and *sparsification*. In the case of spectral clustering, sampling-based approaches include the Nyström method [53] and randomized SVD algorithms [32].

The approach of [53] capitalizes on the Nyström column-sampling methods and, from the insights gained by the comparison, a novel technique, that follows a non uniform selection of columns, is presented and outperforms existing techniques. In [32] the proposed randomized SVD is essentially picking a few columns of the data matrix with respect to a certain probability distribution and then doing Principal Component Analysis on the selected features

Concerning graph sampling, the goal is to produce a graph of smaller size (nodes and edges), preserving a set of desired graph properties (e.g., degree distribution, clustering coefficient) [55].The work by Maiya and Berger-Wolf [63], presents a method – based on the notion of expansion properties – to sample a subgraph that preserves the community structure, i.e., contains representative nodes of the communities . Then, the community membership of the nodes that do not belong to the sample can be expressed as an inference problem.

Unlike the aforementioned methods that sample both nodes and edges, the graph sparsification algorithm presented in [80] reduces only the number of edges (focusing on inter-community edges) in order to improve the running time of a clustering algorithm. In contrast to the above methods, our approach keeps the structure of the graph intact, without excluding any structural information from the clustering process.

A different approach for sampling can be found in the work of [91] where spectral clustering is applied on the centroids produced by simpler clustering algorithms (k-means and random projection trees), the assignment, of the other data points, in the final clusters is decided by their correlation to the centroids. The same idea is found in [23] where “Landmarks" are chosen, with the utilization of k-means, as representatives of the original data. A similar approach, using a simple clustering algorithm, is reported in [62], where the optimization is carried out by minimizing the cost of calculating the similarity matrix based on an estimation

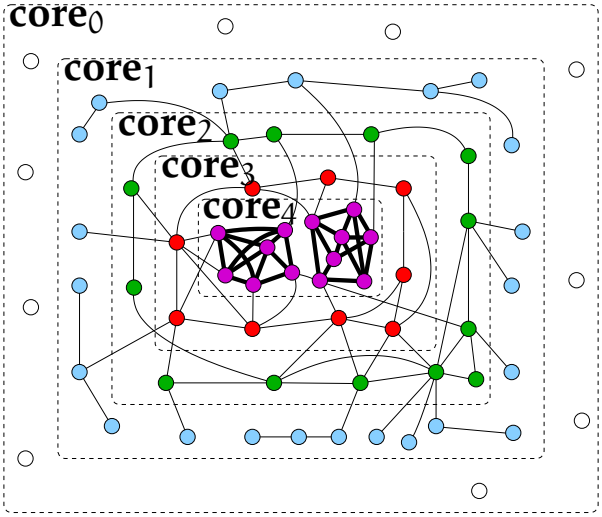


Figure 4.1: A graph G of degeneracy 4 and its cores. The different colors express the partition of the vertices of the graph to layers V_4, V_3, V_2, V_1, and V_0. Fat-edges indicate parts of a clustering of the graph.

process of the similarities between two points. The basic idea is that if in multiple runs of k-means – with random initialization – two data points are assigned to the same cluster, then those two points have a high similarity.

In a completely different spirit, clustering techniques can also be improved by parallelizing the process like in the work of [22]. Here the idea was to conduct spectral clustering on a distributed environment by “breaking down" the eigenvector computation in terms of partitioning the similarity matrix.

4.3 THE PROPOSED METHOD

In this section, the proposed graph clustering framework is described. The proposed methodology capitalizes on the concept of degeneracy to improve the efficiency of graph clustering. The main idea behind this approach is that the k-core decomposition preserves the clustering structure of a graph and therefore the “best” k-core subgraph can be used as good starting point for a clustering method. Furthermore, the decomposition provides an hierarchical organization of the nodes in the graph, that can “guide” the clustering process.

A simple toy example of a graph can be seen in Figure 4.1. There can be seen a visual representation of the intuition behind the proposed frame work. As it can be seen, there are two “core” clusters in purple color. Once every layer (other than the last one) is removed then the two basic communities are easily separable. The rest of the nodes can be easily assigned (in this example) to the two clusters with simplistic procedures.

4.3.1 *The Framework*

Spectral Clustering is a method that requires significant computing resources as it involves matrix eigenvalue decomposition. Many sophisticated clustering algorithms have a high computational complexity as well.

Suppose there is an algorithm that takes as input a graph G and outputs a partition of $V(G)$ into a number of sets that form a clustering of G. As in this section the main focus is on the general aspects of the methodology, the attributes of such an algorithm are discussed any further and it is, abstractly, named it **Cluster**. It is also assumed that it runs in $O(n^3)$ steps as, in our experiments, the **Cluster** algorithm is the spectral algorithm of [74]. The aim is to define a procedure that uses **Cluster** and accelerates the algorithm without any significant loss in its accuracy. This procedure is called *CoreCluster* and is presented below together with the subroutines that it uses.

Procedure *CoreCluster*(G).
Input: A graph G.
Output: A partition of $V(G)$ into clusters.

1. $k := \delta^*(G)$.
2. $q := 0$.
3. Let $V_k, \ldots, V_0$ be the core expansion sequence of G.
4. For $i = 0, \ldots, k$, let G_i be the i-core of G,
5. Let $S_k = V_k$.
6. Let $\mathcal{A}_k = \{C_1^k, \ldots, C_{\rho_k}^k\} =$ **Cluster**$(G[S_k])$.
7. **for** $i = k-1$ **to** 0 **do**
8. $\quad S_i =$*Select*$(G_i, \mathcal{A}_k \cup \ldots \cup \mathcal{A}_{i+1}, V_i)$,
9. $\quad$ let $\mathcal{A}_i = (C_1^i, \ldots, C_{\rho_i}^i) =$ **Cluster**$(G[S_i])$.
10. **Return** $\mathcal{A}^k \cup \cdots \cup \mathcal{A}^0$.

Initially, *CoreCluster* performs k-core decomposition to obtain the core expansion sequence of the graph. Then, algorithm **Cluster** is applied to the k-core subgraph, creating the first clusters. The procedure *Select* takes as input the, so far, created clusters, i.e., the sets in $\mathcal{F}_{i+1} = \mathcal{A}_k \cup \cdots \cup \mathcal{A}_{i+1}$ and the i-layer V_i and tries to assign each of the vertices of V_i in some cluster in $\mathcal{A}_k \cup \cdots \cup \mathcal{A}_{i+1}$. After this update, the procedure *Select* returns the unassigned vertices.

The choice of the selection procedure considers the way the vertices of V_i are adjacent with the vertices of the clusters in $\mathcal{A}_k \cup \cdots \cup \mathcal{A}_{i+1}$. As this selection can be done with several heuristic approaches,it is not specified it in this section. A detailed description of such a procedure is given in Section 4.3.2 where the experimental part of this work is presented.

The *CoreCluster* can be essentially seen as a "meta-algorithmic procedure" in the sense that it can be applied to any clustering algorithm. The discussion that follows in the next two sections, argues that this indeed can improve the clustering argument in time without any significant expected loss in its performance.

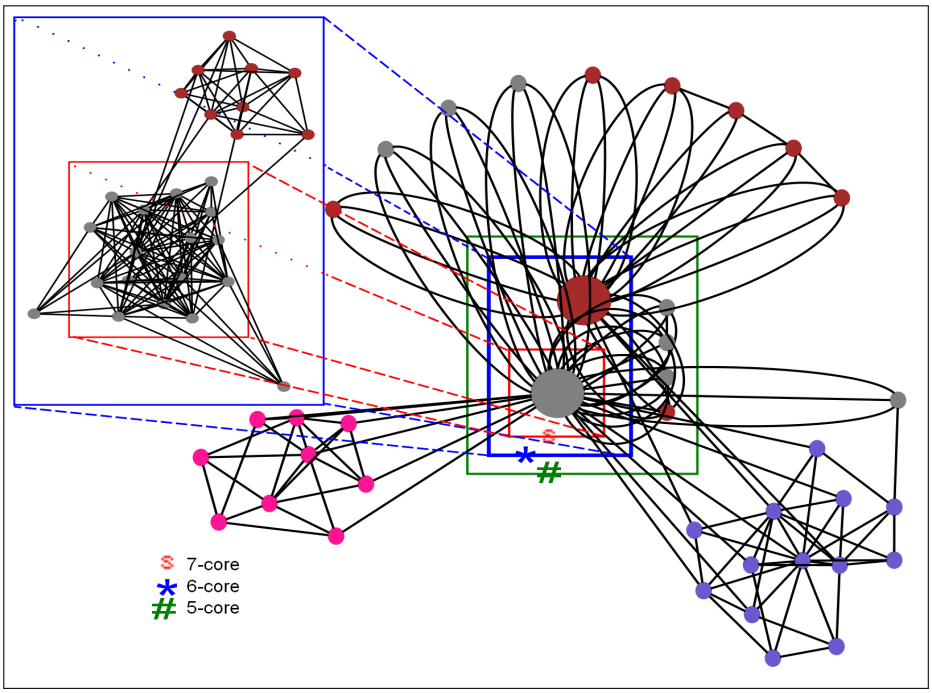


Figure 4.2: An example of the operation of the *CoreCluster* procedure for a portion of a graph obtained from the experiments. This graph consists of the core sequence members V_7, V_6, V_5, and V_4 that are included in the black, green (#), blue (∗), and red (s) squares respectively. Vertices of different colors correspond to different clusters.

4.3.2 *Selection procedure*

In this section, the selection procedure (*Select*()) (in Line 8 of *CoreCluster*) is described. The procedure takes as input the so far created clustering $\mathcal{F}_{i+1} = \mathcal{A}_k \cup \cdots \cup \mathcal{A}_{i+1}$ and the vertex set V_i. We describe below how this procedure assigns some of the vertices of V_i to the clusters in $\mathcal{F}_{i+1}$ and outputs the remaining ones.

First of all a pair $(G, V, \mathcal{F})$ is called a *candidate triple*, if G is a graph, and $\mathcal{F} \cup \{V\}$ is a partition of $V(G)$. Given a candidate triple $(G, V, \mathcal{F})$, the following property is defined on the vertices of V:

$$\mathbf{P}^{\alpha,\beta}(v) = \exists C \in \mathcal{F} : \frac{|N_G(v) \cap V(C)|}{|N_G(v)|} \geqslant \alpha \quad \text{and} \quad |N_G(v)| \geqslant \beta, \tag{4.1}$$

where $\alpha > 0.5$ and β is a positive integer. Notice that, as $\alpha > 0.5$, the truth of $\mathbf{P}^{\alpha,\beta}(v)$ can be certified by a unique set C in $\mathcal{F}$. We call such a set the *certificate* of v. This defined parameter has fixed values through the experiment that were found experimentally to provide better results over all data.

Procedure *Select*$(G, \mathcal{F}, V)$.
Input: A candidate triple $(G, V, \mathcal{F})$
Output: A subset S of V and a partition $\mathcal{F}'$ of $V(\mathcal{F}) \cup (V \setminus S)$.

1. while $\mathbf{P}^{\alpha,\beta}(v)$ is true for some $v \in V$,
2. set $\mathcal{F} \leftarrow (\mathcal{F} \setminus \{C\}) \cup \{C \cup \{v\}\}$ where
3. C is the certificate of v
4. and set $V \leftarrow V \setminus \{v\}$.
5. set $V^1 = N_G(V(\mathcal{F}))$ and $V^2 = V \setminus (V^1 \cup \mathcal{F})$.
6. if V^2 is either empty or an independent set of G,
7. then
8. $\mathcal{F} \leftarrow$ *assign*$(G, \mathcal{F}, V, V^1)$
9. $\mathcal{F} \leftarrow$*assign*$(G, \mathcal{F}, V, V^2)$
10. return $\emptyset$
11. else return $V^1 \cup V^2$.

Before we the *assign* routine is provided, there is a need for some definitions. Given a candidate triple $(G, \mathcal{F}, V)$ and a vertex $v \in V$ the following is defined

$$\mathbf{span}(v) = \max\{|N_G(v) \cap V(C)| \mid C \in \mathcal{F}\}. \tag{4.2}$$

Also defined, $\mathbf{argspan}(v)$ as a minimum size $C \in \mathcal{F}$ with the property that

$$|N_G(v) \cap V(C)| = \mathbf{span}(v). \tag{4.3}$$

Procedure *assign*$(G, \mathcal{F}, V, S)$.
Input: A candidate triple $(G, V, \mathcal{F})$ and a subset S of V
Output: A partition $\mathcal{F}$

1. while $S \neq \emptyset$,
2. let $l = \max\{\mathbf{span}(v) \mid v \in S\}$
3. let $L = \{v \in S \mid \mathbf{span}(v) = l\}$
4. for every $v \in L$,
5. set $C' = C \cup \{v\}$ where $C = \mathbf{argspan}(v)$
6. set $S \leftarrow S \setminus \{v\}$
7. set $\mathcal{F} \leftarrow (\mathcal{F} \setminus \{C\}) \cup \{C'\}$
8. return $\mathcal{F}$.

The selection procedure first tries to assign vertices of V to clusters of $\mathcal{F}$ using the criterion of the property $\mathbf{P}^{\alpha,\beta}$ that assigns a vertex to a cluster only if the vast majority of its neighbors belong in this cluster. The quantification of this "vast majority" criterion is done by the constants α and β that, in the implementations regarding the presented experiments, are chosen to be $\alpha = 0.8$ and $\beta = 5$ (this choice appears to work optimally in our experiments). This first selection is done in lines 1–4 of procedure *Select*. The vertices that cannot be assigned are partitioned into two groups: V^1 contains those that have neighbors in vertices that are already classified in the clusters of $\mathcal{F}$ and V^2 contains the rest. As the vertices in V^2 have no neighbors in the clusters, they have at least k neighbors out of them, it is most likely that they may not enter to any existing cluster in the future, unless, possibly, they are completely disjoint. If this is not the case, a further (milder) classification is attempted by the *assign* procedure that first classifies the vertices in V^1 in the existing clusters and then we do the same for the vertices in V^2. We stress that this last selection has been useful in our experiments in cores of low rank (where many independent vertices may appear). The procedure *assign* is a heuristic that classifies each vertex to the cluster that has the majority of its neighbors.

In simple terms, the *CoreCluster* framework applies the **Cluster** algorithm at the highest k-core and then it iterates from the highest to the lowest core trying to apply the following logic: Assign with a simple criterion, all the nodes that can be assigned, to the existing clusters and apply the **Cluster** algorithm to the remaining nodes (in order to create new clusters).

Examples of the choices of the selection procedure can be extracted by Figure 4.2 that is a instance of a graph derived from our experimental data (in particular from the D1 dataset – see section 4.4.2). In this graph, the 7-core (i.e., the graph delimited by the red square) consists exclusively of the vertices of the "grey" cluster. The set V_6 contains all the vertices of the 6-core that are not in the 7-core. All but two of these vertices are sparsely connected with the grey cluster, while they exhibit a strong interconnection between them. Therefore, they form the red cluster, while the remaining two vertices, that have all of their neighbors in the gray cluster, become a member of it. A similar assignment to the gray and red clusters is happening for the vertices in V_5, i.e., the vertices in the 5-core that do not belong in the 6-core. Finally, the remaining vertices of the graph (the vertices in V_4) either clearly belong to the existing clusters or are forming two more clusters, the ping and the purple ones. The experimental evaluation indicates that a similar clustering behavior characterizes the entire data set that has been considered and that this is indeed captured by the *CoreCluster* procedure.

4.3.3 *Expected time*

Procedure *CoreCluster*(G) runs algorithm **Cluster**(G) on the subgraphs induced by the subsets $S_k, \ldots, S_0$. Each of S_i is the subset of vertices of V_i that cannot be assigned to already existing clusters according to the selection procedure (Step 8). That way, the selection step makes it possible for some of the vertices in V_i to be incorporated in the already computed clusters, reducing the burden of the computation of **Cluster**$(G[S_i])$. However, the real speed up of the algorithm is based on the fact that *CoreCluster*(G) now runs in $k+1$ disjoint subgraphs of G instead from G itself. As the i-th selection phase requires $O(|V(G_i)|^3)$ steps, the running time of *CoreCluster*(G) is bounded by

$$\sum_{i=k,\ldots,0} O(|S_i|^3) \leqslant \sum_{i=k,\ldots,0} O(|V_i|^3) \leqslant O(k \cdot n_{max}^3), \tag{4.4}$$

where $n_{max} = \max\{|V_k|,\ldots,|V_0|\}$. In the above bound, the first equality holds only in the extremal case where no selection occurs during the selection phases. Clearly, the general bound in (4.4) is the best possible when $|V_k|,\ldots,|V_0|$ tend to be equally distributed (which would accelerate the running time by a factor of k^2).

According to the first inequality of (4.4) the running time of the algorithm is proportional to $(k+1) \cdot n_{max}^3$ where $n_{max} = \max\{|S_k|, \ldots, |S_0|\}$. Let $\rho_G = \max\{\frac{|V(G|)}{|S_i|} \mid i = 0,\ldots,k\}$ and $\mu_G = \max\{\frac{|V(G)}{|V_i|} \mid i = 0,\ldots,k\}$ and observe that $\rho_G \geqslant \mu_G$. Notice that the discrepancy between ρ and μ is a measure of the acceleration of the algorithm because of the selection phases. Concluding, the acceleration of *CoreCluster* is upper bounded by

$$\sum_{i=k,\ldots,0} O(|S_i|^3) = O\left(\frac{k}{\rho^3} \cdot n^3\right). \tag{4.5}$$

The above estimation is purely theoretical and its purpose is to expose the general complexity contribution of our algorithmic machinery. In practice, the acceleration given by the *CoreCluster* framework can be *much better* and this also depends on the heuristics that are applied for the selection phase (see Section 4.3.2).

4.3.4 *Quality of the CoreCluster framework*

The intuition behind of this framework is is that the core expansion sequence $V_k, V_{k-1}, \ldots, V_0$ gives a good sense of direction on how to do clustering in an incremental way. In fact, the initial guess $(C_1^k, \ldots, C_{\rho_k}^k)$ is a way to divide the "densest" part of G to clusters. After that, the procedure considers V_{k-1} as the remaining vertices of the $(k-1)$-core G_{k-1}, and tries to assign them one by one to the already existing clusters $C_1^k, \ldots, C_{\rho_k}^k$. The vertices for which such an as-

signment is not possible, form the set S_{k-1} and the **Cluster** is now applied on $G[S_{k-1}]$.

As the algorithm continues, the existing clusters grow up and the vertices for which this is not possible, are grouped to new clusters. The fact that this procedure approximates satisfactorily the result of the application of **Cluster** to the whole graph is justified by the observation that the early i-cores (i.e., i-cores where i is close to k) are already dense, and therefore *sufficiently coherent*, to provide a good starting clustering that will expand well because of the selection criterion. In fact, the subgraphs obtained by the k-core decomposition, provide an $(1/2)$-approximation algorithm for the Densest-Subgraph problem [5].

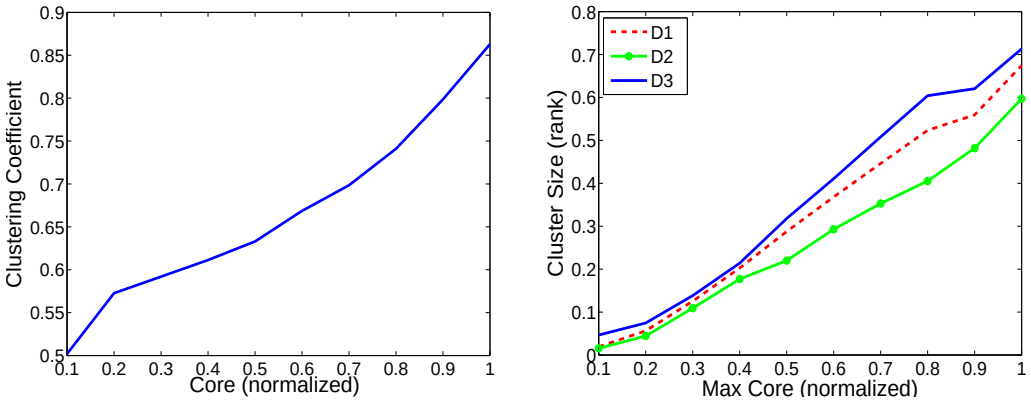
This makes it quite unexpected that an initial cluster of $\mathbf{core}_i(G)$ (where $i = k, \ldots, k-q$, for moderately small values of q) is split into two parts of different clusters of the whole graph. Therefore, the ordering of the core expansion sequence is indeed correctly indicating in which parts of the graph one should look for "good guesses" of the densest parts of the clustering.

Figure 4.3 depicts the above intuition for the datasets we used for our experiments (as they are described in Section 4.4.2). Figure 4.3a displays the average clustering coefficient of a k-core subgraph with regards to the core index value k (normalized by the maximum k of each graph). This indicates that, overall, a clearer clustering can be given at the maximum k-core.

In Figure 4.3b the "survival" behavior of the clusters with regards to the core index(k/k_{max} for each graph) is depicted. The clusters of each graph are ranked by size giving to the largest cluster a size rank value of 1.0 and then assigning to the rest of the clusters a rank of $(ClusterSize_i)/MaxSize$ for each $cluster_i$ within a graph. Then, each $cluster_i$ is tracked to see how far it can be traced in the k-core layers. It can be observed that the rank of a cluster (i.e., size) is also positively correlated with the core index. In conclusion, the above two observations lead us to the perception that the maximum k-core will hold the vertices from the biggest clusters and at the same time those clusters will be the easiest to distinguish from one another. Next, the above observations are theoretically justified.

4.3.4.1 *Theoretical justification*

The goal of this subsection is to provide a theoretical justification of the *CoreCluster* framework. The objective is to show that at the best k-core of the graph, i.e., for $k = \delta^*(G)$ (graph degeneracy), the "best" clusters of the graph G are preserved and therefore they can be used as *seed subgraphs* for a clustering algorithm. The claim is that the decomposition identifies subgraphs that progressively correspond to the most central regions and connected parts of the graph. This can be shown using the measure of local clustering coefficient [88] of the nodes in the graph: nodes with high clustering coefficient at G, are those who finally "survive" at the k-core



(a) (b)

Figure 4.3: a. Average clustering coefficient compared to k-core index normalized by the maximum value. b. Cluster size, normalized by maximum size, compared to "survival" k-core index.

subgraph by the pruning (decomposition) procedure. Typically, these nodes are representative of the clustering structure in the original graph, in the sense that they belong to the best clusters, and therefore the k-core decomposition preserves the clustering structure of a graph.

The claim is based on the following Theorem:

Theorem 1 ([42]). *Let* $G = (V, E)$ *be a graph with heavy-tailed degree distribution, and let* C_G *be the (global) clustering coefficient of* G*. Then, there exists a* k*-core in* G *for* $k \geqslant C_G \frac{d_{max}^{\beta}}{2}$*, where* $\beta < 1$ *is a constant such that most edges are incident to a node with degree at least* d_{max}^{β} *(typically* $\beta = 2/3$*), where* d_{max} *the maximum degree of the nodes.*

The above Theorem implies that graphs with heavy-tailed degree distribution and high global clustering coefficient C_G, have large cores. Next is proven that the claim for the relationship between the local clustering coefficient C_v, $\forall v \in V(G)$ and the k-core subgraph; that way, the selection of the k-core is justified as good seed subgraph for starting the clustering procedure.

Claim 1. *Let* G *be a graph with heavy-tailed degree distribution. The contribution of each node* $v \in V(G)$ *to the* k*-core decomposition of the graph is proportional to the local clustering coefficient* C_v*.*

Proof. The global clustering coefficient C_G for the entire graph is given by the average of the local clustering coefficients C_v, $\forall v \in V(G)$, i.e., $C_G = \frac{1}{n}\sum_v C_v$, where $n = V(G)$. Then, from Theorem 1 we have that:

$$k \geqslant C_G \frac{d_{max}^{\beta}}{2} = \left(\frac{1}{n}\sum_v C_v\right) \frac{d_{max}^{\beta}}{2} = \underbrace{\left(\frac{1}{n}\frac{d_{max}^{\beta}}{2}\right)}_{\alpha} \sum_v C_v$$

$$\Rightarrow \; k \geqslant \alpha \sum_v C_v,$$

where parameter α captures some global characteristics of the graph (that depend on the total number of nodes and the maximum degree). Therefore, nodes with high clustering coefficient (in the original graph) are more likely to be found it the best k-core ($k = \delta^*(G)$) subgraph, since they tend to be more robust to the degeneracy process. □

Based on the above claim, the nodes that survive in the k-core subgraph potentially capture the best clusters of the graph, due to their high local clustering coefficient (in the entire graph). Thus, the k-core subgraph can be used as good starting point for the clustering task.

4.4 EXPERIMENTAL EVALUATION

In this section are presented the experiments that were carried out using the framework on several real and synthetic graph datasets. Initially, the consistent and significant execution time amelioration – that was achieve with the framework – is verified (compared to a baseline method) and next the quality of the produced clustering results is measured and compared for several graph datasets.

4.4.1 *Spectral algorithm*

As the baseline and the basis for the *CoreCluster* framework (algorithm **Cluster**), the Ng-Jordan-Weiss spectral clustering algorithm is used as it is described in [74]. The basic idea of this algorithm is to keep the top k eigenvectors of the normalized adjacency matrix and perform k-means clustering on the rows of the matrix composed from these eigenvectors. Each row in the new matrix corresponds to a data point and usually the k is the same as the number of clusters we are looking for. In this variation k-means++ [7] is used for its advantage of performing better seeding during the initialization process and, since an automatic choice for k is desired, k is define by noticing the "eigen gap" as it is suggested in [77].

4.4.2 *Datasets description*

While real networks are the objective, actual datasets lack ground truth which leaves only evaluation metrics of the quality of clustering as an option and not direct comparison. On the other hand, artificial networks offer ground truth and a large variety of properties that can be parametrized to produce different "types" of networks.The evaluation of the framework is conducted on both real and artificial networks in order to have complete and decisive results.

4.4.2.1 *Artificial Networks*

The graph generator by Fortunato [54] is exploited to produce graphs with a clustering structure which is available to the tester (ground truth). This graph generator provides a wide range of input parameters. The parameters in Table 4.1 are used and tuned them various combinations in order to get a wide range of graphs with different features. Thus, the testing is credible as it is evaluated in essentially hundreds of graphs with different properties and quality of clustering structure. The parameters used are:

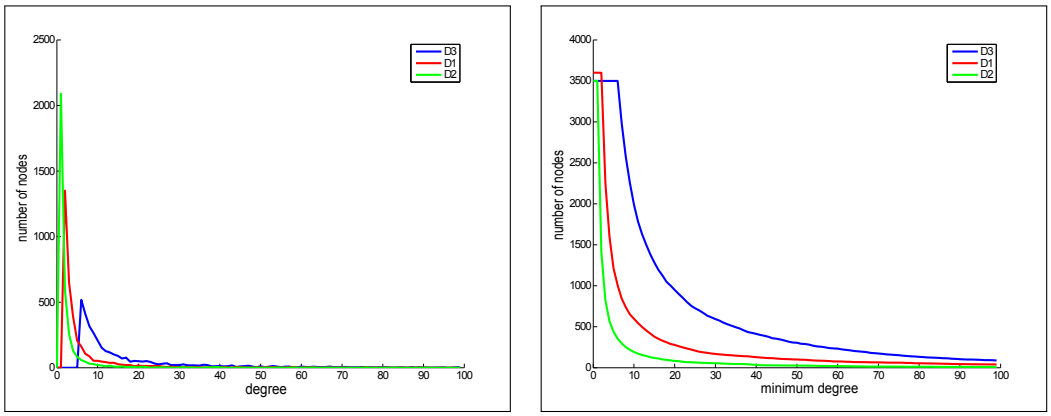
- N is the size of the graph,
- max_d is the maximum node degree,
- min_d is the minimum node degree and
- μ is the mixing parameter representing the overlapping between clusters, i.e., each node shares a fraction $1-\mu$ of its links with the other nodes of its community and a fraction μ with the other nodes of the network.

The graphs produced by the generator contain inherent clusters and the cluster assignment is offered by the generator, enabling thus usage of these data sets for evaluating graph clustering algorithms.

Table 4.1 depicts the various parameters' values for the three main different settings in our experiments. As we see the most important parameter is the min_d, as it is the one differentiating the overall density of the graph. In Figure 4.4 we see the link distribution of the produced datasets.

4.4.2.2 *Real Networks*

In the case of real networks evaluations are performed to a subset of the *Facebook 100* dataset [86]. This is a collection of friendship networks of *Facebook* from 2005, for 100 US Universities (i.e., 100 individual networks). The evaluations were not performed to the full extend of this dataset as hardware limitation did not allow an evaluation, with spectral clustering, of networks with more than 13K nodes



(a) (b)

Figure 4.4: a. Link Distribution in the D1, D2, D3 data sets b. minimum Link Distribution in the D1, D2, D3 data sets.

	D1	D2	D3
max_d(node max degree)	10%, 30%, 50%,	10%, 30%, 50%	200 edges
min_d (node min degree)	~5 (the absolutely minimum)	7	20
μ (mixing parameter)	1% – 43% (in 7 equal steps)	3% – 43% (in 6 equal steps)	3% – 43% (in 6 equal steps)
N (graph size in nodes)	600–3600	3500–5500	3500–5500

Table 4.1: Parameters' values for the artificial graphs.

(the *CoreCluster* framework could handle much larger networks). About half of the networks from this dataset were used for the final evaluation.

4.4.3 *Time performance*

It is evident that the gain in execution time using the proposed framework is very significant (i.e., at least three orders of magnitude for graph sizes above 3000 nodes) and increases exponentially with the graph size. Figures 4.5a - 4.5d(d) depict the specific execution times for the three data sets D1, D2, D3 and *Facebook* in that order. On the other hand, as on can see in Figure 4.6, the execution time of the proposed framework increases rather linearly with the graph size – thus showing good scaling features.

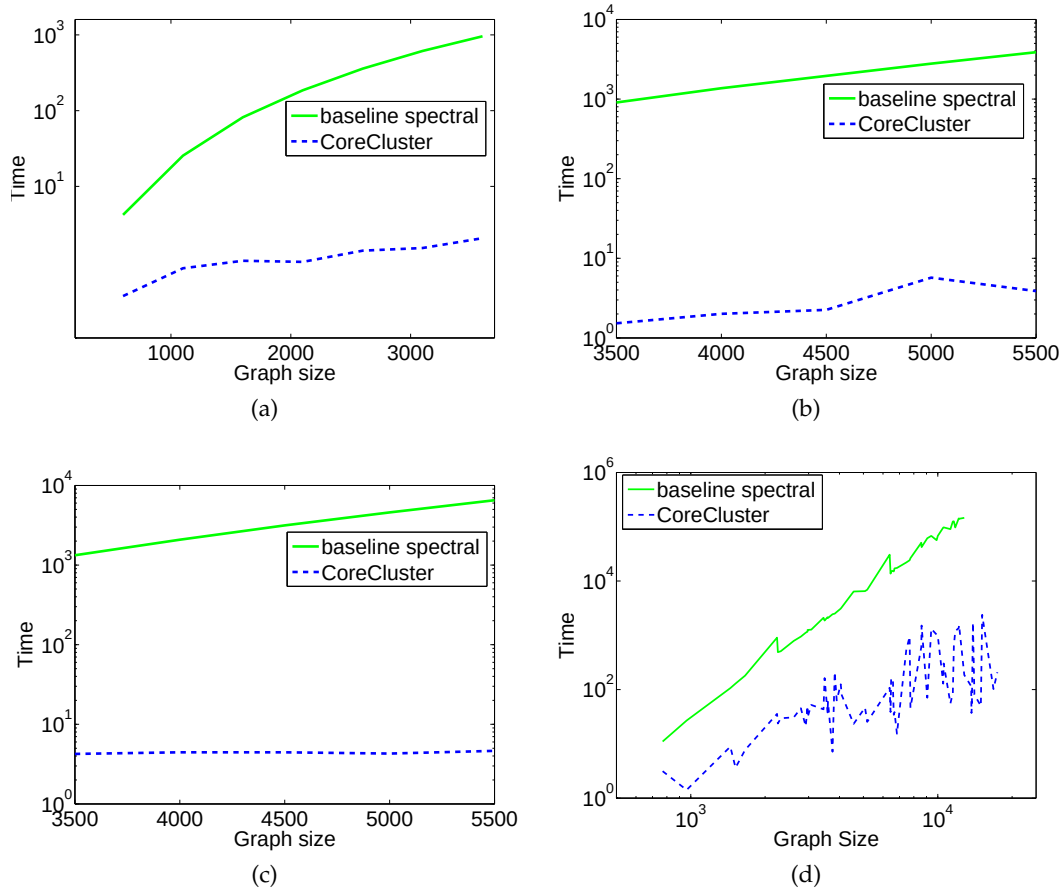


Figure 4.5: Execution time of base line spectral graph clustering and of our framework for various graph sizes for the data sets a.D1, b.D2, c.D3 and d.Facebook respectively.

4.4.4 *Clustering quality evaluation*

The experimental setup is the following. For each of the graphs at hand are run:

i as baseline approach, the Ng-Jordan-Weiss [74] spectral graph clustering algorithm and

ii the *CoreCluster* framework on the datasets described in Section 4.4.2.

Following, are described the methods and metrics for evaluating the results on artificial and real networks.

ARTIFICIAL NETWORKS

The quality of the clustering results in are measured terms of the widely used Normalized Mutual Information [65] (NMI). In Figure 4.7, displays the comparative performance (in terms of NMI values) of the plain spectral algorithm as compared to the performance of our framework (*CoreCluster*) for different graph

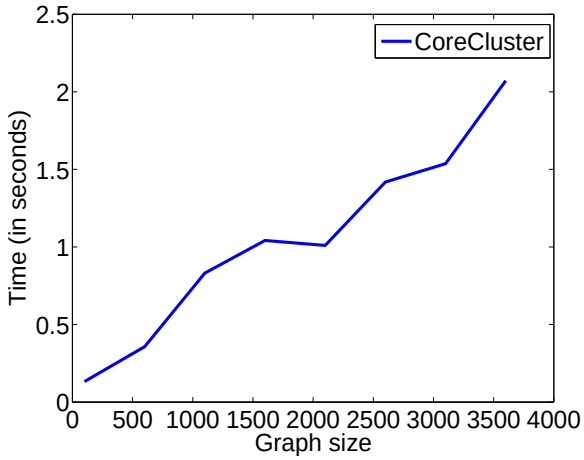
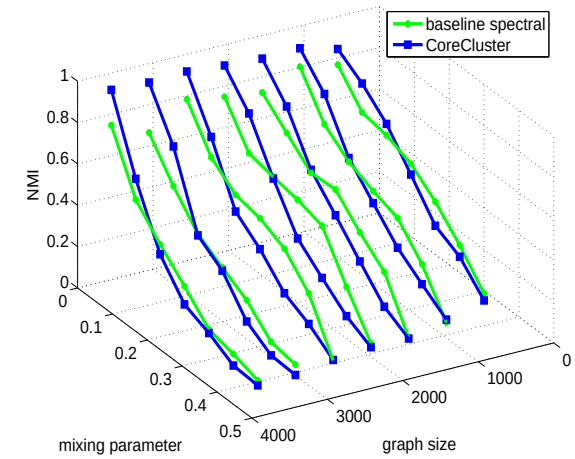


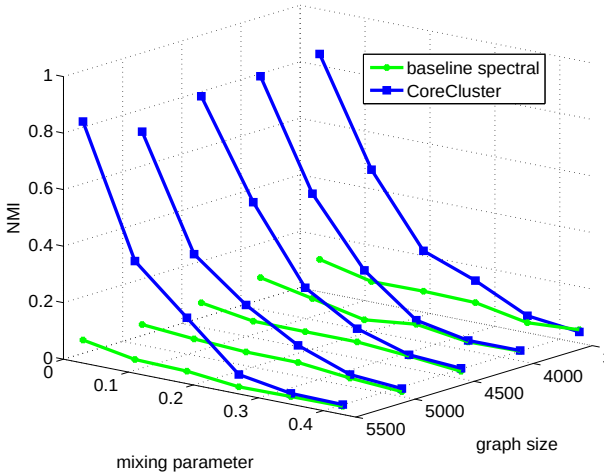
Figure 4.6: Time execution of the *CoreCluster* framework for various graph sizes.

sizes and mixing parameter values. Each point represents the average NMI value for all the graphs produced for each different combinations of parameter values, whose ranges appear in Table 4.1. In order to promote statistical significance of the results for each of the aforementioned combinations, the generator was run ten times and the corresponding graphs were computed. It is noticeable that the proposed approach performs almost perfectly (with NMI>0.92) and generally outperforms the quality of the spectral algorithm applied directly on the graphs in most cases, especially as the graph size grows (this happens for mixing parameters values generally smaller than 0.2). For larger values of the mixing parameter, plain spectral clustering performs better (even though the absolute quality is low). Of course the counterargument here is that for larger values of the mixing parameter the overlap of the clusters is such that it basically prevents the definition of a clustering structure – and therefore perhaps it is meaningless to search for clusters in these cases.

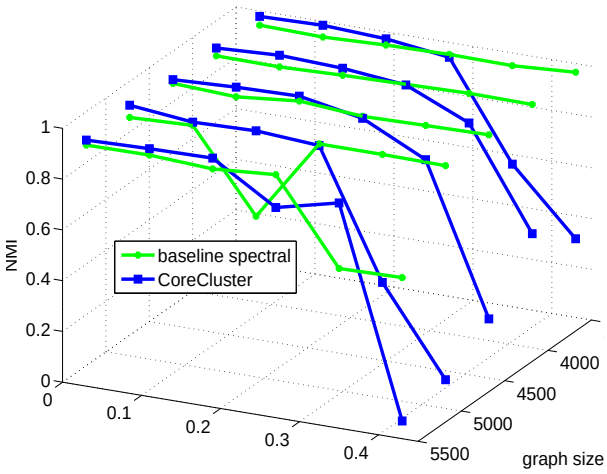
The proposed algorithm performs even better – in comparison to the spectral algorithm – in dataset D2, which resembles well real graphs as they are known to have low $\mathtt{min}_d$ value. In this case (see Fig. 4.7b), *CoreCluster* performs excellently (NMI values close to 1) for small values of the mixing parameters while, on the contrary, the plain spectral algorithm performs really bad (NMI values close or inferior to 0.2). The deterioration of performance when μ (mixing parameter) increases is expected as the clustering structures vanish.



(a)



(b)



(c)

Figure 4.7: Clustering quality comparison in terms of NMI values for the graph datasets a.D1, b.D2 and c,D3.

Another set of experiments was performed with the dataset D3 matching the parameters in [54], where a series of different algorithms are compared. This dataset is characterized by a high number of minimum links per node (20, see Table 4.1) – resulting in a relatively dense graph. In this case, (see Fig. 4.7c the performance of plain spectral clustering was very good for a wide range of mixing parameter values. The *CoreCluster* algorithm is matching or outperforming the plain spectral clustering performance for small values of mixing parameters while its performance decreases significantly for larger mixing parameter values. This is explained by the fact that the graph is quite dense, which makes the usage of the k-core structure questionable, as the partitioning on which the *CoreCluster* procedure is based is poor for lower values of k. Of course, it has to be stressed that, in all the above cases, the execution time of *CoreCluster*, especially for large graphs, is 3-4 orders of magnitude smaller that those of the plain clustering algorithms – achieving essentially the same or even better clustering quality for reasonable values of the mixing parameters.

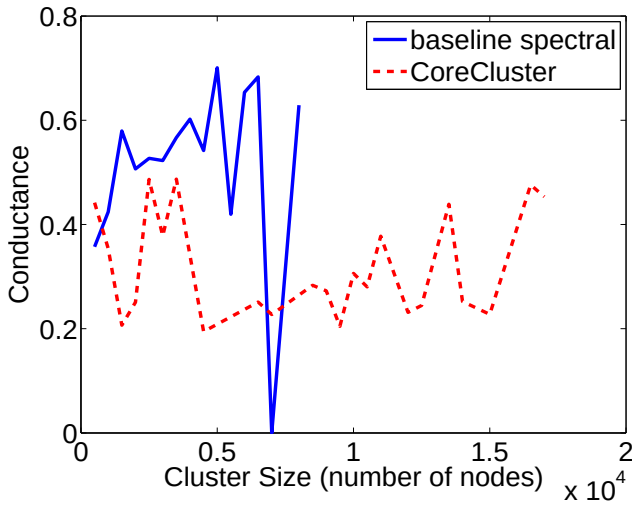


Figure 4.8: Clustering quality comparative performance (*Facebook100*) in terms of conductance (lower values are better).

FACEBOOK100

The networks of this dataset lack ground truth, and for this reason the results are evaluated with the evaluation criterion of conductance. Given a graph G and a cut $(S, \overline{S})$ conductance is defined as

$$\phi(S) = \frac{\sum_{i \in S, j \notin S} A_{ij}}{\min(a(S), a(\overline{S}))},$$

where A_{ij} are the entries in the adjacency matrix $\mathbf{A}$ of G and

$$a(S) = \sum_{i \in S} \sum_{j \in G} A_{ij}$$

Informally, conductance measures (for a cluster) the ratio of internal to external connectivity. It has been used widely to examine clustering quality (e.g., [58]) and has a simple and intuitive definition. In Fig. 4.8, the reader can see the comparison of conductance values versus different sizes of detected communities by the two methods. Conductance has values in the range $(0, 1)$ with lower values indicating better clustering quality.

For better presentation (in Fig. 4.8) the detected cluster sizes (in terms of number of nodes) have been aggregated into bins of 500 (e.g., $0 - 500$, $501 - 1000$, etc.) and the average conductance for each bin has been provided. This plot essentially provides the comparison of average clustering quality between the baseline and *CoreCluster* for different cluster sizes. Before commenting the comparison, it is important to note that – for both methods – clusters with less than 10 nodes were excluded as they were trivia with regards to the clustering criteria for large scale graphs. Moreover, *CoreCluster* was evaluated to a larger subset of *Facebook100* including networks that could not be evaluated with the baseline spectral, due to limitations of hardware memory. Consequently, results exist of clusters up to 8K nodes (from networks of up to 13K of nodes) for the baseline and results of clusters up to 16K (from networks of up to 23K of nodes) for *CoreCluster*.

Moving on to the comparison, in Fig. 4.8 we can see that *CoreCluster* displays better clustering quality than the baseline, with the exception of the first bin. The difference there is negligible and only sightly surpassed by the baseline's conductance value. For the last two bins of the baseline, we should note that there was only one cluster found for each with the one having 0 conductance consisting of the entire network (i.e., the whole graph was found as one cluster). In fairness, we could consider an "in between" value but it would be still worse than the corresponding conductance of *CoreCluster*. Overall, *CoreCluster* displays a quite low conductance regardless of cluster size. From the results, the argument can be made that *CoreCluster* provides better clustering than the baseline in a much faster time.

4.4.5 *Degeneracy features vs. running time*

In this subsection, the gain in execution time is discussed with regards to the graph features – especially with regards to its size deterioration in the application of the k-cores algorithm. As stated in Section 4.3.3, running the spectral clustering algorithm in the k-cores, with k decreasing from k_{max} to 1 , decreases the execu-

tion time significantly – as in Relation (4.4). In Section 4.3.3, it is also stressed that if the core sizes change in roughly equal parts, as k decreases from k_{max} to 1, the execution time is accelerated by a factor of k_{max}^2. Here the above statement is verified experimentally. Indeed the *CoreCluster* method was run on each k-core G_k (k ranging between k_{max} and 1) and the running time was measured for graphs having different degeneracy features – with regards to their size decay. The intuition is that the closer to linear is the decay rate, the faster will be the *CoreCluster* method. The k-core size decay is presented for three different graphs, selected as representative from the D1, D2 and D3 sets respectively, such that they are of the same size (3500) and that they present relatively high quality of NMI value - meaning that the clustering structure is sound and is computed correctly by the clustering algorithm. The other graph parameters are: D1 (μ:0.15, min_d:6, max_d:350), D2 (mu:0.19, min_d:<5, max_d:1000) and D3 (mu:0.19, min_d:20, max_d:200). The contribution of the selection phase to the running time is also analyzed.

The results appear in Fig. 4.9 where the reader can see the normalized (k-core size/original graph size) k-core size decay with regards to the normalized core index (k/k_{max} for each graph). It is clear that D3 and D1 decays behave closer to a linear mode which would ideally speed up the proposed algorithm, while D2 has a clearly more exponential behavior. Thus, the ranking of the three graphs with regards to proximity to linear decay behavior is: D1, D3, D2, with D1 and D3 being practically very close to each other. It is interesting to observe that the *CoreCluster* procedure execution time on these graphs verifies the original intuition, as well as the findings of Section 4.3.3. Also, it is inverse to the decay ordering above: the execution time for D1 (0.86 sec) and D3 (1.1 sec) are significantly smaller that the respective of D2 (2.9 sec).

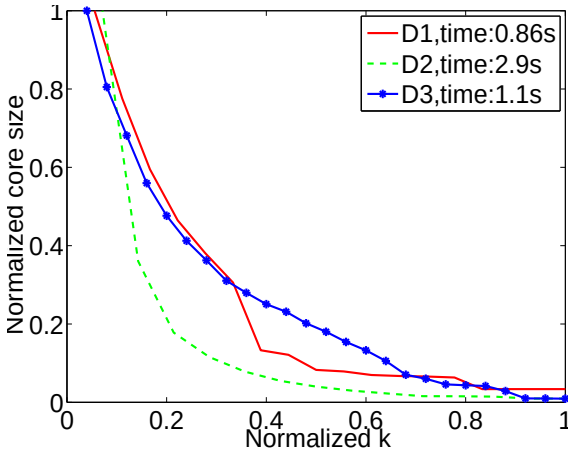


Figure 4.9: Normalized k-core behavior vs. time: Core size decay for representative D1, D2, D3 graphs with core clustering execution times.

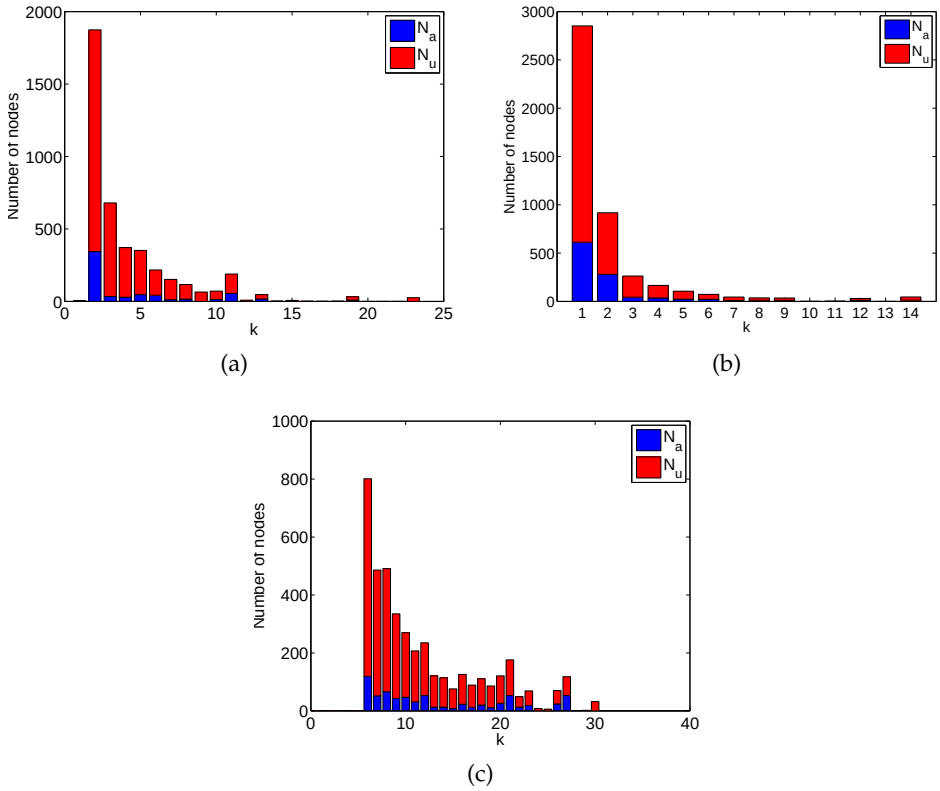


Figure 4.10: Core-clustering incremental load for each step of the core expansion sequence.

For the above mentioned graphs the incremental load to the core-clustering algorithms is presented for each step of the core expansion sequence as it evolves with k (the k-core index). This load has two constituents:

a the nodes assigned to existing clusters (N_a)

b the unassigned nodes that are the input to the spectral clustering algorithm (N_u).

In Figure 4.10 one can see, for each graph and for each step in the core expansion process, the additional load to the core clustering algorithm consisting of the N_a and N_u sets of nodes. There it is evident that, in all cases, there is a significant number of nodes that are processed in the last step ($k = 2$ for D1, $k = 1$ for D2, $k = 6$ for D3). In the case of D1 and D2 – that are sparse as the values of min_d are 6 and 5 respectively – the loads for dense cores is minimal (i.e., $k > 3$) and drops exponentially with k. On the other hand, for D3 which is a dense graph (as $min_k = 20$), the incremental load is almost linear with k and the portion of non assigned nodes is in all cases significant – contributing thus to decreasing the execution time of the clustering. It is worth stressing that the execution times reported above for the core clustering algorithm (D1: 0,86 sec, D3: 1.1 sec D2: 2.9 sec) are well explained by the distribution of values in the above figure. Indeed in the

case of D2 that has by far the largest execution time the algorithm has to process in the last step an additional load of 2800 nodes (about 800 of them are directly assigned to clusters while the rest 2000 have to be processed with the expensive spectral clustering algorithm). On the contrary, for D1, the additional largest load is about 1900 nodes (from which ≈ 400 are assigned immediately to clusters). Finally, in the case of D3, the last-step additional load ($k = 6$) involved only about 800 nodes (≈ 100 of which we assigned immediately). On the other hand, for D3, the next steps accumulate a additional remarkable cost for the core expansion subsequence from 7 to 21, for which the additional load is not negligible (in the area of few hundreds nodes each time).

4.4.6 *Conclusion*

An effort for optimizing the efficiency of graph clustering has been articulated here. This is achieved by capitalizing on the intuition that the extreme k-core a graph preserves the clustering structure of the original graph, while it is much faster to execute clustering on this degenerate graph due to its much smaller size. The key points of this work are:

- **Graph Clustering Framework.** The *CoreCluster* framework that initiates clustering on the highest rank core of the graph and then incrementally clusters the graph's nodes in the subsequent lower rank cores. Furthermore, *CoreCluster* could be potentially combined with any clustering algorithm.
- **Performance Analysis.** An analytical description on why the *CoreCluster* framework scales-up the clustering process and why the k-core decomposition provides good starting subgraphs – that preserve the clustering structure – for the clustering task.
- **Experiments.** A validation of the framework through experiments on a multitude of synthetic (with diverse properties) and real world graphs. The framework is decreasing the execution time of the clustering process by orders of magnitude, especially as the graph's size increases, while the quality of the clustering results is not compromised or even improving.

5

Epilogue

5.1 conclusions

Cohesion and collaboration in graphs are cornerstone features for the evaluation of complex networks, especially with the advent of large scale applications such as the Web, social networks, citations graphs etc. In this dissertation, methodologies and metrics are introduced for the evaluation these concepts on both at the macroscopic (graph or sub-graph) and at the microscopic (node) level of communities. While a variety of measures has been defined to that end, the ones defined and utilized throughout this work take into account the collective collaborative nature of networks with community structure –a property not captured by individual node metrics or by other community evaluation metrics. This is achieved by capitalizing on the degeneracy features of the networks.

Degeneracy has been explored into simple graphs with the k-core structure. Even though this is sufficient for simplistic graph models of real networks, there is a need of greater expressibility for social (and other type of) networks that contain more complex relationships than a simple/equal binary one. For this reason the core concept – and "accompanied" structures: core sequence, cells, cell sequence – has been extended to more complex graph structure. Specifically, k-cores have been extended to weighted (fractional k-cores), directed (D-cores) and signed directed (S-core) graphs. Each type, of the aforementioned ones, signifies different relationships between individual members of the networks. Thus the concept of collaboration is adapted to the semantics of each occasion; e.g. in signed networks collaboration can also be seen as trust/distrust evaluation.

The extensions of degeneracy displayed great interest by themselves but also created a rich setting for new concepts and structures of visualizing and organizing the graph structure (e.g core frontiers, collaboration indices etc). The novel structures, metrics and methodologies were articulated firstly in a theoretical manner, adopting valid terminology from related work, and then explored experimentally on real-world large scale networks which provided interesting and intuitive results.

Finally, as the connection between community structure and the core concept is very close, the application of the k-core structure in community detection was

explored by utilizing it as a preprocessing step and a heuristic function in order to accelerate a graph clustering algorithm of high complexity.

5.2 FUTURE DIRECTIONS ON GRAPH MINING AND DEGENERACY

The methodologies for community evaluation that were developed have a great potential for future applications (as it was shown in the Data Exploration part) but have also a great potential for further research. In particular :

- Altering the weight function to a combination of various factors. For example including the H-Index of an author could be used to evaluate authors or communities (cores) where the collaborative strength is mixed with one's standing in research.
- The exploration of a weighted digraph is also interesting. Extending degeneracy to weighted digraphs, would potentially deal with issues equivalent to the undirected case and would also create a more sophisticated and detailed model for collaboration evaluation.
- Another subject still unexplored is the temporal evolution of cores (of any extension) to capture collaboration evolution and other aspects on how communities might grow.
- As k-cores were used for optimizing the execution time of clustering simple graphs, D-cores (and the other core extensions as well) could be used for the same reason on directed graphs (or the equivalent graph for the other extensions). In the D-core case the decomposition of the graph is two-dimensional thus creating an area where one might seek to find the best "path" to use in the clustering procedure.
- The case of the signed networks can also spark a few new directions.
  - Firstly there is the "logical" step of utilizing a weight function over the signs.
  - Since clustering in signed networks is still quite new as a research field, it would be a big contribution to develop a such a framework that would based on degeneracy.
  - Finally, there is a great selection of models for link formation on directed graphs. Based on those equivalent models could be built for signed networks in order to be used in order to test metrics and evaluation tools like degeneracy. This would be of great use as there are not a lot of real-world signed network data available for such purposes.

List of Figures

List of Tables

Bibliography

[1] Leman Akoglu, Pedro O. S. Vaz de Melo, and Christos Faloutsos. Quantifying reciprocity in large weighted communication networks. In *PAKDD (2)*, 2012.

[2] J. Ignacio Alvarez-Hamelin, Luca Dall'Asta, Alain Barrat, and Alessandro Vespignani. Large scale networks fingerprinting and visualization using the k-core decomposition. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 41–50, Cambridge, MA, 2006. MIT Press.

[3] José Ignacio Alvarez-Hamelin, Luca Dall'Asta, Alain Barrat, and Alessandro Vespignani. k-core decomposition: a tool for the visualization of large scale networks. *CoRR*, cs.NI/0504107, 2005.

[4] Yuan An, Jeannette Janssen, and Evangelos E. Milios. Characterizing and mining the citation graph of the computer science literature. *Knowledge and Information Systems*, 6:664–678, 2004. ISSN 0219-1377.

[5] Reid Andersen and Kumar Chellapilla. Finding dense subgraphs with size bounds. In *WAW*, pages 25–37, 2009.

[6] T. Antal, PL Krapivsky, and S. Redner. Dynamics of social balance on networks. *Physical Review E*, 72, 2005.

[7] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *SODA*, pages 1027–1035, 2007.

[8] Gary D. Bader and Christopher W. V. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, pages –1–1, 2003.

[9] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[10] Albert-Laszlo Barabasi, Reka Albert, and Hawoong Jeong. Scale-free characteristics of random networks: The topology of the world-wide web, 2000.

[11] Vladimir Batagelj and Andrej Mrvar. Pajek - analysis and visualization of large networks. In *Graph Drawing*, volume 2265 of *Lecture Notes in Com-*

puter Science, pages 8–11. Springer Berlin / Heidelberg, 2002. ISBN 978-3-540-43309-5.

[12] Vladimir Batagelj and Matjaz Zaversnik. Generalized cores. *CoRR*, cs.DS/0202039, 2002.

[13] Vladimir Batagelj and Matjaz Zaversnik. An o(m) algorithm for cores decomposition of networks. *CoRR*, cs.DS/0310049, 2003.

[14] Michael Baur, Marco Gaertler, Robert Görke, Marcus Krug, and Dorothea Wagner. Generating Graphs with Predefined k-Core Structure. In *Proceedings of the European Conference of Complex Systems (ECCS'07)*, October 2007.

[15] Béla Bollobás. The evolution of sparse graphs. In *Graph theory and combinatorics (Cambridge, 1983)*, pages 35–57. Academic Press, London, 1984.

[16] Bela Bollobas, Christian Borgs, Jennifer Chayes, and Oliver Riordan. Directed scale-free graph. In *Proc. 14th ACM-SIAM Symposium on Discrete Algorithms*, pages 132–139, 2003.

[17] Bélaa Bollobás and Oliver Riordan. The diameter of a scale-free random graph. *Combinatorica*, 24:5–34, January 2004. ISSN 0209-9683. doi: 10.1007/s00493-004-0002-2.

[18] Ulrik Brandes, Patrick Kenis, Jürgen Lerner, and Denise van Raaij. Network analysis of collaboration structure in wikipedia. In *International conference on World wide web, (WWW)*, 2009.

[19] Pierce G. Buckley and Deryk Osthus. Popularity based random graph models leading to a scale-free degree sequence. *Discrete Mathematics*, 282:53–68, 2001.

[20] Shai Carmi, Shlomo Havlin, Scott Kirkpatrick, Yuval Shavitt, and Eran Shir. Medusa - new model of internet topology using k-shell decomposition, 2006.

[21] Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *Proceedings of the Third International Workshop on Approximation Algorithms for Combinatorial Optimization*, APPROX '00, pages 84–95, London, UK, 2000. Springer-Verlag. ISBN 3-540-67996-0.

[22] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y. Chang. Parallel spectral clustering in distributed systems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(3):568–586, 2011.

[23] Xinlei Chen and Deng Cai. Large scale spectral clustering with landmark-based representation. In *AAAI*, 2011.

[24] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70(6):066111, Dec 2004. doi: 10.1103/PhysRevE.70.066111.

[25] Colin Cooper and Alan Frieze. A general model of web graphs. *Random Struct. Algorithms*, 22:311–335, May 2003. ISSN 1042-9832. doi: 10.1002/rsa.10084.

[26] Chavdar Dangalchev. Residual closeness in networks. *Physica A: Statistical Mechanics and its Applications*, 365(2):556 – 564, 2006. ISSN 0378-4371.

[27] Cristobald de Kerchove and Paul Van Dooren. The pagetrust algorithm: How to rank web pages when negative links are allowed? In *SDM*, 2008.

[28] Reinhard Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, third edition, 2005. ISBN 978-3-540-26182-7; 3-540-26182-6; 978-3-540-26183-4.

[29] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin. Structure of Growing Networks with Preferential Linking. *Physical Review Letters*, 85(21):4633–4636, November 2000. doi: 10.1103/PhysRevLett.85.4633.

[30] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes. k-core organization of complex networks. *PHYS.REV.LETT.*, 96:040601, 2006.

[31] Eleni Drinea, Eleni Drinea, Mihaela Enachescu, Mihaela Enachescu, Michael Mitzenmacher, and Michael Mitzenmacher. Variations on random graph models for the web.

[32] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Mach. Learn.*, 56(1-3), 2004. ISSN 0885-6125.

[33] David Easley and Jon Kleinberg. *Networks, crowds, and markets*, volume 8. Cambridge Univ Press, 2010.

[34] Leo Egghe. Theory and practise of the *g*-index. *Scientometrics*, 69(1):131–152, 2006.

[35] P. Erdős and A. Rényi. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 5:17–61, 1960.

[36] Paul Erdős. On the structure of linear graphs. *Israel J. Math.*, 1:156–160, 1963. ISSN 0021-2172.

[37] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. *SIGCOMM Comput. Commun. Rev.*, 29 (4):251–262, August 1999. ISSN 0146-4833. doi: 10.1145/316194.316229.

[38] Santo Fortunato. Community detection in graphs. *Phys. Rep.*, 486(3-5):75–174, 2010. ISSN 0370-1573.

[39] Eugene C. Freuder. A sufficient condition for backtrack-free search. *J. Assoc. Comput. Mach.*, 29(1):24–32, 1982. ISSN 0004-5411.

[40] Diego Garlaschelli and Maria I. Loffredo. Patterns of link reciprocity in directed networks. *Physical Review Letters*, 93, 2004.

[41] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, June 2002. ISSN 0027-8424. doi: 10.1073/pnas.122653799.

[42] David F. Gleich and C. Seshadhri. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *KDD*, 2012.

[43] R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *International conference on World Wide Web, (WWW)*, 2004.

[44] John Healy, Jeannette Janssen, Evangelos Milios, and William Aiello. Characterization of graphs using degree cores. In *Algorithms and Models for the Web-Graph: Fourth International Workshop, WAW 2006*, volume LNCS-4936 of *Lecture Notes in Computer Science*. Springer Verlag, Banff, Canada, Nov. 30 - Dec. 1 2008.

[45] J. E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences of the United States of America*, 102(46):16569–16572, 2005.

[46] Svante Janson and Malwina J. Luczak. Asymptotic normality of the k-core in random graphs. *Ann. Appl. Probab.*, 18(3):1085–1137, 2008. ISSN 0378-8733. doi: 10.1016/0378-8733(83)90028-X.

[47] Vasileios Kandylas, S. Upham, and Lyle Ungar. Finding cohesive clusters for analyzing knowledge communities. *Knowledge and Information Systems*, 17: 335–354, 2008. ISSN 0219-1377. 10.1007/s10115-008-0135-5.

[48] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, December 1998. ISSN 1064-8275.

[49] Lefteris M. Kirousis and Dimitrios M. Thilikos. The linkage of a graph. *SIAM J. Comput.*, 25(3):626–647, 1996. ISSN 0097-5397.

[50] Guy Kortsarz and David Peleg. Generating sparse 2-spanners, 1993.

[51] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 57–, Washington, DC, USA, 2000. IEEE Computer Society. ISBN 0-7695-0850-2.

[52] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Extracting large-scale knowledge bases from the web. In *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, pages 639–650, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1-55860-615-7.

[53] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. On sampling-based approximate spectral decomposition. In *ICML'09*, pages 553–560, 2009.

[54] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78, 2008. doi: 10.1103/PhysRevE.78.046110.

[55] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *KDD*, pages 631–636, 2006. ISBN 1-59593-339-5.

[56] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *International conference on Human factors in computing systems, (CHI)*, 2010.

[57] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *International conference on World wide web, (WWW)*, 2010.

[58] Jure Leskovec, Kevin J. Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *WWW*, pages 631–640, 2010.

[59] Don R. Lick and Arthur T. White. k-degenerate graphs. *Canad. J. Math.*, 22: 1082–1096, 1970. ISSN 0008-414X.

[60] Haifeng Liu, Ee-Peng Lim, Hady W. Lauw, Minh-Tam Le, Aixin Sun, Jaideep Srivastava, and Young Ae Kim. Predicting trusts among users of online communities: an epinions case study. In *ACM Conference on Electronic commerce, (EC)*, 2008.

[61] Tomasz Luczak. Size and connectivity of the k-core of a random graph. *Discrete Math.*, 91(1):61–68, July 1991. ISSN 0012-365X. doi: 10.1016/0012-365X(91)90162-U.

[62] Dijun Luo, Chris H. Q. Ding, Heng Huang, and Feiping Nie. Consensus spectral clustering in near-linear time. In *ICDE*, pages 1079–1090, 2011.

[63] Arun S. Maiya and Tanya Y. Berger-Wolf. Sampling community structure. In *WWW*, pages 701–710, 2010.

[64] Silviu Maniu, Bogdan Cautis, and Talel Abdessalem. Building a signed network from interactions in wikipedia. In *Databases and Social Networks, (DBSocial)*, 2011.

[65] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[66] David W. Matula. A min–max theorem for graphs with application to graph coloring. *SIAM Reviews*, 10, 1968.

[67] David W. Matula, George Marble, and Joel D. Isaacson. Graph coloring algorithms. In *Graph theory and computing*, pages 109–122. Academic Press, New York, 1972.

[68] Mary McGlohon, Leman Akoglu, and Christos Faloutsos. Weighted graphs and disconnected components: patterns and a generator. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 524–532, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. doi: http://doi.acm.org/10.1145/1401890.1401955.

[69] J. Balthrop M.E. Newman, S. Forrest. Email networks and the spread of computer viruses. *Phys Rev E Stat Nonlin Soft Matter Phys*, 66, 2002.

[70] Abhinav Mishra and Arnab Bhattacharya. Finding the bias and prestige of nodes in networks based on trust scores. In *International conference on World wide web, (WWW)*, 2011.

[71] James Moody and Douglas R. White. Structural cohesion and embeddedness: A hierarchical concept of social groups. *American Sociological Review*, 68(1):pp. 103–127, 2003. ISSN 00031224.

[72] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69(6):066133, Jun 2004. doi: 10.1103/PhysRevE.69.066133.

[73] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69, 2004.

[74] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

[75] Spiros Papadimitriou, Jimeng Sun, Christos Faloutsos, and Philip S. Yu. Hierarchical, parameter-free community discovery. In *Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II*, ECML PKDD '08, pages 170–187, Berlin, Heidelberg, 2008. Springer-Verlag.

[76] Boris Pittel, Joel Spencer, and Nicholas Wormald. Sudden emergence of a giant k-core in a random graph. *J. Combin. Theory Ser. B*, 67(1):111–151, 1996. ISSN 0095-8956. doi: 10.1006/jctb.1996.0036.

[77] Marzia Polito and Pietro Perona. Grouping and dimensionality reduction by locally linear embedding. In *NIPS*, pages 1255–1262, 2001.

[78] Béla Bollobá s, Oliver Riordan, Joel Spencer, and Gábor Tusnády. The degree sequence of a scale-free random graph process. random structures and algorithms. 18:279–290, 2001.

[79] Venu Satuluri and Srinivasan Parthasarathy. Scalable graph clustering using stochastic flows: applications to community discovery. In *KDD*, pages 737–746, 2009.

[80] Venu Satuluri, Srinivasan Parthasarathy, and Yiye Ruan. Local graph sparsification for scalable clustering. In *SIGMOD*, pages 721–732. ISBN 978-1-4503-0661-4.

[81] Stephen B. Seidman. Network structure and minimum degree. *Social Networks*, 5(3):269 – 287, 1983. ISSN 0378-8733.

[82] Ma Ángeles Serrano and Marián Boguñá. Topology of the world trade web. *Phys. Rev. E*, 68, 2003.

[83] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000. ISSN 0162-8828.

[84] Mauro Sozio and Aristides Gionis. The community-search problem and how to plan a successful cocktail party. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 939–948, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0055-1.

[85] George Szekeres and Herbert S. Wilf. An inequality for the chromatic number of a graph. *J. Combinatorial Theory*, 4:1–3, 1968.

[86] Amanda L. Traud, Peter J. Mucha, and Mason A. Porter. Social structure of facebook networks. *CoRR*, 2011.

[87] Stanley Wasserman and Katherine Faust. *Social Networks Analysis: Methods and Applications*. Cambridge: Cambridge University Press., 1994.

[88] D. J. Watts and S. H. Strogatz. Collective dynamics of'small-world'networks. *Nature*, 393(6684), 1998.

[89] Scott White and Padhraic Smyth. A spectral clustering approach to finding communities in graph. In *SDM*, 2005.

[90] Stefan Wuchty and Eivind Almaas. Peeling the yeast protein network. *PROTEOMICS*, 5(2):444–449, 2005. ISSN 1615-9861.

[91] Donghui Yan, Ling Huang, and Michael I. Jordan. Fast approximate spectral clustering. In *KDD*, pages 907–916, 2009.