Karlsruhe Institute of Technology

Analysis of Scientific Collaboration Networks: Social Factors, Evolution, and Topical Clustering

Diploma Thesis of

Christian Lorenz Staudt

At the Department of Informatics
Institut für Theoretische Informatik
Lehrstuhl für Algorithmik I
Karlsruhe Institute of Technology

Reviewer: Dorothea Wagner
Advisors: Robert Görke and Andrea Schumm

Duration: February 2011 – August 2011

Web: http://i11www.iti.uni-karlsruhe.de
Contact: christian.staudt@ira.uka.de or clstaudt@lavabit.com

Contents

1. Introduction and Preliminaries

1.1. Introduction and Outline

In *scientometrics*, the quantitative study of science, *network analysis* has become a prominent tool. The kinds of networks most frequently examined have been citation networks (mapping links between publications based on references) and collaboration networks (mapping the collaborative relationships between researchers based on joint publications) [Mil04][New01]. Collaboration networks are at the center of this work, too. Based on the extensive *DBLP* publication database, relations between authors and publications are compiled as graphs, mapping almost the entire field of computer science. This allows us to examine and quantify the collaborative relations between researchers using graph-based methods. The focus on collaboration is supported by empirical results showing that research increasingly happens in the form of teamwork, and publications produced by teams of authors typically reach a greater impact [WJU07][Sol09]. In the course of this work, the collaboration network is explored from several angles. A variety of techniques for the analysis of complex networks, especially social network analysis, yield insights into its composition. Of particular interest here is whether the natural components of the network correspond to separate fields with regard to research content. These analyses give us a general picture of the network and lead up to the central point of this work, the question how the network is shaped by social events in the academic realm: The *Dagstuhl seminars* assemble researchers with the goal of fostering (collaborative) work in cutting-edge areas of computer science. We examine whether such events leave a track in the structure of the network. Furthermore, the significance of *centrality* in the collaborative network is explored, both for authors and for publications.

The structure of the work can be outlined briefly as follows: After some preliminary definitions (Section 1.2), Chapter 2 describes in detail the preparation of source data, leading to the modeling of the collaboration network as different graphs. Chapter 3 briefly highlights some technical aspects of the implementation. In Chapter 4, standard concepts from network analysis are applied in order to report general properties of the resulting network, including connectedness, degree distribution and k-core decomposition. Chapter 5 is concerned with clustering, i.e. partitioning the graph into internally dense subgraphs. After calculating a clustering of the collaboration network, we look into the question whether authors form collaborative clusters according to the topical similarity of their work. As a core element of this work, we then explore the question whether participation in seminars designed to foster collaboration, the *Dagstuhl seminars*, has an effect on the pattern of

collaborations (Chapter 6). Subsequently, we apply *eigenvector centrality* to a comprehensive graph and discuss the ranking of authors and publications (Chapter 7). A concluding overview over the work is provided in Chapter 8.

1.2. Notation and Basic Definitions

This section defines some basic concepts from graph theory which will be used in the course of the work, as well as their notation. Much of this (graphs, bipartite graphs, hypergraphs, etc.) should be familiar to those acquainted with graph theory, and the conventional notation is used.

1.2.1. Graphs

A *graph* is the mathematical representation of a *network* of entities (as *nodes*) and their connections or relations (as *edges*). In this work, we speak of a network when making domain-specific, conceptual statements, and of a graph if the precisely defined mathematical representation is meant. Throughout this work, the conventional notation for graphs is used.

GRAPH **Definition 1.** *A* graph *is a tuple $G = (V, E)$ where V is the set of vertices. An edge in E connects two vertices. The graph is a directed graph if the edges are ordered pairs $E \subseteq V \times V$ or an undirected graph if the edges are unordered pairs $E \subseteq \binom{V}{2}$.*

We use $n = |V|$ and $m = |E|$ as shorthand for the number of nodes and edges in a graph. All graphs considered in this work are undirected. If there is an edge $\{u, v\}$, it is called *incident* to u and v, and the nodes u and v are called *adjacent*. *Weighted graphs* have a weight function which assigns values to edges:

WEIGHTED GRAPH **Definition 2.** *Let ω be a weight function*

$$\omega : \begin{cases} E \to \mathbb{R} \\ \{u, v\} \mapsto x \end{cases} \tag{1.2.1}$$

Then $G = (V, E, \omega)$ is called a weighted graph.

Graphs can be traversed by following edges, and sets of nodes can be defined based on their reachability from other nodes. This leads us to the concepts of *path* and *neighborhood*.

PATH **Definition 3.** *A path P from u to v is a sequence of edges*

$$P(u, w) := (\{u, v_1\}, \{v_1, v_2\}, \ldots, \{v_{k-1}, w\}) \tag{1.2.2}$$

Alternatively, a path can be defined as a sequence of nodes

$$P(u, w) := (v_1, \ldots, v_k) : v_1 = u, v_k = w, \forall i < k : v_i, v_{i+1} \ \in E \tag{1.2.3}$$

A path of a specific length k is written as $p_k(u, v)$. For unweighted graphs, the length of the path is the *distance* between its end nodes.

$$p_k(u, v) := p(u, v) : |p(u, v)| \ = k \tag{1.2.4}$$

NEIGHBORHOOD **Definition 4.** *In a graph $G = (V, E)$, the neighborhood of a node u is the set of its adjacent nodes.*

$$N(u) := \{v : \{u, v\} \ \in E\} \tag{1.2.5}$$

We can generalize the notion of a neighborhood to sets of nodes V:

$$N(V) := \bigcup_{v \in V} N(v) \tag{1.2.6}$$

Definition 5. *The degree of a node is the number of its incident edges.*

DEGREE

$$deg(u) := |\{\{u, v\}, v \in V\}| = |N(u)| \tag{1.2.7}$$

1.2.2. Hypergraphs

A *hypergraph* is a generalized graph, in which an edge can connect more than two nodes.

Definition 6. *A hypergraph is a tuple $H = (V, E)$ where V is a set of nodes and E is a set of hyperedges. A hyperedge is a non-empty subset of V.*

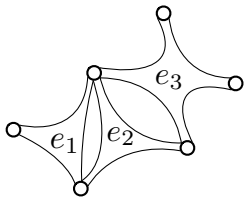


Figure 1.1.: Hypergraph

HYPERGRAPH

1.2.3. Bipartite Graphs

Bipartite graphs will be used throughout this work to model the relationships between researchers and scientific publications.

Definition 7. *A bipartite graph is a graph whose nodes can be divided into two disjoint sets U and V such that every edge in E connects a node in U to one in V. $G = (U, V, E)$*

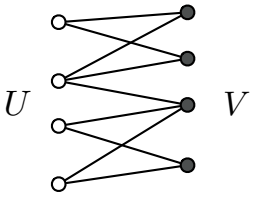


Figure 1.2.: Bipartite graph

BIPARTITE GRAPH

Definition 8. *Given a bipartite graph $G = (U, V, E)$, the* bipartite neighborhood *of a node $v \in V$ is defined as*

BIPARTITE NEIGHBORHOOD

$$N_b(v) := \{\ w \in V : \exists P(v, w) = (v, x, w), x \in U\} \tag{1.2.8}$$

For a nodes $u \in U$, the definition is symmetrical.

Density in Bipartite Graphs

The density of a subgraph defined by a subset of nodes is the number of edges which are present divided by the number of possible pairs. In a bipartite graph, only edges between nodes from the different parts of the graph are allowed. However, can think of density in bipartite graphs, which is analogous to density in general graphs.

Definition 9. *We define* density *in bipartite graphs as the number of pairs which are connected by a path of length two, divided by the total number of pairs.*

DENSITY IN BIPARTITE GRAPHS

$$dens_b(S, G) := \frac{|\{p(u, v) : u, v \in S\ |p(u, v)| = 2\}|}{\frac{|S| \cdot (|S|-1)}{2}} \tag{1.2.9}$$

1.2.4. Clustering

Clustering is the subdivision of a graph's node set into groups, which we formalize as follows:

Definition 10. *A* clustering *$\zeta(G)$ of a graph $G = (V, E)$ is a partition of V into disjoint, non-empty subsets $\{C_1, \ldots, C_k\}$. Each subset is a cluster $C_i \in \zeta$.*

CLUSTERING

ζ is written instead of $\zeta(G)$ when unambiguous. We abbreviate the number of clusters in a clustering with $k = |\zeta|$. If a cluster contains only one node, it is called a *singleton*; accordingly, a *singleton clustering* consisting only of singletons. The other trivial clustering is the *1-clustering*, a cluster which contains all nodes.

2. Data Preprocessing and Modeling

In this chapter, the primary data sources as well as the modeling of the collaboration network based on this data are described. The preprocessing necessary to make such large data sets easily tractable is also covered. Furthermore, preparatory steps needed for the evaluation in Ch. 6 are discussed in Sec. 2.3.

2.1. Data Sources

This section introduces the main data sources, the *DBLP publication database*, as well as the *Dagstuhl seminar database*. A collaboration network representative of the entire field is modeled on the basis of the publication database, while the data provided by the *Schloss Dagstuhl* conference center is the foundation for determining the effect of research seminars on collaboration.

2.1.1. DBLP Publication Database

DBLP, the *Digital Bibliography & Library Project*, is an extensive bibliographic database covering the field of computer science [DBL07]. In the following, the content will be referred to as the *publication data(base)*. As of July 2011, *DBLP* covers over 1.6 million publications. Queries to the database can be performed via web interfaces (see [dbl11][DBL07][dbl]). An XML dump of the entire database is available from the main website. The snapshot of the data used in the following is dated from March 26, 2011. There is ongoing work in updating the database with current publications, and a delay between publication date and entry into DBLP. As of March 2011, publications from 2010 do not seem to be sufficiently covered (although the gap is being closed). Because there is no reason to expect a real break in the strong growth trend of computer science publications, especially since the 1980s (see Fig. 2.1), publications after 2009 are excluded upfront. When parsing the needed information from the database, some adjustments for errors had to be made, e.g., publications without associated authors had to be ignored.

Publications

DBLP contains essentially a large collection of publications. In the following, this set of all proper publications (with at least one author) is denoted as $\mathbf{P}$. This leaves a total of 1 444 336 publications. *DBLP* distinguishes between the following publication types relevant to this work: article, inproceedings, proceedings, book, incollection, phdthesis, and

mastersthesis. Fig. 2.2 illustrates the distribution. Depending on the publication type, different attributes (including references to other elements) are attached to a publication. The attributes relevant to this work are illustrated in Fig. 2.3.

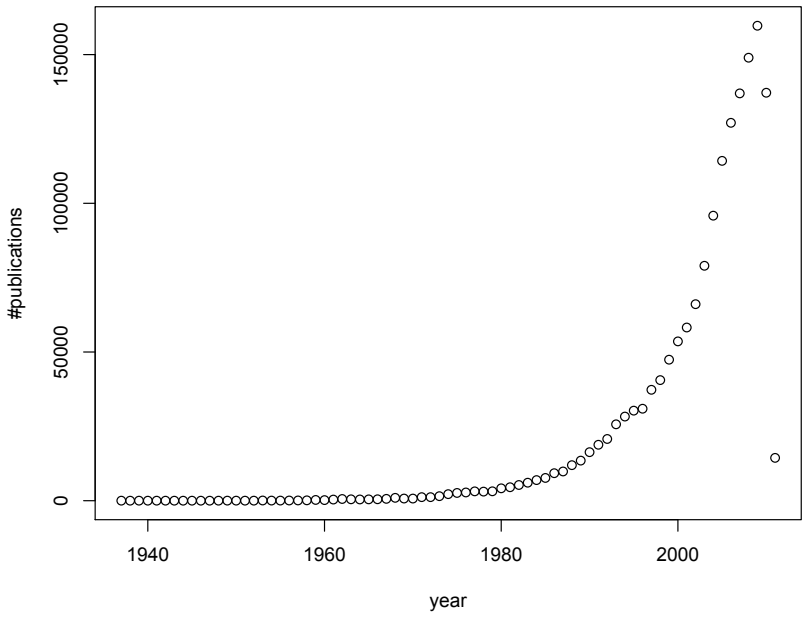


Figure 2.1.: Number of publications per year recorded in *DBLP*

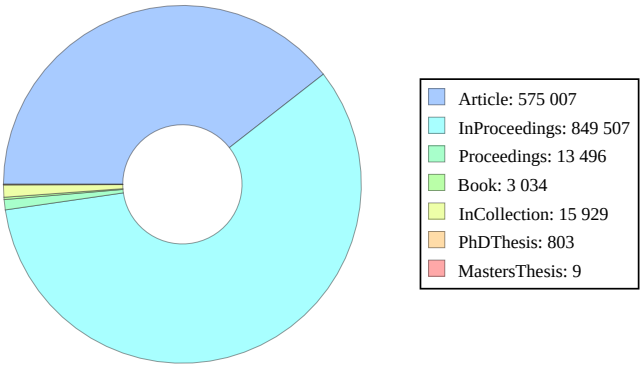


Figure 2.2.: *DBLP*: number of publications by type

Authors

All publications in **P** have at least one author or editor element associated with them. These elements contain only the name of the person. For the present purposes, editors are treated as authors, and mapped onto the same class Author in the data object model (Fig. 2.3). In the following, the set of all authors is denoted as **A**. The total number of authors is thus 852 250.

Conferences

Computer science is arguably a conference-driven field, with contributions to conference proceedings playing a major role. This is also evident from *DBLP*, where inproceedings entries make up the majority of publications (see Fig. 2.2). Therefore, this work employs conferences on several occasions when information about *topical communities* is needed: A conference and the set of participating researchers is treated as defining a subfield of computer science. Conferences are usually periodical events, so we distinguish between a *conference* and its *installments*. We assume that there is at most one installment per year, as the publication data does not allow a finer granularity. The set of all conferences is denoted as **C**. Sub. 2.2.2 covers extracting conference data in more detail.

2.1.2. *Dagstuhl Seminars* Database

In the forefront of this work, the *Schloss Dagstuhl* conference center kindly provided us with comprehensive lists of participants of the *Dagstuhl seminars*. The database was available as a table in comma-separated value format. From the fields of the database (Tab. 2.1) we select only those relevant for associating authors with seminars, i.e. the name of the invited researcher as well as the title and date of the seminar installment. Additionally, the information whether the invited guest actually attended the event is stored. In the following, seminar guests will be matched with authors from the publication database (see Sub. 2.2.3 for details), so that their pattern of publication and collaboration can be examined. Tab. 2.2 lists the number of seminar installments for each year. On average, 52 researchers were invited, 38 attended a seminar, and 14 declined. In the course of this work, we refer to them shortly as *invitees*, *attendees* and *absentees*.

key	description
sem_id	numeric seminar identifier
sem_type	classification of seminar
sem_first	date of first seminar day
sem_last	date of last seminar day
sem_title	seminar title
guest_id	numeric guest identifier
guest_firstname	guest first name
guest_lastname	guest last name
guest_gender	male or female
guest_residence	country of residence
guest_organization	university or institute
guest_organization_id	numeric organization identifier
guest_attended	Did the guest attend the seminar?
guest_organizer	Was the guest the organizer of the seminar?

Table 2.1.: Seminar database schema

2.2. Data Preprocessing

2.2.1. Data Model

To make a large dataset like *DBLP* easily tractable, data imported from the XML format is mapped to an object-oriented data model. The main entities modeled are publications (with several subtypes), authors, and conferences. To facilitate access to the imported data, each model object class maintains a record of its instances. Each instance is identified by a unique key. Given the key, an instance can be retrieved by calling the class method byKey with the key as a parameter. Tab. 2.3 gives an overview of the relevant data objects.

year	installments
2000	32
2001	32
2002	33
2003	45
2004	41
2005	46
2006	50
2007	49
2008	52

Table 2.2.: Number of *Dagstuhl seminar* installments per year

DBLP element	model class	key attributes	description
article	Article	title	journal article
inproceedings	InProceedings	”	published in conference proceedings
proceedings	Proceedings	”	proceedings volume
book	Book	”	general (text)book
incollection	InCollection	”	published in a collection
phdthesis	PhDThesis	”	PhD thesis
mastersthesis	MastersThesis	”	master's thesis
author, editor	Author	name	author or editor
-	Conference	title	conference as a series
-	ConferenceInstallment	title, year	single installment
-	Seminar	title	seminar as a series
-	SeminarInstallment	title, firstDay	single installment

Table 2.3.: Overview of data elements

2.2.2. Importing Conference Data

Publications of the type InProceedings were contributed to academic conferences and published in the respective proceedings. However, *DBLP* does not list conferences as distinct elements. Most of the inproceedings elements in *DBLP* contain a crossref element which indirectly references the conference installment. This reference comes as a string which does not have a uniform format throughout the database, but a typical example would be `conf/3dica/1999`. Therefore, assuming that a conference has at most one installment per year, we attempt to extract the title (abbreviation) of the conference and its year by matching regular expressions on the crossref string. If the string does not yield a year, the year of the publication is used instead. The limitation of this approach is that about 14% of InProceeding entries have to be discarded due to missing crossref strings. For another 0.03% of InProceedings the approach also fails because the entries do not have the expected regular format. Eventually, this method identifies about 2700 distinct conferences in the field of computer science.

2.2.3. Matching Seminar Guests to *DBLP* Authors

Our data sets record a total of 11 625 seminar guests and 852 250 authors in the publication database. In order to evaluate the impact of seminar participation on the collaboration network (Ch. 6), seminar guests must be mapped to authors in the publication database. The

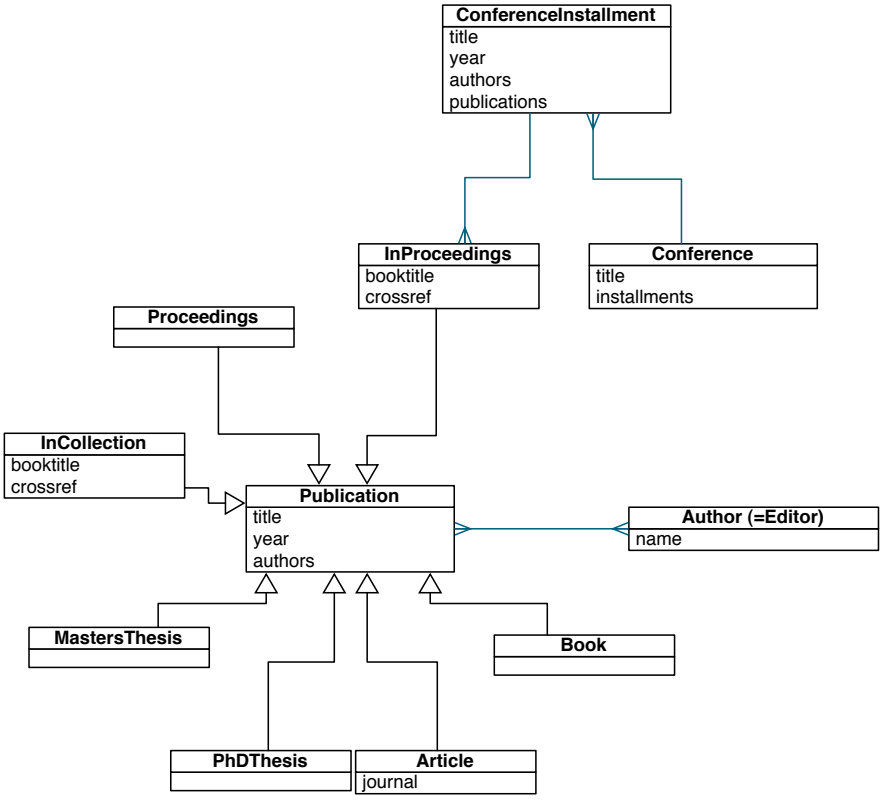


Figure 2.3.: Data model classes and their relations

identity between an author and a seminar participant was established by name. Matching is performed by comparing names for equality using hash tables. This method is, of course, error-prone: "John Doe" in the publication database and "J.X. Doe" in the seminar database might refer to the same actual person, but the identity will be overlooked. On the other hand, different authors with equal names cannot be distinguished due to a lack of unambiguous attributes. With the data at hand, it was not feasible to eliminate this as an error source. It is assumed that errors of this kind do not qualitatively distort the reported findings. Eventually, a matching author in the publication database was found for 72 percent of the seminar guests.

2.3. Preliminary Data Analysis

This section describes further steps of preparation needed for later analyses.

2.3.1. Identifying Area Launchers

This preparatory step is relevant to the analysis described in Ch. 6, in which the aim is to study the effect of joint seminar participation on the collaboration network. In order to detect increased collaboration which can be clearly attributed to the seminars, one central idea was to first identify *area launchers*, seminars intended to bring together a group of researchers who have not collaborated much before. Our assumption, as well as a stated goal of the *Dagstuhl seminars*,[1] is that some seminars are intended to "launch" new areas of research by fostering collaboration between previously unaffiliated researchers, thereby contributing to emerging fields. *Area launchers* are significant to the question at hand due to the following argument: If the researchers involved develop collaborative ties in the

aftermath of such a seminar, it is possible to attribute this more clearly to the seminar rather than existing collaborative relationships, developed, for instance, in the course of a common conference.

The task is therefore to classify a set of seminars as area launchers without special knowledge about the intent of the organizers or the content of the seminar. In fact, we attempt this classification solely from participation data, in the following way: It is assumed that well-established areas of research generally spawn their own dedicated conference, and that the participants of such a conference generally represent the researchers active in this area. By this logic, a seminar is clearly assignable to a particular established area of research if the set of researchers invited to the seminar has a strong overlap with the participants of the respective conference. Furthermore, if researchers attend the same conference, it is probable that they are already familiar with each other as well as each others work. We therefore reason that a seminar is an area launcher if its invitees do not overlap strongly with the participants of any particular conference. We proceed by calculating the overlap of invitees and participants for each pair of seminar and conference.

One has to take into account that most conferences and some of the seminars consist of a series of installments. Clearly, a classification based on topical similarity should involve the conference series and seminar series as a whole, not just single installments. Yet, only those conference installments prior to a seminar installment are relevant to the argument here, namely, that collaborative ties have been developed through prior work in the same field. This leads to Algorithm 1 for calculating the overlap values.

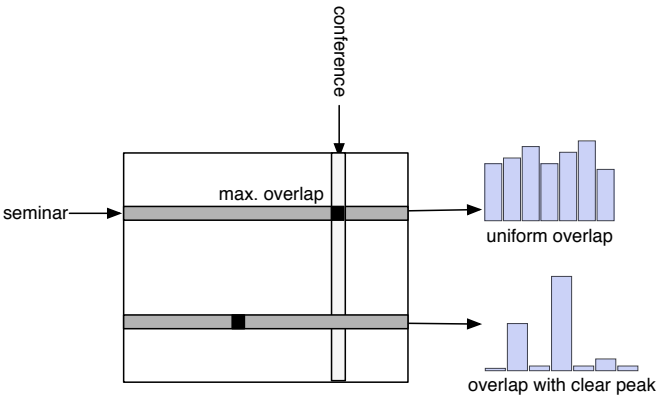


Figure 2.4.: Illustration of the seminar-conference overlap matrix

The overlap algorithm yields a matrix of overlap values (350 seminars by 2752 conferences). Based on these values, it is possible to assign a seminar to the conference with which its overlap is greatest (see Fig. 2.4 for an illustration). Clearly, the participants of some seminars closely match those of a conference, and the topical similarity is evident in many of those cases (see Tab. 2.3.1). We are, however, interested in those seminars where no such clear correspondence can be found. If a seminar row has a clear overlap peak, this means that there is a clear correspondence with one conference in particular. If, on the contrary, no clear peak can be identified, and the seminar has an evenly distributed overlap, we can assume that the seminar brings together researchers previously unaffiliated by joint conference participation. A measure that can be used to distinguish the two extremes is the *Gini coefficient*. The coefficient is known from sociology as a measure of income inequality in a population, but it is generally suited to quantify the inequality of any distribution.

GINI COEFFICIENT

Definition 11. *The* Gini coefficient *is a measure of statistical dispersion that measures the inequality of a distribution, with 0 and 1 expressing maximum equality and inequality, respectively. For values x_i, indexed in non-decreasing order and with a mean value of μ,*

[1]The Dagstuhl webpage states that: "Most seminars discuss an established field within computer science. However, Dagstuhl Seminars are also known for establishing new directions by bringing together separate fields or even scientific disciplines."

Algorithm 1: Calculation of overlap

```
for seminar ∈ seminars do
    for conference ∈ conferences do
        seminarAuthors ← ∅
        conferenceAuthors ← ∅
        preOverlap ← ∅
        for seminarInst ∈ seminar.installments do
            N ← seminarInst.invitees \ seminarAuthors
            conferenceAuthors ← conferenceAuthors ∪ N
            for conferenceInst ∈ conference.installments do
                if seminarInst.year < conferenceInst.year then
                    conferenceAuthors ←
                    conferenceAuthors ∪ conferenceInst.authors
            preOverlap ← preOverlap ∪ (conferenceAuthors ∩ N)
        overlap[seminar, conference] ← |preOverlap| / |seminarAuthors|
```

the Gini coefficient can be calculated trough the following formula [Dam]:

$$G = \frac{\sum_{i=1}^{n}(2i - n - 1)x_i}{n^2\mu} \tag{2.3.1}$$

Fig. 2.5 helps to illustrate the concept of the coefficient: The curve separating the areas A and B plots the cumulative share y of the total value which is contributed by the bottom x percent of the distribution, so that the diagonal line represents perfect equality. The coefficient can now be interpreted as the relative area A between the curve and the line of equality, which increases the more skewed the distribution is.

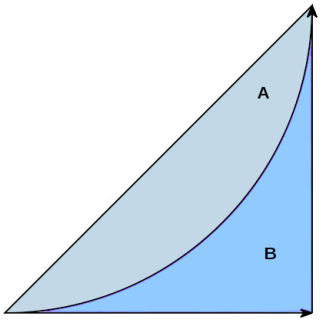


Figure 2.5.: Illustration of the Gini coefficient (taken from [Wik11])

seminar	max. overlap	conference	gini coeff.
Cryptography	0.76842	crypto	0.98349
Symmetric Cryptography	0.70370	fse	0.98362
Circuits, Logic, and Games	0.61290	stacs	0.98964
Graph Drawing	0.60000	gd	0.98523
Multi-Robot Systems	0.56896	robocup	0.96879
Scientific Visualization	0.54687	visualization	0.97847
Theoretical Foundations of Practical Information Security	0.54054	crypto	0.97509
Future Generation Grids	0.53333	europar	0.98266
Cognitive Vision Systems	0.51724	eccv	0.97824
Sublinear Algorithms	0.48214	stoc	0.97580

Table 2.4.: Correspondence between seminars and conferences according to overlap

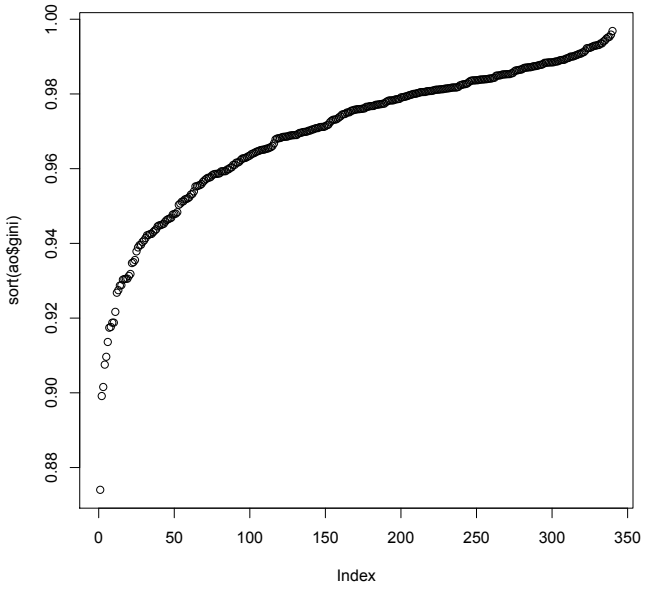


Figure 2.6.: Overlap Gini coefficients per seminar

Selecting a Set of Area Launchers

The Gini coefficient for a seminar's overlap values can be used as an objective measure to automatically classify seminars as area launchers. However, one caveat was noticed: A few seminars invite researchers from many different areas to provide an opportunity for interdisciplinary exchange and reflection, but are not designed to initiate a specialized, innovative area of research. The seminar *Quo vadis Informatik - Innovation dank Informatik* (2006) can be named as an example for interdisciplinary seminars which are not expected to lead to increased collaboration among the participants. Therefore, such seminars are omitted, and from the seminars with low overlap coefficients 10 seminars are selected by hand and classified as area launchers (see Tab. 2.5). All of them are single-installment seminars.

seminar title	overlap Gini coefficient
Software Engineering for Self-Adaptive Systems	0.89913
Organic Computing - Controlled Self-organization	0.90156
Model Engineering of Complex Systems (MECS)	0.90754
The Evolution of Conceptual Modeling	0.90964
Software Engineering for Tailor-made Data Management	0.91361
Event Processing	0.91743
Autonomous and Adaptive Web Services	0.91763
Evolutionary Test Generation	0.92870
Organic Computing - Controlled Emergence	0.93129
Peer-to-Peer-Systems and -Applications	0.93953

Table 2.5.: Selection of seminars with low conference correspondence

2.4. Modeling the Collaboration Network

This section introduces how the relational data concerning authors and publications from the publication database can be represented in the form of graphs, making graph algorithms applicable in the following analyses.

2.4.1. Entities and Relations

The main set of information extracted from the publication data concerns authors and their publications. A shorthand for referring to them is introduced here. Furthermore, the publication database provides data on two main relations, *authorship* and *coauthorship*, formalized here to be used in upcoming definitions. Table 2.6 gives an abstract example for the form in which the data is extracted, which will be used to illustrate how the data is transformed into graphs.

AUTHORS AND PUBLICATIONS

Definition 12. *In the following, the set of all authors is denoted as* $\mathbf{A}$ *and the set of all publications is denoted as* $\mathbf{P}$*. Given an author* $a \in \mathbf{A}$*, the set of her publications is denoted by* $P(a)$*. Conversely, for a publication* p*, the set of its authors is denoted by* $A(p)$*.*

AUTHORSHIP AND COAUTHORSHIP

Definition 13. *Given the set of authors* $\mathbf{A}$ *and the set of publications* $\mathbf{P}$*, the* authorship relation *is defined as*

$$\forall \{a, p\} \in \mathbf{A} \times \mathbf{P} : \quad a \smile p \iff a \text{ is author of } p \tag{2.4.1}$$

The coauthorship relation *between two authors from* $\mathbf{A}$ *is defined as*

$$\forall \{a \in \mathbf{A}, b \in \mathbf{P}\} : a \frown b \iff \exists p \in \mathbf{P} : a \smile p \wedge b \smile p \tag{2.4.2}$$

publication	authors
p_1	a_1, a_2
p_2	a_1, a_3, a_4
p_3	a_1, a_2
p_4	a_2
p_5	a_3, a_4

Table 2.6.: Example of affiliations between publications and authors

2.4.2. Matrices and Graphs

There are several ways of recording and viewing the relational data contained in the publication database. To begin with, we can look at the data on the authorship relation as an $\mathbf{A} \times \mathbf{P}$ matrix $I_{\mathbf{AP}}$ containing binary entries. This *two-mode* matrix (with rows and columns referring to different data sets) is called the *incidence matrix*, and a lookup yields:

$$I_{\mathbf{AP}}(i, j) = 1 \iff a_i \smile p_j \tag{2.4.3}$$

The two-mode data contained in the $\mathbf{A} \times \mathbf{P}$ incidence matrix can be projected onto two one-mode, square *adjacency matrices* (and therefore, graphs): an $\mathbf{A} \times \mathbf{A}$ adjacency matrix $A_{\mathbf{A}}$ and a $\mathbf{P} \times \mathbf{P}$ adjaceny matrix $A_{\mathbf{P}}$ [Sco00]. Note the matrices 2.4.6 corresponding to the example data in table 2.6.

$$A_{\mathbf{P}}(i, j) = 1 \iff \exists a \in \mathbf{A} : a \smile p_i \wedge a \smile p_j \tag{2.4.4}$$

$$A_{\mathbf{A}}(i,j) = 1 \iff a_i \frown a_j \tag{2.4.5}$$

An entry in $A_{\mathbf{P}}$ tells us whether two publications have an author in common, but since the focus of this work is on authors and their collaborative relationships, this adjacency matrix is not considered any further. An entry in $A_{\mathbf{A}}$ gives information about whether two authors are coauthors of any publication. However, information about the cause of the relation (the publication itself) is lost. This information loss could be partly compensated with weighted entries (weighted edges), which inform us about the number of publications and therefore indicate the strength of the coauthorship relation. Par. 2.4.2 elaborates on this option.

$$I_{\mathbf{AP}}: \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \quad A_{\mathbf{A}}: \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad A_{\mathbf{P}}: \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \tag{2.4.6}$$

The incidence matrix $I_{\mathbf{AP}}$ can be interpreted as a *hypergraph* by treating each row (publication) as a *hyperedge* connecting a set of nodes (authors). Such a hypergraph representation would preserve all data, yet would require adapting all graph algorithms used to hypergraphs. Fortunately, a hypergraph can be represented as a standard *bipartite graph* without losing information, by applying a simple transformation: For each hyperedge, introduce a new node and connect it to the nodes incident to the hyperedge. Furthermore, the adjacency matrix $A_{\mathbf{AP}}$ results from the hypergraph incidence matrix $I_{\mathbf{AP}}$ through the following simple combination [Fra] :

$$A_{\mathbf{AP}} = \begin{pmatrix} 0 & (I_{\mathbf{AP}})^T \\ I_{\mathbf{AP}} & 0 \end{pmatrix} \tag{2.4.7}$$

The Authorship Graph $G_{\mathbf{PA}}$

Obviously, each of the adjacency matrices A corresponds to a graph G (see Fig. 2.7 for an illustration using the example data). In the following, the relation between authors and their publications is generally modeled as the bipartite graph $G_{\mathbf{PA}}$ containing nodes for both publications and authors, as this graph is richest in information. The nodes introduced instead of a hyperedge are, of course, the publication nodes $V_{\mathbf{P}}$. The model of this *authorship graph* can be described as follows:

AUTHORSHIP GRAPH

Definition 14. *Formally, the* authorship graph *is a mapping $\mathcal{M}$ from the set of publications* $\mathbf{P}$ *and the set of authors* $\mathbf{A}$ *to the node sets $V_{\mathbf{P}}$ and $V_{\mathbf{A}}$, resulting in a bipartite graph $G_{\mathbf{PA}} = (V_{\mathbf{A}}, V_{\mathbf{P}}, E)$, where*

$$\{v_a, v_p\} \in E \iff a \smile p$$

Throughout this work, we deal with the correspondence between domain objects (authors, publications) and graph objects (nodes and edges). Equations are written with $\cong$ instead of $=$ if the left-hand side refers to domain objects and right-hand side refers to objects in the network, and both can be mapped onto each other using the functions $\mathcal{M}$ and $\mathcal{M}^{-1}$. Formally, this translates to:

$$X \cong Y \quad \iff \quad \mathcal{M}(X) = Y \quad \wedge \quad X = \mathcal{M}^{-1}(Y) \tag{2.4.8}$$

The Coauthorship Graph $G_{\mathbf{A}}$

In some instances, calculations and formulations can be simplified if they are performed and expressed in terms of the coauthorship relation data stored in $A_{\mathbf{A}}$. Information about the publications as the causes of the coauthorship relation is lost, but the information might not be required. Likewise, this matrix corrsponds to a graph, the *coauthorship graph* $G_{\mathbf{A}}$. $G_{\mathbf{A}}$ is a graph composed of overlapping cliques, since each multi-author publication contributes a clique, and is defined as follows:

COAUTHORSHIP GRAPH

Definition 15. *The* coauthorship graph *is a mapping $\mathcal{M}'$ from the set of authors $\mathbf{A}$ to the node set $V_{\mathbf{A}}$, resulting in the graph $G_{\mathbf{A}} = (V_{\mathbf{A}}, E)$, where*

$$\{v_a, v_b\} \in E \iff a \frown b \tag{2.4.9}$$

Weights

When projecting the two-mode data as the matrix $A_{\mathbf{A}}$, and its graph $G_{\mathbf{A}}$, information about specific publications as the cause of coauthorship relations is lost. However, it is possible to assign weights to the relations, based on the number of publications. This raises the question how to assign meaningful weights to edges, more specifically, how to distribute the weight of a publication over the resulting links between its coauthors. Distributing a weight of 1 per k-author publication over $\frac{1}{2}\cdot k\cdot(k-1)$ edges results in negligible edge weights for publications with many authors. Assigning each author an incident weight of 1 results in publications contributing different weights of $\frac{k}{2}$. Since each weight assignment scheme had drawbacks and was likely to add complexity, and questions requiring information on copublications could be better answered in terms of the bipartite graph $G_{\mathbf{PA}}$ (see also Ch. 7), only unweighted graphs were considered in the course of this work.

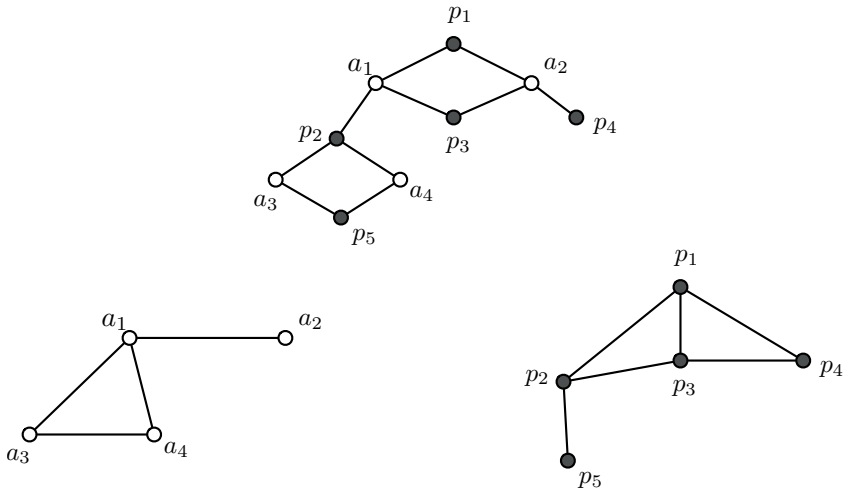


Figure 2.7.: $G_{\mathbf{PA}}$ (above), $G_{\mathbf{A}}$ (left), and $G_{\mathbf{P}}$ (right) for the example from Tab. 2.6

Resulting Graphs

Tab. 2.7 shows the number of nodes and edges in the graphs constructed from the full publication data set.

graph	n	m
$G_{\mathbf{PA}}$	2 296 586	3 775 881
$G_{\mathbf{A}}$	852 250	2 785 037

Table 2.7.: Size of resulting graphs

3. Implementation Notes

This chapter deals with the specifics of implementing and performing the network analyses, giving some information about the custom code written, the usage of existing tools and libraries, and the development process.

3.1. Custom Code

All code for parsing and transforming domain data, assembling and analyzing graphs and writing intermediary output is written in the Python programming language. The Jython interpreter is used, a Python implementation written in Java and for the Java Virtual Machine (JVM). Python was chosen as a language because it favors ease and speed of development in the light of frequently changing requirements. Jython was the interpreter of choice because of the option to integrate Java libraries (like XOM and JUNG), including graph tools developed at the institute [jyt]. Since Jython compiles Python to JVM bytecode, any Java class can be used from Python code. The code base of about 4500 lines of Python can roughly be divided into a section for the data model as well as reusable tools and algorithms on the one hand, and workflows on the other hand. Workflows are mostly linear programs aiming at a specific problem, e.g. building the collaboration network, determining connected components of the graph and writing the resulting information to disk. The interactive Jython interpreter functions as the main user interface, and textual output enabled some monitoring of running calculations (including conveniences like progress indicators). The implementation frequently relies on built-in modules like csv (CSV files input/output), subprocess (spawning processes like the C++ clusterer), itertools (advanced iteration and combinatorics), and datetime (processing date and time). Development took place in the Eclipse IDE, using the PyDev plugin for Python, as well as CDT (C/C++ Development Tooling) for the C++ parts.

Computing

The workflows implemented were designed to run on the institute's computation servers, which provided enough resources to make the large raw data sets and networks tractable. (For example, the primary server makes available 32 GB of RAM and 8 *Xeon* cores at 2.6 GHz). However, as parallelism in terms of tasks or data was not straightforward to achieve (and therefore, not very worthwile for one-time analysis workflows), computation was often limited to a single core.

Complications

The size of the data sources was complicating the development process. For many network analysis algorithms and workflows, a meaningful test data set (e.g., a 1-percent sample of the publication database resulting in a sparsely connected graph) was already making testing on the notebook used for development impractical due to high memory requirements and high running times. Testing and debugging therefore had to be done remotely on the computation servers, via the command line. This limited the use of the IDE and its graphical debugging tools.

XML Processing API

XML input and output is handled by the XOM library, which provides a tree-based API for Java and "strives for correctness, simplicity, and performance, in that order" [xom]. Using XOM, the entire publication database (about 800 MB of XML text) is parsed into a document object model (DOM) tree taking up approximately 8 GB of memory. The XOM-based parser follows the usual approach of processing a node and recursively descending to the child nodes.

Data Objects

After the import phase, the memory footprint of a workflow program is about 17 GB for the full publication database. On the machine mentioned above, the XML processing and import phases take roughly 10 minutes together. It was not viable to avoid this import time and store only the imported data on disk using a persistence format such as pickle, because the memory requirements of the export operation exceeded the capacity of the machine.

Graph API

The implementation relies on JUNG 2, the *Java Universal Network/Graph Framework* as an API for graph processing [jun11]. The library provides common graph data types and algorithms [OFS+05]. The class UndirectedSparseGraph was employed as the graph implementation. It was also used as the base class to create a BipartiteGraph class which provides methods convenient for working with bipartite graphs. JUNG relies heavily on generics, and thus allows arbitrary classes to serve as the node and edge classes in the graph. Custom node and edge classes implemented for this work are quite rudimentary. Instead of complex node and edge classes, data objects are mapped to nodes and edges using a Transformer, a pattern which is frequently employed across the JUNG library. The node class merely handles indexing.

Graph Construction

After the import phase, constructing $G_{\mathbf{PA}}$ takes an additional 20 minutes, raising the memory footprint to about 20 GB. For $G_{\mathbf{A}}$, the values are roughly equal.

Centrality

JUNG implements a wide variety of network analysis algorithms, among them centrality algorithm such as PageRank in edu.uci.ics.jung.algorithms.scoring.PageRank [jun]. A run of the PageRank algorithm on $G_{\mathbf{PA}}$ took about 10 minutes.

Clustering

The C++ implementation of the sLocal clusterer (see 1.2.4) used is derived from the implementation by fellow student David Lisowski, and was only slightly modified [Lis11]. It is highly optimized and able to cluster the full $G_{\mathbf{PA}}$ graph ($\sim$ 2 mio. nodes) in ca. 6 minutes, with a memory consumption of 3 GB. As input, the clusterer takes a list of weighted edges, contained in a text file where each entry is of the form (i_u, i_v, w) with i_u, i_v integer idenfiers of the nodes u,v, and w is a floating point weight. As output, it returns a mapping (i_u, i_C) from integer node identifier to cluster identifier. For this work, the clusterer was embedded in a Python module, which handles exporting of the graph as an edge list to a temporary file, starting the clusterer binary as a child process, and reading the output into a more object-oriented clustering representation.

3.2. Postprocessing and Plotting

Postprocessing of result data, statistical analysis and plotting was performed mainly with Mathematica [mat], and occasionally with R [rpr]. A basic interface was designed for loading a results file, processing the data and displaying the relevant plots. For a preliminary visualization, the tabular content of a CSV file is plotted using the ListPlot3D function, with rows and columns on two axes and the values on the third. The resulting "landscape" allows a first inspection of the distribution and development of values (Fig. 3.1, but the data is presented as 2D plots for evaluation. In general, Mathematica was used for plotting, as well as some statistical postprocessing (e.g., finding a linear fit to a doubly logarithmic plot of the degree distribution, see Sec. 4.2).

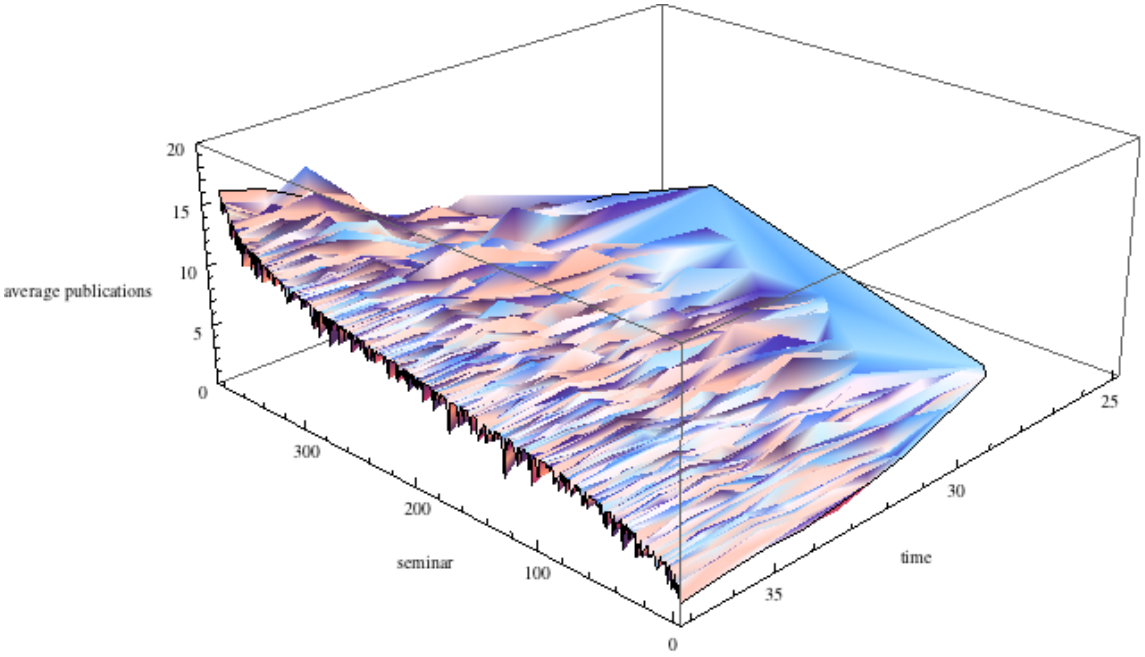


Figure 3.1.: Tabular data as 3D plot

4. General Network Properties

This chapter summarizes several analyses performed on the collaboration network (degree distribution, connectedness, and k-core decomposition) in order to gain some general insights into the network structure of scientific collaboration.

4.1. Connectedness

4.1.1. Connected Components

A basic topological property of graph is the number and composition of its *connected components*.

CONNECTED COMPONENT

Definition 16. *For an undirected graph $G = (V, E)$, a* connected component *is a subgraph $G' = (V', E')$ in which any two nodes from V' are connected by a path.*

The number and size of connected components are determined as a characteristic property of the network. Furthermore, it is desirable that the graph is as connected as possible for properties such as centrality (Ch. 7) to be meaningful. $G_{\mathbf{A}}$ is smaller than $G_{\mathbf{PA}}$, yet contains all information needed when connectedness is concerned. Thus, we determine the connected components in $G_{\mathbf{A}}$ by breadth-first search. As illustrated in Fig. 4.1, $G_{\mathbf{A}}$ is dominated by a single large connected component. This usually termed a *giant component* and has been observed for colllaboration networks before [New01]. Aside from the 6% of the authors without collaborations, about 14 % of author nodes are distributed over a multitude of small components, containing only 3 nodes on average. It seems likely that each of these components is formed by a single publication. We conclude that, in general, authors who have worked on multiple publications and were part of more than one collaborative team join the large connected component. Thus, the collaboration network as a whole is well-connected.

4.1.2. The Small World of Computer Science

Small world networks are a well-known phenomenon in network science. For such networks, the graph is far from fully connected, yet most nodes can be reached from one another by following a relatively small number of edges. Specifically, the typical distance between two randomly chosen nodes is roughly logarithmic in the total number of nodes [Str98]:

$$d(u, v) \sim \log n \tag{4.1.1}$$

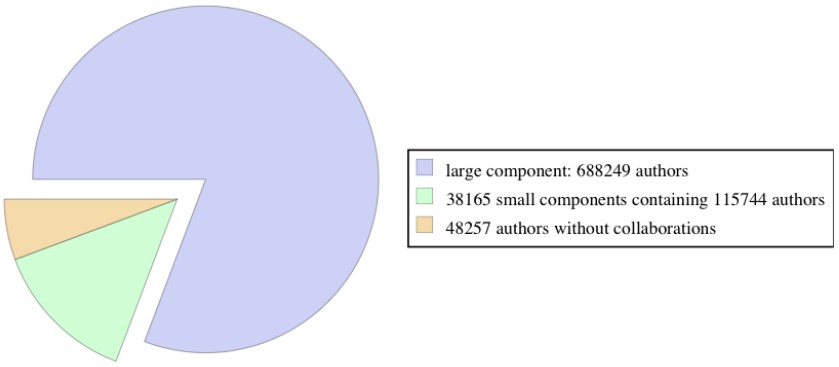


Figure 4.1.: Proportions of connected components in $G_{\mathbf{A}}$

The phenomenon is often described as *six degrees of separation*, applied to the social graph of all human acquaintances. When collaboration is concerned, the concept has been popularized by trivia games like *Six Degrees of Kevin Bacon* (distance in the collaboration network of film actors) and the *Erdös number* (distance in the coauthorship network of scientists). It is very likely that the collaboration network at hand belongs to the class of small world networks, too. As a rough test of the small world property in our collaboration network, distances for some random pairs of nodes were calculated on $G_{\mathbf{A}}$. For a sample of 20 randomly chosen node pairs, three pairs were not connected. The average node-to-node distance for the remaining pairs was 6.58 (suggesting six degrees of separation in this network, too). Additionally, the "collaborative distance", the minimum number of hops in terms of the coauthorship relation, between a few prominent computer scientists was determined and compiled in Tab. 4.1.

	Knuth	Dijkstra	Berners-Lee	Tanenbaum	Turing	van Rossum
Knuth		3	4	4	∞	4
Dijkstra			4	4	∞	4
Berners – Lee				4	∞	3
Tanenbaum					∞	1
Turing						∞
van Rossum						

Table 4.1.: Collaborative distance between several exemplary computer scientists

4.2. Degree Distribution

Many empirically observed complex networks have been categorized as *scale-free networks* [Alb99]. The term scale-free refers to the fact that it is not possible to pick a node of 'typical' scale or degree. Consequently, measures such as the average degree for all nodes convey very little information about the network. In a scale-free network, the distribution of degrees follows a *power law*. A power law is present when the frequency of an event varies as a power of some attribute of that event. This results in the characteris-

tic skewed, long-tail distributions. These are often the underlying distributions to popular "80-20-rules", rules of thumb in various fields which report 20 % of the causes contribute 80 % of the outcome (with 80-20 being, essentially, an arbitrary threshold). In particular, the phenomenon of *participation inequality* (communities being divided into few highly prolific contributors and many small contributors, despite low barriers to participation) is often framed in this way. We continue with an analysis of the degree distribution in the collaboration network. As a first test, relating the frequency and the attribute to each other on a doubly logarithmic plot may point to a power law distribution, since such a distribution would yield a straight line.

SCALE-FREE NETWORK

Definition 17. *A* scale-free network *is a network in which the frequency $P(k)$ of nodes with degree k follows a power law with coefficient γ*

$$P(k) \sim k^{-\gamma} \tag{4.2.1}$$

Thus, a feature of scale-free networks are very high degree nodes termed *hubs* (although there is no degree threshold which distinguishes hubs and non-hubs). The nodes connected to these hubs may in turn function as smaller hubs, leading to a self-similar structure of the network. For empirically observed networks, it is common to find that $2 < \gamma < 3$.

Many real-world networks, including the collaboration network of Hollywood actors and links on the world wide web appear to be scale-free [AB02]. As an example from the field of scientometrics, the scale-free property has been shown for citation networks [Pri65]. Relevant to the network at hand, there is even *"Price's Law"*, which states that 25% of scientific authors are the source for 75% of published papers [dSP86]. Accordingly, we expect to see the familiar pattern of few highly prolific contributors and many smaller contributions, both in terms of publications and collaborations, in the *DBLP* data. This expectation is confirmed in the following.

We consider the histogram $h(k)$ for the degree distribution of the coauthorship graph G_A (Fig. 4.2 and 4.3), where the degree is equivalent to the number of coauthors. In order to determine γ, the logarithm is applied on both sides ($\log k \mapsto \log h(k)$), and a linear regression (least squares) on the resulting value pairs is performed. This yields the coefficients γ shown in Tab. 4.2. Histograms for author and publication nodes in $G_{\mathbf{PA}}$ are included in the Appendix (Fig. A.1)

graph	nodes	γ
$G_{\mathbf{A}}$	$V_{\mathbf{A}}$	2.889
$G_{\mathbf{PA}}$	$V_{\mathbf{A}}$	2.760

Table 4.2.: Scale-free network coefficients

The process of *preferential attachment* has been proposed as the mechanism at work in the formation of scale-free networks [AB02]. In this model, nodes are added and create links to existing nodes with a probability proportional to their degrees. It is straightforward to imagine a similar kind of *cumulative advantage* in the network of (co)publications. Established authors have more routine and opportunities for publishing and acquiring collaborators. As researchers move up in the academic hierarchy (which requires a history of publications), they commonly become coauthors to works of their students. One might also conjecture that researchers select coauthors based on reputation, which is in turn tied to the history of publications. All of these diverse factors can contribute to a network formation process similar to preferential attachment.

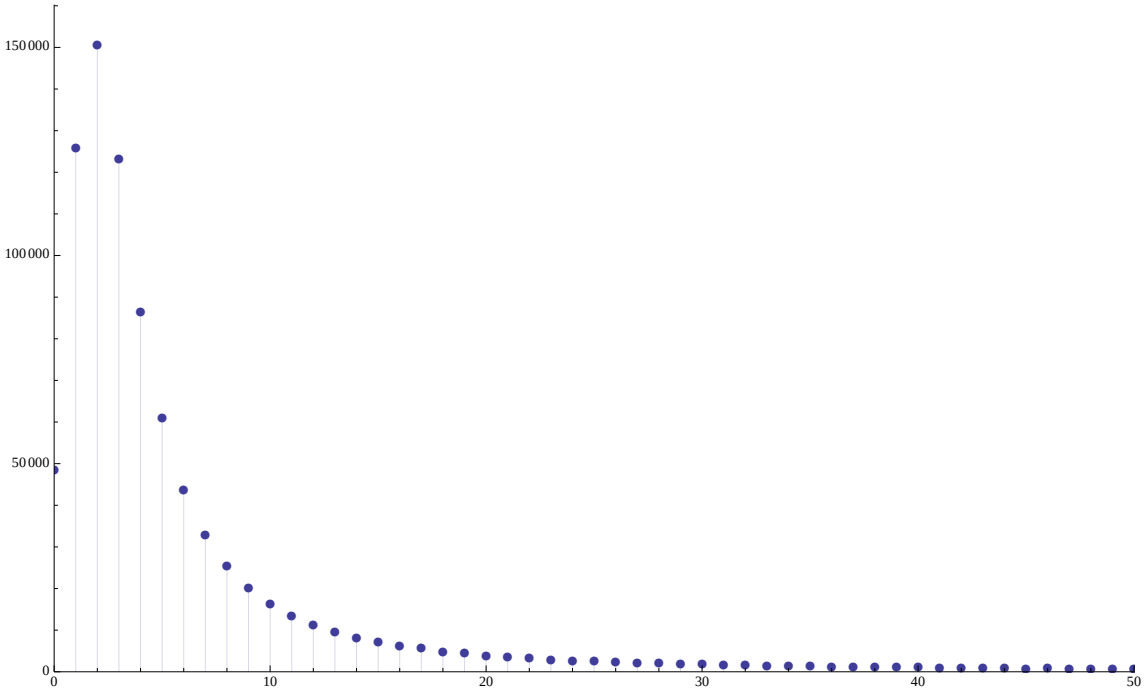


Figure 4.2.: Histogram for the number of coauthors, using $G_{\mathbf{A}}$

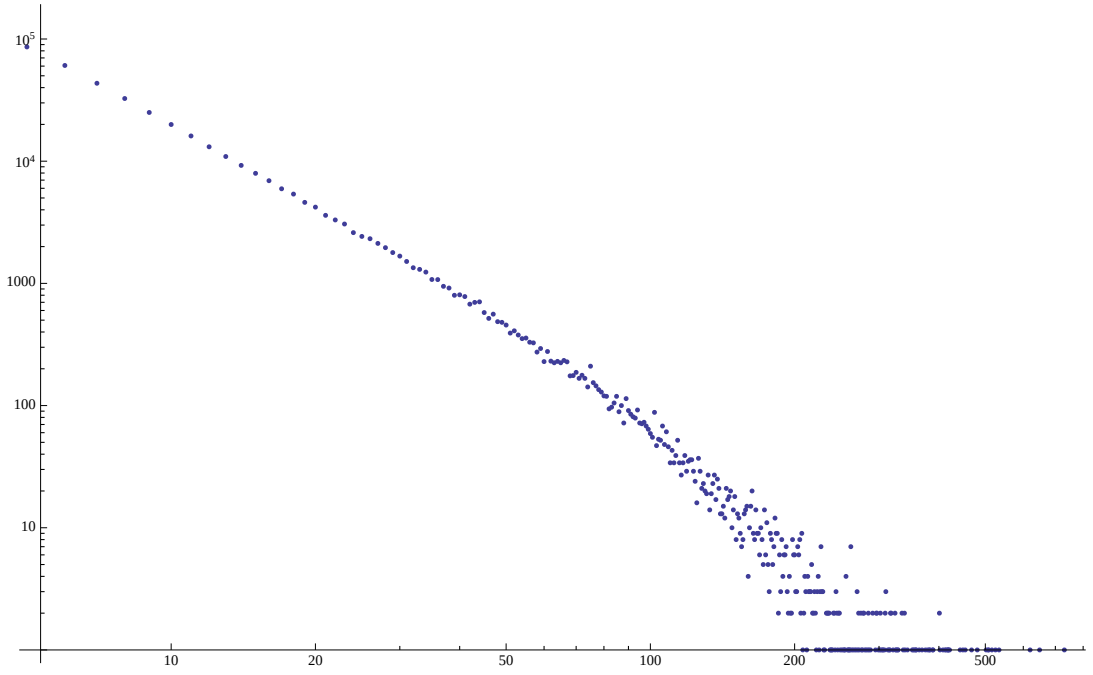


Figure 4.3.: Histogram for the number of coauthors in $G_{\mathbf{A}}$, doubly logarithmic scale

4.3. Core Decomposition

After identifying connected components of the graph, the internal structure and contours of the components can be examined in more detail. We can imagine that the graph is composed not only of connected components, but of successively more cohesive layers. One approach for identifying such a nested structure based on degree is *k-core decomposition.*

Definition 18. *A k-core C_k is a maximal subgraph in which each node is adjacent to at least k other nodes.* k-CORE

A connected component is thus equivalent to a 1-core. A k-core decomposition also lets us categorize nodes according to the highest-order core in which they are contained, assigning a *core number* to each node.

Definition 19. *The core number k_v of a node $v \in V$ is the highest value of k for which there is a k-core which contains v.* CORE NUMBER

We arrive at the k-core by iteratively removing nodes with degrees $1 \ldots k-1$. In each iteration, a certain fraction of nodes is removed, leaving a smaller core of more connected nodes. This process also yields the *core collapse sequence*, a list of the numbers (or fractions) of nodes removed in each iteration [Sei83]. A uniform sequence indicates uniform density and cohesiveness of the graph. An irregular sequence points to the presence strongly cohesive groups of nodes embedded in shells of more peripheral, weakly connected nodes [Sco00]. In the following, a core decomposition of $G_{\mathbf{A}}$ is performed, indicating whether the pattern of collaboration is regular, or if it is marked by strongly connected, clique-like groups of authors surrounded by more peripheral researchers. A core decomposition of $G_{\mathbf{PA}}$ would be less interesting, as publication nodes do not accumulate links over time and typically have a low degree, and author nodes have only connections to publication nodes.

Algorithm 2: computeCoreNumbers

Input: graph $G = (V, E)$
Output: core-numbers K_G of all vertices in G
$D \leftarrow$ array of degrees for all $v \in V$
sort V in increasing order by degree D
$J \leftarrow$ array where entry $J[i]$ is the minimum index j such that for all $r \geq j$, vertex $V[r]$ has degree at least i
for $v \in V$ *in sorted order* **do**
  $K[v] \leftarrow D[v]$
  for $u \in N(v)$ **do**
    if $D[u] > D[v]$ **then**
      if $u \neq w$ *at position* $J[D[u]+1]$ **then**
        swap u, w in V
      $J[D[u]+1] \leftarrow J[D[u]+1]+1$
return K

Algorithm 2 was implemented and applied for the calculation of core numbers, as described in detail in [Erl05]. Fig. 4.4 shows a histogram of the resulting core numbers in $G_{\mathbf{A}}$ with two logarithmic axes. The histogram indicates a regular core decomposition sequence, and therefore an overall uniformity of the network. A look into the table of core numbers for authors shows that publications with unusually high author counts are the cause of very high core numbers: The innermost core, a 112-core, results entirely from a single publication, [AAA+02]. The same holds for the next smaller 100-core. Yet, core numbers

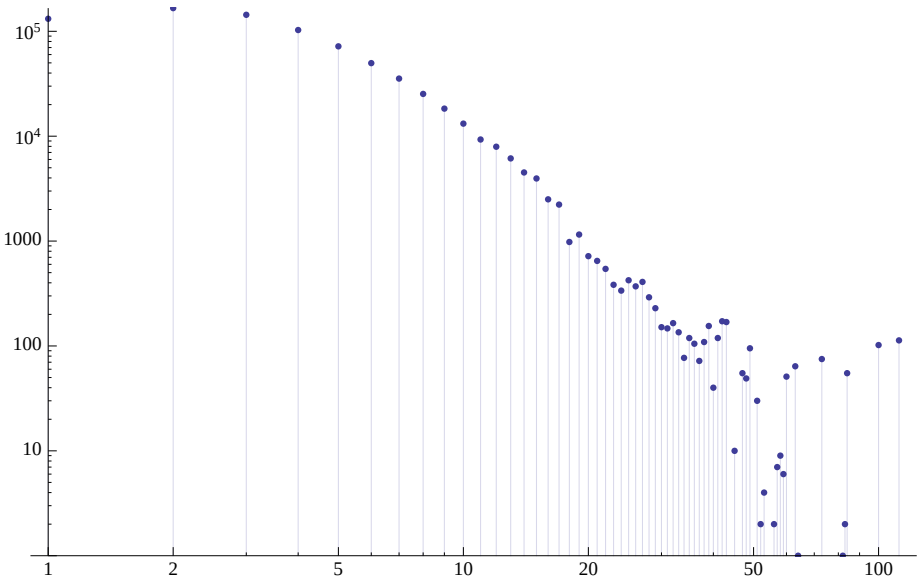


Figure 4.4.: Histogram of core numbers in $G_{\mathbf{A}}$

of about 50 are reached by accumulating coauthors from many publications. Tab. 4.3 also lists some examples of core numbers for a few select computer scientists. It is obvious that the core numbers point to inner circles of highly connected collaborators. Whether this can be used to consistently rank authors according to their influence on the field remains unclear. For another approach to this question, see also Ch. 7.

author	k
Alan M. Turing	1
Edsger W. Dijkstra	6
Donald E. Knuth	11
Guido van Rossum	8
Tim Berners-Lee	15
Andrew S. Tanenbaum	48

Table 4.3.: Core numbers in $G_{\mathbf{A}}$ for a few well-known computer scientists

5. Clustering

5.1. Modularity-driven Clustering

5.1.1. Basics

Clustering is concerned with partitioning the node set into disjoint subsets (clusters), the result of which is called a *clustering* (see 1.2.4 for formal definition and notation). The task is to decompose the graph into 'natural groups' of nodes which are clustered together, according to some clustering paradigm. The predominant approach is the *intra-cluster density versus inter-cluster sparsity* paradigm, according to which an ideal clustering should identify groups of nodes which are internally densely connected, while only sparse connections exist between the groups. One of the primary measures of clustering quality based on this paradigm is *modularity* [Gir04]. *Modularity* is based on *coverage*, a simpler quality index which divides the number (or weight) of edges contained within clusters to the total number (or weight). Maximizing *coverage* means minimizing the number of inter-cluster edges. *Coverage* maps onto $[0, 1]$, with the *singleton clustering* and the *1-clustering* occupying the two extremes.

Definition 20. *For a graph $G = (V, E)$ and a clustering ζ of G,* coverage *is defined as* COVERAGE

$$cov(G, \zeta) := \sum_{C \in \zeta} \frac{|E(C)|}{|E|} \tag{5.1.1}$$

The fact that the *1-clustering* achieves optimal *coverage* but rarely constitutes a meaningful result is an obvious shortcoming of the *coverage* index. *Modularity* remedies this by looking at the statistical significance of the clustering. We obtain *modularity* by subtracting from *coverage* its expected value. This is, roughly speaking, the expected *coverage* the clustering would achieve if the graph had the same degree distribution but was randomly connected. Now the *1-clustering* is bound to have an index value of 0, because it achieves the same *coverage* for the actual edge structure of the graph as can be expected by chance. *Modularity* maps onto $[-1, 1]$.

Definition 21. *For a graph $G = (V, E)$ and a clustering ζ of G,* modularity *is defined as* MODULARITY

$$mod(G,\zeta) := \mathcal{C}(G,\zeta(G)) - \mathbb{E}[\mathcal{C}(G,\zeta(G)] \tag{5.1.2}$$
$$= \sum_{C\in\zeta} \frac{|E(C)|}{|E|} - \sum_{C\in\zeta} \frac{\left(\sum_{v\in C} deg(v)\right)^2}{(2\cdot|E|)^2}$$

As *modularity*-maximization is $\mathcal{NP}$-hard, it is commonly maximized through heuristics [Wag08]. Several efficient heuristic algorithms exist, one of which is described and applied in the following section [Wag10] .

5.1.2. Clustering Algorithm

For all clustering purposes, we employ an algorithm termed sLocal. sLocal is a *modularity*-maximizing heuristic based on locally greedy agglomeration [Lef08][Rot09]. sLocal considers the nodes in turn, moves them to the best neighboring cluster and contracts the graph for the next iteration. Algorithm 3 describes the algorithm in pseudocode: $\Delta_{\mathcal{M}}(u,v)$ denotes the improvement in *modularity* which can be achieved by $\mathsf{move}(u,v)$, i.e. moving u to the cluster of v. The operation $\mathsf{contract}(G,\zeta)$ returns a *contracted graph* where each contracted node corresponds to a cluster in the final clustering [Sta10].

Algorithm 3: sLocal clustering algorithm

Input: graph G
Output: clustering $\zeta(G)$
$\hat{G}^0 \leftarrow G$
repeat
  $\zeta \leftarrow \{\{u\} : u \in V\}$
  repeat
    for u **in** V **do**
      if $\max_{v\in N(u)} \Delta_{\mathcal{M}}(u,v) \geq 0$ **then**
        $w \leftarrow \arg\max_{v\in N(u)} \Delta_{\mathcal{M}}(u,v)$
        $\mathsf{move}(u,\zeta(w))$
  until *no more changes*;
  $\hat{G}^{h+1} \leftarrow \mathsf{contract}(\hat{G}^h,\zeta)$
until *no more changes*;
$\zeta(G) \leftarrow \mathsf{unfurl}(\hat{G}^{h_{max}})$
return $\zeta(G)$

5.1.3. Clustering of $G_{\mathbf{PA}}$

Applying sLocal to $G_{\mathbf{PA}}$ results in a clustering $\zeta_{G_{\mathbf{PA}}}$ with of 86 761 clusters, achieving a modularity of 0.896896. See 3.1 for some implementation details. Fig. 5.1 gives an overview for the sizes of the largest clusters. The majority of clusters, containing only a handful of nodes, are omitted in the plot. Many of them are likely corresponding to the many tiny components of the graph, while the dominant connected component is divided into several large clusters (see Sec. 4.1).

5.2. Modularity-driven Clusters and Topical Clusters

With a clustering of the entire collaboration network at hand, we attempt to reach some conclusions on the significance of such a modularity-driven clustering in the context of

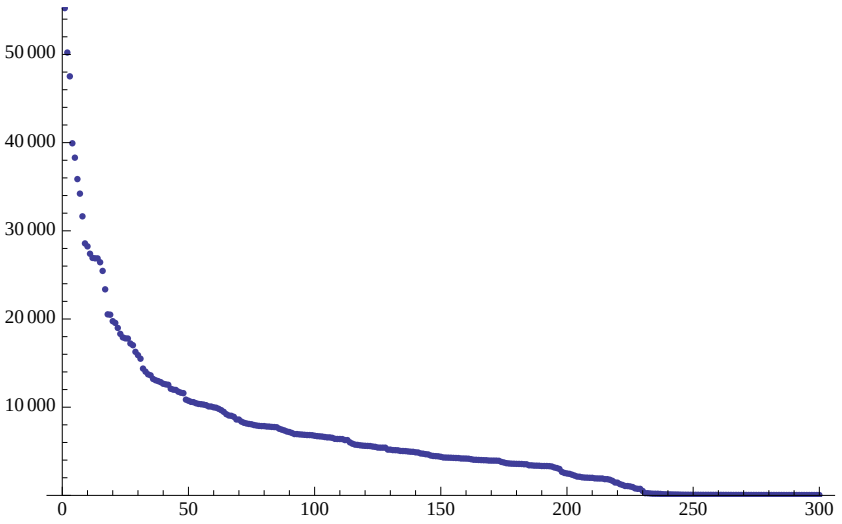


Figure 5.1.: Sizes for the 300 largest clusters

collaboration networks. The partition found by maximizing modularity identifies groups of authors who are densely connected through collaborative ties (as well as groups of publications connected through common authors), with sparse connections between the groups. Our hypothesis is that we can infer a topical similarity from these connections. More specifically, we conjecture that researchers form collaborative ties around distinct areas of research, which is reflected in the clustering structure of the graph.

We call such a hypothetical area of research, around which a strongly connected community of researchers coalesces, a *topical cluster*. Furthermore, we assume that a reasonable approximation of topical clusters can be found on the basis of conferences, as distinct research areas are likely to spawn their own dedicated conference (see Par. 2.1.1). Therefore, the set of authors participating in a conference can be taken as a topical cluster (with the caveat that this is somewhat of an abuse of the term, as the sets are likely not disjoint and do not cover the entire author set). Likewise, the set of publications which appeared in the proceedings of a conference can be treated as a topical cluster. The benefit of considering publications is that this yields an actual partition of the publications with the type InProceedings, with disjoint subsets.

The task is now to evaluate the similarity between the two partitions, one given as the modularity clustering, the other defined by conferences. We apply overlap measures to each pair of sets, arriving at a matrix of overlap values.

In the following, we describe and formalize how overlap values are calculated for pairs of topical clusters and modularity clusters. Two qualitatively different overlap measures are applied, the *Jaccard index* [Jac01] and the *overlap coefficient* [Gör10]. The Jaccard index favors exact match of the two sets (as opposed to, e.g. containment of one set within the other). However, since modularity clusters and conference-based topical clusters are likely not of equal size, we employ a second overlap measure, termed *overlap coefficient*, which is less dependent on equally sized sets and also treats containment of one set in the other set as a strong match.

Definition 22. *Let A and B be two sets. The* Jaccard overlap index *of A and B is defined as* JACCARD INDEX

$$J(A, B) := \frac{|A \cap B|}{|A \cup B|} \tag{5.2.1}$$

OVERLAP COEFFICIENT

Definition 23. *Let A and B be two sets. The* overlap coefficient *of A and B is defined as*

$$O(A, B) := \frac{|A \cap B|}{\min(|A|, |B|)} \tag{5.2.2}$$

Not all *modularity* clusters are included in the calculation: Following the assumption that modularity clusters can coincide with topical clusters (and thus, very small modularity-clusters likely result from "singleton" publications not connected to the main component), as well as to reduce computation time, a cut-off size of 100 nodes is chosen, corresponding roughly to the 250 largest clusters.

Formally, the overlap is calculated for the following sets: For a conference $c \in \mathbf{C}$, we denote the set of publications which appeared in the proceedings of the conference as P_c, and their authors as A_c. The corresponding node sets in $G_{\mathbf{PA}}$ are denoted as $V(P_c)$ and $V(A_c)$. The large modularity clusters selected are the node sets $M_i \in \{C \in \zeta(G_{\mathbf{PA}}) : |C| \geq 100\}$. For the overlap caluclation, we restrict the nodes in the modularity clusters to $M_i^{\mathbf{P}} := \{v \in M_i : \mathcal{M}^{-1}(v) \in \mathsf{InProceedings} \subset \mathbf{P}\}$ (with $M_i^{\mathbf{P}} \neq \emptyset$) and $M_i^{\mathbf{A}} := \{v \in M_i : \mathcal{M}^{-1}(v) \in \mathbf{A}\}$, respectively. The calculation yields two matrices $J_{\mathbf{P}} : M_i^{\mathbf{P}} \times V(P_c)$ and $O_{\mathbf{P}} : M_i^{\mathbf{P}} \times V(P_c)$ for the publications-view, and two matrices $J_{\mathbf{A}} : M_i^{\mathbf{A}} \times V(A_c)$ and $O_{\mathbf{A}} : M_i^{\mathbf{A}} \times V(A_c)$ for the author-view, respectively.

	$J_{\mathbf{P}}$	$O_{\mathbf{P}}$	$J_{\mathbf{A}}$	$O_{\mathbf{A}}$
max. max.	0.19770	1.0	0.16920	1.0
mean max.	0.00865	0.24082	0.01390	0.22832
max. mean	0.00254	0.01173	0.00283	0.01500
mean mean	0.00020	0.00386	0.00038	0.00394

Table 5.1.: Key figures for conference and modularity cluster overlap matrices

Tab. 5.1 shows some key figures for the values in the two matrices. A maximum O value of 1 shows at least one modularity cluster is entirely contained in the topical cluster (or vice versa). Fig. 5.3 also gives a complete picture of the distribution of the overlap values. We can see that it makes little difference whether author nodes or publication nodes are considered.

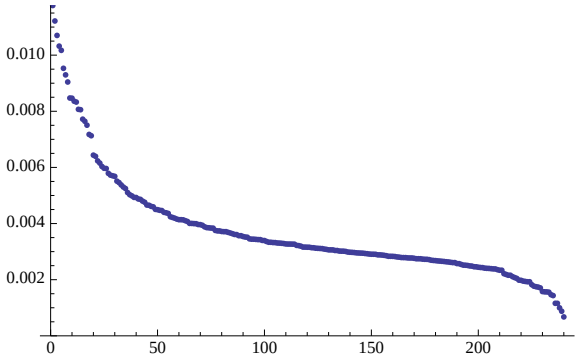
In order to give these figures more informative value, we decided to establish a baseline for the overlap values by calculating the overlap matrix with a random clustering instead of a modularity-driven clustering. The random clustering is constructed by copying the size distribution of the 250 largest modularity clusters (to achieve fairness with respect to the size-dependent overlap measures), but randomly assigning nodes from $G_{\mathbf{PA}}$ to the clusters R_i. Without a qualitative difference between publication nodes and author nodes, only the latter are considered. This yields two more overlap matrices, $J_{\mathbf{A}}^R : R_i \times V(P_c)$ and $O_{\mathbf{A}}^R : R_i \times V(P_c)$. Accordingly, the values contained are presented in Tab. 5.2 and the plots in Fig. 5.2.

With the random clustering for comparison, it is evident that the maximum J overlap is significantly better for modularity clusters, roughly by a factor of 10 for the best matches (compare Fig. 5.2(a) and Fig. 5.3(e)). It is therefore clear that a more than coincidental relation between modularity clusters and topical clusters exists. However, since the maximum overlap is only 0.198 in J_P and 0.169 in J_A, the matches are far from exact. When containment is counted as a match (O overlap), the difference becomes even clearer, with

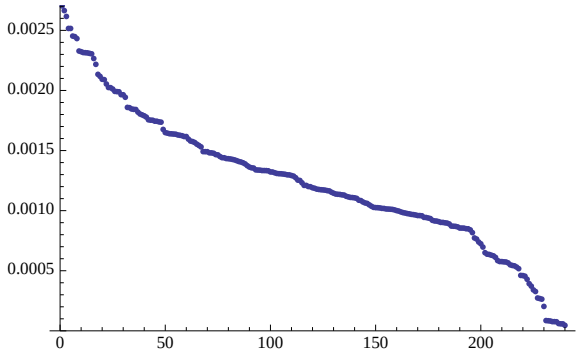
the best matches achieving an overlap of on average 0.228, versus 0.044 for random clusters. In conclusion, the overlap of modularity clusters and topical clusters as defined by conferences is clearly non-random. However, the lower than expected values indicate that the correspondence is not very strong, and factors other than joint conference participation are influential in shaping the cluster structure of the graph.

	$J^R_{\mathbf{A}}$	$O^R_{\mathbf{A}}$
max. max.	0.01176	1.0
mean max	0.00372	0.04404
max. mean	0.00270	0.023563
mean mean	0.00123	0.00506

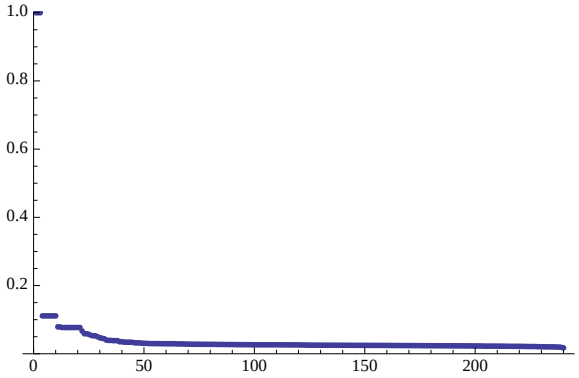
Table 5.2.: Key figures for conference and random cluster overlap matrices



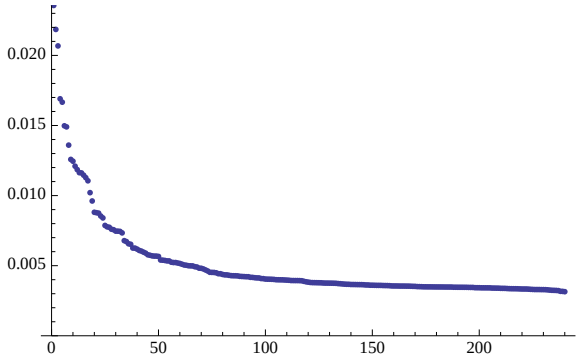
(a) maximum J overlap for authors



(b) mean J overlap for authors

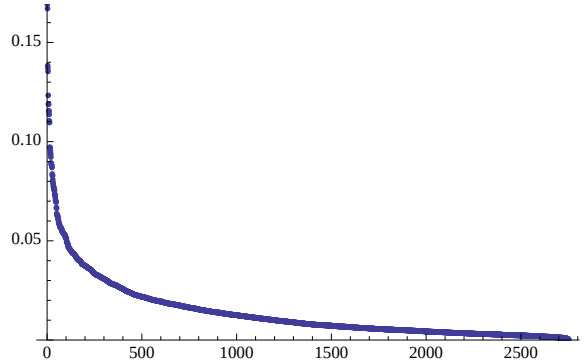


(c) maximum O overlap for authors

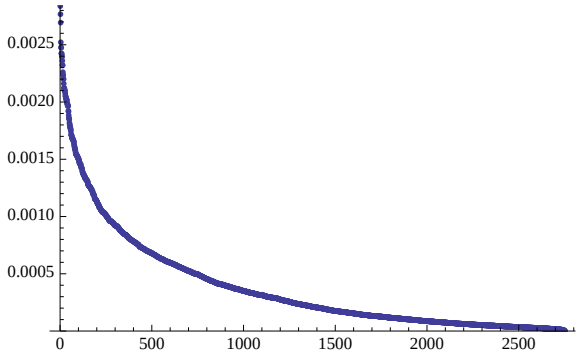


(d) mean O overlap for authors

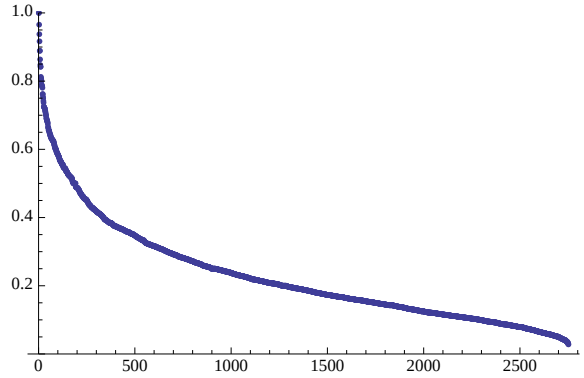
Figure 5.2.: Overlap values of conference clusters and random clusters



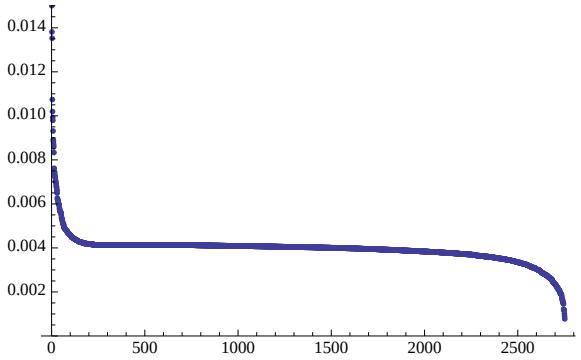
(a) maximum J overlap for authors



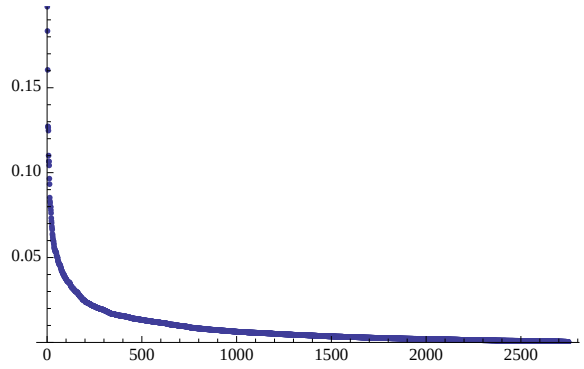
(b) mean J overlap for authors



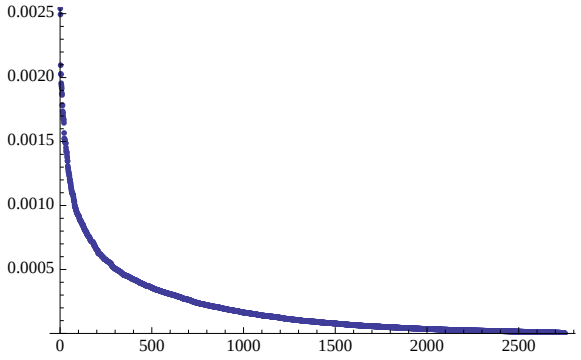
(c) maximum O overlap for authors



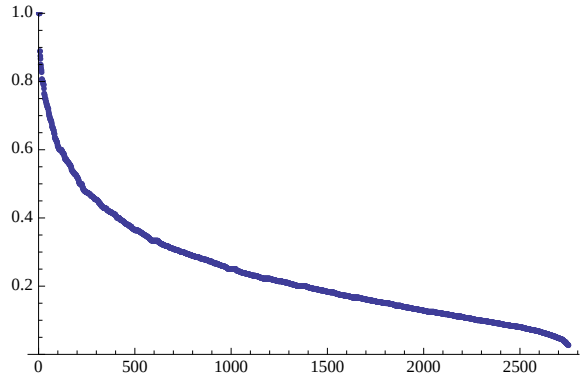
(d) mean O overlap for authors



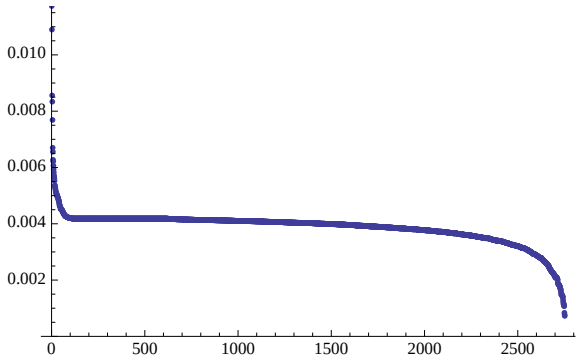
(e) maximum J overlap for publications



(f) mean J overlap for publications



(g) maximum O overlap for publications



(h) mean O overlap for publications

Figure 5.3.: Overlap values of conference clusters and modularity clusters

6. Impact of Seminar Participation

The following chapter describes our approach to the question whether participation in the *Dagstuhl seminars* leads to visible changes in the structure of the collaboration network.

6.1. Measuring the Intensity of Collaboration

In this section, we derive and introduce graph-based measures intended to quantify the amount of collaboration occurring between authors, as well as their publication output in general.

6.1.1. Designing Measures

In order to determine the effect of any factor or event on the publication output and collaboration among researchers, we must be able to quantify the latter. Modeling the relations between authors and publications as a graph allows us to employ well-known concepts from graph theory. An array of such measures is presented in the following section. To begin with, it is useful to distinguish between *copublication* and *coauthorship*. When using the term *copublication*, we focus on the relation between authors and publications, and refer to publications which the respective author has coauthored with other authors. With *coauthorship*, we focus on the relation between authors, and refer to the authors with which the respective author has coauthored publications.

Numbers of Publications and Copublications

The first two measures are based on the bare quantity of publications, the size of the following set.

Definition 24. *Given a set of authors $A \subseteq \mathbf{A}$, the set of their publications $P(A)$ is equal to* PUBLICATIONS

$$P(A) := \bigcup_{a \in A} P_a \cong \bigcup_{a \in A} N(v_a) \tag{6.1.1}$$

This set is used to measure the general publication output of all authors in a group, including single-author publications, yielding the first measure:

ap **Definition 25.** *For a set of authors A, the* average number of publications *(ap) is defined as:*

$$ap(A) := \frac{|P(A)|}{|A|} \tag{6.1.2}$$

COPUBLICATIONS **Definition 26.** *The set of* copublications *for an author a consists of publications which were written as collaborations with another author:*

$$CP(a) := \{p \in P(a) : \exists b \in \mathbf{A} : b \smile p\} \tag{6.1.3}$$

For a set $A \subseteq \mathbf{A}$, the aggregated copublications are

$$CP(A) := \bigcup_{a \in S} CP(a) \tag{6.1.4}$$

Given a set of authors A, the set of their copublications $CP(A)$ is equal to the set of neighbors of their respective nodes with degree larger than one:

$$CP(a) \cong \{v_p \in N(v_a) : deg(v_p) > 1 \} \tag{6.1.5}$$

$$CP(A) \cong \bigcup_{a \in S} \{v_p \in N(v_a) : deg(v_p) > 1 \} \tag{6.1.6}$$

acp **Definition 27.** *For a set $A \subseteq \mathbf{A}$, the* average number of copublications *is defined as:*

$$acp(A) := \frac{|CP(A)|}{|A|} \tag{6.1.7}$$

Number of Coauthorship Relations

The following sets and measures focus on the coauthorship relation between authors.

COAUTHORS **Definition 28.** *The set of coauthors for a given author $a \in \mathbf{A}$ are those authors with which a has authored a collaboration.*

$$CA(a) := \{b \in \mathbf{A} : b \frown a \} \tag{6.1.8}$$

This can be generalized for a set of authors A:

$$CA(A) := \bigcup_{a \in A} CA(a) \tag{6.1.9}$$

The set of coauthors corresponds to the *bipartite neighborhood* of the respective author nodes:

$$CA(a) \cong N_b(v_a) \tag{6.1.10}$$

$$CA(A) \cong \bigcup_{a \in A} N_b(v_a) \tag{6.1.11}$$

It is straightforward that a collaboration measure can be based on the number of coauthors:

Definition 29. *For a set of authors A, the* average number of coauthors *is* *aca*

$$aca(A) := \frac{|CA(A)|}{|A|} \tag{6.1.12}$$

$$aca(A) \cong \frac{|\bigcup_{a \in A} N_b(v_a)|}{|V_A|} \tag{6.1.13}$$

Thus, the *aca* measure yields the number of coauthors for a typical author in the given group. This may strongly depend on the author's publication output in general, but also helps to distinguish between authors publishing with a wide range or narrow range of collaborators.

Coauthorship Density

While the previous measures are simply aggregate and average, we now introduce several density-based measures. Comparing the amount of coauthorship relations present to the maximum possible amount, we arive at *coauthorship density*, illustrated in Fig. 6.1.

Definition 30. *For a set of authors A, the* coauthorship density *defined as* *cad*

$$cad(A) := \frac{\{\{a,b\} \in \binom{A}{2} : a \frown b\}}{|\binom{A}{2}|} \in [0,1] \tag{6.1.14}$$

Clearly, this corresponds to density in bipartite graphs. The measure has a maximum of 1, at which all pairs of nodes are connected by at least one bipartite path, forming a clique in terms of the coauthorship relation.

$$cad(A) = dens_b(V_A, G) \tag{6.1.15}$$

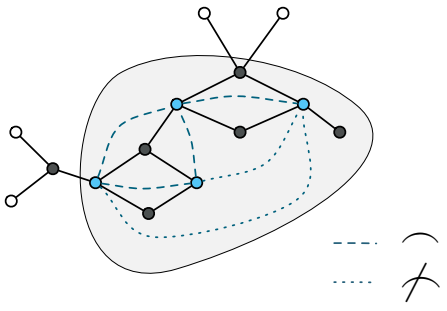


Figure 6.1.: Illustrating *cad*: $cad(A) = 2/3$

Internal versus External Collaboration

Measures like *acp* and *aca* show how much collaborative work is published by the authors belonging to a particular group. In order to determine whether the given group is, internally, a cluster or hotspot of increased collaboration in its local environment, we can also compare the amount of collaboration within the group to the amount of collaboration of its members with the outside world. Based on the copublication sets defined above, we can define the sets of internal and external copublications (see Fig. 6.2 for an example).

Intra- and Extra Copublications

Definition 31. *Give an author a belonging to a set of authors A. Then a's set of* intra-copublications *with respect to A is defined as*

$$CP_{intra}(a, A) := \{p \in CP(A) : \exists b \in A : a \smile p, b \smile p\} \tag{6.1.16}$$

Likewise, the set of intra-copublications of a set of authors is defined as:

$$CP_{intra}(A) := \{p \in CP(A) : \exists a, b \in A : a \smile p, b \smile p\} \tag{6.1.17}$$

Trivially, *extra-copublications* are the complementary set:

$$CP_{\text{extra}}(A) := CP(A) \setminus CP_{\text{intra}}(A) \tag{6.1.18}$$

With respect to the bipartite graph, these sets are found as follows:

$$CP_{\text{intra}}(A) \cong \{v_p \in V(CP(A)) : \exists v_a, v_b \in V_A : \{v_a, p\} \in E, \{p, v_b\} \ \in E\} \tag{6.1.19}$$

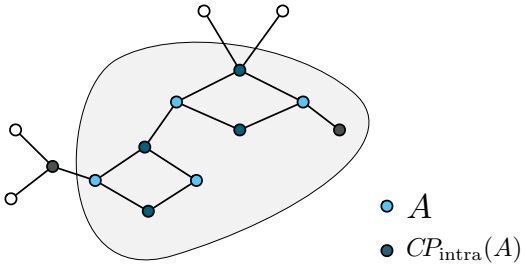


Figure 6.2.: Illustrating $CP_{\text{intra}}(A)$

This leaves us with another collaboration measure, namely the ratio of internal and external copublications:

cpr_{intra}

Definition 32. *The* internal copublication ratio *for a set $A \in \mathbf{A}$ is defined as*

$$cpr_{intra}(A) := \frac{|CP_{intra}(A)|}{|CP(A)|} \tag{6.1.20}$$

The external copublication ratio *would be the complementary value:*

$$cpr_{extra}(A) := 1 - cpr_{intra}(A) \tag{6.1.21}$$

Alternatively, a stricter definition of inner copublications could be applied: A copublication is associated with a given group of authors only if all of its authors belong to that group. The resulting sets and measure are analogous to those above. However, this definition was seen as overly restrictive, so the measure was not implemented and applied.

STRICTLY INTERNAL COPUBLICATIONS

Definition 33. *For a set of authors $A \subset \mathbf{A}$, the set of strictly internal copublications is the subset of copublications belonging to A for which all authors are in A.*

$$CP_{s\text{-}intra}(A) := \{p \in CP(A) : A(p) \subseteq A\} \tag{6.1.22}$$

$$CP_{s\text{-}extra}(A) := CP(A) \setminus CP_{s\text{-}intra}(A) \tag{6.1.23}$$

$$CP_{s\text{-}intra}(A) \cong \{v_p \in V(CP(A)) : N(v_p) \subseteq V_A\} \tag{6.1.24}$$

Contribution to Modularity

As more complex measure to identify clusters of collaboration, *modularity* (previously covered in Ch. 5) can be applied. Recall that this clustering quality index is defined as

$$\begin{aligned} mod(G,\zeta) &:= \mathcal{C}(G,\zeta(G)) - \mathbb{E}[\mathcal{C}(G,\zeta(G)] \\ &= \sum_{C\in\zeta} \frac{|E(C)|}{|E|} - \sum_{C\in\zeta} \frac{\left(\sum_{v\in C} deg(v)\right)^2}{(2\cdot|E|)^2} \end{aligned} \tag{6.1.25}$$

As the latter formulation shows, modularity is calculated as a sum over the set of clusters. This means that we can refer to the partial modularity which a cluster C (or any subset of nodes) contributes to the clustering as

$$pmod(C) := \frac{|E(C)|}{|E|} - \frac{\left(\sum_{v\in C} deg(v)\right)^2}{(2\cdot|E|)^2} \tag{6.1.26}$$

In order to associate a modularity value with a set of authors, we treat the set of author nodes and all adjacent publication nodes as a subgraph, and arrive at the measure pm applicable to a set of authors:

$$pm(A) := pmod(V_A \cup V_{P(A)}) \tag{6.1.27}$$

Interpretation of values for pm is not as straightforward as for the previous measures. Since modularity favors clusterings with cluster sizes around $\sqrt{n}$, and author sets examined will be much smaller, the contribution of such sets to overall modularity will be very small, and the values will have to be compared to several decimal places.

6.1.2. Summary of Measures

In summary, the following 6 measures will be applied to quantify the amount of collaboration, as well as publication output in general. Tab. 6.1 also summarizes the definitions.

- $ap(A)$: average number of publications for author set A
- $acp(A)$: average number of copublications for author set A
- $aca(A)$: average number of coauthors for author set A
- $cpr_{\text{intra}}(A)$: fraction of copublications for author set A which are internal to A
- $cad(A)$: coauthorship density for author set A
- $pm(A)$: partial modularity for the subgraph defined by author set A and its adjacent publications

A better understanding of theses measures can be gained if we consider not only their formal definitions, but the questions they might help to answer:

- $ap(A)$: What is the general productivity of an average author from the group?
- $acp(A)$: What is the productivity of such an author in terms of collaborations?
- $aca(A)$: With how many other authors does an average author from the group collaborate?
- $cpr_{\text{intra}}(A)$: Do the authors collaborate more often within the group or outside of the group?
- $cad(A)$: How close is the group to a collaborative clique, i.e. a group in which all authors have collaborated with each other?
- $pm(A)$: Does the group of authors correspond to a significant cluster (in terms of the *intra-cluster density vs inter-cluster sparsity* paradigm) in the graph?

measure	definition
$ap(A)$	$\frac{\lvert P(A)\rvert}{\lvert A\rvert}$
$acp(A)$	$\frac{\lvert CP(A)\rvert}{\lvert A\rvert}$
$aca(A)$	$\frac{\lvert CA(A)\rvert}{\lvert A\rvert}$
$cpr_{\text{intra}}(A)$	$\frac{\lvert CP_{\text{intra}}(A)\rvert}{\lvert CP(A)\rvert}$
$cad(A)$	$\frac{\{\{a,b\}\in \binom{A}{2} : a \frown b\}}{\lvert\binom{A}{2}\rvert}$
$pm(A)$	$mod_p(V_A \cup V_{P(A)})$

Table 6.1.: Overview of collaboration measures and their definitions

We consider this set of measures suitable to examine whether seminars have the desired effect of fostering collaboration between the participants, as well as leading to an increase in published research in general. One kind of measures simply aggregates and averages, yielding group averages for general publication output (ap), the collaborative publication output (acp) and the number of collaborators (aca). The other group of measures is concerned with some notion of collaborative density, either the density of the group alone (cad), or relating the density within the group to its local environment in the graph (cpr_{intra}, pm). If the seminars examined are able to alter the structure of the network as supposed, the effect should register in terms of these measures.

6.2. Evaluation Setup

6.2.1. Time-Decomposed Authorship Graph

Sec. 2.4 introduced the basic ideas behind the modeling of publication/author data as networks. In Ch. 4, network properties were analyzed after compiling the entire set of publications and authors available into a single large graph $G_{\mathbf{PA}}$ (or $G_{\mathbf{A}}$, respectively). In order to determine whether certain events as social factors have effects detectable in terms of the network, we need to introduce time. More precisely, we need to be able to track groups of authors over the course of time, using a sequence of graphs in which each graph in the sequence represents a current snapshot of the authorship relations. In the following, the construction of this graph is described.

Roughly speaking, we approach this by ordering publications by publication year in ascending order, and then successively including publications from a narrow time segment to construct the next graph in the sequence. Because each graph in the sequence is meant to represent the current state of the collaboration network, links should expire after a certain period of time. At each iteration, we remove the publications (and therefore, links between authors) from the previous time segment. It can be argued that author nodes should also expire, as many authors have certainly retired over the time period captured by the publication data, thereby dropping out as potential collaborators. However, no straightforward rule for retiring author nodes presented itself. Removing an author node after the last publication of the author was ruled out, because collaboration measures are only evaluated on the authors present in the network. It is clear that when studying the development of collaborative ties over time, authors ending their research careers (possibly as a result of the event studied) should count as much as authors who cease to publish or collaborate only temporarily. Removing the former kind of authors then would hide possible negative effects of the event. So rather than introducing a more complex expiration scheme for author nodes, it was decided to simply aggregate them over time, even if this suggests an opportunity for collaboration in some cases where this opportunity did not actually exist.

Consequently, Alg. 4 is used for constructing the graph sequence. The following definitions are used: Let $t(p)$ yield the publication date of publication p. Then the publications from a time segment $[y, z], z \geq y$ are defined as

$$\mathbf{P}_{[y,z]} := \{\ p \in \mathbf{P} : y \leq t(p) \leq z\} \tag{6.2.1}$$

The corresponding authors to these publication are defined as

$$\mathbf{A}_{[y,z]} := \{\ a \in \mathbf{A} : \exists p \in \mathbf{P}_{[y,z]} : a \smile p\} \tag{6.2.2}$$

The graph sequence constructed on the basis of a sliding time segment, with parameters width w and increment s, can be defined as follows:

TIME-DECOMPOSED AUTHORSHIP GRAPH

Definition 34. *The* time-decomposed authorship graph *is a sequence of graphs $\mathcal{G}^{w,s}_{\mathbf{PA}}$ where each graph in the sequence is constructed from the publications up to $\mathbf{P}_{[y,y+w]}$ and the authors up to $\mathbf{A}_{[y,y+w]}$ according to Alg. 4, using a sliding time segment with width w and increment s.*

Algorithm 4: Construction of the time-decomposed network

Input: publications $\mathbf{P}$, authors $\mathbf{A}$,
Output: graph sequence $\mathcal{G}^{w,s}_{\mathbf{PA}}$
$\mathcal{G} \leftarrow ()$
$G \leftarrow (\{\}, \{\})$
while $y + w \leq y_{\max}$ **do**
  remove all publication nodes from G
  select $\mathbf{P}_{[y,y+w]}$ from $\mathbf{P}$
  add $V_{\mathbf{P}_{[y,y+w]}}$ to G
  add $V_{\mathbf{A}_{[y,y+w]}}$ if not present
  connect $V_{\mathbf{P}_{[y,y+w]}}$ and $V_{\mathbf{A}_{[y,y+w]}}$ according to authorship
  $\mathcal{G} \leftarrow \mathcal{G} + G$
  $y \leftarrow y + s$
return $\mathcal{G}$

Finally, this leaves the question which parameters w and s to select for the present evaluation. Several parameters were tried, but it was determined that the finest granularity of $w = 1$ and $s = 1$ is best suited to observe the (possibly short-term) effects of seminars. As all seminars took place in the 2000s, this also increases the number of available data points after the seminar. For simplicity, non-overlapping intervals were chosen.

6.2.2. Tracking Collaboration Measures for Author Groups

We hypothesize that joint participation in a seminar leads to increased collaboration between the participants. This would be measurable as increased values for the collaboration measures (cad, cpr_{intra}, pm) on the respective subgraph. Additionally, we measure whether seminar participation leads, individually, to a higher general publication output for the participants (ap, acp, aca). In order to test this conjecture, the seminar-related groups, as well as other reference groups, are tracked within the graph sequence $\mathcal{G}_{\mathbf{PA}}$: For any author set A (belonging to any of the author classes described in Sub. 6.2.3), a subset $A' \subseteq A$ has corresponding nodes $V_{A'}$ in the graph G_y. For all measures m, we evaluate $m(V_{A'})$ (or $V_{A'} \cup V_{P(A')}$) in the case of pm), yielding a sequence of values for each group (called *tracking curves* in the following). If only a few authors are present as nodes, this may lead to artificially high values (like a cad value of 1 if only two authors are present which happen to have collaborated). Therefore, we begin evaluation only if $|A'| \geq 0.25 \cdot |A|$.

6.2.3. Classes of Author Sets

Here we describe the classes of authors from which the author sets tracked are taken. They include groups related to the seminars itself, as well as reference groups used for comparison.

seminar attendees (At_s) For each seminar s, the set of researchers who attended the seminar.

seminar absentees (Ab_s) For each seminar s, the set of researchers who were invited to the seminar but did not attend. For some seminars, the set was empty or very small, so these are only included if the have a comparable size.

random samples (RS_i) Contains randomly assembled sets of authors with the size of a typical seminar.

connected samples (CS_i) Contains sets of authors found by aggregating nodes from $G_{\mathbf{PA}}$ in a breadth-first search from a random initial node until the typical size of a seminar is reached.

all authors (A) A single set containing all authors.

Obviously, the author sets in At are of primary interest when studying the effect of the seminar as a social event. The absentees Ab form an important reference class: It can be assumed that all invited researchers are seen as highly relevant to the topic by the seminar organizers, and likely to be actively and visibly publishing around the time of the seminar. It is also likely that the selected researchers show above-average publication careers anyway. Therefore, if we observe an effect on the network structure, and want to attribute it to attendance at the seminar rather than other elusive factors, the difference between attendees and absentees is the most important evidence. The other classes provide reference groups: Author groups from RS are randomly composed, so the probability that any collaborative ties exist within a group is very low. However, measures like the average publication count can be applied. The CS class consists of author groups who are connected through collaborative ties in the aggregate graph $G_{\mathbf{PA}}$. Members of such a sample group

presumably work on similar topics. Furthermore, collaborative ties can be expected to be present in each step of the time-decomposed network. Therefore, this class allows us to track the career of a randomly selected group of authors with a known probability of collaboration, but without the influence of a seminar. Finally, the entire set of authors **A** is also tracked in order to determine how the network as a whole evolves in terms of the collaboration measures. If, for example, the collaboration network as a whole was becoming denser over time, this would have to be taken into account when evaluating all tracking results.

6.3. Evaluation

In this section, result plots for tracking author groups in the time-decomposed network are presented and discussed.

6.3.1. Result Plots

To begin with, a few remarks to help with the reading of the plots: For the plots concerning seminars, one can define the year of the seminar as $t = 0$ and thereby align the curves to make a comparison of the values before and after the seminar easier. For all other classes, $t = 0$ corresponds to the year 1936, the year of the earliest publications in *DBLP*, while the latest year is 2009. Different plot marker colors indicate different author sets, however, curves for individual groups are usually not distinguishable due to the large number of groups. Upward or downward trends in the measured values as a whole should be distinguishable from the bulk of the plot markers. Because plots for seminars are aligned at the time of the seminar (all taking place in the 2000s), there is less density at the ends of the t-axis.

Average publication output remains rather constant

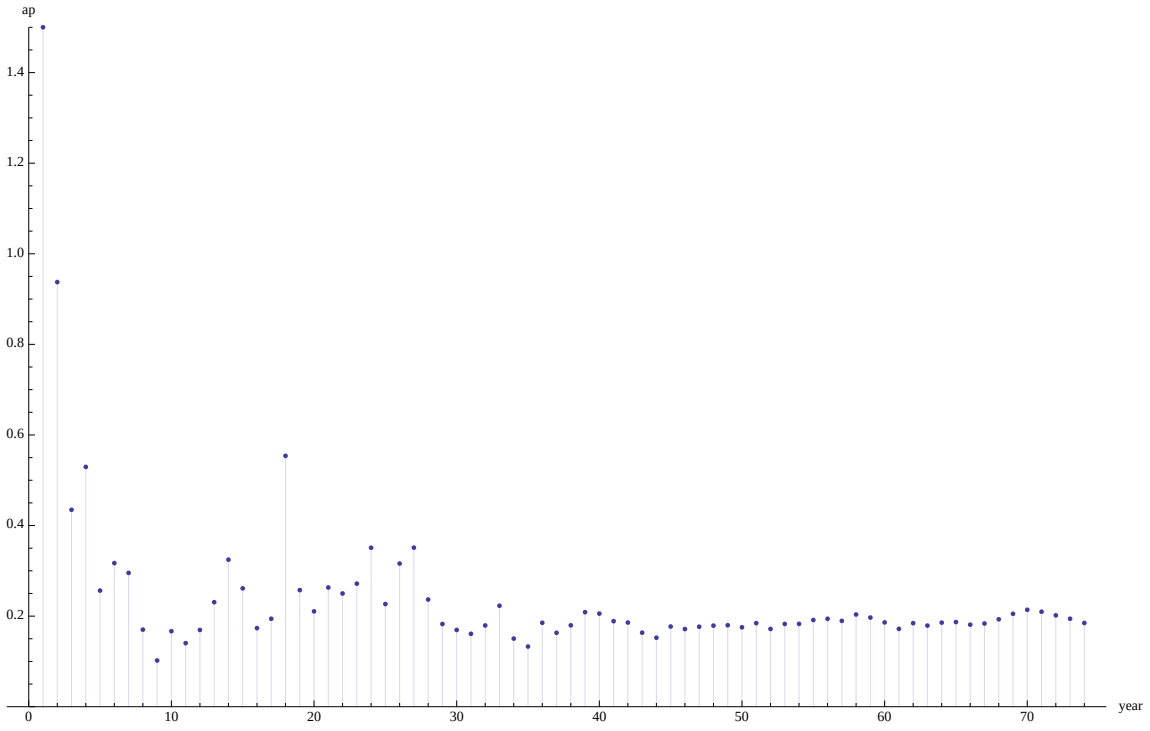
For the set of authors as a whole, average publication output and average number of coauthors remain stable over time (Fig. 6.3), even as the graph grows at an increasing rate as author nodes are accumulated (see Fig. 2.1 for the growth rate of publications). Initial fluctuations are present where the graph is still very small. Both *ap* and *aca* then even out at about 0.2. This provides us with a comparatively stable baseline for the following results.

Randomly grouped authors as a baseline for publication output

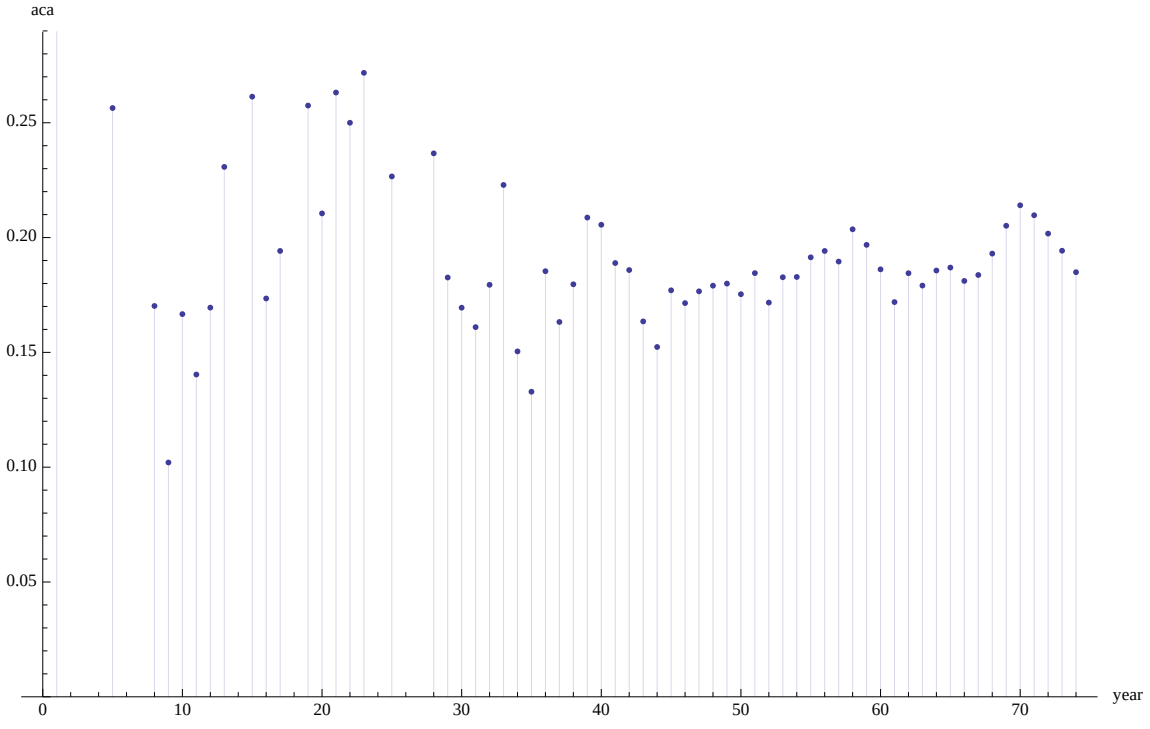
As another reference class, we evaluated the randomly compiled author groups *RS*. Both *ap* and *aca* are, on average, in the range of 0.6 - 0.8, showing that there are typically inactive authors in any given time frame (Fig. 6.4). Values are present only in the right section of the time axis, easily explained by the rapid increase in the number of authors over time. In contrast to all other author classes in the following, there is no upward or downward trend over time. For a single author, we can reasonably expect increases (and maybe decreases) over time, as there are different career stages and periods of activity. Yet, for randomly grouped authors, these differences seem to cancel each other out. As expected, there is no collaboration between authors in the random samples, so the collaborative measures cpr_{intra}, *cad*, and *pm* are omitted.

Connected sample groups

Fig. 6.5 shows a clear difference from radomly grouped authors, with a significantly higher productivity. This can be explained by the fact that *CS* includes comparatively well-connected nodes, from a section of the graph where breadth-first search was able to collect

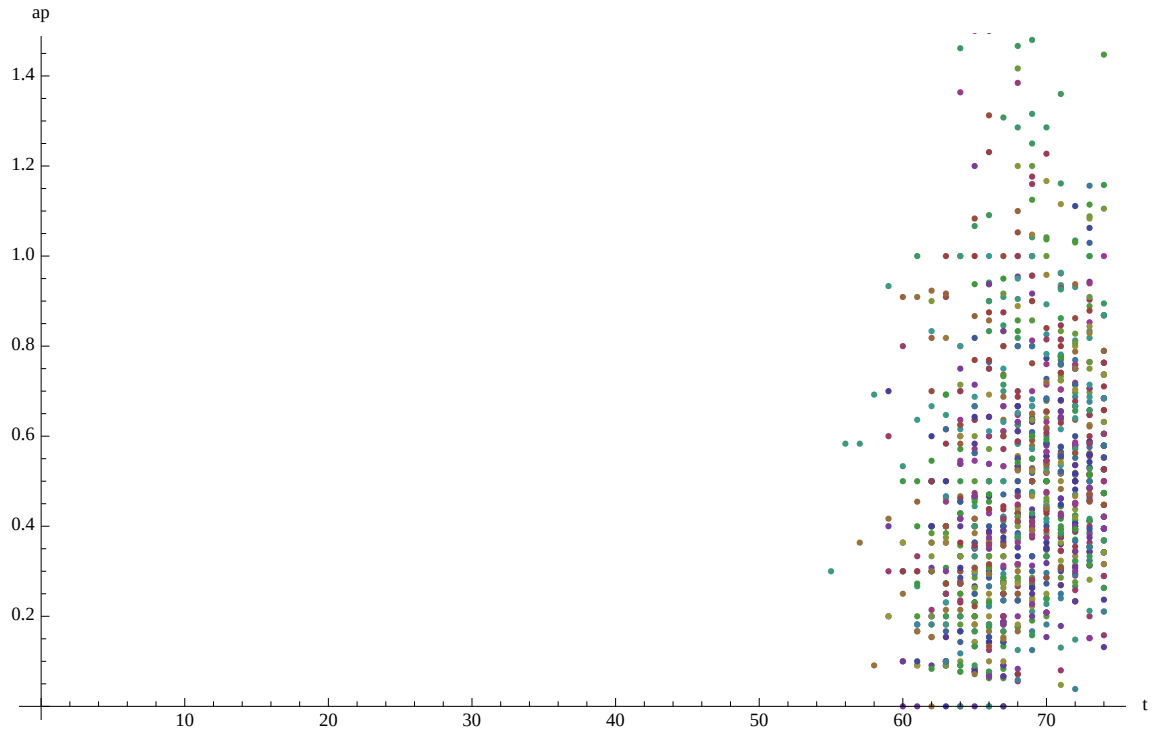


(a) *ap*: **A**

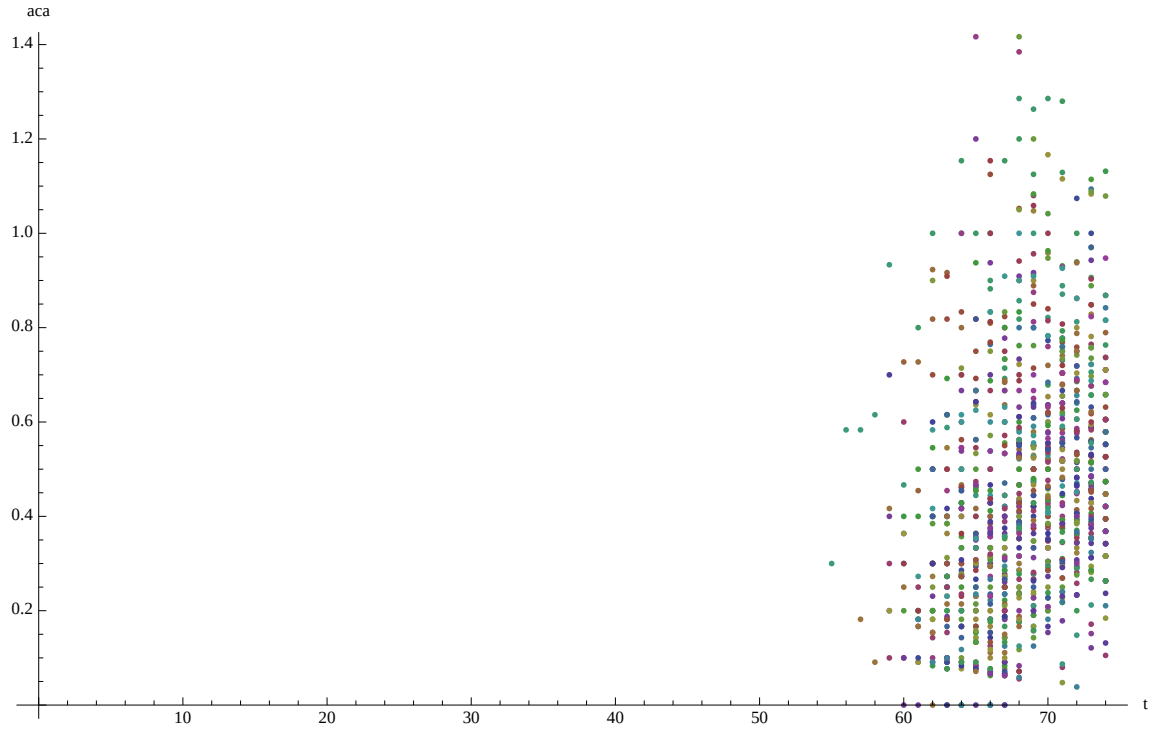


(b) *aca* : **A**

Figure 6.3.: *ap* and *aca* values for **A**

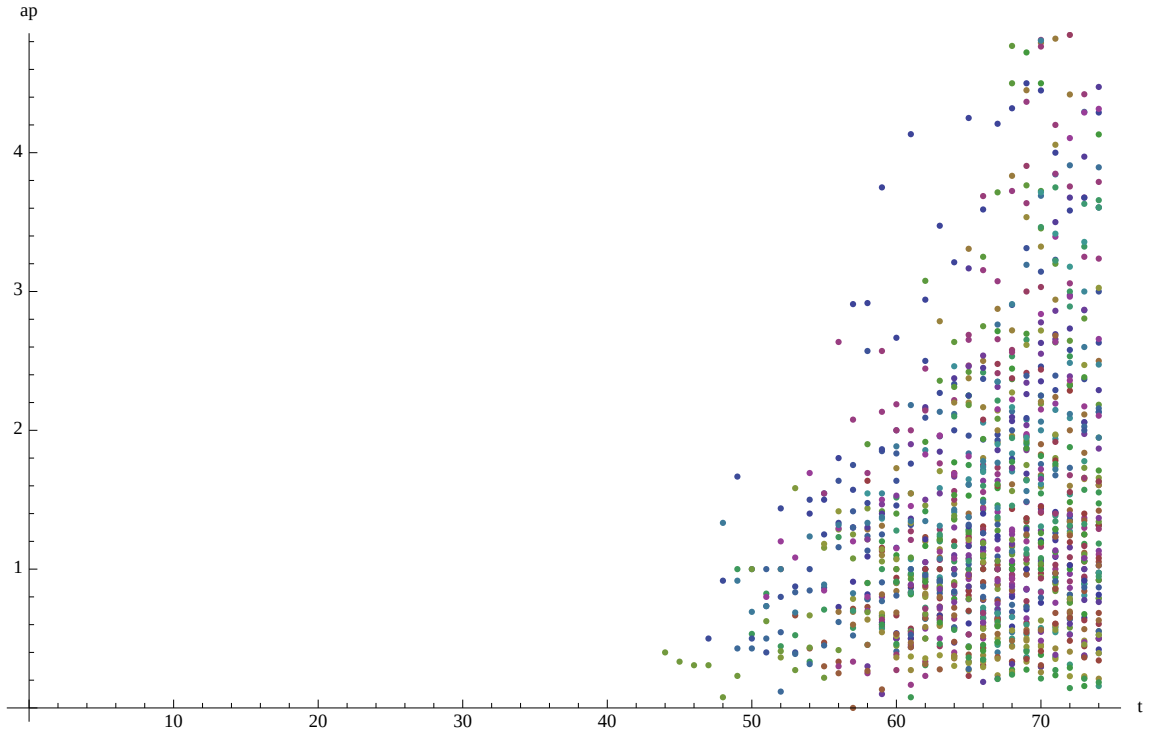


(a) *ap*: *RS*

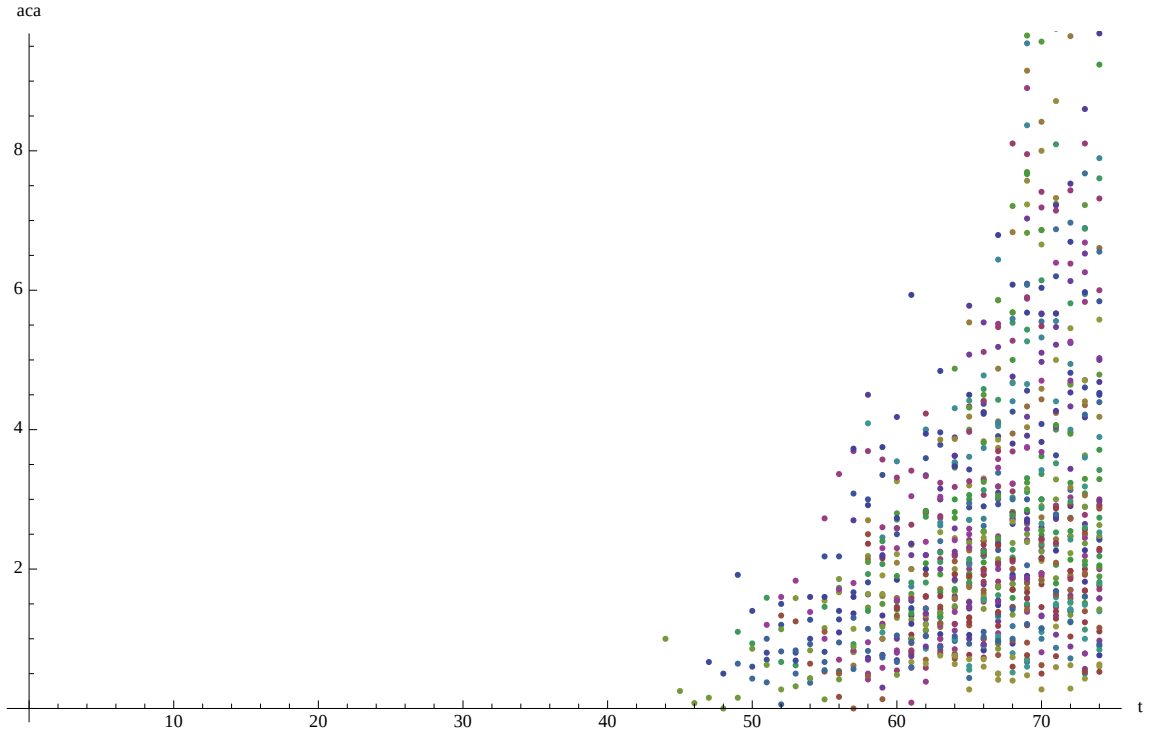


(b) *aca*: *RS*

Figure 6.4.: *ap* and *aca* values for *RS*

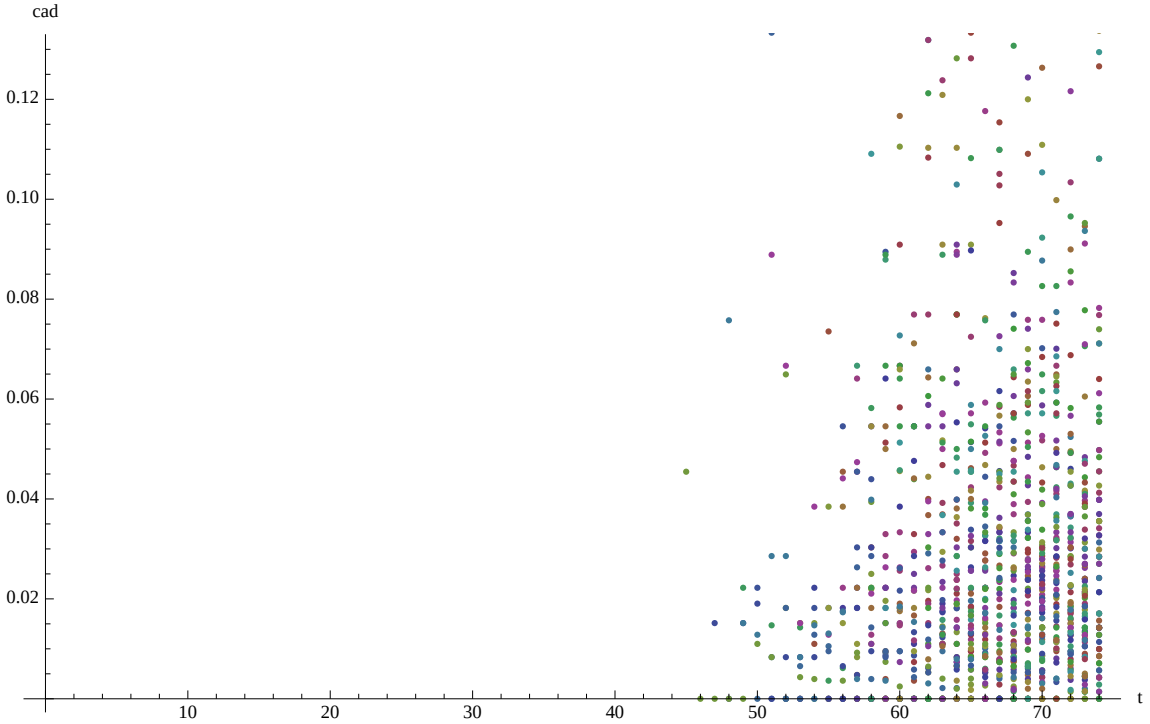


(a) *ap*: *CS*

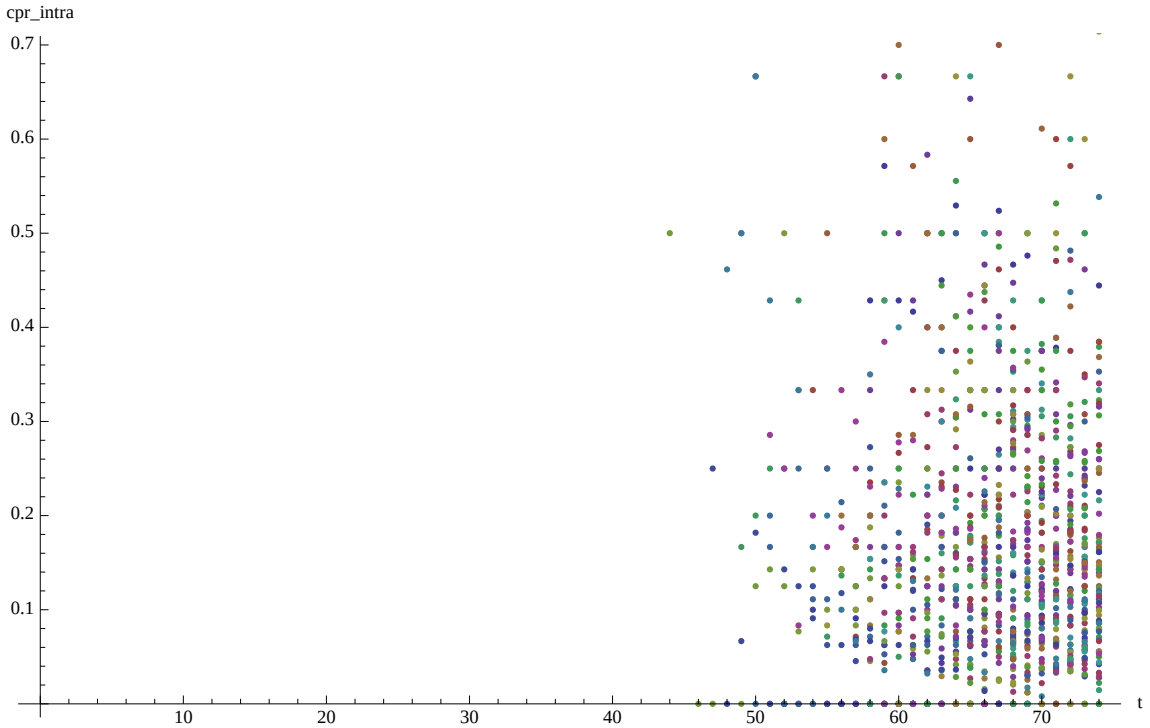


(b) *aca*: *CS*

Figure 6.5.: *ap* and *aca* values *CS*



(a) *cad*: *CS*



(b) cpr_{intra}: *CS*

Figure 6.6.: *cad* and cpr_{intra} for connected sample groups

a sufficiently large set of authors. Very low degree nodes are likely to be unconnected to the giant component and unlikely to be included in *CS*. Furthermore, breadth-first search finds high-degree nodes with a higher probability. There is a very visible upward trend. A possible factor leading to this course can be described as follows: If there is an underlying preferential-attachment-process (as indicated by the power-law degree distribution described in Sec. 4.2), then nodes gain connections over time according to degree. Overall cpr_{intra} remains clearly below 0.5, showing that these sample groups are just sections from greater collaborative clusters. Values from Fig. 6.6 serve as a baseline for the collaborative density measures.

Attendees and absentees are equally productive

The effect of seminar participation is mainly judged by comparing attendees and absentees. The number of coauthors (see Fig. 6.7) is quite similar for both groups, while the outliers of absentees surpass the attendees. The average number of publications is even more similar for both attendees and absentees (Fig. B.2 in the Appendix). In general, it became clear that plots for the *acp* measure convey basically the same information as *aca*, showing roughly the same values multiplied by a factor of 2-3, about the average number of authors per publication. The plots are therefore omitted here, and the measure neglected in the following. An increase over time can be observed, which is present for nearly all measures. After the time of the seminar, a slight decrease is visible.

Attendees form a more cohesive group

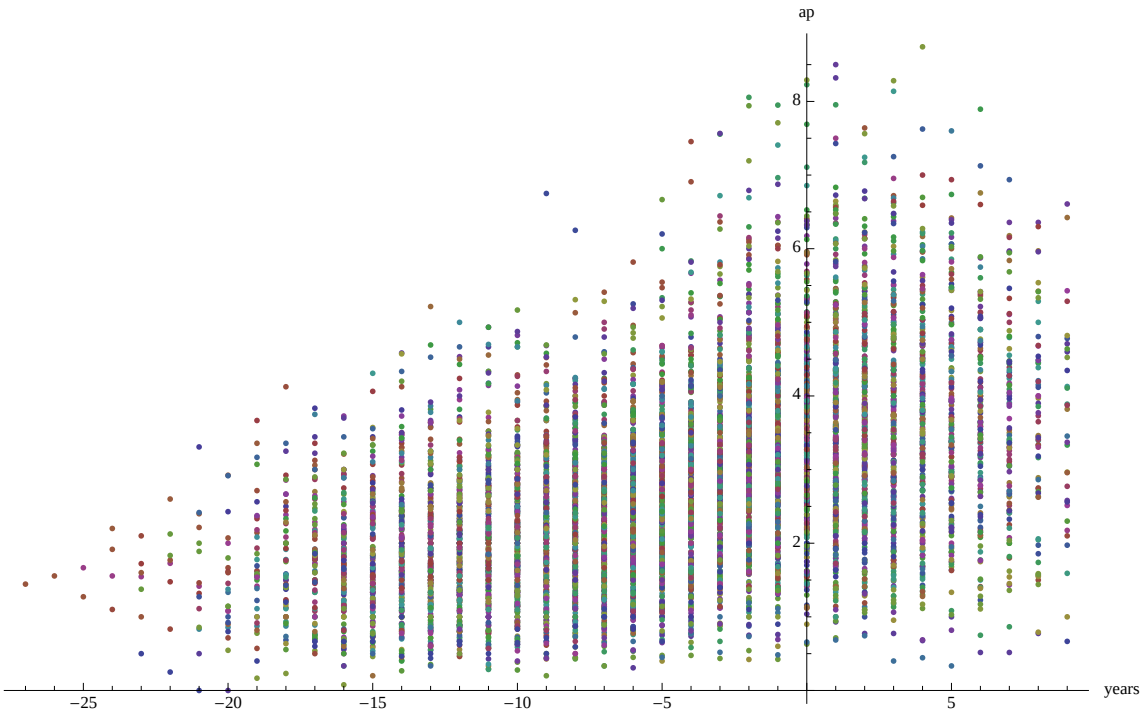
For seminar attendees, a larger fraction of their collaborations are internal to the seminar group, both before and after the seminar (Fig. 6.8). This indicates that attendees already come from a more cohesive group. Values for *cad* agree with this interpretation (Fig. B.3 in the Appendix): Clearly, those who choose to attend the seminar form a denser subgraph in the collaboration network. There seems to be no lasting increase in collaboration after the seminar, but a downward trend for both attendees and absentees. Absolute cpr_{intra} values for attendees reach those of the connected samples (Fig. 6.6), while absentees remain clearly below.

Modularity-based values are inconclusive

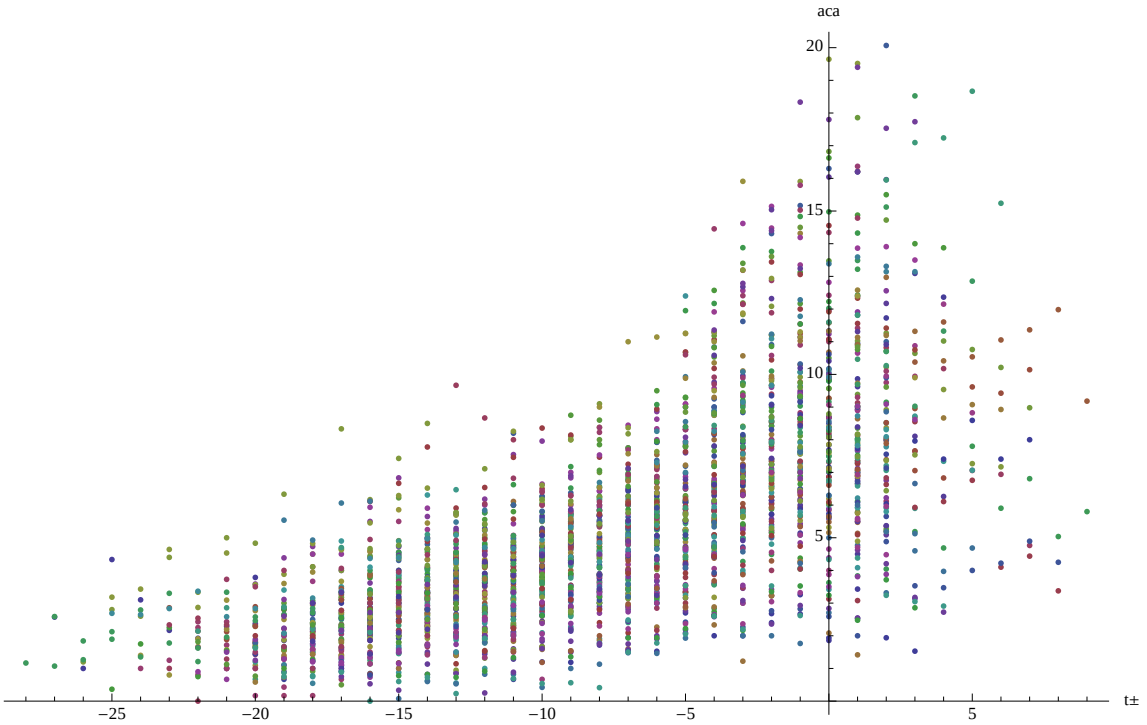
The modularity-based *pm* values (Fig. 6.9) are very similar for attendees and absentees. No seminar effect can be discerned. There is a clear downward trend, which is most likely due to the general growth of the graph: For a graph with a growing number of nodes, the modularity contributed by a comparatively stable subset of nodes becomes smaller over time. Increasing or stable values (while the graph is growing) would explicitly show that the subgraph defined by the author set becomes more significant as an internally dense and externally sparse cluster. It is possible that there is such a development here, but the growth of the graph is stronger so that it remains hidden. Furthermore, the absolute values are hard to interpret. Therefore, the measure is neglected compared to others.

Area launchers are not exceptional

Subsequently we take a closer look at 10 preselected area launcher seminars only. For this subset of seminars, we expect comparatively less collaboration before the seminar, and therefore possibly a stronger increase after. This effect would be most clearly captured by the measures cpr_{intra} (Fig. 6.10) and *cad*. (Fig. B.4 and Fig. B.5 in the Appendix show *aca* and *cad* for area launchers.) Fewer data points make it difficult to discern trends, but now individual seminars are distinguishable according to plot maker colors. The plots in Fig. 6.10 support our reasoning about area launchers (see Sub. 2.3.1), namely that the

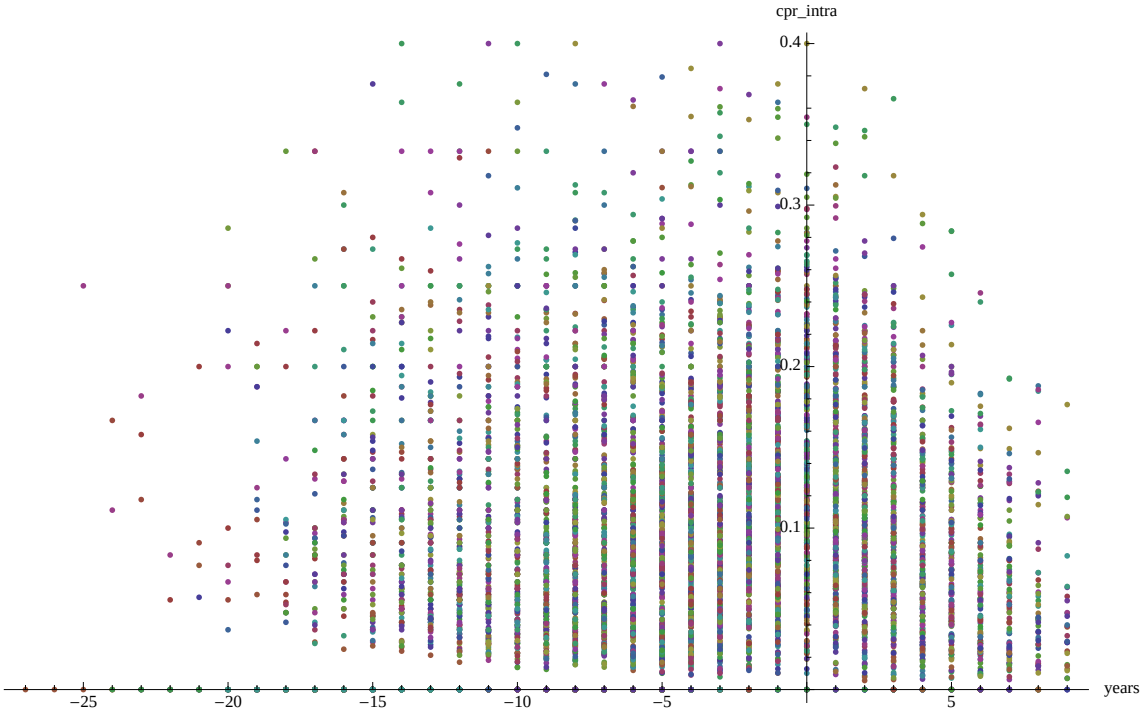


(a) *aca*: *At*

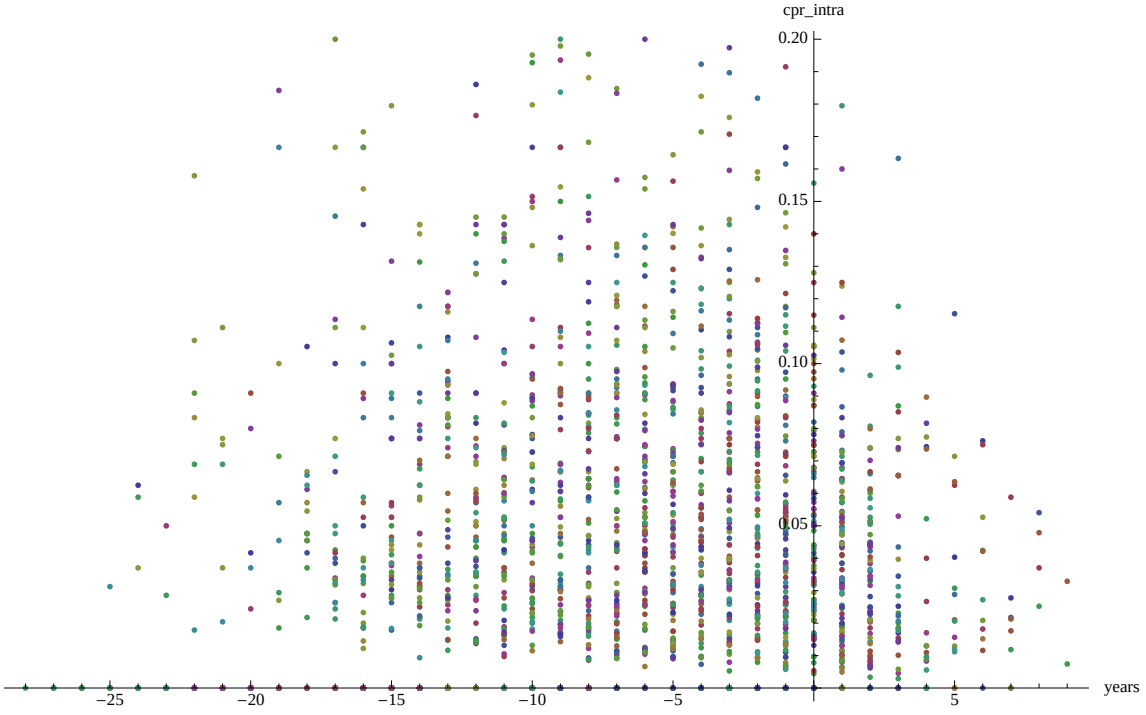


(b) *aca*: *Ab*

Figure 6.7.: *aca* for seminar attendees and absentees

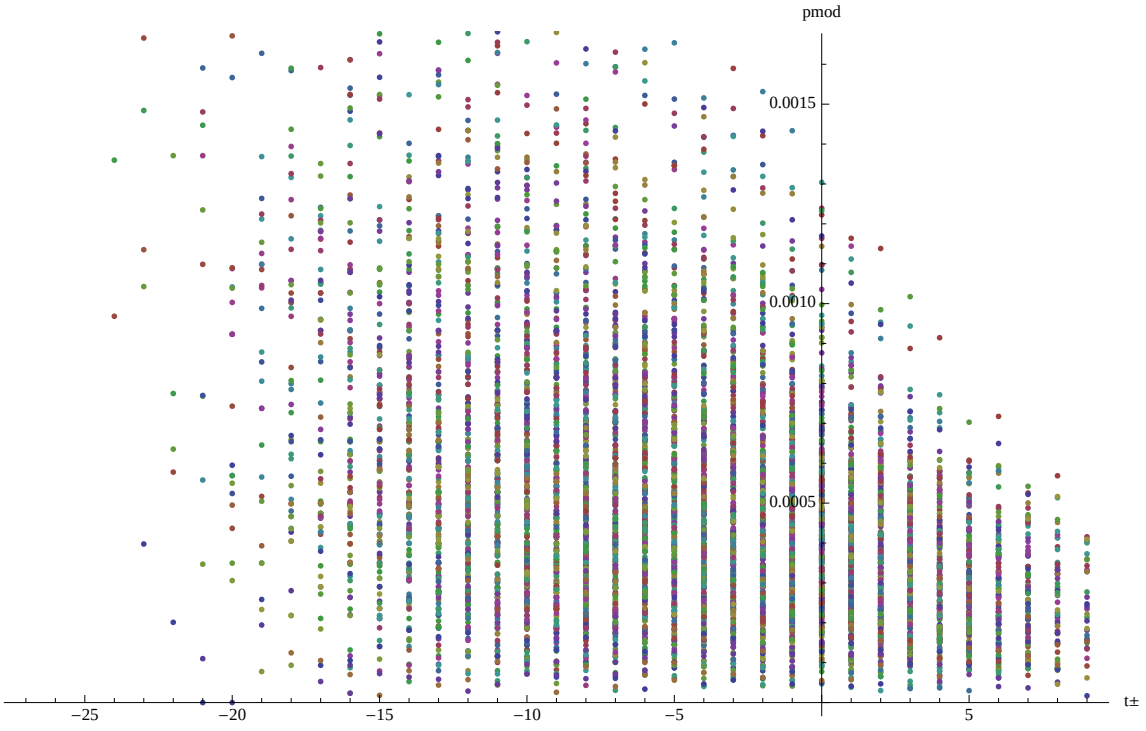


(a) cpr_{intra}: *At*

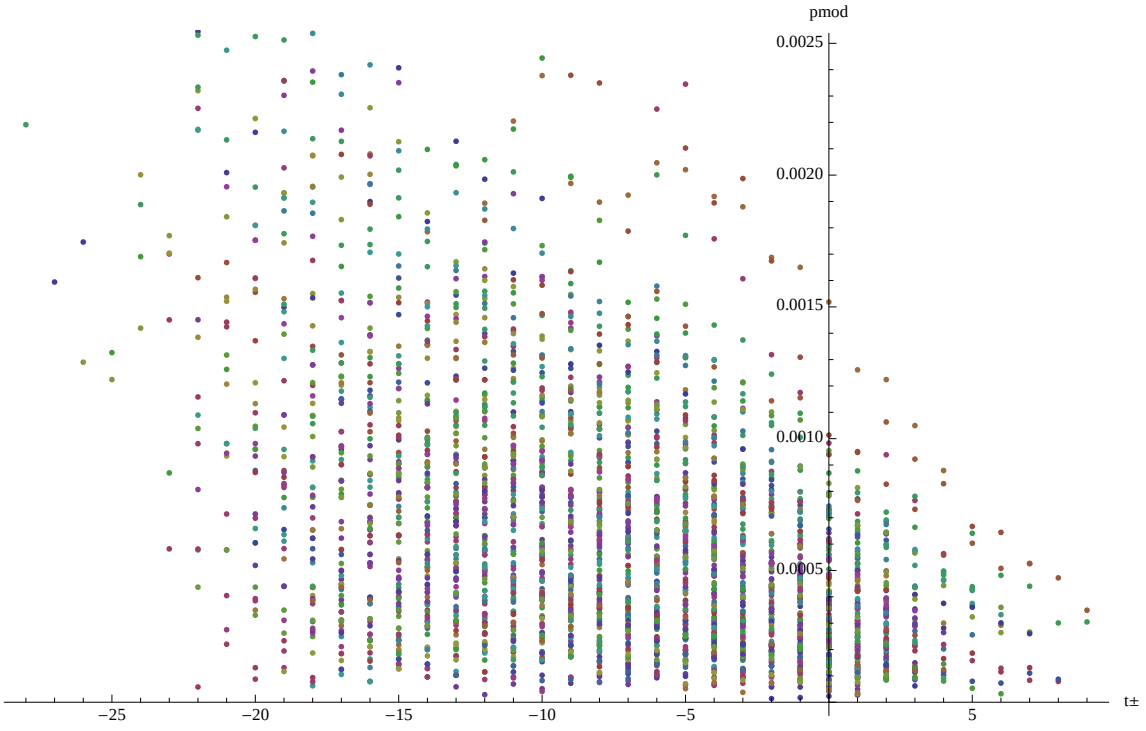


(b) cpr_{intra}: *Ab*

Figure 6.8.: cpr_{intra} for seminar attendees and absentees



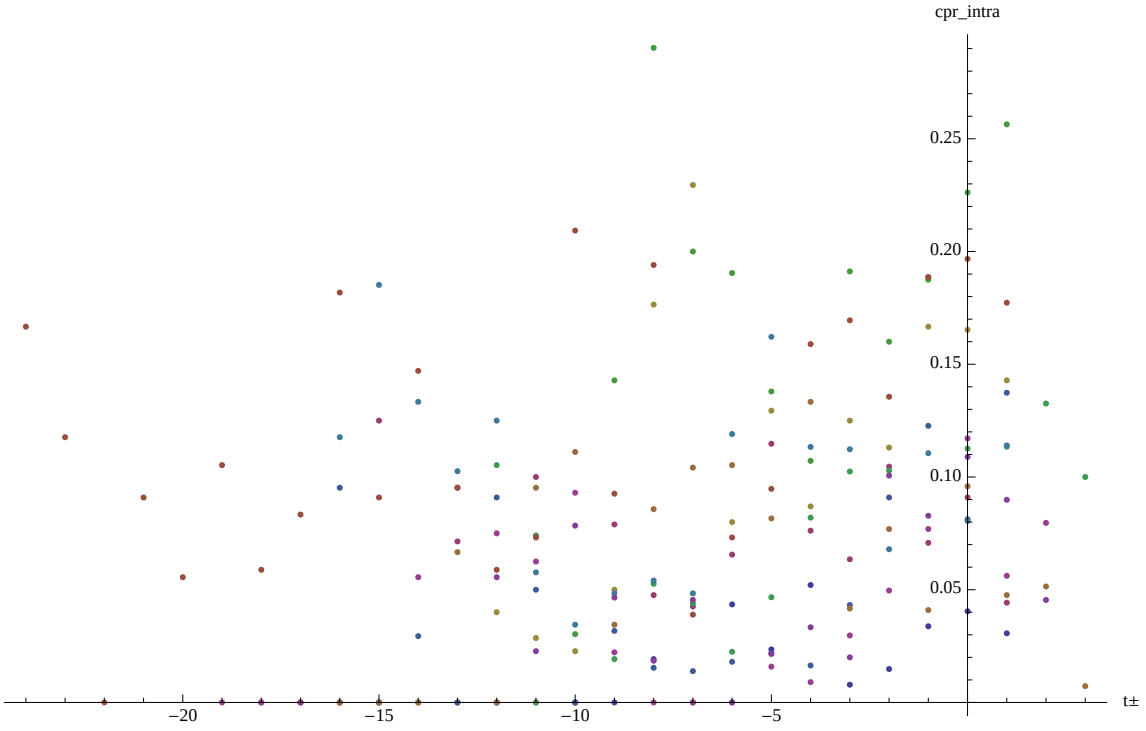
(a) *pm*: *At*



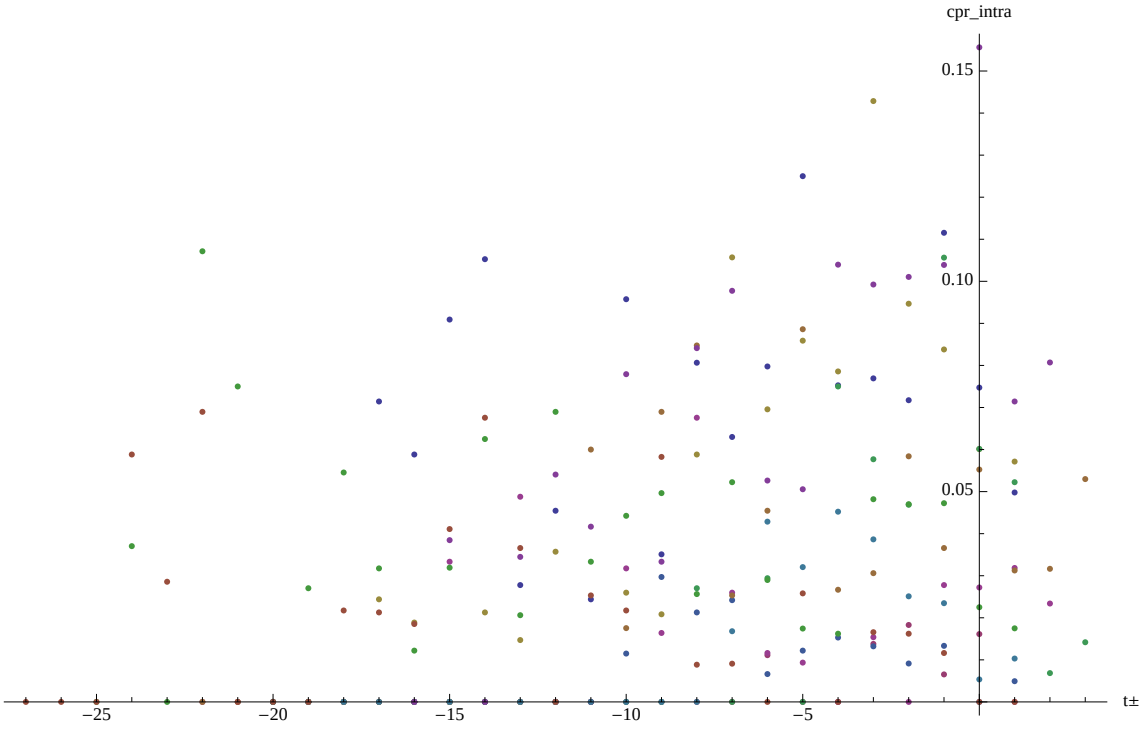
(b) *pm*: *Ab*

Figure 6.9.: *pm* for seminar attendees and absentees

authors invited have a comparatively low probability of collaboration in the time prior to the seminar: Values for cpr_{intra} are generally in the lower range compared to the set of all seminars (Fig. 6.8). Still, a visible change after the time of the seminar is missing. The influence of area launcher seminars does not seem to differ from the other seminars.



(a) cpr_{intra}: *At*



(b) cpr_{intra}: *Ab*

Figure 6.10.: cpr_{intra} for attendees and absentees of area launchers

6.3.2. Summary of Results

Generally, both seminar attendees and absentees are more productive in terms of publications and collaborations than randomly selected authors. Yet there is little difference between attendees and absentees in terms of their productivity. Invited researchers are already actively publishing, with an upward trend, prior to the time of the seminar. Absentees are as productive or more productive as attendees - other commitments may be a reason for absence at seminars. For cpr_{intra} and *cad*, attendees are consistently better than absentees. This indicates that those who attend are already a tightly connected collaborative group before the seminar, possibly influencing their decision to participate. The general trend over time is an increase up to the seminar and a slight decrease afterwards for both classes of researchers. A possible explanation for the increase and decrease over time is the following: When inviting researchers to the seminars, there is a bias towards researchers who are currently most active. Invitations to seminars occur at a period of peak activity. There is, however, no significant change of course at the time of the seminar (e.g., either significant short-term increase in collaboration directly after the seminar or long-term increase). Most importantly, attendees and absentees do not seem to differ in this respect. While the focus on area launcher seminars supported our assumption that the invited researchers had collaborated less, a significant structural change after the seminar was not visible. These results suggest that a single event like a seminar is not influential enough to alter the network structure of collaboration for the group of participants in ways observable with our measures. Clearly, other factors have more influence on the structure. Rather in the opposite direction, the network structure might be employed to predict who will attend the seminar and who will decline, since the participants evidently come from a more cohesive group.

6.3.3. Future Methodological Improvements

Our methods could not detect a seminar effect for the bulk of participants. Yet, there might be visible effects on a finer scale. For instance, it may be possible to distinguish different types of authors with regard to publication behavior: Different career stages should be distinguishable from the form of the publication and collaboration curves. There should be different durations of the productive phase. We could distinguish researchers retiring from academia after graduation or pursuing a long-term academic career. Such a career-based typology of authors could then be used to analyze in more detail. First of all, we could answer the question whether certain types of authors are present or absent at the seminars. Furthermore, this could enable us to detect previously invisible effects on the course of the researcher's career, e.g. whether the publication behavior after seminar participation is still typical. Lack of time prevented us from such analyses, but these seem to be promising starting points for future work.

7. Centrality Analysis

7.1. Introducing Centrality

Centrality refers to a family of measures designed to distinguish between peripheral and central nodes in a graph by assigning a score to each node. One of the simplest centrality measures conceivable would be *degree centrality*, being simply the degree of the node normalized by the upper bound for the degree.

Definition 35. *For a node $v \in V$, degree centrality is defined as* DEGREE CENTRALITY

$$x_{\deg}(v) := \frac{\deg(v)}{n-1} \tag{7.1.1}$$

Such a notion of centrality is purely local and based on the sheer quantity of links, regardless of the targets. Accordingly, a node with many links to peripheral nodes is ranked as more central compared to a node which has fewer links, but to nodes which are in turn well-connected. To better capture this notion of centrality, a centrality score can be defined recursively, in such a way that links to more central nodes contribute more to the centrality score of the node in focus, an approach called *feedback centrality* [Erl05].

7.1.1. Eigenvector Centrality

As an instance of *feedback centrality*, we discuss and apply *eigenvector centrality* [Bon72]. Let A be the adjacency matrix of a graph $G = (V, E)$, hence $A(i,j) = 1 \iff \{v_i, v_j\} \in E$. Let x be a vector with entry x_i being the centrality score of node v_i. If the centrality score of a node is proportional to the scores of its neighbors, then this can be expressed as

$$x_i = c \sum_{j=1}^{n} A(i,j) x_j \tag{7.1.2}$$

where $c \neq 0$ is a constant. Transposed as $\frac{1}{c}x = Ax$, this equation corresponds to the eigenvector equation $Ax = \lambda x$. It follows that the centrality vector x satisfying the equation is an eigenvector to an eigenvalue $\lambda = \frac{1}{c}$. Furthermore, the *Perron-Frobenius theorem* implies that only the greatest eigenvalue $\lambda_{\max}$ satisfies the requirement that all entries in x are positive and thus valid centrality scores. Therefore, finding the eigenvector to $\lambda_{\max}$ yields centrality values with the desired properties.

An algorithm for a variant of eigenvector centrality has been introduced as PageRank, and drives the search engine of *Google* by providing a ranking of webpages based on link-analysis [Pag98]. A requirement for strict eigenvector centrality is that the graph is connected. This is avoided in PageRank by introducing a probability for random jumps between nodes, α, which can also be interpreted as adding auxiliary edges to make the graph connected. According to the original paper, α should be chosen in the range $[0.1, 0.2]$, and we select 0.1. We use the JUNG implemenatation of PageRank for calculating centrality values in $G_{\mathbf{PA}}$.

7.1.2. Applying Eigenvector Centrality

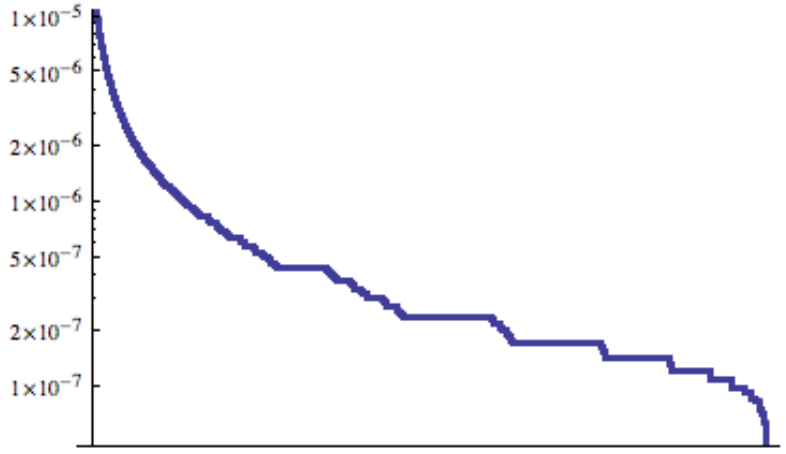
In this section, we discuss a ranking of authors and publications according to eigenvector centrality. The question is whether eigenvector centrality on $G_{\mathbf{PA}}$ allows conclusions about scientific significance, and helps to identify influential authors and publications (to the extent that there can be an objective concept of influence in science). For centrality scores calculated through PageRank on a web link graph, hyperlinks to pages are interpreted as votes concerning the quality and significance of the recipient page. Because votes from highly ranked pages carry more weight, it is possible to infer that a high rank indicates a webpage with high quality, influence or significance. Eigenvector centrality has been applied to bibliometrics, e.g. in the form of *Eigenfactor* impact scores, ranking scientific journals according to eigenvector centrality in a citation-based graph [eig11]. In this case, references are similarly treated as votes of significance.

Interpreting eigenvector centrality in $G_{\mathbf{PA}}$ is not as straightforward, since links cannot directly be interpreted as votes in the sense described above. Yet, one can argue in the following way: In the beginning of the algorithm, centrality scores are equally distributed. For the first iteration, the centrality of publication nodes depends on the number of authors, which is a fairly low and constant number for most publications, and not related to scientific significance. On the other hand, the centralities of author nodes depend at first on the volume of their publications. Here we can include the assumption that authors who are prolific in terms of output are generally also researchers with a high influence on the field. While the algorithm converges, publications coauthored by prolific researchers acquire more centrality, and authors in turn acquire centrality by collaborating on such central publications.

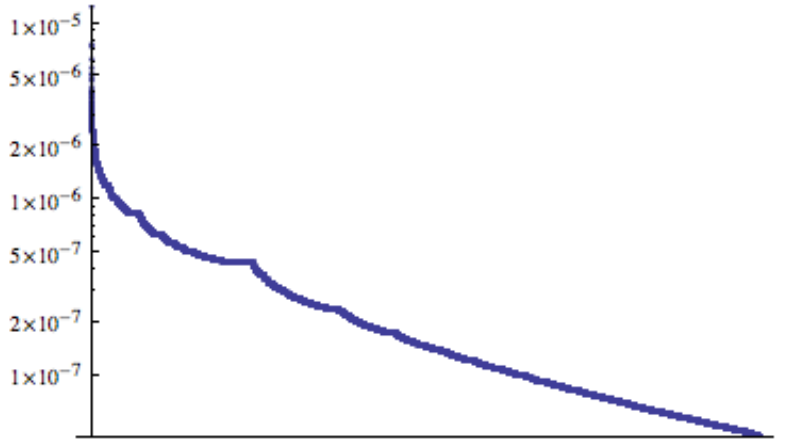
Calculating centrality scores for authors is one of the instances in which modeling the collaboration network as the bipartite graph $G_{\mathbf{PA}}$ (corresponding to a hypergraph) has advantages. This choice was motivated in Sec. 2.4 by the fact that information about the cause of a coauthorship relation is lost in graphs like $G_{\mathbf{A}}$, but strictly speaking, this information could have been associated with links in the form of weights or arbitrary attributes. However, this does not allow instances of the attributes to carry centrality scores as well [BCHJ04]. When viewing a publication as an attribute of a coauthorship relation, influencing the importance of the link, we are able to assign centrality to the attribute by including publication nodes in the graph. Consequently, centrality scores express the concept that authors are central to the collaboration network if they have worked on many central publications.

Plots of the centrality values for author and publication nodes are presented in Fig. 7.1. As expected, the distribution is highly skewed. A table of the 100 most central authors (Tab. C.1) and the 100 most central publications (Tab. C.2) is included in the Appendix. Concerning the author ranking, readers familiar with the field will probably recognize several names as prominent researchers in the field. After this admittedly superficial estimation, it is possible to argue that influential researchers generally achieve high eigenvector centrality in $G_{\mathbf{PA}}$. If this was further corroborated, it would make eigenvector centrality in

the bipartite collaboration graph a promising tool for studying the role and impact of collaborating individuals in science. In contrast, entries at the top of the publication ranking appear to be works with unusually high author counts [Sol09]. This suggests that degree is the main factor behind publication centrality, calling into question whether centrality in $G_{\mathbf{PA}}$ is meaningful as an impact measure for publications. However, a more detailed analysis of the ranking would be needed to confirm or reject the reasoning about central publications as an influence on author centrality.



(a) authors



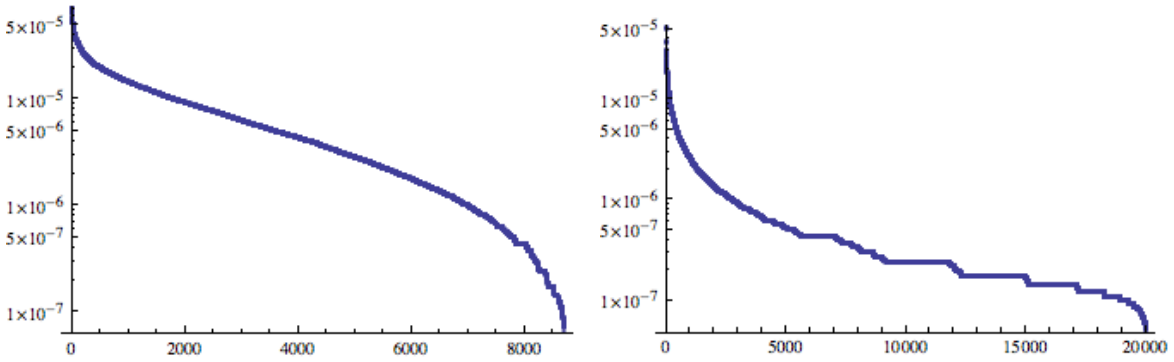
(b) publications

Figure 7.1.: Eigenvector centrality scores in $G_{\mathbf{PA}}$ (logarithmic scale)

Eigenvector centrality is most conclusive if the graph has a single clear center and periphery. If the graph has multiple, equally sized centers instead, each corresponding to an eigenvector of the adjacency matrix A, then the nodes in the largest and densest center are strongly overvalued by the algorithm applied here. Examining the largest eigenvalues of A and their ratios can provide information on the structure of the graph in this respect, and help to evaluate the eigenvector centrality values. Unfortunately, the Colt numerics library included with JUNG could not handle a sparse adjacency matrix of 2.3 million nodes and 3.7 million edges, so this test had to be omitted.

With Ch. 6 in mind, we perform an additional test on the centrality data, in order to determine whether researchers invited to the *Dagstuhl seminars* are more central in the network than other authors. We compare the centrality scores for researchers invited

to seminars to a random sample of authors not invited. The plots in Fig. 7.2 already suggest a marked difference. The median centrality scores are $3.8 \cdot 10^{-6}$ and $2.4 \cdot 10^{-7}$ for invitees and others, respectively. A *Mann-Whitney test* (a non-parametric hypothesis test with the null-hypothesis that the median of both distributions is equal) confirms that the difference is highly significant (with p-value of $\approx 10^{-2554}$). We conclude that *Dagstuhl* invitees are significantly more central than other authors. Assuming that authors are selected by the conference organizers according to some criterion of scientific influence or significance as domain experts, this result further supports the idea that eigenvector centrality in the authorship graph $G_{\mathbf{PA}}$ can serve as an objective measure of influence in scientific publishing.



(a) seminar invitees (b) random sample from authors not invited

Figure 7.2.: Centrality scores for seminar guests and others

8. Conclusion

In the course of this work, a real-world social network was explored through a variety of approaches from the toolbox of network analysis. Many of the aspects studied would have merited a more in-depth analysis, however, we aimed at providing a general overview of several network properties. These insights gave us a general picture of the network, helpful with regard to studying the impact of seminars. Before this set of analyses was feasible, much effort had to go into custom code converting raw data into an intermediate form, and finally graph form. This leaves room for improvement in the form of general tools helping with network data extraction.

The general graph properties indicate that the network of collaborations in computer science is in many respects a typical social network: It has participation inequality (visible as a power-law degree distribution), with a few highly prolific authors and many smaller contributions. It also shows a high degree of connectedness, and mostly short paths between arbitrary pairs nodes. Its texture in terms of degree is quite regular, as indicated by the core decomposition.

Since we have previously worked on modularity-driven graph clustering as an algorithmic problem, we were interested in applying it to another real-world data set, in the hope that this adds an example on how meaningful conclusions can be drawn from a modularity clustering. Assuming that authors cluster together according to topical similarity of their work, and using conferences as an approximation for the sub-fields of computer science, we compared the resulting network partitions. Overlap values suggest that conferences are influential in creating the clustered structure of the network, but that it is by no means segmented strictly according to conferences.

The central point of this work was the question how (academic as well as social) events shape the structure of the network. The *Dagstuhl seminars* provided the appropriate data, as events with the explicit goal of facilitating cutting-edge research. In the forefront, we selected area launcher seminars, bringing together previously unaffiliated researchers, according to an objective criterion based on the correspondence between seminars and conference. We designed several measures intended to capture structural changes in the graph and quantify the effect of a seminar. Using these measures, different classes of author groups (seminar attendees and absentees, connected samples, random samples) were tracked in a graph changing over time. Many of our analyses show that researchers invited to the seminars are, in fact, above-average in terms of publication output and collaborative behavior (as well as network centrality). However, much of what distinguishes them in

structural terms applies before as well as after the time of the seminar. Assuming that our measures designed to capture any effect are adequate, the impact of seminars on the collaboration network structure is not significant. The general trend of increase towards and decrease after might be due to a selection bias, in the sense that researchers who are currently very active are more likely to be invited. We must conclude that a single event like a seminar is not influential enough to alter the network structure of collaboration for the group of participants in obvious ways. There might still be more subtle effects on the collaborative behavior of researchers which evade our methods.

Finally, we calculated eigenvector centrality scores, in order to identify authors and publications central or peripheral to the network. Modeling the network as a bipartite graph of authors and publications, thereby allowing publications to carry and transfer centrality values, turned out to be appropriate here. The resulting author ranking is promising, indicating that an analysis of roles and influence on the field can be based on eigenvector centrality in the bipartite graph of authors and publications.

9. Acknowledgements

We frequently stressed the importance of collaboration in science, and this thesis was no exception: I thank my advisors Andrea Schumm and Robert Görke for their constant support, as well as Tanja Hartmann for her input, and Ulrik Brandes for expert advice. On this occasion, I would also like to thank Dorothea Wagner for inviting me to join her institute as a research assistant, allowing me to gain valuable insights into science in the subsequent years (again, with important advice by Robert Görke). Finally, I want to thank all those who are creating and collaborating on the informational and technological commons, including free software and openly accessible texts, without which much research, including this work, would be hard to imagine.

Bibliography

[AAA+02] N. Adiga, G. Almási, G. Almasi, Y. Aridor, R. Barik, D. Beece, R. Bellofatto, G. Bhanot, R. Bickford, M. Blumrich *et al.*, "An overview of the bluegene/l supercomputer," 2002.

[AB02] R. Albert and A. Barabási, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.

[Alb99] A.-L. B. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[BCHJ04] P. Bonacich, A. Cody Holdren, and M. Johnston, "Hyper-edges and multidimensional centrality," *Social networks*, vol. 26, no. 3, pp. 189–203, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378873304000024

[Bon72] P. Bonacich, "Factoring and Weighting Approaches to Status Scores and Clique Identification," *Journal of Mathematical Sociology*, vol. 2, pp. 113–120, 1972.

[Dam] C. Damgaard. "Gini Coefficient." From MathWorld–A Wolfram Web Resource, created by Eric W. Weisstein. [Online]. Available: http://mathworld.wolfram.com/GiniCoefficient.html

[dbl] Completesearch dblp. [Online]. Available: http://dblp.mpi-inf.mpg.de/dblp-mirror/

[DBL07] "DBLP - DataBase systems and Logic Programming," 2007, http://dblp.uni-trier.de/. [Online]. Available: http://dblp.uni-trier.de/

[dbl11] (2011) DBLP Faceted Search. [Online]. Available: http://dblp.l3s.de/

[dSP86] D. de Solla Price, *Little science, big science... and beyond*. Columbia University Press New York, 1986.

[eig11] (2011) Eigenfactor.org. [Online]. Available: http://www.eigenfactor.org

[Erl05] U. B. Erlebach, Ed., *Network Analysis: Methodological Foundations*, ser. Lecture Notes in Computer Science. Springer, February 2005, vol. 3418. [Online]. Available: http://springerlink.metapress.com/content/nv20c2jfpf28/

[Fra] M. Fratz. Eigenvektorzentralität auf Hypergraphen. [Online]. Available: www.inf.uni-konstanz.de/algo/lehre/ws08/projekt/ausarbeitungen/fratz.pdf

[Gir04] M. E. J. N. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 026113, pp. 1–16, 2004. [Online]. Available: http://link.aps.org/abstract/PRE/v69/e026113

[Gör10] R. Görke, "An Algorithmic Walk from Static to Dynamic Graph Clustering," Ph.D. dissertation, Fakultät für Informatik, February 2010. [Online]. Available: http://digbib.ubka.uni-karlsruhe.de/volltexte/1000018288

[Jac01] P. Jaccard, "Etude comparative de la distribution florale dans une portion des alpes et du jura," 1901.

[jun] JUNG 2.0 API Documentation. [Online]. Available: http://jung.sourceforge.net/doc/api/index.html

[jun11] (2011) JUNG Project Website. [Online]. Available: http://jung.sourceforge.net/

[jyt] Jython. [Online]. Available: http://www.jython.org/

[Lef08] V. B.-L. G. L. Lefebvre, "Fast unfolding of community hierarchies in large networks," March 2008, aRXIV: 2008arXiv0803.0476B. [Online]. Available: http://arxiv.org/abs/0803.0476

[Lis11] D. Lisowski, "Modularity-basiertes Clustern von dynamischen Graphen im Offline-Fall," Master's thesis, Karlsruhe Institute of Technology, 2011. [Online]. Available: http://i11www.iti.uni-karlsruhe.de/_media/teaching/theses/da-lisowski.pdf

[mat] Wolfram Mathematica. [Online]. Available: http://www.wolfram.com/mathematica/

[Mil04] Y. A. J. E. Milios, "Characterizing and Mining the Citation Graph of the Computer Science Literature," *Knowledge and Information Systems*, vol. 6, no. 6, pp. 664–678, 2004.

[New01] M. Newman, "Scientific collaboration networks. i. network construction and fundamental results," *Physical Review E*, vol. 64, no. 1, p. 016131, 2001.

[OFS+05] J. O'Madadhain, D. Fisher, P. Smyth, S. White, and Y. Boey, "Analysis and visualization of network data using JUNG," *Journal of Statistical Software*, vol. 10, pp. 1–35, 2005. [Online]. Available: http://jung.sourceforge.net/doc/JUNG_journal.pdf

[Pag98] S. B. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, 1998.

[Pri65] D. Price, "Networks of scientific papers." *Science (New York, NY)*, vol. 149, p. 510, 1965.

[Rot09] A. N. Rotta, "Multi-level Algorithms for Modularity Clustering," in *Proceedings of the 8th International Symposium on Experimental Algorithms (SEA'09)*, ser. Lecture Notes in Computer Science, J. Vahrenhold, Ed., vol. 5526. Springer, June 2009, pp. 257–268. [Online]. Available: http://www.springerlink.com/content/qugv7708h3806230/

[rpr] R project. [Online]. Available: http://www.r-project.org/

[Sco00] J. Scott, *Social Network Analysis - a Handbook*, 2nd ed. SAGE Publications, 2000.

[Sei83] S. B. Seidman, "Network Structure and Minimum Degree," *Social Networks*, vol. 5, pp. 269–287, 1983.

[Sol09] J. Solomon, "Programmers, professors, and parasites: Credit and co-authorship in computer science," *Science and engineering ethics*, vol. 15, no. 4, pp. 467–489, 2009.

[Sta10] C. Staudt, "Experimental Evaluation of Dynamic Graph Clustering Algorithms," February 2010, student Project, Studienarbeit. [Online]. Available: http://i11www.iti.uni-karlsruhe.de/projects/spp1307/dyneval

[Str98] D. J. W. H. Strogatz, "Collective Dynamics of "Small-World" Networks," *Nature*, vol. 393, pp. 440–442, 1998.

[Wag08] U. B. D. G. G. H. N. Wagner, "On Modularity Clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 2, pp. 172–188, February 2008. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TKDE.2007.190689

[Wag10] R. G. M. S. Wagner, "Modularity-Driven Clustering of Dynamic Graphs," ITI Wagner, Department of Informatics, Karlsruhe Institute of Technology (KIT), Tech. Rep., 2010, informatik, Uni Karlsruhe, TR 2010-5. [Online]. Available: http://digbib.ubka.uni-karlsruhe.de/volltexte/1000016558

[Wik11] Wikipedia. (2011) Gini coefficient. [Online]. Available: http://en.wikipedia.org/wiki/Gini_coefficient

[WJU07] S. Wuchty, B. Jones, and B. Uzzi, "The increasing dominance of teams in production of knowledge," *Science*, vol. 316, no. 5827, p. 1036, 2007.

[xom] XOM. [Online]. Available: http://xom.nu/

List of Tables

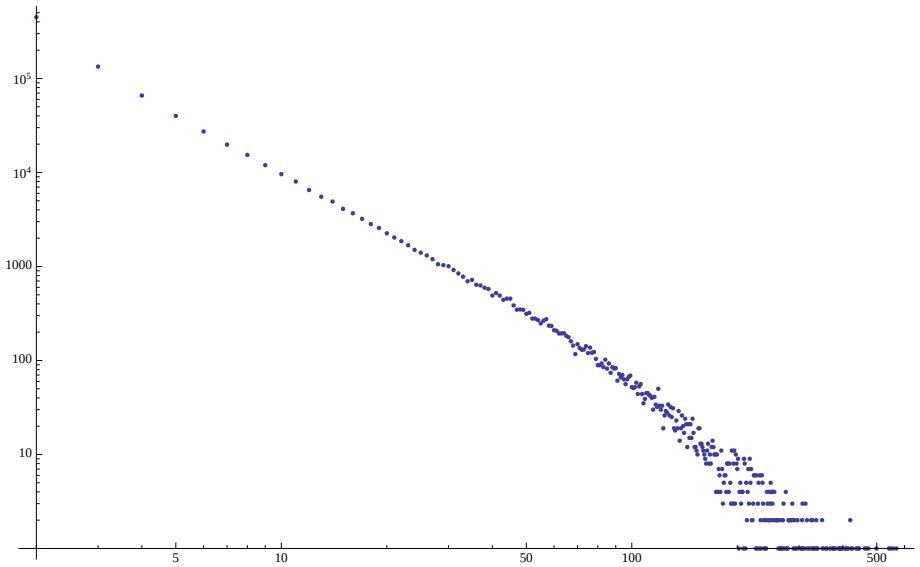
List of Figures

List of Algorithms

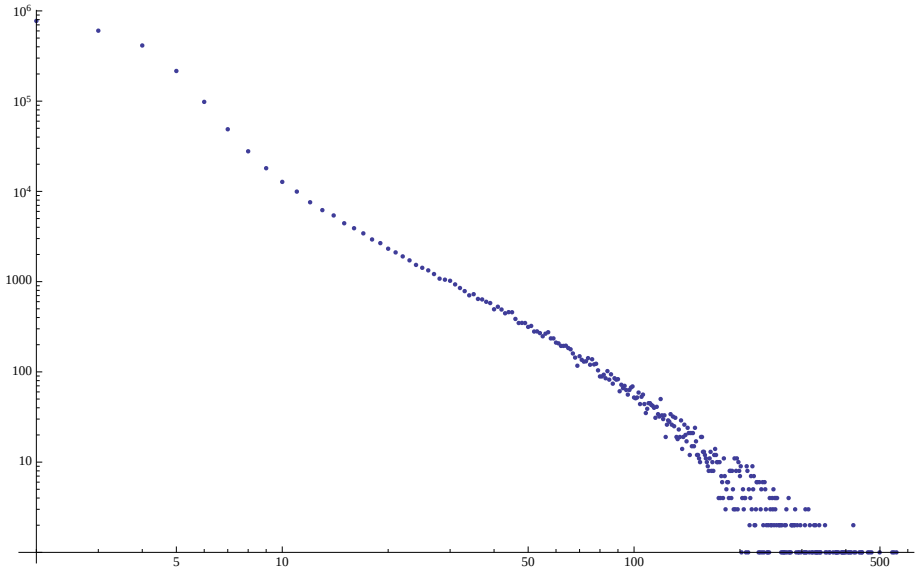
Appendix

A. Degree Distribution: Additional Plots

Here we include the degree histograms for $G_{\mathbf{PA}}$ referenced in Ch. 4, Sec. 4.2.



(a) Histogram of author node degrees in $G_{\mathbf{PA}}$

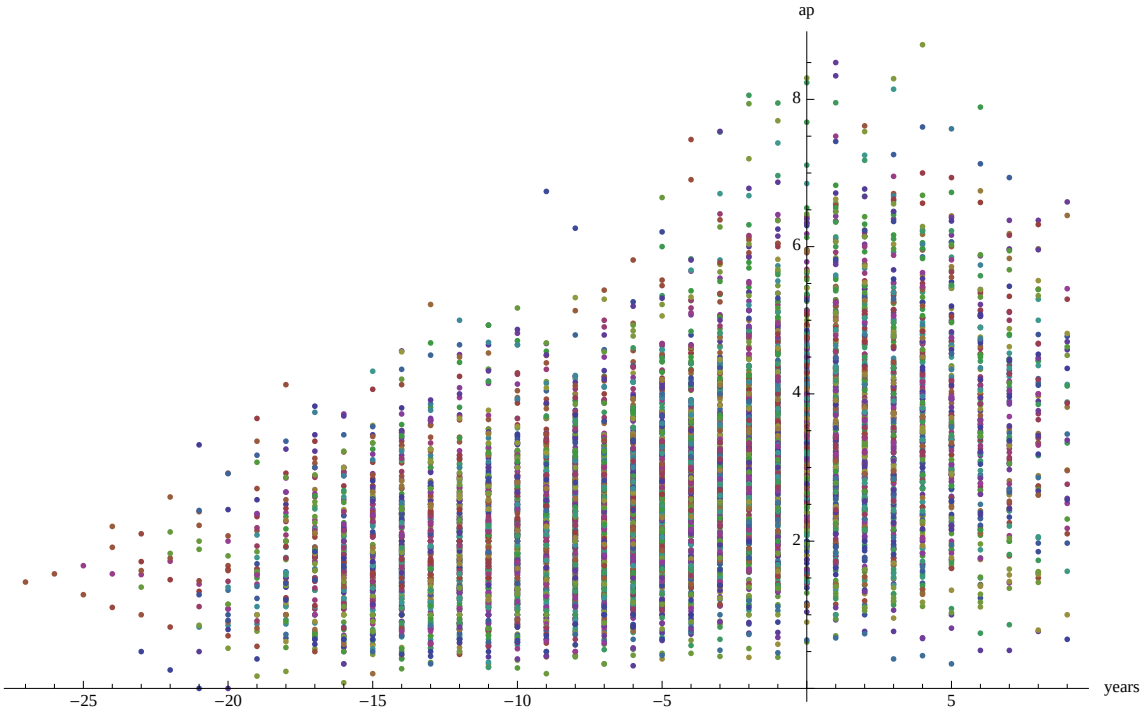


(b) Histogram of publication node degrees in $G_{\mathbf{PA}}$

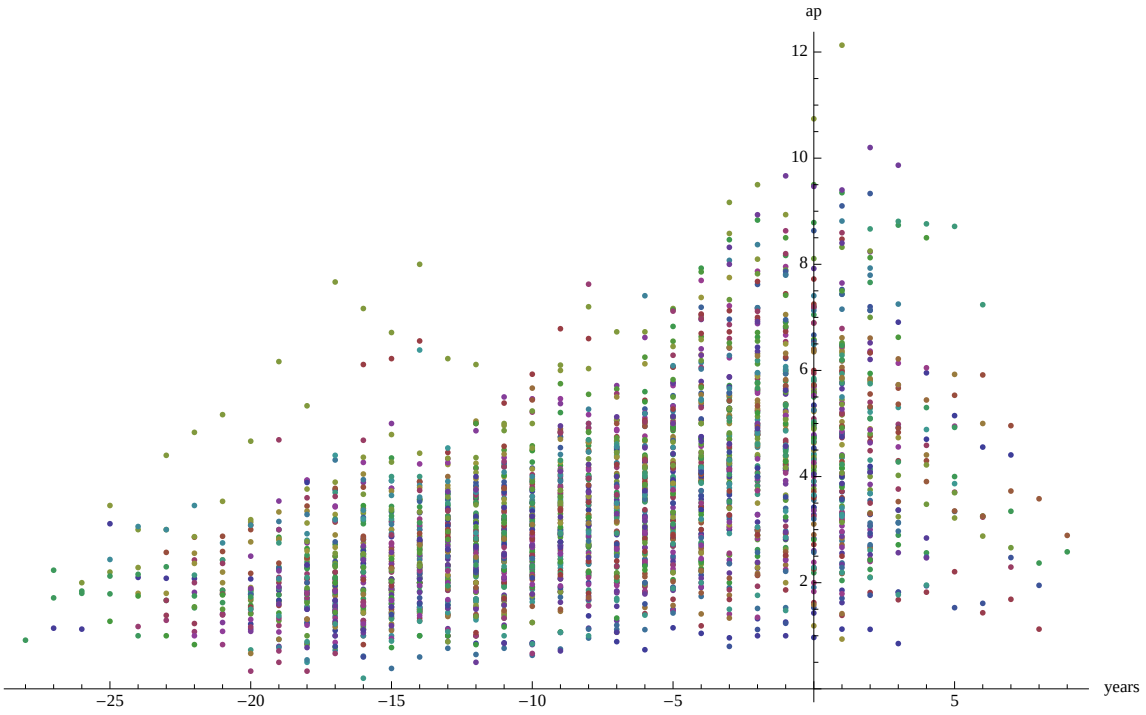
Figure A.1.: Degree histograms for $G_{\mathbf{PA}}$

B. Seminar Impact: Additional Plots

In the following, we include additional plots discussed in Ch. 6, Sec. 6.3.

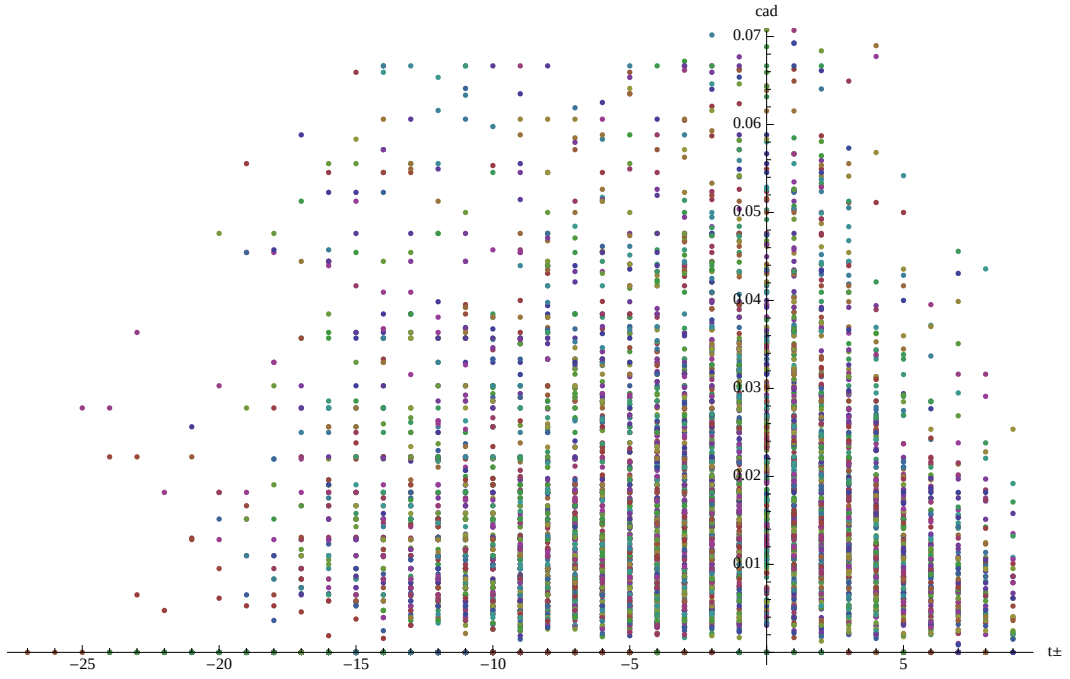


(a) *ap*: *At*

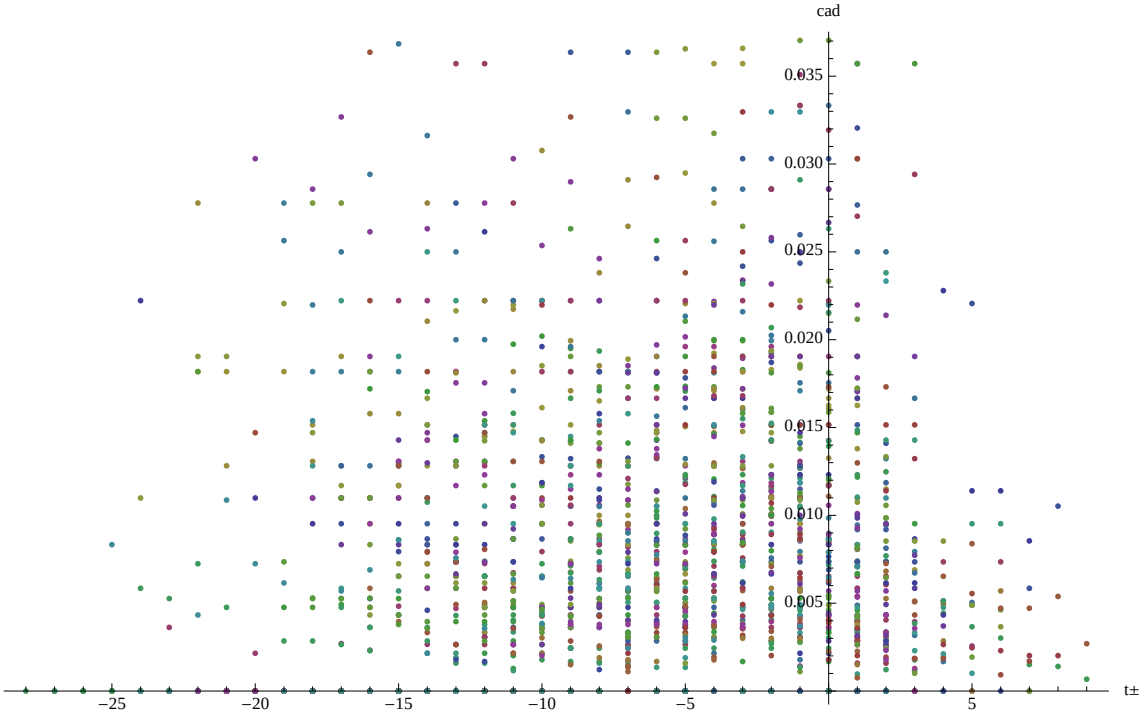


(b) *ap*: *Ab*

Figure B.2.: *ap* values for seminar attendees and absentees

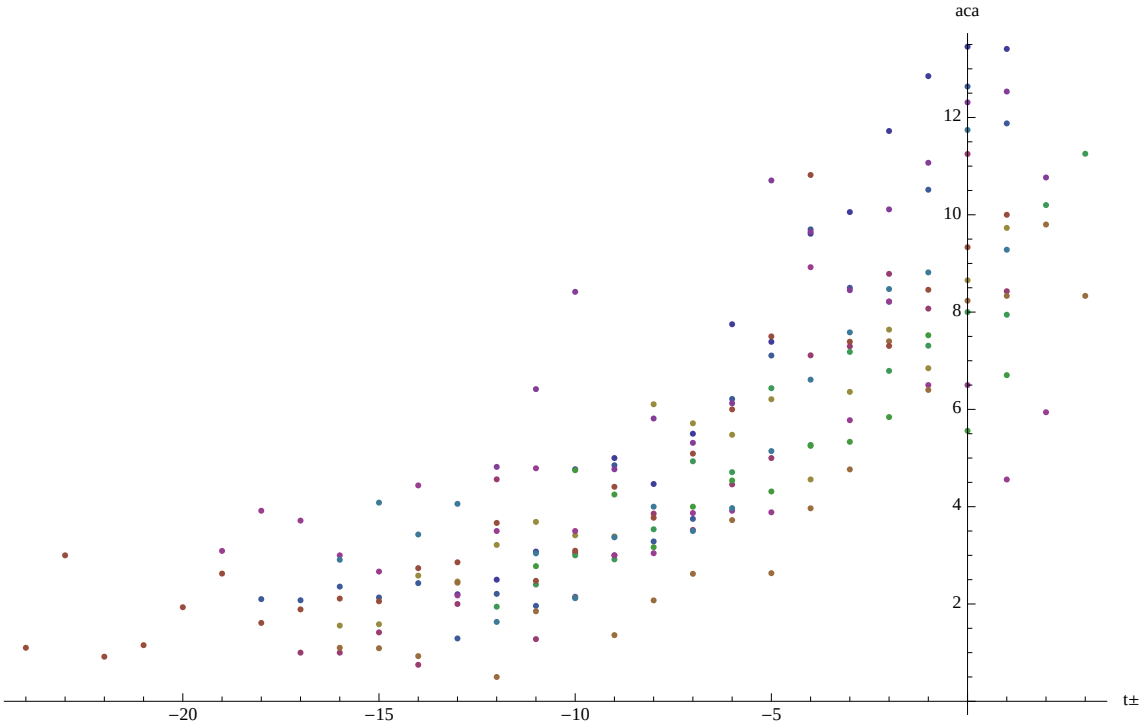


(a) *cad*: *At*

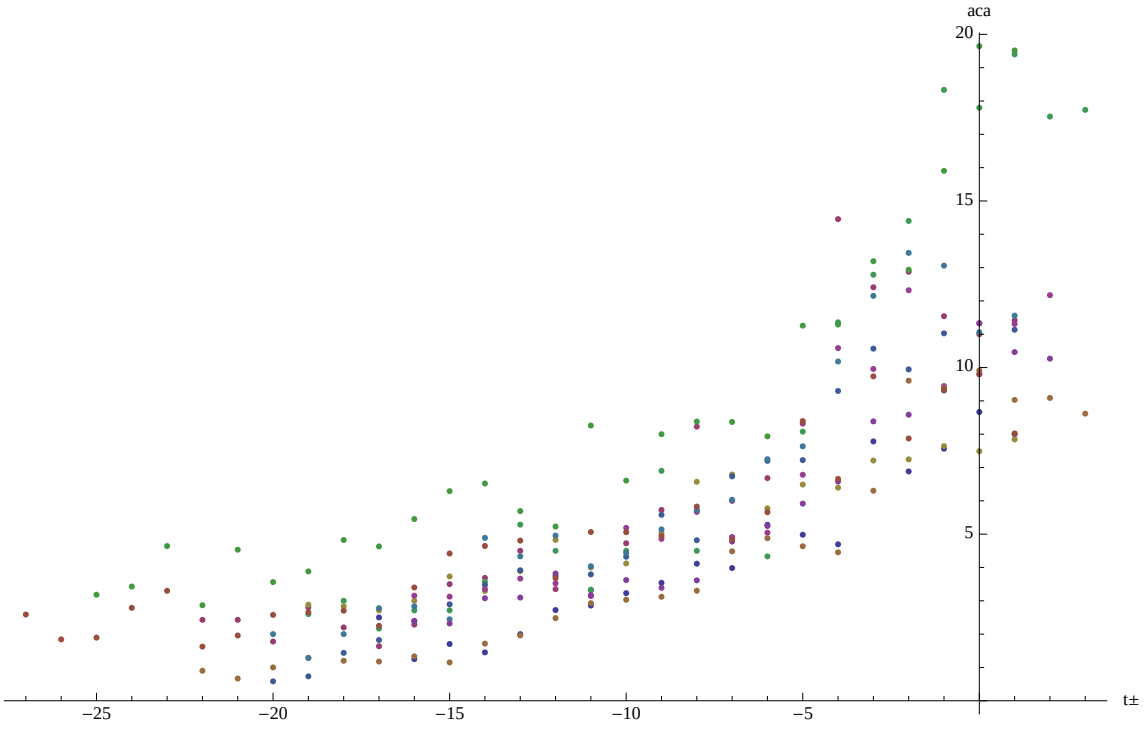


(b) *cad*: *Ab*

Figure B.3.: *cad* for seminar attendees and absentees

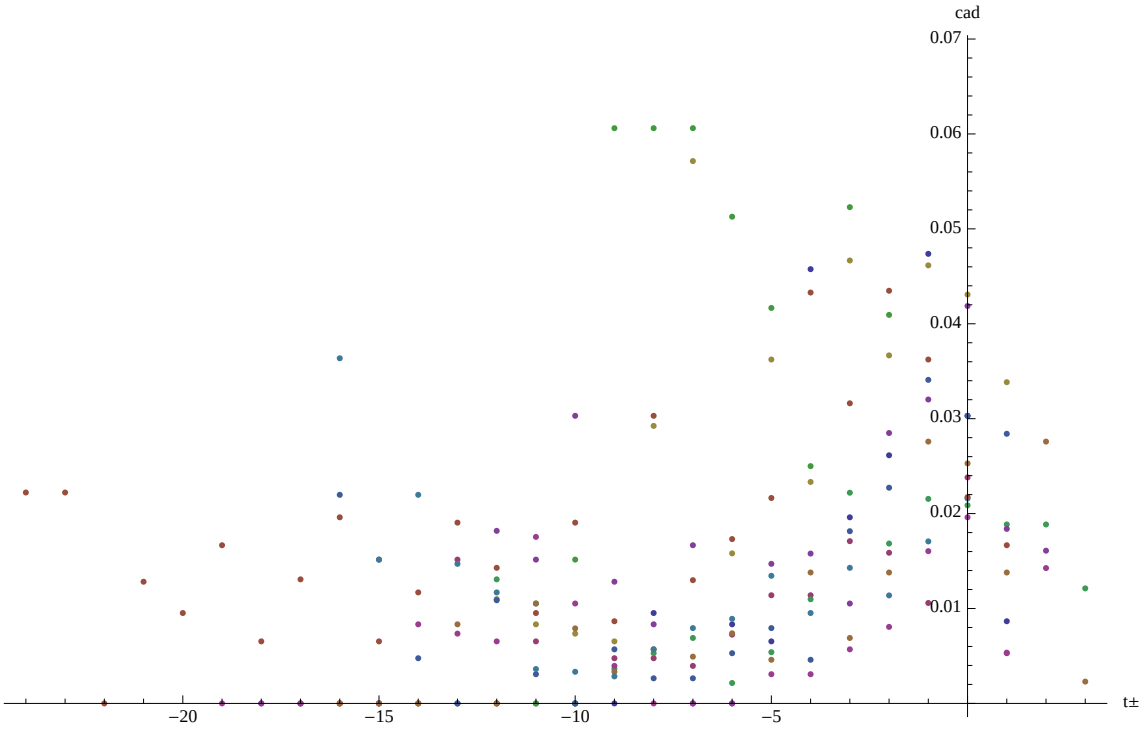


(a) *aca*: *At*

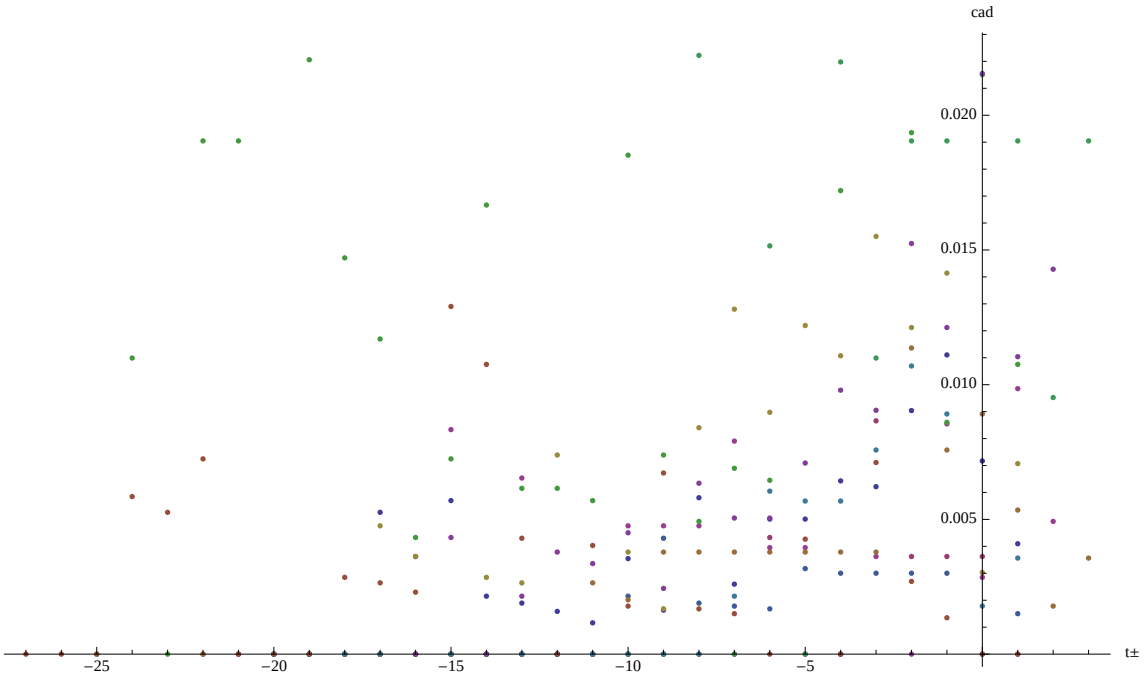


(b) *aca*: *Ab*

Figure B.4.: *aca* for attendees and absentees of area launchers



(a) *cad*: A_t



(b) *cad*: A_b

Figure B.5.: *cad* for attendees and absentees of area launchers

C. Centrality: A Ranking of Authors and Publications

The following tables list the top entries of the eigenvector centrality ranking described in Ch. 7.

centrality	*author*
0.0000976232	Diane Crawford
0.0000945441	Robert L. Glass
0.0000908697	Chin-Chen Chang
0.0000830777	Edwin R. Hancock
0.0000791401	Grzegorz Rozenberg
0.0000782901	Joseph Y. Halpern
0.0000775409	Sudhakar M. Reddy
0.0000769387	Philip S. Yu
0.0000750894	Moshe Y. Vardi
0.000074737	Ronald R. Yager
0.0000736573	Elisa Bertino
0.0000725425	Bill Hancock
0.0000724209	Thomas S. Huang
0.0000701698	David Eppstein
0.000070088	Kang G. Shin
0.0000685396	Noga Alon
0.0000676021	Micha Sharir
0.0000668928	Irith Pomeranz
0.0000665116	Christos H. Papadimitriou
0.0000663964	Witold Pedrycz
0.0000648648	Jie Wu
0.0000647047	Edmond Bianco
0.0000635781	Hermann A. Maurer
0.0000635442	Ming Li
0.0000628694	Azriel Rosenfeld
0.0000604239	Jun Wang
0.0000600533	Vladik Kreinovich
0.0000599181	Vishwani D. Agrawal
0.0000596144	Peter G. Neumann
0.000059516	Hamid R. Arabnia
0.0000591013	Hector Garcia-Molina
0.0000576606	Wei Zhang
0.0000573284	Ben Shneiderman
0.0000568801	Saharon Shelah
0.0000567995	Alberto L. Sangiovanni-Vincentelli
0.0000566617	Anil K. Jain
0.0000561621	Jiawei Han
0.0000559378	Christoph Meinel
0.0000552472	Gheorghe Paun
0.0000550194	Manfred Broy
0.0000548357	Sajal K. Das
0.0000543492	Kurt Mehlhorn
0.0000538785	Joseph O'Rourke
0.0000533677	H. Vincent Poor
0.0000532086	Li Zhang
0.0000531458	Viktor K. Prasanna
0.0000530604	David B. Lomet
0.0000530491	Sushil Jajodia
0.0000522742	Hans-Peter Seidel
0.0000522446	Oded Goldreich
0.0000520347	Gerhard J. Woeginger
0.0000518036	Horst Bunke
0.0000517968	Wen Gao
0.0000517156	Bertrand Meyer
0.0000513871	Oscar H. Ibarra
0.0000513354	Mahmut T. Kandemir
0.0000508769	Josef Kittler
0.000050765	Moti Yung
0.000050676	Richard T. Snodgrass
0.0000506571	Jack Dongarra
0.0000504149	Wei Wang
0.0000499426	Won Kim
0.00004994	Yan Zhang
0.0000498101	Mario Piattini
0.0000496485	Munindar P. Singh
0.0000495518	Michel Raynal
0.0000492644	Arto Salomaa
0.00004918	David Peleg
0.0000489602	Stephan Olariu
0.0000489379	Azzedine Boukerche
0.0000486519	Kaushik Roy
0.0000485932	Michael Stonebraker
0.0000484309	Paul G. Spirakis
0.0000483599	Gerhard Weikum
0.0000482455	Greg Goth
0.0000479213	Ajith Abraham
0.0000479009	Ramesh Jain
0.0000478941	Hartmut Ehrig
0.0000478117	Shusaku Tsumoto
0.0000471247	Ugo Montanari
0.0000470027	Ricardo A. Baeza-Yates
0.0000467766	Wil M. P. van der Aalst
0.000046751	John H. Reif
0.000046678	Gilbert Held
0.000046611	Xin Li
0.0000464657	Massoud Pedram
0.0000464584	Henri Prade
0.0000462825	Qing Li
0.0000462405	John Mylopoulos
0.000046045	Jeffrey D. Ullman
0.0000457791	Bart Preneel
0.0000453032	Friedrich L. Bauer
0.000045131	Makoto Takizawa
0.0000448056	Robert Endre Tarjan
0.000044294	Bruno Courcelle
0.0000442254	Marek Karpinski
0.000044108	Didier Dubois
0.0000437875	Donald F. Towsley
0.0000437502	Nicholas R. Jennings
0.0000433289	Niraj K. Jha

Table C.1.: The 100 highest ranking authors according to eigenvector centrality

centrality	publication
0.0000225345	The Biomolecular Interaction Network Database and related tools 2005 update.
0.000015423	The Grid2003 Production Grid: Principles and Practice.
0.0000137444	Humanoid Robots in Waseda University-Hadaly-2 and WABIAN.
0.0000132675	An overview of the BlueGene/L Supercomputer.
0.0000130997	GermOnline, a cross-species community knowledgebase on germ cell differentiation.
0.0000129758	Human protein reference database - 2006 update
0.0000129175	Construction of an open-access database that integrates cross-reference information from the transcriptome and proteome of immune cells.
0.0000125121	The Long-Term Ecological Research community metadata standardisation project: a progress report
0.0000118705	PATRIC: The VBI PathoSystems Resource Integration Center
0.0000113686	GeneFarm, structural and functional annotation of Arabidopsis gene and protein families by a network of experts
0.0000111262	Mascot animations
0.0000108194	INFN-CNAF activity in the TIER-1 and GRID for LHC experiments
0.000010306	A space-based end-to-end prototype geographic information network for lunar and planetary exploration and emergency response (2002 and 2003 field experiments)
9.92234522762e-6	Interoperation of world-wide production e-Science infrastructures
9.57617089018e-6	Run Control and Monitor System for the CMS Experiment
9.47586672441e-6	High Speed, High Capacity ATM Optical Switches for Future Telecommunication Transport Networks (Invited Paper)
9.42921362405e-6	CandidaDB: a genome database for Candida albicans pathogenomics
9.18051191515e-6	System Level Policies for Fault Tolerance Issues in the FERMI Project
8.97304451958e-6	CAD Methodology for the Design of UltraSPARC-I Microprocessor at Sun Microsystems Inc
8.91626278559e-6	Hierarchical power distribution and power management scheme for a single chip mobile processor
8.79600449357e-6	HMDB: a knowledgebase for the human metabolome
8.62069363856e-6	Autonomic Management of Large Clusters and Their Integration into the Grid
8.59972149965e-6	AgriBMPWater: systems approach to environmentally acceptable farming
8.59646105864e-6	HMDB: the Human Metabolome Database
8.59250767697e-6	Human protein reference database as a discovery resource for proteomics
8.53440715915e-6	Extending a Monoprocessor Real-Time System in a Multiprocessing Environment, DSP-Based
8.52921170024e-6	Validated 90nm CMOS Technology Platform with Low-k Copper Interconnects for Advanced System-on-Chip (SoC)
8.340649455e-6	UTGB/medaka: genomic resource database for medaka biology
8.21210671559e-6	Adaptive Image Content-Based Exposure Control for Scanning Applications in Radiography
8.10519863584e-6	CAD utilities to comprehend layout-dependent stress effects in 45 nm high- performance SOI custom macro design
8.09182474707e-6	OWLS: a ten-year history in optical wireless links for intra-satellite communications
8.01562468315e-6	IBM experiments in soft fails in computer electronics (1978-1994)
7.84963786292e-6	IUPHAR-DB: the IUPHAR database of G protein-coupled receptors and ion channels
7.82903274359e-6	Q-Chem 2.0: a high-performance ab initio electronic structure program package
7.81594941361e-6	Science and technology in the region: The output of regional science and technology, its strengths and its leading institutions
7.55955055012e-6	Evola: Ortholog database of all human genes in H-InvDB with manual curation of phylogenetic trees
7.54536143152e-6	The digital tipping point
7.41445832051e-6	A Distributed, Heterogeneous Control System for the ALICE TPC Electronics
7.342067702e-6	ProtozoaDB: dynamic visualization and exploration of protozoan genomes
7.33697758325e-6	An advanced multichip module (MCM) for high-performance UNIX servers
7.32764696318e-6	A 50-Gb/s IP router
7.29615612043e-6	New designs for MRI contrast agents
7.19428076404e-6	MamMiBase: a mitochondrial genome database for mammalian phylogenetic studies
7.18939827578e-6	DIRAC - Distributed Infrastructure with Remote Agent Control
7.02855066291e-6	Description of the HYPERMEDIA ACTS-361 Project: Continuous Audiovisual Market in Europe
6.99952015731e-6	Lipid/Polymer Nanoparticles as Tools to Improve the Therapeutic Activity of Existing and Emerging Anticancer Drug Combinations
6.96686298706e-6	Techno-Economic Evaluation of Narrowband and Broadband Access Network Alternatives and Evolution Scenario Assessment
6.8688914763e-6	ABINIT: First-principles approach to material and nanosystem properties.
6.82373890396e-6	The feasibility of on-chip interconnection using antennas.
6.76840787553e-6	CHEOPS: Really Using a Satellite.

6.76765424852e-6	Atmospheric Water Vapor Effects on Spaceborne Interferometric SAR Imaging: Comparison with Ground-based Measurements and Meteorological Model Simulations at Different Scales
6.76275567299e-6	UltraSPARC-I Emulation
6.71866849315e-6	AutoCSA, an algorithm for high throughput DNA sequence variant detection in cancer genomes
6.68601132289e-6	Ultralow-power SRAM technology
6.62417851717e-6	Stanley: The robot that won the DARPA Grand Challenge
6.60763411429e-6	Pathbase: a database of mutant mouse pathology
6.60668408752e-6	A 800 MHz System-on-Chip for Wireless Infrastructure Applications
6.59783696321e-6	Human Protein Reference Database - 2009 update
6.59674839087e-6	Implementation of a Distributed Architecture for Managing Collection and Dissemination of Data for Fetal Alcohol Spectrum Disorders Research
6.58489850338e-6	STING Millennium: a web-based suite of programs for comprehensive and simultaneous analysis of protein structure and sequence
6.55519602949e-6	Functional verification of the POWER4 microprocessor and POWER4 multiprocessor system
6.51945975461e-6	Foundation of rf CMOS and SiGe BiCMOS technologies
6.50033198346e-6	Mixed signal integrated circuits based on GaAs HEMTs
6.49768830778e-6	A Multiphysics and Multiscale Software Environment for Modeling Astrophysical Systems
6.48573168365e-6	Multimedia Manager: Query by Image Content and Its Applications
6.47453800264e-6	Wireless Sensor Networks for Home Health Care
6.47449334326e-6	The Rat Genome Database (RGD): developments towards a phenome database
6.44924683857e-6	The Dark Energy Survey Data Management System
6.41169109278e-6	A Multilevel-Cell 32MB Flash Memory
6.32753607713e-6	A Large-Scale, Flip-Flop RAM Imitating a Logic LSI for Fast Development of Process Technology
6.31371958202e-6	Automatic Exposure Control in Digital Mammography: Contrast-to-Noise Ratio Versus Average Glandular Dose
6.31371958202e-6	A standard format for Les Houches Event Files
6.29441900796e-6	Ensembl 2009
6.29343562534e-6	EU Project Resolution - Reconfigurable Systems for Mobile Local Communication and Positioning
6.29194813519e-6	The National Transport Code Collaboration Module Library
6.28759384582e-6	A CMOS SoC for 56/18/16 CD/DVD-dual/RAM applications
6.23207665639e-6	An integrated multi-model approach for air quality assessment: Development and evaluation of the OSCAR Air Quality Assessment System
6.19615376912e-6	Vbfnlo: A parton level Monte Carlo for processes with electroweak bosons
6.19615376912e-6	Laboratory Information Management Software for genotyping workflows: applications in high throughput crop genotyping
6.19615376912e-6	Standby power reduction and SRAM cell optimization for 65nm technology
6.1310430428e-6	ArrayExpress update - from an archive of functional genomics experiments to the atlas of gene expression
6.12477452557e-6	GRIDA3 - a shared resources manager for environmental data analysis and applications
6.09064163836e-6	IntAct - open source resource for molecular interaction data
6.08885163829e-6	LDLR Database (second edition): new additions to the database and the software, and results of the first molecular analysis
6.0508293615e-6	CTdatabase: a knowledge-base of high-throughput and curated data on cancer-testis antigens
5.9871478795e-6	Subretinal Microelectrode Arrays Allow Blind Retinitis Pigmentosa Patients to Recognize Letters and Combine them to Words
5.97081929438e-6	Assessment of algorithms for high throughput detection of genomic copy number variation in oligonucleotide microarray data
5.9610221433e-6	SUSY Les Houches Accord 2
5.9560458126e-6	Adapting SAM for CDF
5.921833539e-6	Design and implementation of a point-of-care computerized system for drug therapy in Stockholm metropolitan health region - Bridging the gap between knowledge and practice
5.91530210495e-6	On the detection of multiple-binding modes of ligands to proteins, from biological, structural, and modeling data
5.88917636875e-6	SNP mining porcine ESTs with MAVIANT, a novel tool for SNP evaluation and annotation
5.86999938946e-6	New approaches to genomic analysis using single molecules
5.8695820666e-6	SNORTEX (Snow Reflectance Transition Experiment): Remote Sensing Measurement of the Dynamic Properties of the Boreal Snow-forest in Support to Climate and Weather Forecast: Report of IOP-2008
5.86281778285e-6	QPACE: Quantum Chromodynamics Parallel Computing on the Cell Broadband Engine
5.8565191985e-6	Dynamical twisted mass fermions with light quarks: simulation and analysis details
5.75854768774e-6	A comparison of three computational modelling methods for the prediction of virological response to combination HIV therapy
5.75854768774e-6	Battery-powered, Wireless MEMS Sensors for High-Sensitivity Chemical and Biological Sensing
5.72589051749e-6	The MERIS Water Products: Performance, Current Issues and Potential Future Improvements
5.69406988559e-6	Building a distributed robot garden

Table C.2.: The 100 highest ranking publications according to eigenvector centrality