Master Thesis

A Tie Strength Model For Reconstructing Collaboration Networks on GitHub

A study of the Ruby on Rails project network

August 21, 2014

Author:
Arno Gregorian
a.gregorian@students.uu.nl

Supervisors:
prof. dr. ir. R.W. Helms
dr. A. J. Feelders

Utrecht University, Faculty of Science,
Department of Information and Computing Sciences,
Business Informatics.

Universiteit Utrecht

Abstract

Social network analysis based on follow relations, on GitHub and other social networking websites, ignores the strength and time of relations. This paper introduces an approach to reconstruct weighted social networks covering extensive periods of time using communication data.

A tie strength model is suggested to determine edge weights that reflect the intensity of collaborations. The approach and the model are tested by investigating the 10 year history of Ruby on Rails, an open source software project.

The results show that for a group of 283 software developers, social network analysis based on communication data is more successful at finding collaborators compared to analysis based on follow relations. The model for tie strength results in a significantly better ranking of the intensity of collaborations compared to rankings based on sums of activity.

The approach in this paper allows GitHub activities to be combined into meaningful project networks that can be used to answer more fundamental questions about open source software development, e.g. determining if some or any aspect of OSS communities and development are constant and predictable.

Keywords - Social network analysis, open source, software, tie strength, weighted networks, collaborations, GitHub, Ruby on Rails

Acknowledgements

I would like to thank my supervisors for having the patience and giving me the freedom to explore a broad topic. Finding a specific focus turned out to be harder than expected, but the productive meetings with Remko Helms helped me rethink and contextualize the questions I was struggling with. Remko Helms also communicated some of my ideas with collogues, which led to interesting responses for which I am grateful. Thanks also to Ad Feelders for his interest and willingness even with short notice to review my work. I would also like to thank my fellow colloquium students for suggesting interesting papers and giving me challenging questions. Last but no least I would like to thank my brother Pablo Gregorian for reviewing some of my ideas about the model for tie strength and my father dr. Thadeus Gregorian for his discussions and advice.

Contents

Chapter 1
Introduction

Online social networks encourage communication between members. In online social networks, relations (edges) are often introduced after people (nodes) intentionally decide to connect. Examples are: confirming a friendship request on Facebook or following another open source software (OSS) developer on GitHub: an online platform for sharing and developing software. Wejnert [1] correctly observes that manipulating edges in online social networks is easy. People can remove relations at will and can easily increase them by approaching an ever larger group of people. Even if the edges would turn out to be accurate, they would still be based on a single identical event and can therefore hardly be distinguished by strength. Network analysis and reconstruction based on communication in time is therefore needed and important. Social network analysis (SNA) has been a part of sociology for almost a century [2]. With the introduction of social networking websites SNA has attracted researchers from a wide range of disciplines including Business Informatics (BI) researchers interested in OSS communities. Here too network reconstruction methods struggle with the issues of manipulation, accuracy and determination of the strength of relations.

Before discussing how this thesis proposes to deal with these problems, it is important to argue why SNA research methods deserve attention as a part of BI and OSS research, apart from the already mentioned problems. There are only a few fundamental and successfully replicated theories about OSS communities and development. One is the repeatedly found skewed distribution of contributions and connectedness of individuals in OSS projects [3]. The majority of OSS projects seems to consist of a relatively well-connected small group of people doing most of the work. At the same time a large disconnected group of people comes and goes while suggesting small improvements. These distributions are interesting but are not evidence of OSS projects having predictable life cycles with causal relationships determining their survival. This observation about survival might seem out of place, but it is not when looking at recent related research. Researchers have suggested to view OSS projects and communities as ecosystems [4]. The term ecosystem stems from ecology and has been used by business researchers [5] and now is being applied in BI research. In ecology the function of the term ecosystem is *'to emphasize obligatory relationships, interdependence and causal relationships'* [6]. The assumption that a hypothetical OSS community is as interdependent and reliant on causal relationships as an ecosystem is speculation at best. Using the term ecosystem should therefore be done reluctantly until clear evidence of causal relations and interdependence is found. What such evidence should be is debatable. This is where the use of reconstructed social networks becomes relevant and where this thesis ties in with current research.

Qualitative research methods of BI, such as expert interviews and case studies, can give a detailed insight into one or a small set of OSS projects. Describing a large set or system of OSS projects using these methods is however hard, firstly as it is time consuming and secondly transcripts and case study observation are hard to combine into testable and replicable theories. Collaboration networks of OSS projects can in comparison, when reconstructed in the same way for a wide range of projects, be compared directly. The accumulation of results about such networks could therefore be evidence for replicable and general theories, where qualitative measures might fail. But before this is possible, a well-defined and tested reconstruction method is needed for weighted OSS collaboration networks which this thesis aims to deliver.

Thesis Overview, Goal and Structure

The constructive part of the thesis starts by creating a novel social network reconstruction method based in part on existing research, which is tested by comparing its accuracy against other reconstruction methods. The reconstructions result in four networks, two networks based on follow relations and two networks based on communication data. The networks all portray the open Ruby on Rails project (Rails: a web development framework) with its members and their interpersonal relations, in different ways. The accuracy of the networks is tested using a reference network based on survey results completed by Rails members.

The goal of this thesis is to deliver a social network reconstruction method that enables researchers to test hypothesis about OSS in an objective, replicable an reliable manner. Let's take the example of a researcher trying to answer the question: '*Do contributors in small OSS projects have the same average number of collaboration partners compared to a large OSS project?*'. Currently a researcher will have to interview or survey a statistically large enough set of contributors, both in small and large OSS projects, before being able to make a comparison. With the approach presented in this thesis a researcher can select and retrieve a substantial set of projects (in the thousands) from GitHub instantly, create two subsets of large and small projects, reconstruct their social networks and compute the average degree of nodes in the networks. This without having to wait for survey responses while having access to the full population of projects. Additionally other researcher can easily recreate and test the research, as the information is openly available to all.

The thesis starts with an exploration of related literature in chapter 3, resulting in a definition for network activity and a model for interpersonal tie strength in social networks using network activity in chapter 4. In chapter 5 the steps of CRISP-DM are used to structure the research approach including the steps for business and data understanding, data preparation, modeling and evaluation. Chapter 6 contains the main results of the network accuracy comparison. The chapter also includes a general exploration of the data with two partial replications of related literature. Chapter 7 concludes the thesis by answering the main research questions, lastly the results and conclusions are discussed together with possible future research directions.

Chapter 2
Problem Statement

The reconstruction of social networks and its analysis is a welcome addition to the set of research methods available to BI researchers. There is however no standard approach for reconstructing OSS networks or open online social networks in general. The characteristics of online social networks found by researchers are therefore hard to compare and generalizable results and theories are hard to come by. Another problem is that reconstructed networks often contain edges that cannot be distinguished by strength or importance and the networks ignore the influence of time on network properties. Knowing the relative strength of relations in social networks is important as it is essential for discovering clusters or communities in networks [7]. Similar problem statements about the strength of relationships have been made by Toivonen, et al. (2007), Leskovec et al. (2007), Song, et al. (2005) and Barrat, et al. (2004). A reconstruction method is therefore needed that can be applied in a wide range of situations and accounts for the strength of relations and the influence of time on social networks.

Research questions:

Main Research Question: Can communication data be used to reconstruct weighted social networks over time?

The main research question is split into two separate question, the first question looks at existing research to create a novel social network reconstruction method using communication data. The second question uses the novel and other methods to reconstruct and compare social networks of the Rails OSS project. The results of the comparison will show which method most closely resembles a reference network based on survey results completed by Rails users and contributors.

Question 1: What is the relation between activities connecting people in a social network and the strength of their relationships?

The thesis starts with the assumption that modelled communication meta-data is sufficient to create accurate social networks portraying the relative strength between people over time. First a definition for appropriate communication meta-data for reconstruction is introduced, defined as network activity data. Secondly the network activity data is modeled according to results from related literature to calculate the relative strength of personal relations in OSS project networks.

> ***Sub question 1.1: What is appropriate continuous data for social network reconstruction?***
>
> ***Sub question 1.2: How can continuous network activity data be used to model the strength of personal relations according to existing research?***

Finally the accuracy of the novel reconstruction method for OSS project networks is compared to other frequently used reconstruction methods. Multiple ways of calculating accuracy are selected and used to compare the reconstruction methods in a fair way to a reference network based on survey results.

Question 2: Are social networks reconstructed using modeled network activity data more accurate than networks based on follow relations and in what way?

Scope definition

With 5 million users it is far outside the scope of this thesis to investigate the full population of GitHub. A single project called Rails is chosen as scope or population for this thesis, because of its relatively long and open recorded history on GitHub. The scope of the project only includes communication data created by and aimed at Rails users and contributors.

Chapter 3
Related Literature

This chapter contains relevant literature concerning SNA, OSS and tie strength research. As this thesis aims to introduce a novel social network reconstruction method for OSS, firstly relevant SNA will be discussed followed by OSS literature that combines OSS and SNA. Properties and models about social networks are often represented in graphs using terms and concepts from graph theory. The basic terminology for undirected weighted graphs is used in this thesis: G(V,E,W) in which V is a set of nodes (synonyms include points as used by Freeman [8], vertices and actors) and E a set of edges (synonyms include lines and ties) that connect nodes and W is a set of edge weights that are attached to every edge in E in graph G. When two nodes are directly connected by one edge they are adjacent. A undirected graph is defined by its symmetric adjacency matrix denoted by A, where elements of A $a_{i,j}$ (=$a_{j,i}$) are real valued functions of nodes *i* and *j*, where (i,j = 1, …, *n*).

Social Network Analysis

There are several general review articles on social network analysis. Freeman [2], for example, provides an excellent survey of the history of SNA. In this section only literature on the reconstruction of social networks and tie strength models will be considered. The narrative that emerges from the research is that SNA is a strong tool for making sociological and organization hypothesis measurable and testable. OSS researchers assert the unique nature of OSS communities and attempt to prove hypothesis about them, again when SNA is applied progress is made.

Social network analysis assumes that social life and its properties and effects are culminations of relations between people. More formally social networks are sets of nodes (people) connected by edges that represent different types of relations. By assuming such a network structure, research approaches using SNA attempt to explain social phenomena by means of network properties. SNA does not focus only on one attribute and ignores others, rather it considers each attribute in turn to be part of another network. A network of education experts might be connected by a network of collaborating research groups, this network can be enriched with other attributes, for example a geographic network with edges between nodes when they work in the same geographical location. The attribute of location is thus not ignored, but considered as a new network.

As network structures can be claimed to exist in many different social situations, a wide array of different applications from different scientific fields is found. For example networks of researchers based on co-authorship of publications, networks of related webpages or as in the case of this thesis, the relations of OSS project contributors. The reconstruction of social networks creates some practical problems. Firstly in the early stages of reconstruction a boundary has to be defined (boundary problem), determining which nodes should be included in the network analysis. Secondly the problem of assessing the properties of large networks, where only network data of a small sample of nodes can be gathered (sampling problem). It might even be unknown what the real size of the full population is, for example estimating the spread of the HIV virus by studying networks of a small sets of patients [9]. The boundary and sampling problem can be seen as related, as the boundary of a social group results in the criteria for sampling. Two approaches to deal with these initial problems have emerged, the ego-centric and saturated network data approach [10]. The ego-centric approach surveys nodes of a social networks with questions about their attributes, relations and attributes of relations. The resulting network from such a set of surveys contains important properties about individuals, for example the number of people they feel connected to (degree). It is however hard to also survey these connected contacts and subsequently their contacts, by having a large enough network of individuals, statistical inferences about the population become possible by using standard statistical procedures. This ego-centric approach is ideal for studying sociological phenomena, as the networks contain relatively accurate and in depth information from surveys. The lack of data about a larger population creates the need for saturated social networks. Saturated networks are based on data taken from a complete population, the network is therefore strictly speaking no longer a sample, which makes statistical analysis easy. Creating such saturated networks is however costly and time consuming [1], they also tend to be shallow, as they often only focus on one type of network attribute. The saturated networks are used because they are a reliable source for determining networks structures such as average degree, density and diameter.

Computer-based social networks in theory seem the ideal source for both ego-centric and structural analysis, the different social networking websites such as Facebook and Twitter contain rich and saturated network data which is accessible through automated means. People using such network sites however also have the means to manipulate and distort their social network and personal attributes. For example, increasing personal connections (degree) is relatively easy, by continuously sending friendship requests to more people with whom the ego might not even share a real-world connection. The accuracy of such computer-based networks can therefore be challenged [1].

Tie Strength

SNA assumes that the edges presented for analysis reflect their supposed attributes correctly. One of the hardest attributes to reflect is the strength of interpersonal ties. The strength of these ties are important for social network reconstruction methods as they are often necessary for algorithms to discover communities in social networks [7]. The research field of *tie strength* [11] identifies concepts that make up the strength of ties between people. As defined by Granovetter *"The strength of a tie is a (probably linear) combination of the amount of time, the emotional intensity, and intimacy (mutual confiding), and the reciprocal services which characterize the tie"*. Tie strength brings together multiple concepts from sociology such as *"intensity, intimacy, duration, reciprocal services, structural, emotional support and social distance"* [12]. All these topics describe in part the strength of ties. Another thesis on the matter of tie strength by Granovetter states that weak links are probably to be found between communities whereas strong links are probably found within communities. The appeal of such hypotheses is that they can be measured and tested. A way of testing this thesis is for example by calculating the so called overlap of edges in a social network:

$$Overlap_{i,j} = \frac{n_{i,j}}{(k_i - 1) + (k_j - 1) - n_{i,j}}$$

Eq. (1) as appeared in Toivonen et al. (2007) [13]

Where $n_{i,j}$ is the number of common adjacent nodes of nodes i and j and k_i and k_j are the degree of the nodes i and j. If the Granovetter thesis holds, nodes connected with edges of relative high weights should also have a relatively high overlap. For example, when the overlap of an edge is plotted against the weight of the edge and this is done for all edges, it is expected that in a real world network the overlap of edges is positively linearly correlated with edge weight. The example of overlap illustrates how a general hypothesis about a broad concept such as tie strength can be made precise and tested using SNA. An example of research confirming this thesis can be found in Onnela et al. [14].

A recurring research approach when researching the factors of tie strength is regressing from the overall tie strength between people, as determined by the people themselves, to possible individual (usually measurable) factors. Using such egocentric data has been described by terms such as ‘key informant sampling’, ‘purposive sampling’ and ‘judgment sampling’ [15]. An example is the work by Marsden and Campbell who asked subjects from three different cities to describe three of their closest relationships considering indicators such as closeness, duration, frequency, breadth of discussion and mutual confiding. The results suggest that closeness, a measure of the *‘emotional intensity of a relationship’,* is the strongest predictor of tie strength. Marsden and Campbell (1984) hypothesize, starting with Granovetter’s assumption that tie strength is a linear combination of different indicators, that Eq. (2) models tie strength in social networks:

$$\mathrm{Y}_1 = \sum_{i=2}^{7} \beta_{1i} \mathrm{Y}_i + \mathrm{u}_1$$

Eq. (2) as appeared in Marsden & Campbell (1984) [16] page 491

The function sums over the seven regression parameters Y_i that are predictors of tie strength: neighbor, co-worker, kinship, shared organizations, difference in occupational prestige and absolute difference in years of education. The function result is Y_1, the actual tie strength.

Next to the predictors of tie strength, five indicators that actually make up the strength of a tie are also modelled:

$$\mathrm{X}_1 = \lambda_{11} \mathrm{Y}_1 + \varepsilon_1$$
$$\ldots$$
$$\mathrm{X}_5 = \lambda_{51} \mathrm{Y}_1 + \varepsilon_5$$

Eq. (3) as appeared in Marsden & Campbell (1984) [16] page 490

The five indicators are closeness, duration, frequency, breath of discussion and mutual confiding. The regression coefficient $\lambda_{11} \ldots \lambda_{51}$ are to be found by regression from a dataset, $\varepsilon_1 \ldots \varepsilon_5$ are the proportions of the indicators. The overarching assumption is that both the indicators and the predictors should be used to determine the tie strength (Y_1). The results showed that in order to make the models fit the results the indicators and predictors cannot be assumed to be independent, which undermines the idea of separation in indicators and predictors. The insights about what is most powerful in predicting a strong connection is valuable, the lack of a clear model which predicts the strength of a relation, using simple meta-data about communications between people, that can be tested and replicated leaves room for improvement. More recent research about online social networks finds the time since the last activity as the strongest predictor of tie strength, in second place the time since the first communication. These findings support the assumption that the age and recency of edges should be considered when calculating centrality in social networks. This is however misleading, these factors are powerful predictors because they indirectly hold the information if there was any communication at all. Simple heuristics such as defining a tie as ‘strong’ if two actors have communicated a *n* number of times gives at best an accuracy of 61%, while more complex models are able to predict tie strength as determined by the actors by nearly 90% [12]. Gupte & Eliassi-Rad [17] determine tie strength by using co-attendance of events and develop a set of axioms for tie strength in general. A baseline axiom is proposed where the strength of ties between pairs is assumed to be 0 untill an event brings the pair together, another axiom argues that the strength of a tie cannot increase indefinitely. The added strength due to a single event is proposed to have diminishing returns for each additional event.

General Network Types

As mentioned before SNA has been around for the better part of a century, the accumulation of SNA research results has led to some general network types with their respective properties. As some of these recurring network types are also expected to be present in OSS project networks they are discussed here. SNA research focused on large real-world networks has repeatedly found a type of network called a small-world network, also known as a network with six degrees of separation. This property suggests that real-world networks have small diameters, this was first suggested by Milgram [18] who introduced the problem *'Starting with any two people in the world, what is the probability that they will know each other?'* [18]. The degree of nodes in real-world networks have been successfully described by heavily tailed power laws. Such distributions of degree have for example been found on the internet [19] and phone call graphs [20], and are referred to as scale-free networks. Scale-free networks seem to have a statically large amount of 'hubs', nodes with an above average degree [21].

Most research of large real-world networks uses static networks or a small set of snapshots showing the network during different time windows. Recent research on time evolution of graphs suggest that the out-degree of networks grows over time and counter to standard believe the node diameters actually shrink as the network grows [22]. This type of dynamic SNA uses a sliding window filter to document the changes of a network during a specific timespan by creating a set of chronological network reconstructions that represent the network during a time window. Using such sets of networks to answer statistical questions is not obvious, but its review is outside the scope of this thesis. A paper by Snijders is frequently referenced and describes the statistical evaluation of such dynamic networks in great detail [23]. What is important to discuss is the implementation of Snijders statistical approach to dynamic networks, such as the work on homophily by Kossinets and Watts [24]. The subject of homophily is not relevant, but Kossinets and Watts, as one of the few, accurately describe network reconstruction and a more serious approach to calculating the weight of edges in social networks. Take for example the function for calculating the weight of an edge *i,j* during the time window (t, τ) based on message exchange:

$$W_{i,j}\,(\mathrm{t},\tau) = \frac{1}{\tau}\sqrt{M_{ij}^{\tau}(t)M_{ji}^{\tau}(t)}$$

Eq. (4) [24]

Where M_{ij}^{τ} is the message count from node i to j and $M_{ji}^{\tau}(t)$ vice versa. This weight measures gives the geometric average of the number of messages exchanged between nodes. It also highlights the measures a researcher has to determine when dealing with a dynamic network, i.e. the appropriate values for the timespan t and time window τ. An important measure is the step size, a step size equal to the window size means that time windows will ignore everything preceding it, while a smaller step size will include parts of the previously examined time window.

Social Network Parameters

Researchers use different data to reconstruct social networks and broad set of data creates difficulties in comparing research results, as was mentioned in the problem statement. The calculation of network properties is however precise and includes a set of important concepts of which centrality is probably the most well-known example. It is outside the scope of this thesis to review these properties in detail and such reviews have been published repeatedly over the years [25] [8] [2]. Only a small set of properties are discussed, as they will be mentioned repeatedly in the chapters to come:

Centrality

The concept of centrality has been applied in many different research fields in analyzing networks. Explaining communication performances in small groups of people [26], finding essential scientific papers [27] or researching medical innovations [28]. In each application the concept of centrality might mean something else completely, from power, independence to wealth or influence. Different authors have attempted to define centrality more precisely by categorizing the mathematical properties of different centrality measures. Freeman [8] concludes that centrality measures can be categorized by three conceptual foundations: degree, betweenness and closeness. These three measures will be discussed briefly.

The degree of a node is the simplest measure of centrality and counts the number of adjacent nodes of the node. An additional operation might be to divide the result by the number of nodes in the network minus one node, the node of interest. This would be the normalized measure of degree. Calculating the degree of a node in a weighted network only changes the function for finding degree slightly. The function for calculating the weighted degree of a node in a weighted network is:

$$Weighted\ Degree_i = \sum_{\mathrm{j \in V}} W_{i,j}$$

Eq. (5)

Where the weighted degree of node *i* is the sum of edge weights: $W_{i,j}$ which have a value of ≥ 0 given all nodes in V.

Betweenness (or Anthoinisse-Freeman measure) makes the assumption that nodes in strategic positions to influence flows of information between relatively large sets of nodes should be considered as central. Or in the terms discussed before a node’s betweenness centrality is determined by the number of shortest paths that go through it.

Closeness centrality makes the assumption that nodes that are relatively close to all other nodes are central. Summing over the distance of a node to all other nodes in a graph results in the closeness centrality. Closeness centrality is related to degree, instead of summing over adjacent nodes it sums over all nodes. This actually gives a measure of decentrality, the value grows as the nodes are further apart. A property that unifies degree, betweenness and closeness is that in an undirected, unweighted star-shaped graph the center node that solemnly connects all the other nodes of the star, is most central.

Cliques and Clustering

While centrality is used by SNA researcher to describe the hierarchy of social networks [29], it has little to say about collaborating groups of nodes. Just a centrality measures can help in finding central nodes, other measures exist to discover groups of nodes that are connected given the set of edges in the network, an example of such a group is a clique. As defined by Luce & Perry (1949) [30] a clique is "*A subset of the group forms a clique provided that it consists of three or more members each in the symmetric relation to each other member of the subset, and provided further that there can be found no element outside the subset that is in the symmetric relation to each of the elements of the subset*.". Counting the cliques in a graph can yield a more interesting insight into the structure of the graph compared to the density (number of existing links divided by the number of possible links) or average centrality of the graph. Cliques as described by Luce & Perry might be an interesting way to find groups of friends in a social network, which gives an intuitive goal at what these types of measures are trying to achieve.

Another way of describing the connectedness of a network is by calculating the clustering coefficient. This coefficient assumes that the more triangles a network includes, the more clustered the network is. The clustering of node v_i in an unweighted graph is calculated as such:

$$Clustering_i = \frac{2t_i}{degree_i(degree_i - 1)}$$

Eq. (6) as appeared in Onnela et al. (2007) [31]

Where t_i is the number of triangles around node *i*. The maximum outcome of 1 if all the adjacent nodes of v_i are connected and the lowest outcome is 0 if none of the adjacent nodes of *i* are connected. Numerous different grouping mechanisms of nodes exist, such as the just discussed cliques and clusters, other include communities, modules and cohesive groups [32]. In real-world networks its more likely that communities overlap instead of being isolated groups. It is outside the scope of this thesis to discuss the algorithms for finding communities, one of these algorithms called k-clique-community is however applied to test the prediction made by Crowston & Howison about the rising number of communities in growing networks. The definition of a k-clique-community is the union of all cliques with size k. The assumption is that a community consists of multiple connected sub graphs that share many nodes between them [32]. The algorithm sets out to find these connected sub graphs which consist of cliques with the minimum size of k.

Open Source Software

In the SNA literature groups of people are modeled as networks to explain and describe their structure, it is then no surprise than that SNA is often applied in organizational research. OSS communities are especially interesting as their goals and management are often fundamentally different from popular research subjects such as companies, governments and groups of students.

OSS lacks a clear definition, different organizations and user groups however all mention a set of important similar properties: the source code must be available to read, must be distributed freely and may be enhanced by anyone [33] [34]. OSS has been around since the 1980s and finds its roots in the free software movement started by Richard Stallman in 1983. Stallman launched the GNU General Public License, a license that guarantees users can share and change all versions of a program [35]. Stallman called this license "copyleft", developers using other free software are obligated to share their derivative works openly. The most well-known OSS project is the Linux operating system, a white paper by McPherson et. al. [36] estimates that the Linux kernel would approximately cost $10.8 billion to build commercially in 2008. This shows that OSS is mature, it might have started as groups of hobbyists sharing code, but the success of WebKit browsers, the Amazon Kindle reader or Google's mobile operating system Android (which uses the Linux kernel) might never have seen the light of day without the OSS tradition. Gradations of software openness have been made, for example the Open Governance Index [37], by comparing the relative success of well-known OSS projects the results do not suggest that more openness creates more success (a larger user base), more open projects do outperform the lesser open projects when it comes to longevity.

OSS is known for its interconnected network of developers, users, products and services. Organizing efficient software development is hard, intuitive 'improvements' such as adding more developers does not always increase development [38]. This effect is sometimes referred to as Brook's law: "*Too many cooks spoil the broth*"[1]. The interest in OSS projects is therefore not unexpected, as these projects seem to successfully produce software without having strong or hierarchical organizational structures. A famous quote from Linus Torvalds, who started work on the Linux kernel, Git and GitHub, "*Given enough eyeballs, all bugs are shallow*" contradicts Brook's law. Open source software is therefore a bigger concept than just sharing source code openly and freely, it is an organizational paradigm that welcomes feedback from anyone and is much more an evolutionary process than a process of setting and achieving goals. An important concept in this thesis and for OSS is the concept of open source collaboration. The general definition of collaboration, working with another person or group in order to achieve a goal, is too broad for this context. A more elaborate definition is found in Neus & Scherf (2005) who define the process of open source collaboration as '*a meritocratic philosophy that invites feedback from everyone, regardless of official status or formal training and frequent releases of interim versions to encourage testing, feedback and quick evolution of solutions.*' [39]. With this definition in mind, when considering just interpersonal collaborations, the term collaboration refers in this thesis to: giving feedback, testing proposals and working towards the quick evolution of ideas and source code with fellow project members.

Social Network Analysis in OSS Research

Reconstructing social networks based on communication data is not a new idea in OSS research. Bird et al. [3] reconstruct a social network based on emails from Apache mailinglists and introduce edges between

[1] Frederic Brooks was a development manager at IBM and is the author of 'The Mythical Man-Month: Essays on Software Engineering'

actors when they have exchanged at least 150 emails. The results show that the betweenness centrality of actors in the network is strongly correlated with the number of source code changes made by the actor. The analysis of email behaviour shows the recurring phenomenon of power-law distributions in real-world networks. Strong power-laws describe many of the actions and attributes of network actors, such as in-degree, out-degree and the number of source code changes by a single actor. The limitations of Bird et al. [3] are in the use of only a single OSS project, without investigating the changes of the reconstructed project network over time.

Instead of one mailinglist, Crowston & Howison [29] use mailinglists from a set of 52 OSS projects. The mailinglist are and addition to their main dataset of 120 OSS project networks, reconstructed using messages in bug report systems. Crowston & Howison [29] use the reconstructed networks to see if OSS projects have similar network properties. They find a normal distributions of network centalities and conclude from this that OSS projects have a normal variaty of organizational centralization hierarchies. An admitted limitation of their analysis is grouping all interactions over time into one network. The changes over time in an OSS project network are thus not being considered. Crowston & Howison [29] find a negative correlation between out-degree centrality and project size. The authors speculate that this might be the result of increasing modularity of older and larger projects, where a large project might consist of multiple smaller projects. To determine the modularity of a project dynamic networks are needed, based on continuous data as Croston & Howison [29] and for example Kabbedijk & Jansen [40] conclude. There is still no research proposing a definition for modularity in OSS projects or showing the modularity of such a project over time. Bird et al. [3] Crowston & Howison [29] use communication data to reconstruct social networks, but little to no attention is paid to the strength of edges in the networks. Edge weights are however crucial in finding communities or other modules in social networks as Sade [7] shows.

The current state of SNA in OSS projects lacks and struggles with:

1. What time window length and step size to use when investigating dynamic social network of OSS projects.
2. What to base the existence of edges in social network of OSS projects on.
3. How to measure the strength or importance of an edge given a specific timespan.
4. What a module or collaborating team of OSS project members looks like in social network terms. Should it be described as a clique, community, cluster or might it even be impossible to use network properties to identify such groups?

Chapter 4
Network Activity & Social Network Reconstruction

The discussed literature highlights how theories about organizations and OSS can be tested by reconstructing and analyzing social networks. To continue this research this thesis makes the case to only use a particular set of data suitable social network reconstruction called network activity data. In this subsection the arguments for such a specification are introduced together with a novel weighted social network reconstruction method.

In the seminal work 'The mathematical theory of communication' (1948) by Claude E. Shannon, a system of communication is introduced consisting of five parts: an information source, a transmitter, a channel, a receiver and a destination. The information source produces a message which is consequently communicated to the receiver using the previously named parts. A message traversing from an information source and reaching the destination can be seen as an activity. Whether the message was accurately transmitted, conveyed the desired meaning or affects the conduct of the receiver in the desired way is for now not important. The assumption can be made that the activity confirms a relation or edge between the information source and the destination and therefore qualifies as an network activity. This assumption was made for example by David Berlo [41], similar elements from Shannon's system appear in Berlo's sociological SMCR model. According to this model communication or activity occurs when "*a sender (S) transmits a message (M) through a channel (C) to a receiver (R)*". The concepts of transmitters and receivers are also applied; the sender encodes messages in words which can be decoded when received by a receiver. These models can be extended by adding feedback or confirmation of the communication process. This extension is often called the 'transactional model of communication', which describes human communication in which people play the role of information source and information receiver simultaneously.

The term activity seen from this perspective has a scientific history starting in systems theory and ending up being developed by sociology. To make activity an appropriate term for SNA, an activity should also be able to connect a sender and receiver when the sender and receiver are the same node, this is a self-loop. This is important as SNA is often applied in organizational research where identifying unconnected but active people can be important. This thesis proposes to use the following definition for network activity data:

A set of intentional or unintentional transactions in which one or more actors within a network boundary send and or receive a message, with a specific timestamp and duration, over any available channel.

This definition uses defined terms from Shannon's and Berlo's work such as transaction and message, it also adds definitions from SNA such as network boundary and actors (nodes). The definition sets out to select data appropriate for social network reconstruction from a wide array of available data. For example, the definition excludes follow relations on GitHub or Twitter, because the act of following someone has no obvious duration Another difference is intentionality, where a follow relation clearly shows the intention of one actor to follow another, this definition does not require its activities to have the intention to establish a relation or bond. The definition does not require confirmation or feedback, one way communication suffices. The transaction or communication is defined as the process outlined by Berlo [41] including the steps described previously. In a network setting the edges between nodes serve as channels for activity. The definition allows for loops, for example an author starting a discussion with an opening argument. This action is maybe aimed at an unknown group of people , the first action however shows the activity of the author within a network boundary. The first activity confirming a relation between actors is the moment of edge creation.

Modeling Tie Strength

For most of the discussed SNA literature, edges are introduced between nodes when they intentionally confirm a relation. This is a reasonable choice for modeling relationships if a researcher is interested in historically accumulated networks of conscious relationships. Most SNA research is however interested in more timely and organizational issues, here the assumption that every existing edge is equal to any other in the network given any time is inaccurate. The challenge is to define a reasonable way of calculating an edge weight starting with the assumption that every edge strength is not always equal to all other edge strengths. This research is interested in the relative strength of relations in OSS networks, as there is no research about this specific subject, literature about tie strength in general is used. The resulting model will be tested using an OSS project network, but could also yield positive results in other settings. The following limited set of tie strength factors from literature is used in this thesis:

- **Baseline**: If there has never been any activity, i.e. there is no edge between any pair of nodes, the activity for all the nodes and the network as a whole is 0 [12] [17].
- **Duration & Frequency**: More time spend as a result of more or longer activities increases edge weight. [11] [16].
- **Age**: An older edge has a stronger edge weight compared to a younger but otherwise identical edge [12].
- **Recency**: Recent interactions are more important than older ones, they contribute more to the weight of an edge [12].
- **Diminishing returns**: A strong edge has a smaller increase in weight due to an activity compared to its weaker self, experiencing the exact same activity [17].

This set of assumptions ignores some of Granovetter's previously mentioned factors such as emotional intensity and reciprocal services. The relations however all apply to communication data and can be combined to create a model for tie strength, in this case the strength of collaboration ties in an OSS project. There are gaps in the literature when trying to connect or explain dependence or interdependence of the different factors. It is also important to mention that for example topological influences such as common neighbors, Jaccard's coefficient and preferential attachment are not considered [42]. The focus is only on a single edge in an undirected graph, without directionality it is for example also nearly impossible to model reciprocity. To following steps have been taken to bring these factors together in one model for tie strength:

A undirected graph is defined by its symmetric adjacency matrix denoted by A, where elements of A $a_{i,j}$ ($=a_{j,i}$) are real valued functions of nodes *i* and *j*, where (i,j = 1, …, *n*). The elements of A (the edge weights) can be defined as the sum over all activities c:

$$a_{i,j}= \sum_{c} i^c j^c$$

Eq. (7)

Where i^c has two possible values of 0 or 1, where $i^c = 1$ if node i takes part in activity c. An almost identical function is used by Barrat et al. [21] for calculating the weight of co-authorship relations. This weight measure does not account for the age, relative frequency, duration and recency of activities. A more appropriate measure is:

$$W_{i,j}(t_s, t_w) = A_{i,j}^{a}(t_s) * D_{i,j}^{d}(t_s) * R_{i,j}^{r}(t_s, t_w)$$

Eq. (8)

Where the weight of an edge $W_{i,j}$ is determined by three measures. $A_{i,j}^{a}(t_s)$ is a measure for the relative age of the edge, $D_{i,j}^{d}(t_s)$ a measure combining the relative duration and frequency of activities and $R_{i,j}^{r}(t_s, t_w)$ a measure for the relative recency of activities. The complete timespan of activities is represented by t_s, and the length of a single time windows of investigation is t_w. The measures a, d, r have values between and including 0 and 1, where a value of 0 removes the influence of one of the measure from the total relative weight. The multiplication of these measures is one of many possible combinations. The measures are assumed to be dependent, take for example age, duration and frequency. A young age directly influences the possible frequency and duration values of activities. A young age will allow for fewer activities compared to an old age with activities of the same duration. The multiplication reflects this dependency, however no previous research or evidence suggests that this is indeed the correct relation. A multiplication is chosen as it is the simplest dependency relation and enables the use of a, d, r to change the proportional influence of individual measures.

The model just as in dynamic network requires a timespan t_s and time window t_w. In this thesis for simplicity the time window is always equal to the step size. If for example the time window is 7 days, the first step will only include activities originating during the first 7 days. The second step will only include activities originating in day 8 to 14. A in depth discussion about how to choose such measures is outside the scope of this thesis, Kossinets and Watts provide an good review [24] (page 414).

The individual measures of A, D and R, are now discussed individually and in more detail: Let $a_{a_{i,j}}(t_s)$ be the age of an edge $a_{i,j}$ during a time span of t_s and a_m the maximum age of all edges in E. The relative age of an edge $a_{i,j}$ is defined as:

$$A_{i,j}(t_s) = \frac{a_{a_{i,j}}(t_s)}{a_m}$$

Eq. (9)

An edge with a greater relative age is considered to be stronger and its strength is assumed to increase linearly with time. The age of an edge starts at the beginning of the first activity which creates the edge.

Results of Marsden & Campbell [16] suggest that strong ties have a negative correlation between frequency and duration. Where either the communications are predominately short and frequent or long and infrequent. The measure D values three settings: 1) when the duration is above average with an below average frequency, 2) when the frequency is above average and the duration is below average and 3) when both duration and frequency are above average. The third situation is unlikely, as increases of either duration or frequency by limitations of time make the decrease of the other inevitable.

$$D_{i,j}(t_s) = \frac{\sqrt{\left(\frac{f_a}{f_{ma}}\right)^2 + \left(\frac{d_a}{d_{ma}}\right)^2}}{\sqrt{2}}$$

Eq. (10)

Given edge $a_{i,j}$ during timespan t_s, where f_a is the average frequency and d_a the average duration of edge $a_{i,j}$ and f_{ma} the maximum average frequency and d_{ma} the maximum average duration when considering all activity in graph G, during the complete timespan t_s. D measures the value of activity frequency and duration by measuring the distance from the baseline situation of no activity ($f_a = 0, d_a = 0$). The maximum value of D is $\sqrt{2}$ which reflects the situation when the activity over an edge is both maximally frequent and maximally long, considering all other activities for any edge. Note that the measures in the numerator of the division is simply a vector of the axes average frequency and average duration, where the length of the vector determines the strength of the value $D_{i,j}(t_s)$. Just like the age measure the result is divided the highest possible outcome, in this case the square root of two. The function of the square root also complies with the diminishing returns assumption, the more the frequency and duration grow the lower the relative returns will be for equal increases.

The last measure for recency sums over each time window in the complete timespan of the network. For each time window the frequency and duration are considered and for every window closer to the final window the relative additional weight of the activities increases. The weight of activities can be damped in order to remove weak edges from the network. The measure $\sqrt{R_m}$ removes the value of a combination of duration and frequency. For example, $\sqrt{\left(\frac{1}{10}\right)^2 + \left(\frac{1}{10}\right)^2}$, would cancel out activities lasting one tenth of the average length of all activities and with a frequency of one tenth relative to all activities.

$$R_{i,j}(t_s, t_w) = \frac{\sum_w^s \frac{\sqrt{\left(\frac{f_{t_w}}{f_m}\right)^2 + \left(\frac{d_{t_w}}{d_m}\right)^2} * \frac{t_w}{t_s} - \sqrt{R_m}}{\sqrt{2} - \sqrt{R_m}} \quad if\ f_{t_w} \neq 0}{n_{t_w}}$$

Eq. (11)

Given edge $a_{i,j}$ during time period t_s, by summing over all the complete time windows t_w fitted into timespan t_s where:

- f_{t_w} is the frequency of activities during time window w over edge $a_{i,j}$.
- f_m is the maximum frequency of any edge during any time window during the complete timespan t_s.
- d_{t_w} is the sum of durations of all activities during time window w over edge $a_{i,j}$.
- d_m is the maximum duration of any edge during any time window during the complete timespan t_s.

- R_m is the weakening measure of every activity which has a value between and including 0 and 2. A value of 2 would even cancel out the most frequent and long activity in the last time window.
- n_{t_w} is the number of complete time windows t_w for which $f_{t_w} \neq 0$.

The recency measure is normalized by dividing by the highest possible outcome for each time window ($\sqrt{2} - \sqrt{R_m}$) and is normalized again by dividing by the number of time windows during which the edge has seen activities. The measure also forces edges with infrequent and short communications in the past to compensate the negative outcome of these sums with activity in the nearer past. A negative outcome of the recency measure results in an overall negative weight and removes the edge from the network. These last steps are not based on past research and are in that sense contrived and require a vigorous test for validity.

The recency measure contains the vector present in the duration and frequency measure. The difference between the two occurrences is that in the recency measure the vector is calculated for each individual time window, while in the duration and frequency measure it is calculated using averages for the complete time span. The duration and frequency measure will therefore give the strength of the relation based on those two variables independent from when the activities resulting in the variables took place. The recency does however compensate for the moment the activities took place and allows to even ignore some of the activities if they occurred too long ago and were infrequent or short.

Chapter 5
Research Methods and Data Collection

The majority of data used in this thesis is not created during an experiment, but already exists and needs to be collected, filtered and modelled. It makes sense then to use a data mining research method, in this case CRISP-DM (Cross Industry Standard Process for Data Mining) [43], instead of more well-known research methods such as the empirical cycle. As the name suggests, the method is used in industry as well as in research. The method consists of six steps that do not necessarily have to be taken in a precise order. The iterative steps of CRISP-DM are Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, Deployment. The last step of deployment has no analogous step in this research as it might have in industry and is therefore not performed.

The bird's-eye view of the research is shown in Figure 1. During the first step all data is collected and prepared to create the activity networks, follow networks and to set up and process the survey. This step is preceded by a broad overview of GitHub and Rails to introduce the reader to the basics of both subjects. The second step uses the data collected in the first step to create five networks:

1. **Follow Network:** an unweighted network based on follow relations, where an edge exists between two nodes when at least one of the nodes follows the other.
2. **Mutual follow network:** an unweighted network based on mutual follow relations, where edges only exist when nodes both have decided to follow each other.
3. **Sum Activity Network:** a weighted network based on communication edges where the weight of edges is based on the sum of communications exchanged by two nodes.
4. **Model Activity Network:** a weighted network based on communication edges where the weight of edges is based on the results found by using the model introduced in this paper.
5. **Reference Network**: a weighted network based on the survey results completed by Rails members, where the weight of edges is based on the order by which they have been entered by survey subjects.

The third step makes the most important evaluation of the research, namely the accuracy comparison which will show which network most closely resembles the reference network. The fourth step explores the networks as they can be used to create and continue important existing research. The results of the third and fourth step are used in the fifth step to conclude the research and discuss the outcomes and future directions.

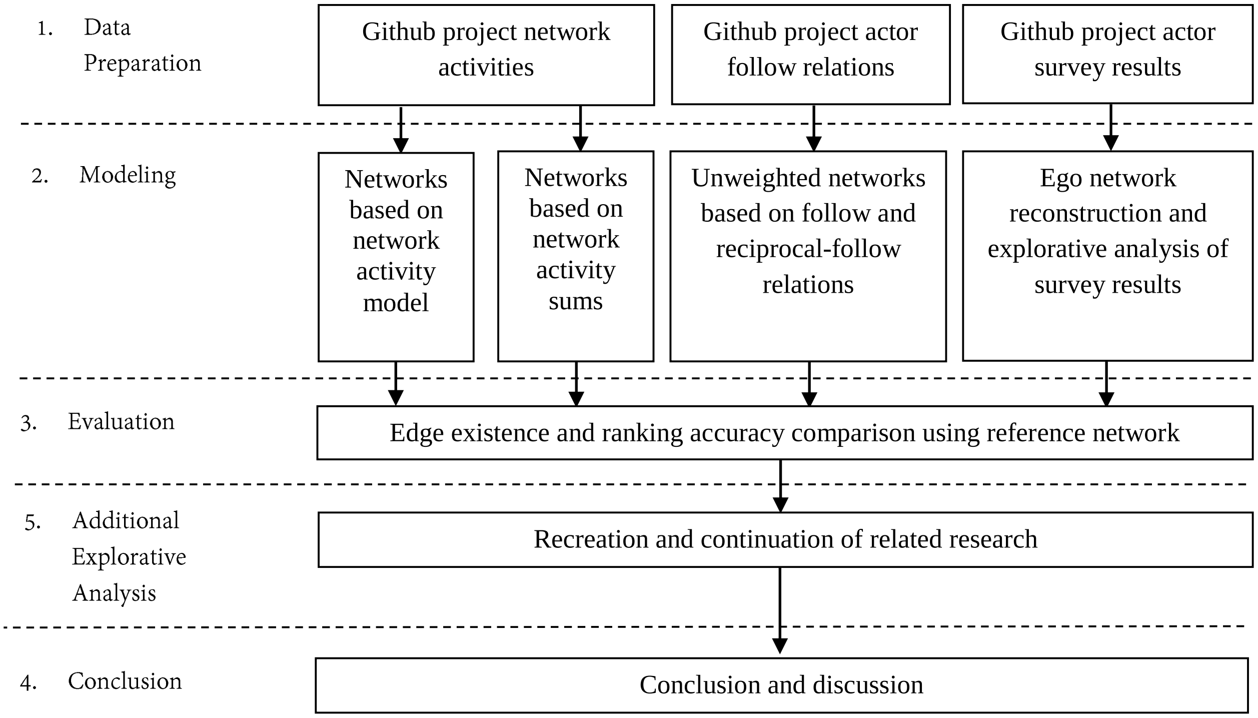

1. Data Preparation
Github project network activities
Github project actor follow relations
Github project actor survey results
2. Modeling
Networks based on network activity model
Networks based on network activity sums
Unweighted networks based on follow and reciprocal-follow relations
Ego network reconstruction and explorative analysis of survey results
3. Evaluation
Edge existence and ranking accuracy comparison using reference network
5. Additional Explorative Analysis
Recreation and continuation of related research
4. Conclusion
Conclusion and discussion

Figure 1 | Bird's-eye view of the research method’s order and steps

Business Understanding

Data used in this research is collected from GitHub [44]. GitHub is an online repository for sharing and developing software with more than 5 million users. The website uses the Git revision control and source code management system, to which it lends it name [45]. GitHub provides users with an API (application programming interface) which makes it possible for researchers to reliably access large sets of data about public projects. GitHub is the ideal data source as it combines a social network of follow-relations with records about development and other activities from Git repositories. Compared to other sites such as SourceForge it also has an easy API instead of a SourceForge’s set of databases with acknowledged inaccuracies and changes in data structures over times.

The Ruby on Rails project is selected as data source, the first proposal of this thesis included a larger set of 10 projects to be investigated. As the filtering and preparation of the data takes time, only Ruby on Rails is selected, as it was the biggest project in the selection and has gone through different phases of openness and has a recorded history of almost ten years. With 2,310 people as listed contributors on GitHub and with 8,704 contributors found in the actual data of the project and more than 132,000 recorded activities Ruby on Rails is an interesting example of a popular OSS project to research.

Ruby on Rails, from now on referred to as Rails, is a web development framework based on the Ruby language (created by Yukihiro Matsumoto). Although the framework has gone through many changes, the underlying architecture of the framework has not changed, this Model-View-Controller architecture is a much applied architecture in web development frameworks (i.e. Django, CodeIgniter). The focus of the framework has been on ‘programmer happiness’. Rails was started by David Heinemeier Hansson in 2003, the source of Ruby was made open in July of 2004, in February of 2005 commit rights to the project were also extended to others than Hansson. In a blog post during August 2006 Hansson announced that Ruby would be included in Apple’s OS X [46]. At the time of writing, Ruby has seen 11 major releases, starting with 1.0 in December 2005 and the latest 4.1 released in April 2014 [47].

Data Understanding

GitHub repositories are used to determine the network boundary. The list of users that have ever contributed to a particular project in a variety of ways will be defined as the population. This is a large scope, it is however needed as it is not clear yet where the most interesting and active users reside. Users can contribute in different ways with different levels of rights. Since a complete discussion of user roles and permissions of GitHub is outside the scope of this thesis, only the most important roles and interactions considered to be network activities are discussed.

Two types of account types can be identified, user accounts and organization accounts. The organization accounts can be considered as the super type for user accounts but are not a mandatory to have as a project. The organization contains owners and teams, the owners have all the possible rights and permissions. The members of teams can receive three types of access to an organization: admin access, write access and read access. Users with write access can write (push) source code changes and perform the standard create, read, update and delete actions. Projects that do not use the organizational structure can add collaborators to the project that have read and write access.

When an repository is open to the public, any GitHub user can ‘fork’ the repository which means to make a local copy of the project. The user can then make adjustments by creating a new branch for the project and add commits to it. A commit consists of changes that have been made to the repository. These changes can be file changes or for example changes in the directory structure of a repository. When a user decides that his or her changes are probably valuable for the original project, the changes can be made into a pull request and send to the original project. Users and project members with write access can now discuss the new pull request and decide to add it to the main branch. This is the most important part of open GitHub projects, the ability for any user to make a local copy, start improving the project and send the changes back to the community for discussion. It is also possible to start a discussion without sending a pull request, but by opening an issue. The issues consist of comments and commits, when an issue is resolved it can be closed by write access team members, otherwise it is labeled as open. This leaves us with the following subset of activities:

1. Sending pull requests
2. Commenting on commits, pull-requests or issues
3. Opening and closing issues
4. Adding commits

The aim is not to create a development history, but to find the collaboration and social history of a project. The comment activities are therefore most interesting, as they are communications between members and fall under the network activity definition outlined in this thesis. Sending a pull request or adding a commit are communications in part, as these action include a description which is the opening statement for a possible discussion about the change. In a clear communication setting such as a chat message or telephone call the active nodes are easy to identify. In a OSS project setting it is however harder to define group activities and activity durations. The following choices are therefore made to be able to use the tie strength model introduced in this thesis.

1. The first action for an issue, pull requests or commit confirms a self-loop with the original author. It reflects the fact that the author is active, whether anyone decides to comment on the action is not a prerequisite for a network activity. If these actions are not included, members of the network that work alone without communicating with others would not appear in the network. In the most extreme case when every member works alone, the network would be empty while in reality the combined activity of the members yields results. A researcher should still be able to create a network of such a project reflecting the un-connectedness, which would not be possible without self-loops.
2. Any comments on an action confirms a relation between the authors of the new and previous comment, a similar decision is made by Crowston & Howison [29]. Sporadically @ mentions are used to direct messages to specific users, when this is the case the message will be directed to the mentioned user(s) instead of the previous commenter.
3. If reasonable distributions of message lengths are found, the character should be used as communication length. This is a reasonable choice when taking into account the assumptions of the model in Eq. (8), as typing a message or speaking it both require time. Simply assuming that typing longer messages will take longer compared to shorter messages keeps the assumptions about frequency and duration valid. This choice is susceptible to outliers, resulting from copying content or code, a thorough data preparation is therefore needed.

As mentioned before, besides source code management, GitHub also includes some social aspects on the website. In order to create the often used social networks based on follow relations, the followers and followings of users are collected. GitHub users can follow other users to track their work, comments and other activities. Although GitHub describes it as ‘Pick a Friend’, it is likely that follow relations on GitHub rather reflect an interest in the professional work of a user than his or her social self. These networks indicate some social status and can therefore be used to recreate the research by Bird et al. [3] to see if users who have contributed more also have more followers.

The GitHub API has some limitation, for example only the top 100 contributing contributors of each project are accessible. Some of the commits and comments are anonymous, as the user that created the commit might have deleted his or her account. It is however possible to get a good view of a projects history by iterating over all the project commits. The commits contain information about the author and committer including their email addresses. By listing all authors and committers by going through the commits the limitation of 100 contributors is easily evaded. The commits that can be accessed through the GitHub API are however a filtered set of commits that leave out some commits, the criteria of this filtering turned out to be a bug. Communications by the author with Robert Sese, a GitHub staff member, confirmed the bug, which was resolved on May 9th 2014 [48]. By copying the .git structure and analyzing the actual files it is also possible to get to the correct amount of commits including their unique identifying keys (sha). By using these keys it is possible with the API to get all the commits with the correct information. As GitHub limits the amount of calls per hour to the API (5000) and this method is quite expensive, as it uses a single call for every commit, the process is slowed down by this approach. It is however the only viable option during the data collection part of this thesis. The following data were collected for Rails:

1. Commits:
   All the commits are collected with the author id, title, description and timestamp, the important part are the comments for each commit. All the comments are saved by collecting the comment

text, author id, author email address and timestamp. The relations will later be established for these comments using the approach described previously.

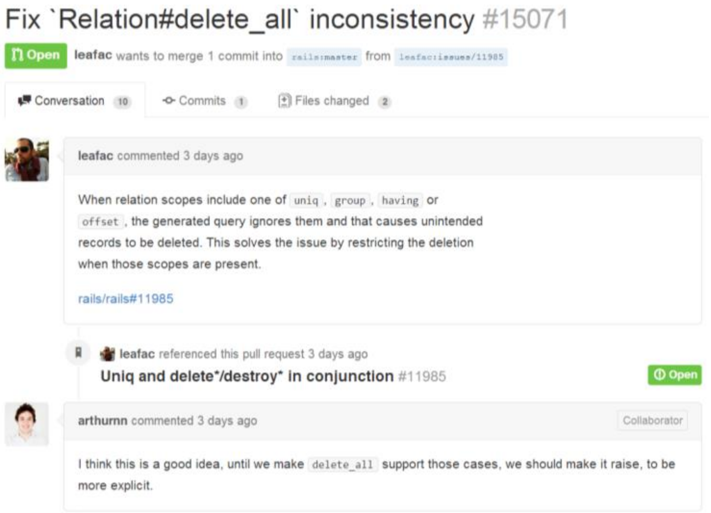
2. Issues, Pull Requests and Commits that are part of a Pull Request:
   The issues are gathered in a similar fashion as the commits and pull requests, both the author and comment details are saved. As issues can also be pull request, that can in turn include commits, comments and review comments, parsing issues is therefore important and will be explained in detail in the data preparation step.
3. Following and Followers:
   For each user that has created a commit or pull request the followers and the following relations are saved. The relations are used to create the network of follow relations within the population of the project.

The result is a complete set of all the comments exchanged between users during the complete time the project has been on GitHub. The reported number of contributors on GitHub will be lower compared to the amount of contributors found in this way, as the authors of commits that have not been included in the project and users that only opened issues are included.

Data Preparation

Data Collection

Collecting all comments is not as straight forward as requesting all of them and iterating over the results. There is a hierarchy of objects that might contain comments that are represented online in a simple manner, but are hard to discover and order when using the API. A project on GitHub can have multiple ‘branches’, which, as the term suggests, are different versions of the same project diverging from the master (e.g. most important) branch of the project. The data collection in this thesis only includes data from the master branch.



Fix `Relation#delete_all` inconsistency #15071

leafac wants to merge 1 commit into `rails:master` from `leafac:issues/11985`

leafac commented 3 days ago

When relation scopes include one of `uniq`, `group`, `having` or `offset`, the generated query ignores them and that causes unintended records to be deleted. This solves the issue by restricting the deletion when those scopes are present.

rails/rails#11985

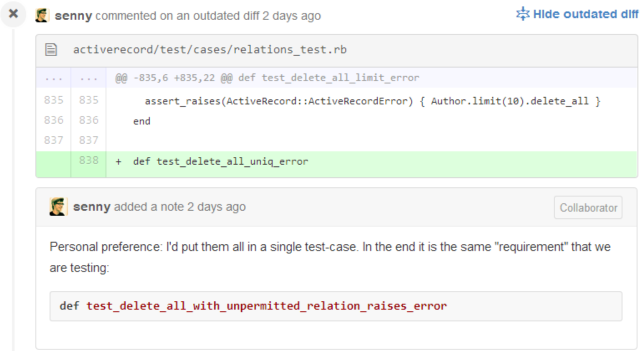
leafac referenced this pull request 3 days ago

Uniq and delete*/destroy* in conjunction #11985

arthurnn commented 3 days ago

I think this is a good idea, until we make `delete_all` support those cases, we should make it raise, to be more explicit.

Figure 2 | Example of a pull request that is simultaneously an issue and all together have one commit without comments (May 15th 2014).



senny commented on an outdated diff 2 days ago

```
activerecord/test/cases/relations_test.rb
@@ -835,6 +835,22 @@ def test_delete_all_limit_error
     assert_raises(ActiveRecord::ActiveRecordError) { Author.limit(10).delete_all }
   end

+  def test_delete_all_uniq_error
```

senny added a note 2 days ago

Personal preference: I'd put them all in a single test-case. In the end it is the same "requirement" that we are testing:

```
def test_delete_all_with_unpermitted_relation_raises_error
```

Figure 3 | Example of a review comment on a commit (May 15th 2014).

Essential is to understand the hierarchies of objects that form GitHub projects: for every issue there is a possible pull request and for every pull request there are possible commits with review comments. This hierarchy results ends with a set of review comments. Review comments are always about source code changes, they are directed at certain lines of code. To illustrate this first hierarchy consider Figure 2, this is a Rails issue with id #15701, as it can be found on the GitHub website. As can be seen at the top of the figure, issues are parent to conversations, commits and file changes. There are however no review comments to see yet, Figure 3 which shows the first review comment. The line highlighted in green is commented on by ‘senny’ who proposes an encapsulated change. This is the end result of the first hierarchy, the collection of all such review comments.

There is however much more to collect, for almost each step in the hierarchy (issue > pull request > commit) there are comments to be found. Figure 2 for example shows a comment by ‘arthurnn’ on the issue opened by ‘leafac’, this is a comment on an issue which is also collected. The first comment by ‘leafac’ is a self-loop as discussed before. The first comments shows the activity of the author, but not yet a collaboration with someone else. Another place where comments can be found is in the set of commits that are not related to a pull request or issue. These commits are made by users with administrative rights to the project, as they do not need to go through the process of opening an issue to apply the commits to the project. These commits also have an opening statement like the issues, which is a message that is directed at the project and confirms an loop activity with the author. If others decide to comment on the commit these comments are saved as activities. However, commits that were already part of an issue have already been stored, these initial commits are therefore not recorded again to prevent duplicate activities. All these steps together create a set of activities, for each activity the following data is saved:

- Source: The GitHub user that has written the message.
- Target(s): The GitHub user that the message is targeted at, if the message contains @ mentions these will be used as targets. This makes it possible that a message is targeted at multiple users.
- Length: The number of characters of the message.
- Message: The actual text of the activity.
- Timestamp: A Unix timestamp indicating when the message was created.
- Type: A value indicating if the activity is a comment on a commit, a review comment or a comment on an issue.

It could be argued that especially the opening message of a commit which was never part of an issue and therefore never part of a conversation is not a social activity. This case is made even stronger by the fact that a large number of commits that are not connected to an issue do not include comments at all. The author of such a commits is however telling other contributors about the work that he or she has done, in a way the message is directed at all contributors that want to keep up to date with the progress of the project. More importantly, the message is a reminder for the author about what he or she has included in the commit. For this reason it confirms a loop with the original author. Another argument for this choice is the ability to discover which people are collaborating with each other by reviewing and commenting on commits and which users are ‘lone wolfs’ and do most of their work on their own without receiving comments.

The amount of source code additions or deletions in commits are not considered, these measures can with some adjustments be used to see which contributors do most of the work. In this case the interest is in social activities and not coding activities, only the messages directed at other are therefore used as activities.

Data Filtering

One of the challenges of creating a representative dataset of activities is accounting for the outliers in message length. Some contain copied and generated error reports, these messages are disproportionally long, but do not reflect the actual time the user has spent writing the message. Luckily most of these error reports or other pasted materials are encapsulated by ‘```’. By removing the text between these characters it is possible to get a dataset with more representative lengths, where length gives a measure of time spend on the creating the activity. More of this type of filtering is applied, the complete list is given bellow:

1. Some messages use ‘~~~’ to encapsulate pasted materials, in these messages ‘~~~’ is converted to ‘```’.
2. By using a regular expression all the data between ‘```’ are removed: '```.*?```'
3. Some messages open the pasted materials by using ‘```’ but do not close this by again using ‘```’, to account for these messages a second regular expression is used: ‘```.*$’
4. Some message encapsulate pasted materials using <pre> tags, to remove these the following regular expression is used: ‘<pre>.*?</pre>’
5. Some message encapsulate pasted materials using <code> tags, to remove these the following regular expression is used: ‘<code >.*?</ code >’
6. Some messages have titles but the actual text is empty, this occurs quite frequently for commits. In these case the title length is used instead of the message length.
7. Some of the messages are completely generated by software, for example the ‘lighthouse-import’ user, all activities by these type of users are ignored. These users are hard to detect in an automated fashion and are therefore done by hand.
8. Some messages include cited emails, these emails are precluded by a short sentence, for example: ‘On Mon, Sep 24, 2012 at 8:14 PM, David Heinemeier Hansson <notifications@github.com> wrote:’ . The email is usually copied in full with at the start of each line a’>’. In these cases the quoted emails are removed from the message.
9. A large set of messages still have excessive message lengths due to pasted materials that are not filtered out by the actions above. These messages are filtered by hand to select the actual text written by the author of the activity. Most of these messages are automatically generated error traces that are not encapsulated in any way.

This is a set of decision that can be challenged, in cases with pasted materials it probably take a while for a reader to get through the message, so the assumption that the copied materials don’t take extra time is incorrect at least for the target audience. Copying text does also take more time than not copying anything. The filtered text should therefore get some value because of the following reason, a large number of messages only consists of copied materials. These would unjustly all be categorized as empty messages. A rather arbitrary rule is therefore applied which adds 1 character in length the total length of the message for each 50 characters removed. This is an unsatisfying approach, leaving the messages empty is however unacceptable.

Survey

As described in the data understanding part, the email addresses of contributors are saved. These are used to send all the contributors to Rails an email with a link to a survey. The aim of the survey is obtain the ego networks of users that are part of the Rails population. The following four questions are included in the survey:

1. Since your first involvement with Ruby on Rails, which Ruby on Rails contributors have you collaborated most with while contributing to Ruby on Rails? Leave empty if you feel you haven't communicated or collaborated with any other Github users of Ruby on Rails.

2. Which GitHub users contributing to Ruby on Rails are overall most important for the management of Ruby on Rails development?

3. Are you planning to work on Ruby on Rails in the future?

- No
- Less than I used to
- As much as I used to
- More than I used to

4. How much time have you approximately spend while contributing to Ruby on Rails?

- Less than 1 hour
- 1 - 9 hours
- 10 - 49 hours
- 50 - 99 hours
- 100 - 249 hours
- 250 - 999 hours
- 1000 or more hours

The first two questions are both followed by 10 fields where usernames can be written down, while typing the API of GitHub is used to suggest results with corresponding gravatars. In this way the subjects can quickly select the people they know and the survey program can link the people to the actual nodes. The third and fourth question use a Likert-scale to ask contributors about their future and past activities in a very broad sense. The third question can be used to see what type of ego networks can be found for different future expectations. The fourth question can be used to validate the approaches to determine how much time a user has spent on the project by looking at the comment and other network activities. As related research suggest that the amount of time has a power law distribution, although not an exact power law increase, a wide and quickly increasing selection of times is presented. At the end of the survey subjects are required to agree with the consent for as shown in Appendix A. Without agreeing to the consent form the survey cannot be submitted. The data of the subjects is pseudonymized, meaning that all subjects receive a random identifying key, that is only linked to the subject in a single table, which is only accessible by the author. The results of the survey discussed in this thesis will therefore refer to the anonymous keys of subjects, instead of their usernames or other identifiable characteristics.

Software

The web framework Django [49] is used to create the survey[2], email the subjects, gather and filter the API data. The choice of using a web framework for all these activities is to enable the direct connection between survey and data mining results. When survey subjects enter a collaborator, it is already known by the PostgreSQL database containing all the API results, which makes the generation of ego networks fast and reliable. Django is based on the Python programming language which has excellent libraries for requesting API results and parsing json. The generated results are additionally analyzed and modeled using Gephi [50]

[2] A live version of the survey can be seen here: http://gitresearch.webfactional.com/survey/1/2/

for visualization, Networkx [51] for the explorative analysis of networks and SPSS 10 for the statistical analysis of the results.

Modeling

As mentioned before, five different type of networks have to be created as the final input for the evaluation. The follow network, mutual follow network an sum activity network have been explained and can only be created in one way.

The model activity network however has variables that can create a set of networks with varying edges and edge weights. These variables are: timespan, time window, step size and the relative influence of age, duration, frequency and recency. For sake of simplicity and because of time constraints only one variation of these variables is used. The timespan should be the complete history as past communications can still have an influence today. The time window should be determined by the average time between communications and the step size should be equal to the time window. These none overlapping steps are chosen as they prevent repeatedly valuing a single communication activity. Other authors however argue to use a small overlap just big enough to not break of threads of communications that are resuming to exist instead of being called into existence. Additionally overlap should also not be too big to be dominated by past communications. This use of a small overlap is underpinned by clear evidence from Kossinets and Watts [24], but is applied and tested specifically in a directed network, which makes application of these results impossible in this thesis.

The fifth and last network which is used as a reference is modeled as an ego network, with at its center the survey subject. Each collaborator mentioned by the subject in the survey is added to network with as edge weight the rank of the collaborator. The first mentioned collaborator gets the highest edge weights, while the last mentioned collaborator gets the lowest edge weight of one. In the evaluation only the ranking will be tested, the specific weight assigned to an edge is therefore not important as long as the correct ranking of weights remains.

Evaluation

Accuracy is calculated in multiple ways, firstly confusion matrixes [52] are created for the edges of each ego-network and its equivalent network in one of the predicted networks. These tables show the true positives (tp), true negatives (tn), false negatives (fn) and false positives (fp) of edges (Table 1). With these results multiple interpretations of accuracy can be calculated:

		Predicted	
		Negative	Positive
Real	Negative	tn	fp
	Positive	fn	tp

Table 1 | Confusion matrix of real and predicted networks

$$Accuracy\ (AC) = \frac{tp + tn}{tp + tn + fp + fn}$$

Eq. (12)

This overall accuracy measures divides all the correct edges by the total number of edges, the disadvantage of this measure is the usually disproportionally high amount of true negatives. The results can therefore approach 1 while the actual amount of true positives can be 0.

$$True\ positive\ rate\ (TP) = \frac{tp}{fn + tp}$$

Eq. (13)

To counter this problem the true positive rate gives an indication of how good the model is at finding the correct edges in the real network of the survey.

$$True\ negative\ rate\ (TN) = \frac{tn}{tn + fp}$$

Eq. (14)

The true negative rate is similar but finds as the name suggests the true negatives. By dividing by the amount of true negatives, the number is more comprehensible compared to the high outcomes of the AC measure.

$$Precision\ (P) = \frac{tp}{fp + tp}$$

Eq. (15)

Although very similar to TP, the precision rate focusses only on the real and predicted positive cases. It calculates the fraction of correct edges in the predicted network compared to all the positive edges given in the predicted network.

$$g - mean\ (GM) = \sqrt{TP * P}$$

Eq. (16) as appeared in Kubat et al. (1998) [53]

$$g - mean_2\ (GM2) = \sqrt{TN * TP}$$

Eq. (17) as appeared in Kubat et al. (1998) [53]

The geometric mean provides another way of coming to an accuracy measure without the amount of true negatives influencing it.

Comparing the accuracy of the rankings is not as straight forward as simply testing the accuracy of edges, as the accuracy and the number of edges heavily influences the results of ranking accuracies. The most extreme example would be a predicted network without edges. This could be interpreted as making no ranking mistakes at all, or making all possible ranking mistakes. As the two follow networks do not have weighted edges, only the activity networks are compared for their ranking accuracy. Each survey subject is asked to rank his or her collaborators by strength of the collaboration. To see which of the two networks most closely resembles this ranking, the absolute difference in rank of each edge in the predicted networks and the real networks is summed up. When a network has the correct rank of an edge, the absolute difference is 0. This measures is therefore a measure of inaccuracy, rather than accuracy:

$$Ranking\ score\ (RS) = |real\ rank - predicted\ rank|$$

Eq. (18)

This score is however biased toward small networks, take for example the case where the real network is small and the prediction network is also small, but has no correct edges. Such a network will outperform networks that might include all the correct edges, but have a large number of false edges. A filtering of edges is therefore needed. If an edge in the real network occurs in the predicted network, the ranking is the absolute difference in weight. If the edge does not occur in the real network, the predicted weight is 0 and the absolute difference in weight with 0 is added to the score. This still, but to a lesser degree, favors smaller networks, as a larger network is more likely to make ranking mistakes. The approach should therefore be to compare the ranking accuracy of the networks with exactly the same edges in each case. The model activity network should use all the edges picked by the sum activity network. If in this case the model has a better ranking score, it will be a clear measure that the model determines relative weight more accurately than a simple sum.

Brewer & Webster (1999) [54] show that people tend to forget even high ranking nodes when surveyed, the claim could be made that the chance that a false positive is statistically actually a true positive is larger than the chance that a false negative is actually a true negative. This suggest that a measure of inaccuracy should come down softer on false positives compared to false negatives. A reason not to include this in a test would be the disadvantage for prediction models that actually account for this effect.

Both approaches for edges and ranking are calculated for the example in Figure 4 and Figure 5, the example is taken from the survey results of a contributor with a pseudonymized id '829070' that is used to identify him or her. The width of the edges shows their relative weight, the thicker the edge the heavier the weight of the edge. Table 2 shows the different results of the different accuracy measures. The amount of nodes in the complete network is 8704, this causes the high results for the accuracy (AC) and true negative rate (TN) and g-mean$_2$ results. The measures that compensate for these high results are TP, P and g-mean. An interesting result is therefore the precision rate, while the true positive rate accurately shows that all of the edges have been discovered, it hides the fact that many more were false identified. This false identification is included in the precision rate, which therefore in the view of the author shows the desired accuracy in the most intuitive way, in this case: 0.6.

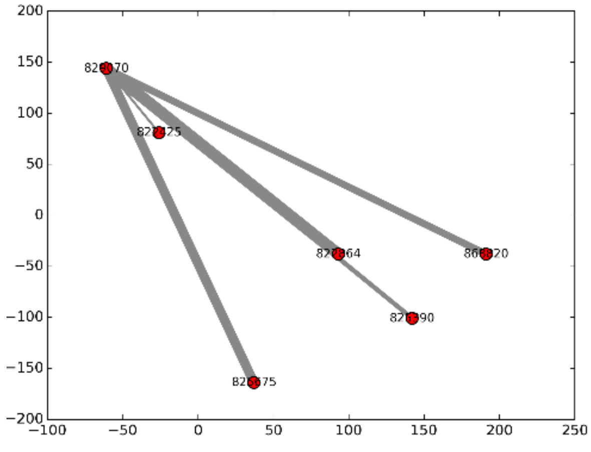


Figure 4 | Example predicted network of ego '829070'

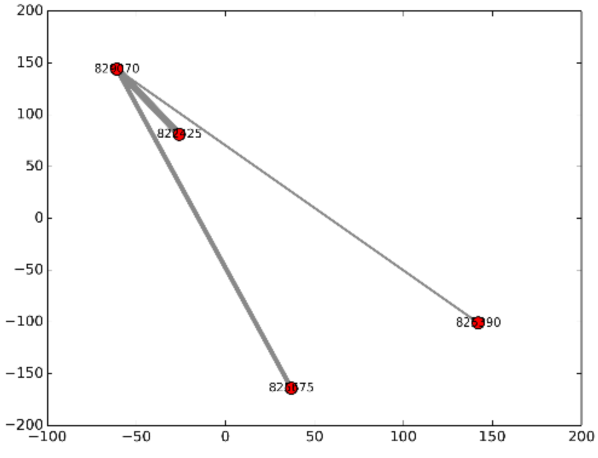


Figure 5 | Example reference network of ego '829070'

	AC	TP	TN	P	g-mean	g-mean$_2$	RS
Predicted Sum Network	$\frac{8704+3}{8704+2+0+3}$ = 0.999770	$\frac{3}{0+3}=1$	$\frac{8704}{8704+2}=$ 0.999770	$\frac{3}{2+3}$ $= 0.6$	$\sqrt{TP*P}$ $= 0.7745$	$\sqrt{TN*TP}=$ 0.9998	|1-2|+|2-4|+|3-1| = 5

Table 2 | Example results of accuracy measures of the two networks in Figure 4 and Figure 5.

The weakness of the just discussed research approach is its generalizability, the goal is to find the best network out of four to describe the existence and strength of OSS developer collaborations. The real GitHub population is in the millions and growing, the population of Rails users can be debated. Gathering a big enough sample size is therefore hard an unlikely to succeed. The generic nature of the model for tie strength introduced in this thesis calls for tests of the model in different network settings with different types of communications, for example networks of emails, Facebook messages or telephone calls. The results of this thesis can therefore only be viewed in a narrow and net yet generalizable context. They serve as an research approach example and what results are attained only apply to the group of survey subjects.

Chapter 6

Results

The results start with the most important result, the accuracy comparison of reconstructed networks created using different reconstruction methods. The results continue with a more broad exploration of the gathered data together with a recreation of the research by Bird et al. and a continuation of the research of Crowston & Howison.

Accuracy Comparison

The accuracy comparison is split up in edge accuracy and ranking accuracy. Figure 6 shows the combined results of the 283 surveys, each node in the graph is pseudonymized. The overarching question of accuracy is simply, which of the four networks most closely resembles this graph and in what way? For each survey the results are used to create ego-networks of the subject and is compared to each of the 4 networks using a set of accuracy measures. The statistical question is to identify the significant differences in the mean scores of the accuracy tests. The scores however violate tests for equality of variances and normality. Non-parametric tests are therefore used to compare the means of the scores. Parametric scores hold to stricter assumptions and are in that sense more powerful and desirable. As the results in this thesis are data mining results and do not deal with the performance of people or groups of people the assumptions of the parametric tests in some cases fail or do not apply. A possible parametric test could be the fixed size, repeated measures, one way ANOVA. Because the score measure is repeated for each model and the significant differences could show which networks perform better. Even when the

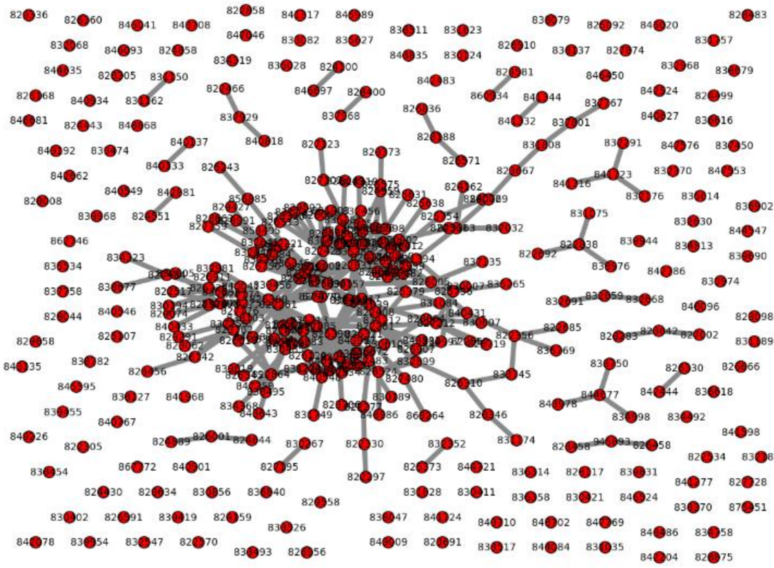


Figure 6 | The complete graph that is created when all survey results are combined.

accuracy measures are logarithmically transformed no normal distributions and equality of variances can be found. A non-parametric test is therefore used which is similar to the earlier mentioned ANOVA; the Friedman test. As the Friedman test only looks at the ranking of results some additional tests are performed. Firstly a pairwise Wilcoxon test is performed to show if the pairwise comparisons are significant. As there are 4 levels a Bonferroni correction is used (0.05/4) to account adequately for Type 1 Errors [55]. Kendall's coefficient is calculated to give a measure of the effect size where 0 means no concordance or agreement at all and 1 is a perfect concordance or agreement [56]. These agreements are among 'judges' in this case social network reconstruction methods, that judge a particular object, in case the ego networks from surveys. When the outcome is 1, the reconstruction methods all perform equally well for each ego network, while values closer to 0 show that the reconstruction methods perform or 'judge' differently for the different ego networks.

As mentioned, to create the model activity network some variables have to be defined. The timespan is the complete history of Rails as found on GitHub and the step size is equal to the window size. The window size should be based on the average time between communications. The 5% trimmed mean of time between communication activities is 4.16 days, the distribution of time between messages is a power law distribution with a mean of 6.15 days. A week is chosen as the time window or sometimes referred to as the time granularity. By choosing a week the majority of average time between activities will be included in a window and establish the appropriate edges for the time window. If a smaller time window is used, for example one day, the majority of nodes that reply to each other taking longer than one day would not be identified as being part of a clique or cluster. If an even larger time window was chosen, for example a month, clusters or cliques could be falsely identified by activities with a time between them that is far above the expected average.

The main aim of the accuracy comparison is to compare the different reconstruction methods, but this also serves another end, which is to answer the question which network probably reflects the whole population most accurately. With a population size of 8,074, 283 respondents and a confidence level of 95%, the margin of error is 5.72% [57], which is higher than the usually highest expectable margin of error of 5%. The population variables, such as the average degree, that are found in the most accurate network, have to interpreted with 5.72% margin of error in mind. The main goal of finding the most accurate network is however not influenced by this margin of error. In the accuracy comparison, the accuracy difference and whether it is significant given the number of surveys is essential and not the estimation of a population variable. The accuracy comparison starts with looking at which network contains most of the edges found in the surveys of Rails members. The accuracy is computed for the measures mentioned in the research methods chapter of this thesis starting with accuracy just below and continuing with true positive rate, true negative rate, precision, g-mean and g-mean$_{2.}$

Edges

Accuracy

The Friedman assumptions are met for all the measures including the current measure of accuracy:

- The group of 283 contributors is a random sample of the total of 8704 contributors.
- The dependent variable is a continuous level variable.
- The groups accuracy measures are measured for more than three different networks.

Hypotheses:

H_0: There is no difference in the mean accuracy of the four different networks. ($\mu_{accuracy\ follow} = \mu_{accuracy\ mutual\ follow} = \mu_{accuracy\ \ sum\ activity} = \mu_{accuracy\ model\ activity}$)

H_a: At least two of the networks have different mean precision rates.

Critical value and rejection region:

The null hypothesis is rejected with p-values ≤ 0.01

Result

The found p-value of $.000 \ \leq 0.01 \ = \propto$, the null hypothesis is rejected.

A Friedman test was conducted to evaluate differences in medians among the

- model activity network accuracy (Mean rank = 3.23),
- sum activity network accuracy (Mean rank = 2.37),
- follow network accuracy (Mean rank = 1.24) and
- mutual follow network accuracy (Mean rank = 3.15).

Test Statistics	
N	283
Kendall's W[a]	.558
Chi-Square	473.876
df	3
Asymp. Sig.	.000
a. Kendall's Coefficient of Concordance	

Table 3

The test was significant χ^2 (3, N = 283) = 473.876, p < .01, and the Kendall's coefficient of concordance of .558 indicated strong concordance among the scores.

Follow-up pairwise comparisons were conducted using a Wilcoxon test and controlling for the Type I errors across theses comparisons at the .05/4 level using the Bonferroni correction. All pairwise comparison are significant except for the accuracy difference between the mutual follow and model activity network.

Test Statistics[a]						
	Sum accuracy – model accuracy	Follow accuracy – model accuracy	Mutual follow accuracy – model accuracy	Follow accuracy – sum accuracy	Mutual follow accuracy – sum accuracy	Mutual follow accuracy – follow accuracy
Z	-12.258[b]	-13.337[b]	-1.014[c]	-12.359[b]	-6.071[c]	-14.191[c]
Asymp. Sig. (2-tailed)	.000	.000	.311	.000	.000	.000
a. Wilcoxon Signed Ranks Test						
b. Based on positive ranks.						
c. Based on negative ranks.						

Table 4

Descriptive Statistics					
	N	Mean	Std. Deviation	Minimum	Maximum
Model accuracy	283	.998943664	.0052411	.92267	1
Sum accuracy	283	.998153645	.0087929	.86661	1
Follow accuracy	283	.992673040	.0165465	.80951	1
Mutual follow accuracy	283	.999425145	.0008344	.99425	1

Table 5

Additional reports about this and following comparison can be found in Appendix C in the same order as presented here. The reports consist of detailed descriptives of the means and the pairwise comparison.

True Positive Rate

A the true positive rate that has no true positive rates results in a division by zero, all networks are interpreted to have a score of 1 when the survey result did not contain any nodes.

Hypotheses:

H_0: There is no difference in the mean true positive rate of the four different networks. $(\mu_{true\ positive\ rate\ follow} = \mu_{true\ positive\ ratemutual\ follow} = \mu_{true\ positive\ rate\ sum\ activity} = \mu_{true\ positive\ ratemodel\ activity})$

H_a: At least two of the networks have different mean true positive rate rates.

Critical value and rejection region:

The null hypothesis is rejected with p-values ≤ 0.01

Result

The found p-value of .000 $\leq 0.01 = \propto$, the null hypothesis is rejected.

A Friedman test was conducted to evaluate differences in medians among the

- model activity network true positive rate (Mean rank = 2.61),
- sum activity network true positive rate (Mean rank = 2.76),
- follow network true positive rate (Mean rank = 2.48) and
- mutual follow network true positive rate (Mean rank = 2.16).

Test Statistics	
N	283
Kendall's W[a]	.121
Chi-Square	102.977
df	3
Asymp. Sig.	.000

a. Kendall's Coefficient of Concordance

Table 6

The test was significant χ^2 (3, N = 283) = 102.977, p < .01, and the Kendall's coefficient of concordance of .121 indicated a weak concordance among the scores. Follow-up pairwise comparisons were conducted using a Wilcoxon test and controlling for the Type I errors across theses comparisons at the .05/4 level using the Bonferroni correction. All pairwise comparison are significant.

Test Statistics[a]

	Sum accuracy – model accuracy	Follow accuracy – model accuracy	Mutual follow accuracy – model accuracy	Follow accuracy – sum accuracy	Mutual follow accuracy – sum accuracy	Mutual follow accuracy – follow accuracy
Z	-4.645[b]	-3.029[c]	-6.576[c]	-4.781[c]	-7.859[c]	-6.028[c]
Asymp. Sig. (2-tailed)	.000	.002	.000	.000	.000	.000

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

c. Based on negative ranks.

Table 7

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum
Model true positive rate	283	.7347	.4173	0	1
Sum true positive rate	283	.7911	.3807	0	1
Follow true positive rate	283	.6486	.4550	0	1
Mutual follow true positive rate	283	.5326	.4900	0	1

Table 8

In order to give a more insightful percentage of correct results the descriptives in Table 9 show the results when surveys with at least one edge are considered. With the Wilcoxon test results it is now possible to conclude that the sum activity network is significantly better at finding true positives given the network found in the survey results.

Descriptive Statistics					
	N	Mean	Std. Deviation	Minimum	Maximum
Model true positive rate	149	.496	.459	0	1
Sum true positive rate	149	.603	.448	0	1
Follow true positive rate	149	.332	.426	0	1
Mutual follow true positive rate	149	.112	.286	0	1

Table 9

True Negative Rate

Hypotheses:

H_0: There is no difference in the mean true negative rate of the four different networks. $(\mu_{true\ negative\ rate\ follow} = \mu_{true\ negative\ ratemutual\ follow} = \mu_{true\ negative\ rate\ sum\ activity} = \mu_{true\ negative\ rate\ model\ activity})$

H_a: At least two of the networks have different mean true negative rate rates.

Critical value and rejection region:

The null hypothesis is rejected with p-values ≤ 0.01

Result

The found p-value of $.000 \leq 0.01 = \propto$, the null hypothesis is rejected.

A Friedman test was conducted to evaluate differences in medians among the

- model activity network true negative rate (Mean rank = 3.22),
- sum activity network true negative rate (Mean rank = 2.33),
- follow network true negative rate (Mean rank = 1.26) and
- mutual follow network true negative rate (Mean rank = 3.20).

Test Statistics	
N	283
Kendall's W[a]	.560
Chi-Square	475.425
df	3
Asymp. Sig.	.000
a. Kendall's Coefficient of Concordance	

Table 10

The test was significant χ^2 (3, N = 283) = 475.425, p < .01, and the Kendall's coefficient of concordance of .560 indicated strong concordance among the scores. Follow-up pairwise comparisons were conducted using a Wilcoxon test and controlling for the Type I errors across theses comparisons at the .05/4 level using the Bonferroni correction. All pairwise comparison are significant.

Test Statistics[a]						
	Sum accuracy – model accuracy	Follow accuracy – model accuracy	Mutual follow accuracy – model accuracy	Follow accuracy – sum accuracy	Mutual follow accuracy – sum accuracy	Mutual follow accuracy – follow accuracy
Z	-12.358[b]	-13.196[b]	-1.865[c]	-12.201[b]	-6.655[c]	-14.217[c]
Asymp. Sig. (2-tailed)	.000	.000	.062	.000	.000	.000
a. Wilcoxon Signed Ranks Test						
b. Based on positive ranks.						
c. Based on negative ranks.						

Table 11

Descriptive Statistics

	N	Mean	Std. Deviation	Minimum	Maximum
Model true negative rate	283	.9990022	.005	.922	1
Sum true negative rate	283	.9981986	.008	.866	1
Follow true negative rate	283	.9927496	.016	.809	1
Mutual follow true negative rate	283	.9995298	.000	.994	1

Table 12

Precision

Hypotheses:

H_0: There is no difference in the mean precision of the four different networks. ($\mu_{precision\ follow} = \mu_{precision\ mutual\ follow} = \mu_{precision\ sum\ activity} = \mu_{precision\ model\ activity}$)

H_a: At least two of the networks have different mean precision rates.

Critical value and rejection region:

The null hypothesis is rejected with p-values ≤ 0.01

Result

The found p-value of $.000 \leq 0.01 = \propto$, the null hypothesis is rejected.

A Friedman test was conducted to evaluate differences in medians among the

- model activity network precision (Mean rank = 2.78),
- sum activity network precision (Mean rank = 2.64),
- follow network precision (Mean rank = 2.34) and
- mutual follow network precision (Mean rank = 2.24).

Test Statistics	
N	283
Kendall's W[a]	.102
Chi-Square	86.382
df	3
Asymp. Sig.	.000
a. Kendall's Coefficient of Concordance	

Table 13

The test was significant χ^2 (3, N = 283) = 86.382, p < .01, and the Kendall's coefficient of concordance of .102 indicated a weak concordance among the scores. Follow-up pairwise comparisons were conducted using a Wilcoxon test and controlling for the Type I errors across theses comparisons at the .05/4 level using the Bonferroni correction. All pairwise comparison are significant expect for the accuracy difference between the mutual follow and follow activity network.

Test Statistics[a]						
	Sum accuracy – model precision	Follow accuracy – model precision	Mutual follow accuracy – model precision	Follow accuracy – sum precision	Mutual follow accuracy – sum precision	Mutual follow accuracy – follow precision
Z	-4.316[b]	-6.986[b]	-5.646[b]	-7.340[b]	-5.672[b]	-.642[c]
Asymp. Sig. (2-tailed)	.000	.000	.000	.000	.000	.521
a. Wilcoxon Signed Ranks Test						
b. Based on positive ranks.						
c. Based on negative ranks.						

Table 14

Descriptive Statistics					
	N	Mean	Std. Deviation	Minimum	Maximum
Model precision	283	.092	.211	1	1
Sum precision	283	.072	.161	1	1
Follow precision	283	.009	.035	1	.5
Mutual follow precision	283	.023	.115	1	1

Table 15

g-mean

Hypotheses:

H_0: There is no difference in the mean g-mean of the four different networks. $(\mu_{g-mean\,follow} = \mu_{g-mean\,mutual\,follow} = \mu_{g-mean\,sum\,activity} = \mu_{g-mean\,model\,activity})$

H_a: At least two of the networks have different mean g-mean rates.

Critical value and rejection region:

The null hypothesis is rejected with p-values ≤ 0.01

Result

The found p-value of .000 $\leq 0.01 = \propto$, the null hypothesis is rejected.

A Friedman test was conducted to evaluate differences in medians among the

- model activity network g-mean (Mean rank = 2.76),
- sum activity network g-mean (Mean rank = 2.66),
- follow network g-mean (Mean rank = 2.35) and
- mutual follow network g-mean (Mean rank = 2.22).

Test Statistics	
N	283
Kendall's W[a]	.101
Chi-Square	85.611
df	3
Asymp. Sig.	.000
a. Kendall's Coefficient of Concordance	

Table 16

The test was significant χ^2 (3, N = 283) = 85.611, p < .01, and the Kendall's coefficient of concordance of .101 indicated strong differences among the scores. Follow-up pairwise comparisons were conducted using a Wilcoxon test and controlling for the Type I errors across theses comparisons at the .05/4 level using the Bonferroni correction. All pairwise comparison are significant expect for the mean difference between the sum and model activity network and the mutual follow and follow network.

Test Statistics[a]						
	Sum g-mean – model g-mean	Follow g-mean – model g-mean	Mutual follow g-mean – model g-mean	Follow g-mean – sum g-mean	Mutual follow g-mean – sum g-mean	Mutual follow g-mean – follow g-mean
Z	-1.987[b]	-6.564[b]	-6.225[b]	-2.974[b]	-6.609[b]	-1.244[b]
Asymp. Sig. (2-tailed)	.047	.000	.000	.003	.000	.214
a. Wilcoxon Signed Ranks Test						
b. Based on positive ranks.						
c. Based on negative ranks.						

Table 17

Descriptive Statistics					
	N	Mean	Std. Deviation	Minimum	Maximum
Model g-mean	283	.140001787423	.2483889651240	0	1
Sum g mean	283	.135780983467	.2219143945380	0	1
Follow g mean	283	.034343584839	.0766898921253	0	.5
Mutual follow g mean	283	.032578786639	.1288158604345	0	1

Table 18

g-mean$_2$

Hypotheses:

H_0: There is no difference in the mean g-mean 2 of the four different networks. $(\mu_{g-mean\,2\,follow} =$

$\mu_{g-mean\ 2\ mutual\ follow} = \mu_{\ g-mean\ 2\ sum\ activity} = \mu_{g-mean\ 2\ model\ activity})$
H_a: At least two of the networks have different mean g-mean 2 rates.

<u>Critical value and rejection region:</u>
The null hypothesis is rejected with p-values ≤ 0.01

<u>Result</u>
The found p-value of .000 $\leq 0.01 = \propto$, the null hypothesis is rejected.

A Friedman test was conducted to evaluate differences in medians among the

- model activity network g-mean (Mean rank = 2.75),
- sum activity network g-mean (Mean rank = 2.67),
- follow network g-mean (Mean rank = 2.39) and
- mutual follow network g-mean (Mean rank = 2.19).

Test Statistics	
N	283
Kendall's W[a]	.100
Chi-Square	85.274
df	3
Asymp. Sig.	.000
a. Kendall's Coefficient of Concordance	

Table 19

The test was significant χ^2 (3, N = 283) = 85.274, p < .01, and the Kendall's coefficient of concordance of .100 indicated weak concordance among the scores. Follow-up pairwise comparisons were conducted using a Wilcoxon test and controlling for the Type I errors across theses comparisons at the .05/4 level using the Bonferroni correction. All pairwise comparison are significant expect for the accuracy difference between the mutual follow and model activity network.

Test Statistics[a]						
	Sum g-mean$_2$– model g-mean$_2$	Follow g-mean$_2$– model g-mean$_2$	Mutual follow g-mean$_2$– model g-mean$_2$	Follow g-mean$_2$– sum g-mean$_2$	Mutual follow g-mean$_2$– sum g-mean$_2$	Mutual follow g-mean$_2$– follow g-mean$_2$
Z	-.155[b]	-3.930[c]	-6.551[c]	-5.454[c]	-7.619[c]	-5.932[c]
Asymp. Sig. (2-tailed)	.877	.000	.000	.000	.000	.000
a. Wilcoxon Signed Ranks Test						
b. Based on positive ranks.						
c. Based on negative ranks.						

Table 20

Descriptive Statistics					
	N	Mean	Std. Deviation	Minimum	Maximum
Model g-mean$_2$	283	.278	.428	0	1
Sum g-mean$_2$	283	.333	.453	0	1
Follow g-mean$_2$	283	.193	.368	0	.99
Follow follow g-mean$_2$	283	.068	.233	0	1

Table 21

Ranking

<u>Real Positive Ranking Score</u>

As was mentioned before, comparing to the actual sum network will always favor the model network. The real networks are relatively small, and with the large number of edges in the sum network, the model network will by design be more accurate. To create an appropriate test, the true positive edges in the activity network are selected as the complete network. For these edges the sum of activities can be calculated as edge weight and the model can be applied to calculate the edge weight. These two sets of ranking can now be compared to the ranking in the surveys by determining the absolute difference in rankings for each survey and each network. A significant lower ranking error score for any of the two networks will show which edge weight better reflects the relative collaboration strength.

The Wilcoxon signed-rank test is used to test the difference, the assumptions of the test are met:

- The data of the paired results are form the same population of survey subjects.
- Each pair was randomly picked from the complete set of 8704 nodes.
- The ranks measure is at least on an ordinal scale.

Hypotheses:
H_0: There is no difference in the mean true ranking score of the sum activity and model activity networks. $(\mu_{true\ ranking\ score\ sum\ activity} = \mu_{true\ ranking\ score\ model\ activity})$
H_a: The mean true ranking score of the sum activity and model activity networks are different. $(\mu_{true\ ranking\ score\ sum\ activity} \neq \mu_{true\ ranking\ score\ model\ activity})$

Critical value and rejection region:
The null hypothesis is rejected with p-values ≤ 0.05

Result
The found p-value of $.046 \leq 0.05 = \propto$, the null hypothesis is rejected.

	N	Mean	Std. Deviation	Minimum	Maximum
Model true ranking error	50	3.70	4.921	0	31
Filtered sum true ranking error	50	4.02	5.180	0	33

Table 22

Test Statistics[a]	
	Filtered sum true ranking error - Model true ranking error
Z	-1.999[b]
Asymp. Sig. (2-tailed)	.046
a. Wilcoxon Signed Ranks Test	
b. Based on negative ranks.	

Table 23

The effect size of the test is $r = \left|\frac{-1.999}{\sqrt{50}}\right| = .28$, which is generally considered small.

Explorative Analysis

In order to make the previously presented accuracy comparison, a dataset of network activities, follow relations and survey results was created. Parts of this dataset can be used to accurately recreate the research of Bird et al. [3] and to continue the research of Crowston & Howison [29]. Additionally the data can be used to give a detailed description of the history and structure of Rails. The explorative analysis starts with the descriptives of network activities, followed by descriptives of the survey results, the four reconstructed social networks and a limited analysis of the dynamic network of Rails. The goal of this additional analysis is to strengthen and replicate related research and to show that SNA of OSS project networks can yield fundamental results as was claimed in the introduction of this thesis.

Network Activities

The network activity data starts with the first commit of David Heinemeier Hansson on November 24, 2004. The last network activity considered in this analysis started on May 15, 2014. In total 8,704 distinct GitHub users were found related to Rails of which 74 were members of the Rails organization on GitHub. Each of these users has either authored a commit, send a pull-request, opened an issue or posted a comment. In total 49,486 commits were found of which 1,938 (3.9%) referred to a non-existing author. It is likely that these authors deleted their account since the time the commits were created. These commits and their authors are assumed to be missing at random. In total 135,807 activities were found of which 25,439 (18.7%) contained @ mentions.

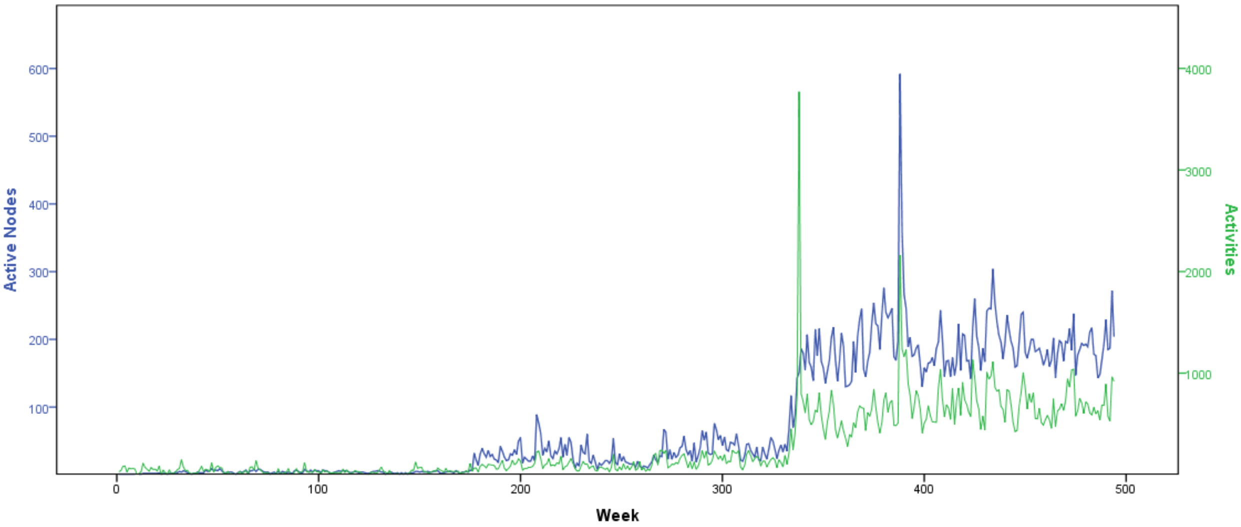


Figure 7 | The number of active nodes and activities during the almost 500 week history of Rails on GitHub. The values represent the sum of active nodes and the sum of activities during a single week.

The first substantial increase in activity and active nodes occurs near week 200, as can be seen in Figure 77, this coincides with the 2.2 release of Rails on November 21, 2008. The next substantial increase occurs near week 350 which coincides with release 3.1 on August 31, 2011. The high peak of activities and active nodes before week 400 occurs just before the 3.2 release on January 20, 2012. The last two years show a rather stable amount of active nodes and activities, where on a weekly basis nearly 200 people are active performing around 600 activities per week.

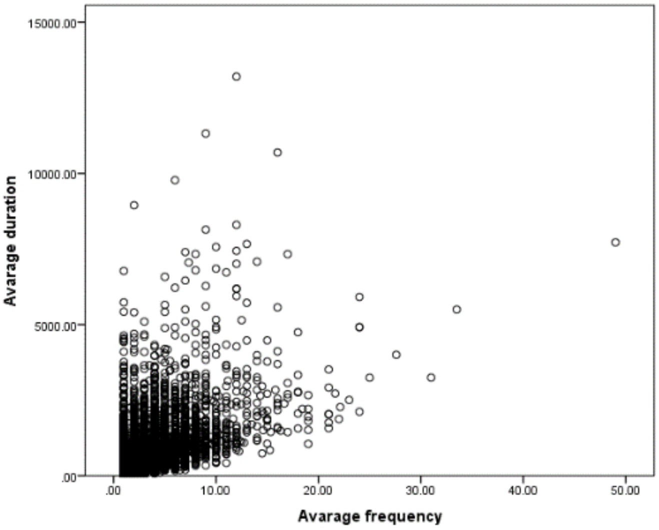


Figure 8 | The relation between average frequency and average duration of activities.

As mentioned before 135,807 activities were found, each with a particular length, source, target and timestamp. Using the sources and targets an adjacency matrix was created with 31,542 edges, 25,978 when self-loops are excluded. For each of the edges the average lengths and frequencies of the messages were calculated and the number of distinct months during which these edges have been used. The relation between average frequency and average duration (Figure 8) shows the properties that were expected. The scatterplot shows diverging lines, where the communications are either relatively shorter and more frequent or relatively longer and infrequent. Not surprisingly the occurrence of high average durations combined with high average frequency are absent. This relation is only strong for edges that have relatively high frequencies and durations. For the majority of edges there is a positive 2-tailed linear Pearson correlation of .604 significant at the .01 level between average frequency and average duration. To summarize, for edges with an above average frequency and duration the negative correlation found by Marsden & Campbell [16] is replicated, for the majority of edges there is however a positive correlation between average frequency and average duration.

Distributions of Personal Metrics

In the survey subjects are asked about a limited set of personal metrics: the collaborative relations they have (degree), the time have approximately spend working on Rails and their future commitment to Rails. For the first two metrics related literature [40] [3] [29] [4] suggests their answers should, when combined, be distributed by a power law. Before looking at the actual results from the survey, it is possible to retrieve the same metrics from modeled GitHub data including additional metrics. In Figures 9, 10 and 11 the distribution of degree, active months and commits are shown on a logarithmic scale. Al have a power law distribution as expected, two relevant questions arise about these distribution. Firstly are these distribution similar for members and non-members of the Rails organization, secondly are the distributions related, e.g. do individuals with a low number of communication partners also have low values for distinct active months and number of commits?

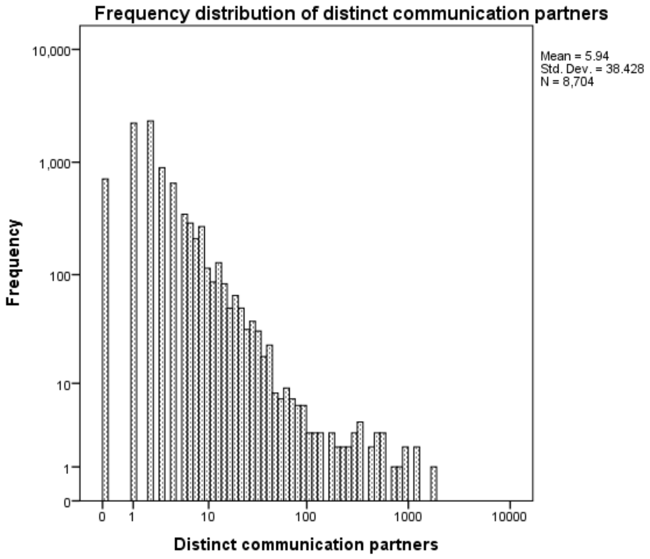


Figure 9 | Degree distribution for Rails population, shown as number of distinct communication partners as found in the communication dataset of Rails, shown on a logarithmic scale.

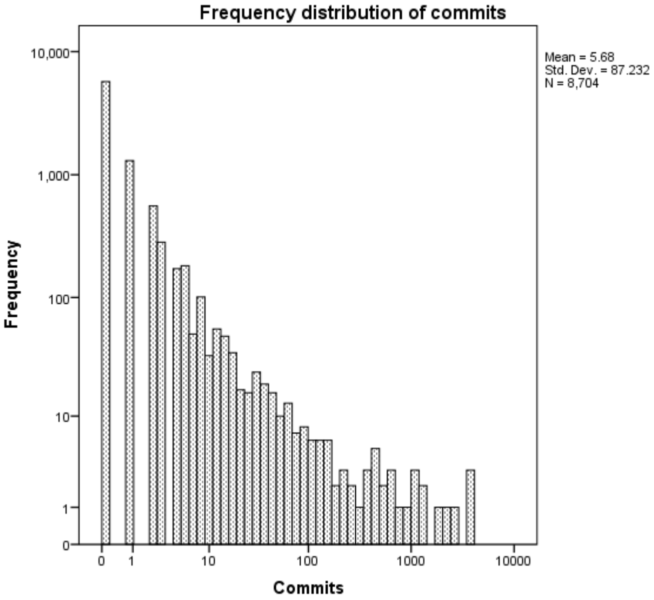


Figure 10 | Commits distribution for Rails population, on logarithmic scale.

Rails members have more administrative rights and most have been invited to become member based on merits and past efforts. It is therefore likely that Rails members should have different distributions of performed work and time spend working on Rails. When the results are logarithmically transformed they indeed seem to have a different distribution, the well-known normal distribution as the results in Table 24 show. The Rails members are a far smaller group, the Shapiro-Wilk test is therefore used to determine the significance while for the larger group of nonmembers the Kolmogorov-Smirnov test is used. The Rails members tests are not significant, the hypothesis that the distributions are normal can therefore not be rejected. Normal distribution for members are found for the number of communication partners, active months and other metrics. These results do not contradict the findings of related research about power law distributions, they however highlight that it is possible that the overall population contains well defined groups of users that have normal distributions for important metrics.

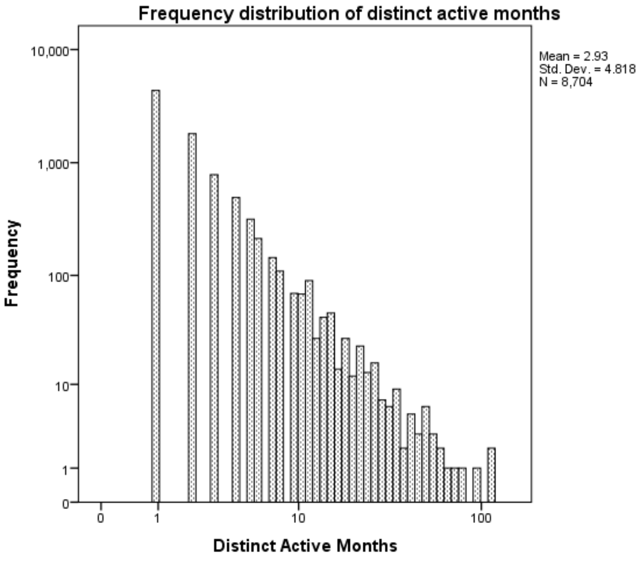


Figure 11 | Active months distribution for Rails population, on logarithmic scale.

In order to answer the second question about relations between the distributions an extensive correlation of different metrics is needed. Some of these metrics need to be calculated based on the different reconstructed social networks. Before the correlation is discussed the descriptives and the characteristics of the different reconstructed networks are therefore discussed.

Tests of Normality						
	Rails Nonmembers Kolmogorov-Smirnov[a]			**Rails Members** Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Log10 sum activity degree	.184	8633	.000	.971	71	.099
Log10 active months	.276	8633	.000	.971	71	.106
Log10 follow in-degree	.243	8633	.000	.971	71	.106
Log10 total messages send and received	.148	8633	.000	.968	71	.070
Log10 total commits	.375	8633	.000	.968	71	.068
*. This is a lower bound of the true significance.						
a. Lilliefors Significance Correction						

Table 24 | Normality tests for different properties for the distinct groups of members and nonmembers.

Networks

Before the individual explorative analysis is presented of the four networks, a visual overview of the networks is shown Figure 10, Figure 11, Figure 12 and Figure 13. The networks are created using Gephi and the OpenOrd layout plugin. After the layout the NoOverlap layout is used to put a margin between the nodes for clarity. Each of these figures only shows the giant component of the network. The colored nodes in the graphs represent the nodes that are official members of the Rails organization on GitHub.

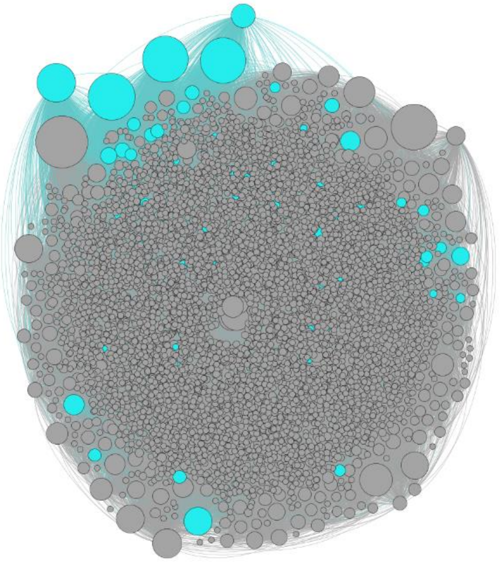


Figure 12 | Follow giant component network of the complete history of Rails. Nodes = 6070, edges = 48830.

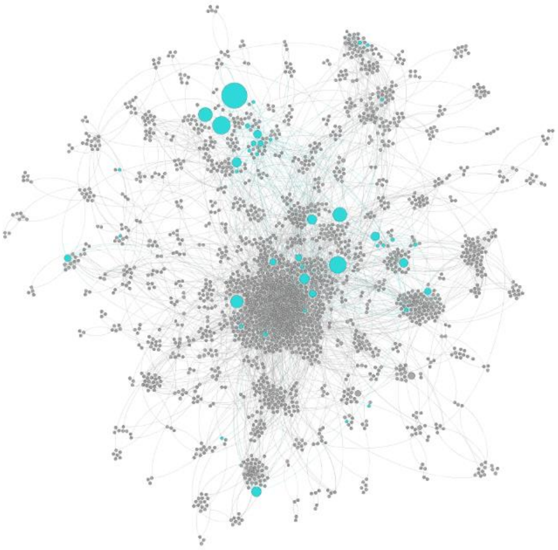


Figure 13 | Mutual follow giant component network of the complete history of Rails. Nodes = 2162, edges = 7408.

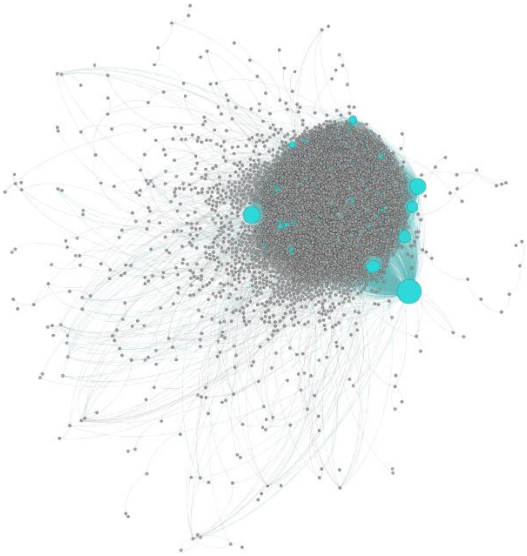


Figure 14 | Sum activity giant component network of the complete history of Rails. Nodes = 7958, edges = 25851.

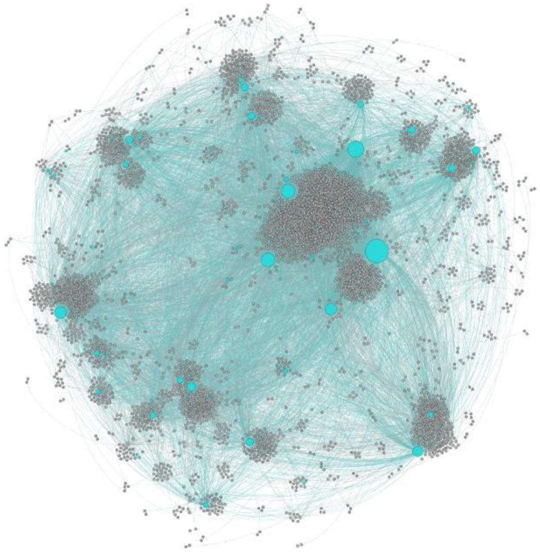


Figure 15 | Model activity giant component network of the complete history of Rails. Nodes = 5100, edges = 13241.

	Follow Network	Mutual Follow Network	Sum Activity Network	Model Activity Network
Nodes	8704	8704	8704	8704
Edges	48872	8212	25978	13348
Average Clustering Coefficient	0.072	0.041	0.458 (weighted)	0.518 (weighted)
Average Degree	5.615	0.943	5.969	3.067
Diameter	14	17	16	11

Table 25 | The descriptives of the four different networks created using GitHub data about Rails.

Follow Network

The main difference between the activity and the follow networks is that the follow networks are directed. Most notably the clustering coefficient drops sharply for the follow networks, suggesting that in a follow network the nodes are part of fewer triangles. The diameter (maximum distance between all pairs of nodes) of the follow network is much greater than the model activity network, even with more than double the amount of edges.

Mutual Follow Network

The mutual follow network is for many metrics the weakest of the networks, the average clustering coefficient shows the almost complete absence of triangles and the diameter shows large distances within the network. The power law degree distribution is however preserved and the presence of hubs still shows the characteristics of a real-world network.

Sum activity network

The 5% trimmed degree mean of the sum activity network is 3.01 (full descriptives in Appendix B5), suggesting that adjusted for high and low outliers, on average contributors to the project will be approached by or send messages to 3 other people in the project. As the arbitrarily selected distribution in Figure 16 shows, even when bi-modal distribution is found on a single day. Even when the group of degree on the left side of the graph is singled out for analyses, a non-normal and skewed distribution is found.

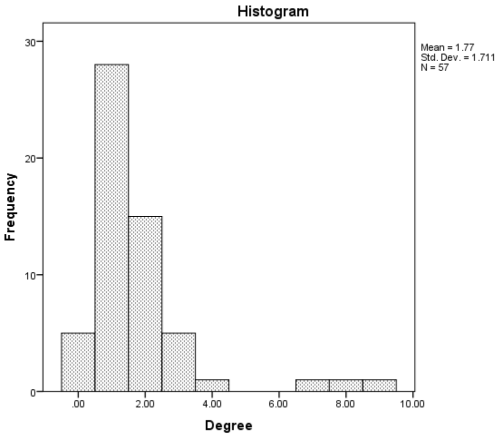


Figure 16 | The degree distribution of users recorded on the 1 day of the 450st week of the Rails project.

Model Activity Network

With Rm = 0.000025, 8704 nodes and 13348 edges, the model activity network reduces the amount of edge in the network roughly by a half compared to the sum activity network. The average degree is much lower than the sum activity network, but is in some respects more reasonable as will be shown in the explorative analysis of the survey results. The increase in clustering stems from the different weighing approach of edges. The network graph visually shows a structure that corresponds closely to what Crowston & Howison (2005) expected, union like structures with central nodes. The choice of Rm value is based on the power law distribution of edge weights, a large number of very weak relations is present in the network. By removing a substantial amount of nodes it will become clear if the technique of removing weak nodes is accurate or not, the choice was to remove about half of the nodes. The value of 0.000025 achieves this.

Comparison of Activity and Follow Networks

Some of the general characteristics of the two network approaches (activity or follow relations) seems to be the same, for example the power laws that describe the degree of the nodes in the networks. To explore these differences and resemblances in more detail an extensive correlation is made of different attributes of the network nodes.

	Activity length	Active months	Messages out	Messages in	Commits	Follow In-Degree	Mutual Follow Degree	Sum Activity Weighted Degree	Model Activity Weighted Degree
Activity length	1.000	.562	.937	.933	.784	.374	.125	.915	.891
Active months	.562	1.000	.559	.583	.610	.459	.233	.459	.538
Messages out	.937	.559	1.000	.990	.864	.397	.142	.955	.889
Messages in	.933	.583	.990	1.000	.911	.437	.140	.920	.888
Commits	.784	.610	.864	.911	1.000	.505	.133	.681	.721
Follow In-Degree	.374	.459	.397	.437	.505	1.000	.403	.293	.344
Mutual Follow Degree	.125	.233	.142	.140	.133	.403	1.000	.131	.116
Sum Activity Weighted Degree	.915	.459	.955	.920	.681	.293	.131	1.000	.891
Model Activity Weighted Degree	.891	.538	.889	.888	.721	.344	.116	.891	1.000

Table 26 | The Pearson correlations of different network node attributes of both the follow and activity networks. All correlations are significant at the 0.01 level (2-tailed).

Description of the variables mentioned in Table 26:

- Activity length: The sum of all activity lengths related to a user.
- Active months: The number of distinct months wherein an user has at least performed one activity.
- Messages out: The total number of messages send by a user.
- Messages in: The total number of messages received by a user.
- Commits: The number of accepted commits in the master branch of Rails by a user.
- Follow in-degree: The number of followers of a user.
- Mutual follow degree: The number of mutual follow relations of a user
- Model activity weighted degree: The sum of weights of the edges of a user in the sum activity network.
- Sum activity weighted degree: The sum of weights of the edges of a user in the model activity network found by applying the model in Eq. (8).

Although lengthy, the correlations show some important relations:

- The large group of users which is active during a small number of distinct months has simultaneously a small number of commits, messages and communication partners. The correlation are significant but not all are strong, for example the correlation of .61 between commits and number of active months. Other relation for example between number of commits authored and number of messages received is much stronger with a value of .911 suggesting that people who do not author commits are very unlikely to receive any messages.

- Follow relations are described by GitHub as ‘Pick a Friend’, from the correlations it seems that indeed the establishment of many mutual follow relations has not much to do with the actual contributions or time spend working on a project, although the time spend working on a project has the highest value of .233.
- Numbers of followers has the highest correlation with the numbers of commits with a value of .505. This has implication for the research of Bird et al. [3] which will be discussed in more detail later.

Survey Results

The distributions found in the Rails dataset contain the expected power law distributions for number of active months, commits and other metrics. The question is whether the survey results also contains these distributions. Before discussing the distributions the descriptives results of the survey are presented. Not all contributors share their email addresses openly, only a set of 2331 contributors were send a survey request via email. 283 (12%) replies were received of which 283 were valid and ready for use. As discussed before, this sample is too small to make confident generalizations about the whole Rails population. With a population size of 8,074, 283 respondents and a confidence level of 95%, the margin of error is 5.72% [57], which is higher than the usually highest expectable margin of error of 5%.

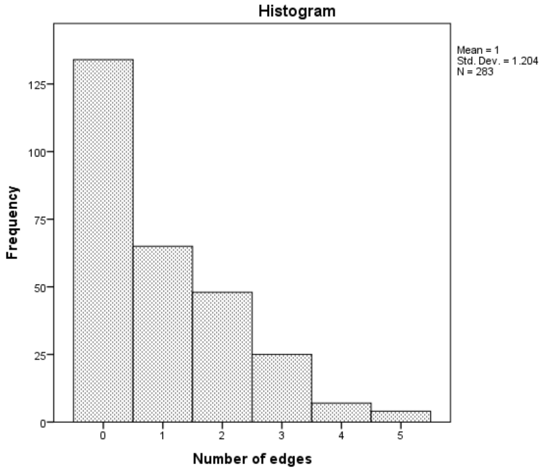


Figure 17 | Degree distribution found in survey results.

A set of subjects commented via email with a similar problems, they did not feel like they had collaborated with anyone or their commits were few and made a long time ago and they did not remember the details. This is exactly in line with the expectation of a power law distribution of the amount of work performed. Some of the (anonymized) responses via email are shown in Appendix B3. The distribution of degree (Figure 17) reflects the earlier power law distributions found in the network activity dataset. The time spend working on Rails distribution in Figure 18 also shows a power law distribution. As the seven different Likert-scale answers of the survey itself contained increasingly larger answers, the visual representation does not clearly reflect the power law distribution.

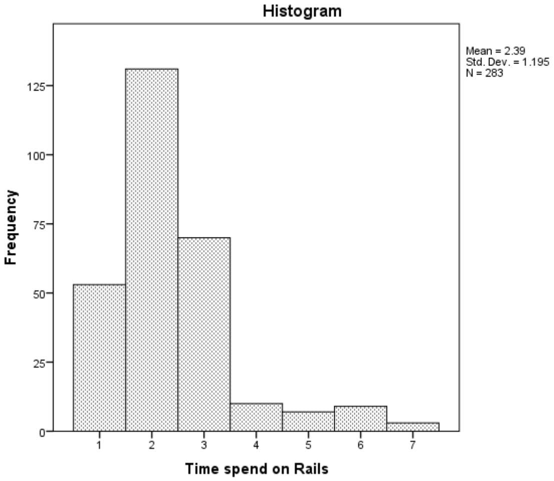


Figure 18 | Time spend on Rails distribution found in survey results.

In a crude correlation shown in Table 27 a Spearman’s rho correlation is used as it is expected that nodes with many connections will tend to leave out increasingly more connections in the survey. This effect as discussed before in the accuracy part of the research methods chapter is better approximated by the monotonic relation handled by the Spearman correlation instead of a linear correlation.

		Survey Result	Sum Activity Network	Model Activity Network	Follow Network	Mutual Follow Network
Survey Result	Spearman’s rho	1	.386**	.398**	.265**	.132**
	Sig. (2-tailed)		.000	.000	.000	.026
**. Correlation is significant at the 0.01 level (2-tailed).						

Table 27 | The number of edges in the survey correlated with the number of edges in the different networks for the same survey subject.

The results in Table 27 suggest that the activity networks more accurately resemble the degree of nodes without taking into account the correctness of the found edges. By summing over the complete history of distinct weeks of the projects it is possible to count the number of weeks during which a particular user has been active. When these sums are correlated to the results of the Likert-scale ‘time spend on project’ question from the survey a significant but weak Pearson correlation (a=0.01) of 0.466 is found (Table 28).

		Time	Weeks	Activities	Activity length sum	Commits
Time	Pearson Correlation	1	.466**	.322**	.326**	.441**
	Sig. (2-tailed)		.000	.000	.000	.000
**. Correlation is significant at the 0.01 level (2-tailed).						

Table 28 | The correlation of the Likert-scale results of the ‘time spend on project’ question of the survey with the active weeks as found in the activity dataset.

This correlation suggests that the sum of distinct weeks during which an actor performed at least one activity has a relationship, although a weak one, with the relative time spend on a project as determined by the actor in the survey.

The survey also included a question about who the most important ‘managers’ were of the Rails network, in total 16 users were at least mentioned twice. This list of users was ranked and the absolute difference in rank was calculated compared to the ranking taken from the four different networks. Table 29 shows that the activity sum ranking was the most accurate in resembling the ranking of the survey results. The ranking is based on the absolute difference in degree for the follow networks and the weighted degree of the activity networks.

Descriptive Statistics						
	N	Minimum	Maximum	Sum	Mean	Std. Deviation
Sum activity network	16	0	375	493	30.81	92.451
model activity network	16	0	771	892	55.75	191.160
Follow network	16	1	547	1253	78.31	133.142
Mutual follow network	16	9	2793	14292	893.25	1189.013

Table 29 | The absolute rank difference between the survey results and four networks of most important managers according to survey subjects.

More about the histograms and distributions of the answers of the survey can be found in Appendix B8.

Recreation of Bird et al. [3]

In ‘Mining Email Social Networks’ by Bird et al. [3] the OSS server Apache email archives are mined to answer questions about social status and the relation between email activity and commit activity. In this research some similar assumptions are made about the relation of communication and social structures. Although this thesis uses data from a different OSS project and the communications are not emails but mostly comments, the basic outset is the same. Given an OSS project with its communication data, how can the data be used to reconstruct the social network of the project and what do the parameters of the reconstructed network tell us about the project?

The Apache email archive data revealed (scale-free) power law distributions for number of metrics, for example messages send, out-degree and in-degree. The communications of Rails also reveal a same real-world or scale-free network. The biggest differences between for example the frequency of the number of messages in Figure 19 of Rails compared to the results of Bird et al. is the relatively high amount of nodes that have send 5 or less messages. The same is true for the degree of unique actors the nodes have communicated with. Bird et al. find power laws both for in-degree and out-degree, the degree results (Figure 20) of Rails show the same phenomenon as with the number of messages, a relatively large number of people have a low degree.

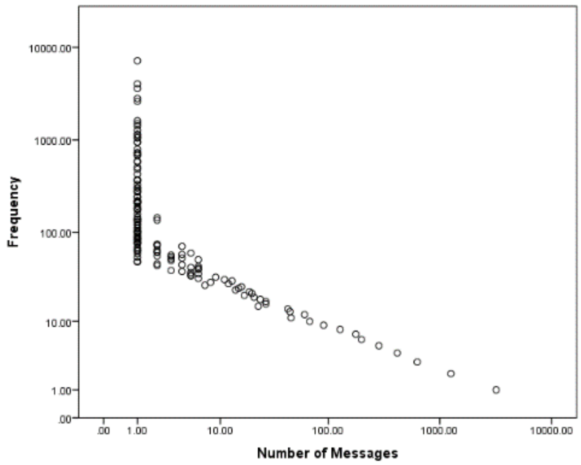


Figure 19 | A recreation of the combination of graphs in Figure 1 of Bird et al. [3] on a logarithmic scale.

Bird et al. observe: ‘*There is a strong relationship between the number of messages sent by an individual, and the number of distinct individuals who respond to that individual ...*’. Also in the Rails data a significant Pearson correlation of .959 is found between the number of messages send and the number of unique correspondents of an actor.

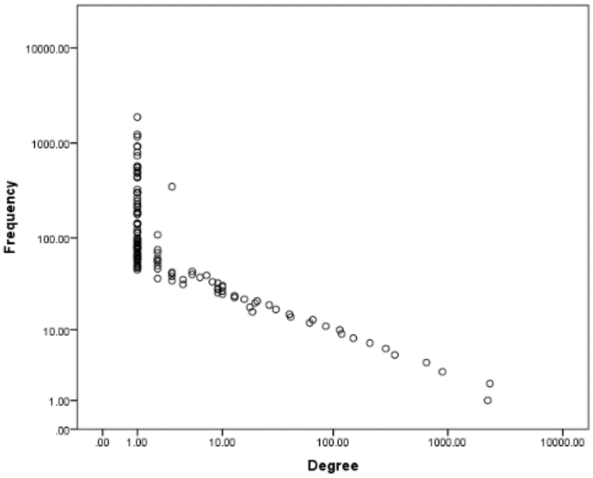


Figure 20 | A recreation of the graph in Figure 1 of Bird et al. [3] on a logarithmic scale.

Bird et al. continue by creating a pruned social network of the most connected nodes in the network of directed emails. When two nodes in the network have at least exchanged 150 emails an edge is introduced between them. This results in a network of 73 nodes. All the nodes included in this sub graph have made at least one source code change. In order to approximate this setting the sum activity network is pruned for edges that have at least exchanged 25 messages, and nodes are only considered when they have at least authored one accepted commit. This leads to a subset of 112 nodes with 312 edges. Where Bird et al. distinguishes between source code changes and document changes, here only commits are considered. There are also some other differences, where Bird et al. have threshold of 150 messages here a minimum of 25 is used. Furthermore a message on GitHub is different in a lot of ways compared to a directed email.

The final network used for determining the betweenness centrality is therefore an undirected one. The results of the correlation and shown in Table 30:

			# Commits	Messages out	Messages in	Follow In Degree	Betweenness Centrality
Spearman's rho	# Commits	Correlation Coefficient	1.000	.323**	.328**	.124	.350**
		Sig. (2-tailed)		.001	.000	.191	.000
	Messages out	Correlation Coefficient	.323**	1.000	.976**	.435**	.725**
		Sig. (2-tailed)	.001		.000	.000	.000
	Messages in	Correlation Coefficient	.328**	.976**	1.000	.477**	.732**
		Sig. (2-tailed)	.000	.000		.000	.000
	Follow In Degree	Correlation Coefficient	.124	.435**	.477**	1.000	.446**
		Sig. (2-tailed)	.191	.000	.000		.000
	Betweenness Centrality	Correlation Coefficient	.350**	.725**	.732**	.446**	1.000
		Sig. (2-tailed)	.000	.000	.000	.000	
**. Correlation is significant at the 0.01 level (2-tailed).							

Table 30 | Correlations of the 112 contributor nodes that have at least on edge over which 25 or more messages were communicated.

The betweenness centrality Spearman's rho correlation of Bird et al. of .757 is in this replication a significant value is found, but much lower: .350. But the betweenness centrality has the strongest correlation of the attributes, most notably, the number of followers (Follow In Degree) correlation with the number of commits is not significant. The correlation of the whole population in Table 26 showed a correlation of .505 between the number of followers and the number of commits by a user. This value is higher than found in the recreation and is based on the whole population, still the correlation cannot be considered to be strong. It is however easier to argue that the number of followers of a user is a sign of social status, compared to the betweenness centrality of a user in a filtered network of highly active users. This .505 correlation of followers and commits is therefore a more interesting result than the more exact replication shown in Table 26.

Continuation of Crowston & Howison [29]

In their 2005 paper titled ‘The social structure of free and open source software development’ Crowston & Howison reconstruct social networks by looking at interactions in bug tracking systems of OSS projects. The size of a project (number of nodes) is correlated with the out degree of the nodes. A significant (a=0.01) correlation of -.39 is found, suggesting that larger projects have declining out degree centrality. Out-degree is described as ‘*number of interactions sent*’, the interpretation is however unclear as it can be interpreted as the sum of out-degree or the number of unique out-degree edges or even something else entirely. About the correlation between out-degree and project size Crowston & Howison note ‘*As projects grow, they have to become more modular, with different people responsible for different modules. In other words, a large project might be an aggregate of smaller projects …*’. Leaving aside what this ‘modularity’ should precisely look like, a prediction is made that as a project grows, it is likely to have multiple modules with at the center of these modules hubs that have a disproportionate high degree. In order to give a possible answer to the question of size and modularity a dynamic network is needed. As mentioned in the related literature, there is no standard approach for reconstructing dynamic networks of OSS projects. The familiar weekly time

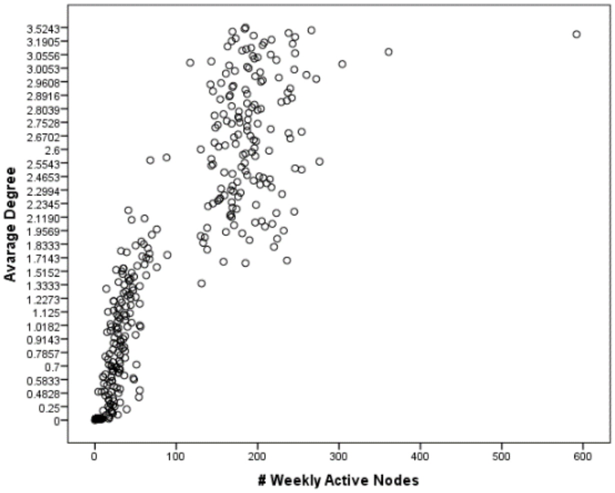


Figure 21 | The number of # active nodes during a week of Rails activity plotted against the average degree of the nodes during the week.

window is therefore used, with a step size of a weak and as timespan the complete history of Rails on GitHub[3]. Crowston & Howison use unfiltered bug reports to reconstruct social networks, the network activity data is most similar to these reports and is therefore used to reconstruct the social networks over time. Crowston & Howison's Figure 11 shows the out-degree plotted against the project size of different project. Leaving the interpretation of out-degree aside, a similar plot is created (Figure 21) but instead of out-degree the average degree of nodes is used. As the number of active nodes grows the average degree also tends to grow, the Spearman's rho correlation has a significant value of .938 (a=0.01). If larger projects are comprised of smaller unconnected sub graphs with internally roughly the same structure, the average degree should also stay roughly the same as the network grows. This is not the case, this first indication therefore seems to contradict the prediction of Crowston & Howison.

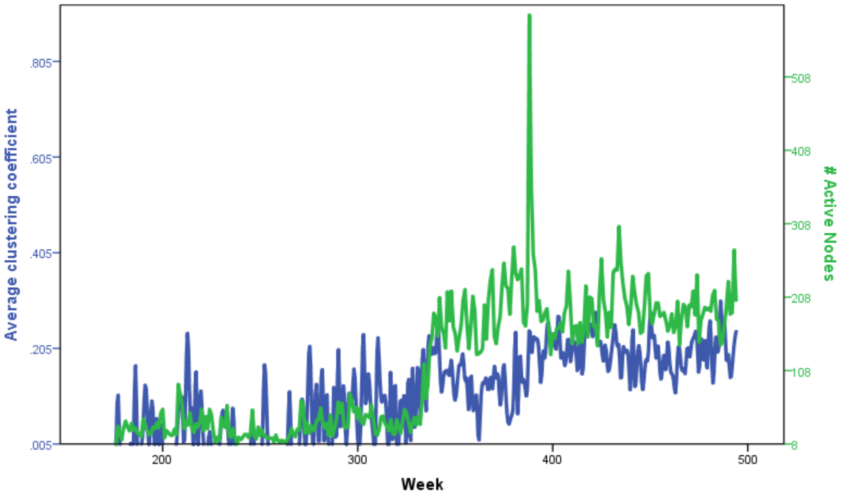


Figure 22 | The number of active nodes during the near 500 weeks of Rails activity plotted against the average clustering coefficient of the nodes during the week.

There are however more sophisticated ways of determining modularity in projects than degree, as discussed in the related literature of this thesis. These other ways include the changes of time in cliques, clusters and overlap. By looking at the average cluster coefficient of nodes over time, the prevalence of triangles is interpreted as a measure for modularity in the network. If the prediction by Crowston & Howison is indeed correct, the expected cluster coefficient could vary during periods of rapid growth or decline, but should otherwise be relatively stable. At the same time the cluster coefficient should not increase significantly when for example the number of active nodes changes from 150 to 200. As the active nodes would be spread over multiple sub graphs where the clustering is assumed to be roughly the same in each sub graph. The clustering coefficient could however grow as communication between the sub graphs

[3] Two animations of the history of Rails created as part of this thesis can be watched online. A first animation shows the relations over the nearly 10 year history of Rails members: https://www.youtube.com/watch?v=68zMRG85paI. A second animation shows the whole population of Rails for the same timespan where Rails members are depicted by blue nodes and non-members as grey nodes: https://www.youtube.com/watch?v=IEBEAlPpT-8

might be required, the number of isolated nodes could also grow, in turn negatively influencing the clustering coefficient. The graph in Figure 22 shows the number of active nodes and the average clustering coefficient found in the activity networks during the nearly 500 weeks of Rails activity. The Spearman's rho correlation of average clustering coefficient and number of active nodes during the weeks has a significant value of .843 (a=0.01). Again suggesting that as the network grows, the connections seem to grow linearly with it.

By looking at cliques the expectation is that collaborations in OSS projects manifest as connected groups of nodes. If this is the correct way to interpret modularity, then as the amount of active nodes grows the size of cliques should also grow and reach a critical size. After reaching this critical size the cliques should on the whole stop growing and in turn the number of cliques should grow. The alternative should be the unstopped growth of a single or low number of cliques that connect the majority of all nodes. The results of Figure 23 show that there is indeed a limit to the size of the size of cliques as found in the sum activity network. The quadratic function describes the relation between the number of active nodes and the size of cliques significantly better than a linear function. The results are calculated using the activity during the near 500 weeks of Rails activity. For each week the number of active nodes is recorded with the maximum size of the largest clique. The extended results of this analysis can be found in Appendix B4.

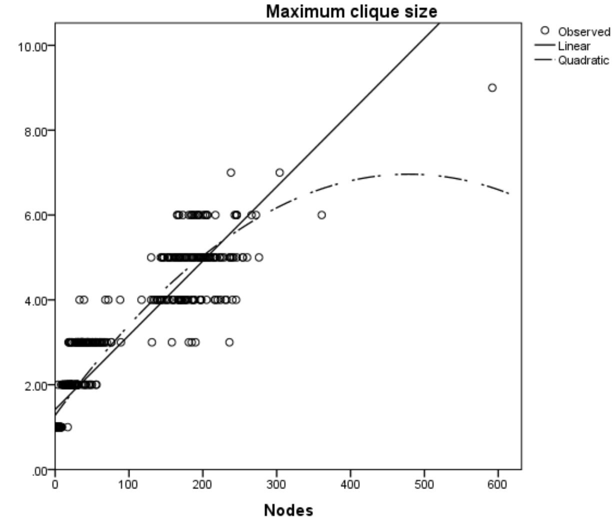


Figure 23 | The number of active nodes plotted against the maximum clique size is significantly more accurately described by a quadratic function compared to a linear

By looking at the degree distribution of the network over time, it is expected that as the network grows, the amount of hubs should also grow. For each week the number of nodes with a disproportionally high degree are counted, this high degree is defined as a degree that is at least five time as high as the average degree. As can be seen in Figure 24 the Pearson correlation of the number of active nodes with the number of nodes with a disproportionally high degree is significant and has a value of .951. The linear regression of the data shows that for roughly every 30 nodes there is a node with a degree five times as high as the average degree. The results of this regression can be found in Appendix B1.

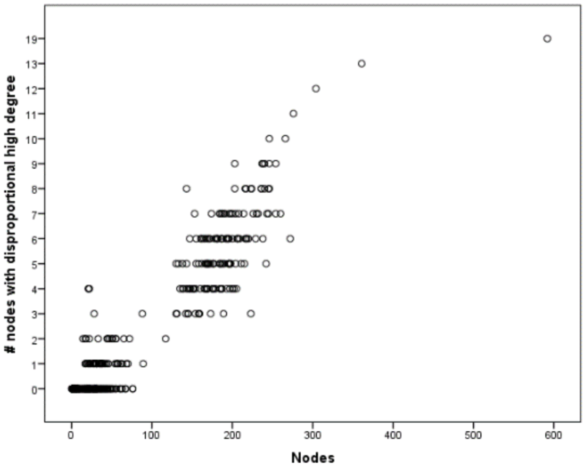


Figure 24 | The number of nodes with a disproportionally high degree plotted against the number of active nodes during a given week.

Overall all the activity network data combined over time seems to suggest that as the amount of active nodes grows, the density also grows, but as Crowston & Howison predicted the number of hubs grows linearly with it. But as the nodes start to connect more, they do not keep making ever bigger cliques. The maximum size of cliques seems to peak just after 6 connected nodes.

Another measure of edge weights and community structures is the overlap of edges as discussed in Eq. (1). The complete network history shows that the group of nonmembers has a non-linear relations between overlap and edge weight. The group of members has a significant positive linear relation (Figure 25), which suggests that in the communities of members the edges within communities are strong compared to edges between communities. The extended results of this analysis can be found in Appendix B2.

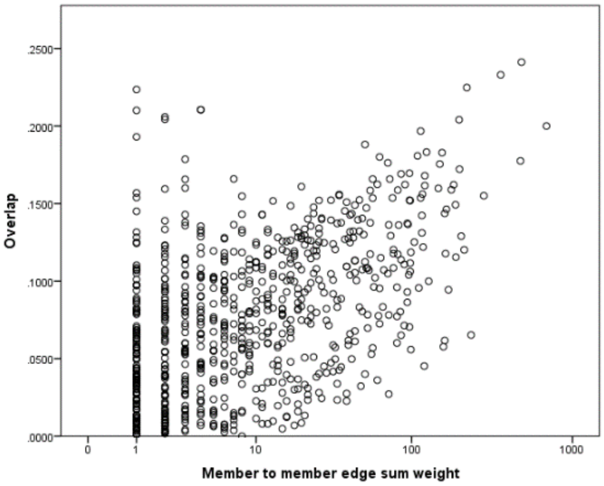


Figure 25 | The values of edge weights determined by the sum of activities against the overlap of the nodes connected by the edge on a logarithmic scale.

To investigate the actual communities the k-clique-communities algorithms is used on the weekly sum networks. With k=4, two things become clear, firstly the number of active nodes is correlated with the number cliques and as time progresses especially after week 350 the number of cliques keeps rising. The linear regression of k-clique-communities (k=4) with the number of weekly active nodes shows that roughly with 50 active nodes one new k-clique-community emerges. This results with more details can be found in Appendix B7.

The last measure of modularity discussed in this analysis is the lack of multiple substantially large disconnected components. During the complete history of Rails, on a weekly basis the active nodes communicated with a large number of other active nodes. The strong correlation between active nodes and maximum size of connected components can be seen in Figure 26 (.983 Pearson corr., significant at a=0.01 level (2-tailed)) . The idea that multiple disconnected groups of developers are working without communicating, even if the communication was weak is incorrect. Rails is therefore in terms of Crowston & Howison a very centralized project.

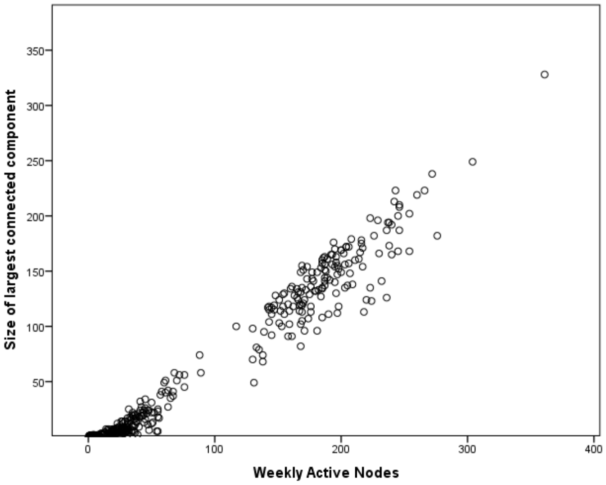


Figure 26 | The relation between the number of weekly active nodes compared to the maximum size of a connected component found between the active nodes.

Chapter 7
Conclusion and Discussion

Conclusion

This thesis answers the question whether communication data can be used to reconstruct weighted social networks over time representing OSS projects and does so in two parts:

Firstly by determining how communications contribute to the strength of interpersonal ties according to a set of factors from related literature. The choice of using communication data is no coincidence, communications occur frequently and in the case of OSS projects on GitHub the communications are openly available for anyone to see and retrieve. To make the selection of appropriate data precise, a definition for appropriate data is introduced, referred to as network activity data:

A set of intentional or unintentional transactions in which one or more actors within a network boundary send and or receive a message, with a specific timestamp and duration, over any available channel.

The definition highlights the importance of communication meta-data, such as the timestamp and duration or length of a message. The combination of known tie strength factors from related research and the availability of communication meta-data leads to the following model for tie strength in OSS projects:

$$W_{i,j}(t_s, t_w) = A_{i,j}^{a}(t_s) * D_{i,j}^{d}(t_s) * R_{i,j}^{r}(t_s, t_w)$$

Eq. (19)

Where the weight of an edge $W_{i,j}$ reflects the strength of a tie and is determined by three measures. $A_{i,j}^{a}(t_s)$ is a measure for the relative age of the edge, $D_{i,j}^{d}(t_s)$ a measure combining the relative duration and frequency of activities and $R_{i,j}^{r}(t_s, t_w)$ a measure for the relative recency of activities. The complete timespan of activities is represented by t_s, and the length of a single time windows of investigation is t_w. The measures a, d, r have values between and including 0 and 1, where a value of 0 removes the influence of one of the measure from the total relative weight. The multiplication of these measures is chosen as it is the simplest dependency relation and enables the use of a, d, r to change the proportional influence of individual measures.

Ties in an OSS project are expected to exist between collaborating members of an OSS project. These interpersonal OSS collaborations are defined as: giving feedback, testing proposals and working towards the quick evolution of ideas and source code with fellow project members.

Secondly by reconstructing an actual OSS project using different reconstruction methods, including the novel method presented in this thesis, an accuracy comparison can be performed. To measure the accuracy of a reconstructed network a reference network is needed. A reference network is created in this thesis by surveying members of an OSS project, in this case the Ruby on Rails project, a web framework based on the Ruby programming language. In total 4 reconstruction of the social network of Rails are created:

1. **Follow Network:** an unweighted network based on follow relations, where an edge exists between two nodes when at least one of the nodes follows the other.
2. **Mutual follow network:** an unweighted network based on mutual follow relations, where edges only exist when adjacent nodes both have decided to follow each other.
3. **Sum Activity Network:** a weighted network based on communication edges where the weight of edges is based on the sum of communications exchanged by two nodes.
4. **Model Activity Network:** a weighted network based on communication edges where the weight of edges is based on the results found by using the model introduced in this paper.

The results show that the sum activity network finds more of the connections mentioned in surveys compared to all other networks. It is able to identify 60% of all edges, the model network finds 50% while the follow network finds 33% and the mutual follow network only 11%. The pairwise comparison of these outcomes shows that the results differ significantly. The model network has a significantly higher precision score (.093) compared to the sum activity model (.073), the follow network (.009) and the mutual follow network (.023). Precision here refers to the ability of a reconstruction method to find the edges mentioned in the survey and exclude false edges which were not mentioned in the survey.

The ranking accuracy comparison, applied to find the reconstruction method that assigns the most accurate weights to ties, shows that the model network is significantly better at ranking important relations compared to the sum activity network. The effect size of .28 for the reconstruction method is however generally considered to be small.

The overarching conclusion is that for the set of Rails survey members, modeled communication data is a more accurate source of information for creating social networks compared to follow relations data. Additionally, the model for tie strength introduced in this thesis is significantly more accurate at sorting strong relations compared to relations who's strength is based on the sum of its activities. These results are limited to the Rails project, but the novel reconstruction method introduced in this thesis can be applied and tested in any social network given its communication meta-data is available.

When the collected data for this thesis is used the recreate research by Bird et al. [3] much weaker but significant correlations between number of source code changes and social status are also found. The most accurate recreation of Bird et al. [3] using betweenness centrality (in a communication based social network) as a measure of social status and number of commits as an indication of a source code contributions, a correlation of .35 is found, which is much lower than the value of .757 found by Bird et al. [3]. When using the number of followers as a measure of social status, a significant correlation of .505 is found with the number of commits created by the same individual. Using the number of followers seems a more reliable measure of social status compared to betweenness centrality, because following requires a conscious directed social action probably representing a positive interest. The finding of Bird et al. [3] that more contributions to the source code of a project lead to a higher social status is not directly contradicted, the strength of the relation is however much weaker in the Rails project.

When using a reconstructed dynamic network based on Rails communications, a prediction of Crowston & Howison [29] can be tested. The prediction states that: '*… a large project might be an aggregate of smaller projects …*'. The Rails project has increased in size over time, from one active user per week to roughly 200 active users per week. If the prediction is correct and the modular elements on a weekly basis of OSS projects are smaller than 200 active people, these modules should show up in the dynamic network of Rails.

Multiple interpretations of modularity are measured in the dynamic network over time: cliques, cluster coefficient and hubs. As the number of active users grows the cluster coefficient increases linearly, so does the number of hubs. The number of cliques also increase with more active users, the cliques also grow in size, but reach a maximum size of 6 members with some outliers. If modularity is interpreted as having multiple disconnected components of reasonable size, then Rails does not consist of modules. When interpreting cliques, clustering or hubs as a sign of modularity, the prediction by Crowston & Howison [29] that larger projects consist of smaller sub-projects, seems to hold.

Discussion

Software development has had an interesting decade, with two general approaches: the agile software development and its counterpart the planned and codified software development. Both approaches have been studied extensively, as summarized by Dingsøyr et al. [58]. The research approaches often take an active role within software development projects, by creating experimental environments to test organizational theories. An example of this is the paired programming experiment, which reveal the different outcomes of solo and paired software developers when given the same task [59]. Other research methods like expert interviews and case studies do not create experimental settings, but are still limited by the number of projects that can be investigated as both methods are time and resource consuming.

Observing software developers as an ecosystem [40] without creating an experimental setting is less popular, maybe because of the problems mentioned in the introduction or the outcomes are less likely to contain practical results. Stallman, the pioneer of OSS, even suggests to avoid the word ecosystem and the view it promotes, as the approach *'implies the absence of ethical judgment*' [60]. Stallman argues that reducing OSS communities to ecosystems means not caring about what should happen and implies that the only interest is in what does happen. It is however these *'nonjudgmental observation*[s]*'* as Stallman calls them, that are lacking and crucial in order to research more fundamental concepts of OSS communities and professional communities in general. Essential empirical observations and definitions are still missing to describe OSS projects and to prove software development theories. Important question are:

a) *What growth patterns can be found in OSS projects?*
b) *How fast can a OSS projects grow in terms of number of developers?*
c) *What should be the definition of a dead or failed OSS project?*
d) *When given the freedom to self-organize, what type of hierarchies emerge within OSS projects?*

Of course there are partial answers, as discussed in the related literature of this paper. GitHub does however provide for the first time the possibility to empirically answer such questions based on a diverse set of hundreds of thousands of projects. Github data is rich in terms of history, connectivity and diversity, ranging from communication data to detailed descriptions of which lines of code were changed by whom and when. The results of this paper show that GitHub communication data is enough to find and rank the majority of collaborative relations for a subset of developers. More importantly is shows that the creation of social networks based on GitHub data enables empirical research into the communities of OSS projects.

Research of GitHub networks should be categorized into three levels:

1. Contributor level
2. Project level

3. Inter-project level

Contributor level research focuses on the behavior and different roles of contributors and their relations to other contributors. The project level describes the properties of projects that emerge from the collective activities of its contributors. The inter-project level looks at how connected, rival or dependent projects interact and influence each other. The approach and model for tie strength in this paper enables research on the contributor level. Future research should focus on the next levels of GitHub networks. The idea of having different levels of systems is again borrowed from ecology, where system levels usually start with genetic systems and end with population and ecosystems [6].

Shortcomings

Although the results of the accuracy comparison are significant for important accuracy measures in favor of the model introduced in this thesis, it is not without its shortcomings. The number of surveys is too low to enable confident estimations about the whole Rails population. It is therefore not possible to say that the outcomes of the accuracy comparison for the complete Rails population would also be significantly in favor of the model introduced in this thesis.

The model for tie strength introduced in this thesis allows for variations in proportional influences of age, duration, frequency and recency. Only the simplest variation of these factors was used, a set of variations should have been used to thoroughly test the model. Retrospectively, the recency measures seems an overly complex factor based on too little related literature and should have been made simpler or removed from the model.

In the creation of the activities dataset a lot of filters have been applied to find the actual text written by the author, this approach needs to be improved as it still needed to much manual inspection of messages. An algorithm that detects natural language could hugely improve the speed by filtering out for example copied error traces. Luckily a substantial amount (18%) of messages included @ mentions, which made determining the targets of a communication easier. The remaining messages were either directed at the author or directed at the previous author. This can lead to false edges where an author was actually replying to another message and not the previous one. The occurrence of these errors could have been tested by investigating a representative sample of comments.

Future Research

Reconstructing social networks representing OSS projects has been done in a variety of ways, Crowston & Howison [29] use emails and bug reports, Bird et al. [3] use emails while Kabbedijk & Jansen [40] use co-authorships. All approaches can be argued and are valid for their research questions, but comparison and generalization of results remains hard when all results stem from different reconstruction methods. A general reconstruction approach, when applied regularly, would make comparisons easier and help in bringing network similarities to light. Whether the general approach represented in this thesis is an acceptable candidate is up for debate, the need for a standard method is however great and should be a topic of future research.

The introduction mentions the question what the evidence should be for proving the existence of interdependence and causal relations determining the survival of OSS projects. This thesis tries to start from the bottom up, by creating the most basic of networks representing a project. Another approach could be to create a large dataset of activity histories using a filtered set of GitHub projects. These activity histories could look like the simplified version of Figure 7 in this thesis. This data is currently accurate and easily accessible following a recent GitHub API update [61]. This dataset could reveal after analysis if there is a limited set of activity histories, say for the first 3 years of a project. If there is indeed a limited set of such histories, the next step would be to see if 1 year of history would be enough to predict the next years, using a model based on a large set of histories. If indeed project activity histories are predictable in such a way, the case for general relations governing OSS projects could be made.

The ratio in Rails between the number of active nodes and nodes with a degree of at least five times the average degree (as mentioned in the results) is found to be consistent and strong regardless of age or number of active users. The same is true for the number of k-clique-communities. This shows that some structural network characteristics remain constant. If again similar structures are found within a large set of projects, the case for general relations governing OSS projects could be made.

Bibliography

[1] C. Wejnert, "Social Network analysis with respondent-driven sampling data: A study of racial integration on campus," *Social Networks,* vol. 32, no. 2, pp. 112-124, 2010.

[2] L. C. Freeman, The Development of Social Network Analysis: A Study in the Sociology of Science, Vancouver: Booksurge Publishing, 2004.

[3] C. Bird, A. Gourley, P. Devanbu, M. Gertz and A. Swaminathan, "Mining Email Social Networks," in *international workshop on Mining software repositories*, Shanghai, China, 2006.

[4] S. Jansen, S. Brinkkemper and M. A. Cusumano, Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry, Cheltenham: Edward Elgar Publishing, Inc., 2013.

[5] J. F. Moore, The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems, New York: HarperBusiness, 1997.

[6] E. P. Odum, Fundamentals of Ecology, Philadelphia: W. B. Saunders Company, 1971.

[7] D. S. Sade, "Sociometrics of Macaca mulatta I. Linkages and Cliques in Grooming Matrice," *Folia Primatologica* , vol. 18, pp. 196-223, 1972.

[8] L. C. Freeman, "Centrality in social networks, conceptual clarification," *Social Networks,* pp. 215-239, 1978.

[9] D. Jackson, J. Kirkland, B. Jackson and D. Bimler, "Social Network Analysis and Estimating the Size of Hard-to-Count Subpopulations," *Connections,* vol. 26, no. 2, pp. 49-60, 2005.

[10] P. V. Marsden, "Network Data and Measurement.," *Annual Review of Sociology,* vol. 16, p. 435–463., 1999.

[11] M. S. Granovetter, "The Strength of weak ties," *American Journal of Sociology,* pp. 1360-1380, 1973.

[12] E. Gilbert and K. Karahalios, "Predicting Tie Strength With Social Media," 2009.

[13] R. Toivonen, J. M. Kumpula, J. Saramäki, J.-P. Onnela, J. Kertész and K. Kaski, "The role of edge weights in social networks: modelling structure and dynamics," in *SPIE Proceedings Vol. 6601,* 2007.

[14] P. J. Onella, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész and A. L. Barabási, "Structure and tie strengths in mobile communication networks," *PNAS,* 2007.

[15] R. B. Rothenberg, "Commentary: Sampling in Social Networks," *Connections,* vol. 18, no. 1, pp. 104-110, 1995.

[16] P. V. Marsden and K. E. Campbell, "Measuring Tie Strength," *Social Forces, Vol. 63, No. 2,* pp. 482-501, 1984.

[17] M. Gupte and T. Eliassi-Rad, "Measuring Tie Strength in Implicit Social Networks," *CoRR,* 2011.

[18] S. Milgram, "The small-world problem," *Psychology Today,* pp. 60-67, 1967.

[19] M. Faloutsos, P. Faloutsos and C. Faloutsos, "On power-law relationships of the Internet," *SIGCOMM,* p. 251–262, 1999.

[20] J. Albello, A. L. Buchsbaum and J. Westerbook, "A functional approach to external graph algorithms," in *Proceedings of the 6th Annual European Symposium on Algorithms*, 1998.

[21] A. Barrat, M. Barthelemy, R. Paster-Satorras and A. Vespignani, "The architecture of complex weighted networks," *Proc. Natl. Acad. Sci.,* p. 3747–3752, 2004.

[22] J. Leskovec, J. Kleinberg and C. Faloutsos, "Graph Evolution: Densification and shrinking diameters," *ACM Transactions on Knowledge Discovery from Data,* 2007.

[23] T. A. B. Snijders, "The Statistical Evaluation of Social Network Dynamics," *Sociological Methodology,* vol. 31, pp. 361-395, 2001.

[24] G. Kossinets and D. J. Watts, "Origins of Homophily in an Evolving Social Network," *American Jounal of Sociology,* vol. 115, no. 2, pp. 405-450, 2009.

[25] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry,* pp. 35-41, 1977.

[26] A. Bavelas, "A mathematical model for group structure," *Human Organization,* pp. 16-30, 1948.

[27] D. J. de Solla Price, "Networks of Scientific Papers," *Science,* pp. 510-515, 1965.

[28] J. S. Coleman, E. Katz and H. Menzel, Medical innovation: a diffusion study, Indianapolis: Bobbs-Merril, 1966.

[29] K. Crowston and J. Howison, "The social structure of Free and Open Source Software development," *First Monday,* no. 10, 2005.

[30] R. D. Luce and A. D. Perry, "A method of matrix analysis of group structure," *Psychometrika,* vol. 14, no. 2, pp. 95-116, 1949.

[31] J.-P. Onnela, J. Saramäki, M. Kivelä, K. Kaski and J. Kertész, "Generalizations of the clustering coefficient to weighted complex networks," *Physical Review,* vol. 75, 2007.

[32] G. Palla, I. Derényi, I. Farkas1 and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature,* no. 435, pp. 814-818, 2005.

[33] Red Hat, Inc., "What is open source software?," Red Hat, [Online]. Available: http://opensource.com/resources/what-open-source. [Accessed February 2014].

[34] The Open Source Initiative , "The Open Source Definition | Open Source Initiative," The Open Source Initiative , [Online]. Available: http://opensource.org/osd. [Accessed 2014 February].

[35] Free Software Foundation, Inc., "GNU General Public License," Free Software Foundation, Inc., 29 June 2007. [Online]. Available: https://www.gnu.org/copyleft/gpl.html#mission-statement. [Accessed February 2014].

[36] A. McPherson, B. Proffitt and R. Hale-Evans, "Estimating the Total Development Cost of a Linux Distribution," The Linux Foundation, October 2008. [Online]. Available: http://www.linuxfoundation.org/sites/main/files/publications/estimatinglinux.html. [Accessed February 2014].

[37] L. Laffan, "Open Governance Index. Measuring the true openness of open source projects from Android to WebKit," 2011.

[38] F. Brooks, The Mythical Man-Month: Essays on Sofware Engineering, 20th anniversary Edition, Addison-Wesley, 1995.

[39] A. Neus and P. Scherf, "Opening minds: Cultural change with the introduction of open-source collaboration methods," *IBM Systems Journal,* vol. 44, no. 2, pp. 215 - 225, 2005.

[40] J. Kabbedijk and S. Jansen, "Steering Insight: An exploration of the Ruby Software Ecosystem," in *Second International Conference on Software Business*, 2011.

[41] D. Berlo, The process of communication, New York: Holt, Rinehart, & Winston, 1960.

[42] D. Liben-Nowell and J. Kleinberg, "The Link-Prediction Problem for Social Networks," *Journal of American Society for Information Science and Technology,* pp. 1019-1031, 2007.

[43] P. Chapman, CRISP-DM 1.0 - Step-By-step data mining guide, CRISP-DM consortium, 2000.

[44] Github, "About," Github, [Online]. Available: https://github.com/about. [Accessed Januari 2014].

[45] S. Chacon, "About," git, [Online]. Available: http://git-scm.com/about. [Accessed February 2014].

[46] D. H. Hansson, "Ruby on Rails will ship with OS X 10.5 (Leopard)," Ruby On Rails, 7 August 2006. [Online]. Available: http://weblog.rubyonrails.org/2006/8/7/ruby-on-rails-will-ship-with-os-x-10-5-leopard/. [Accessed February 2014].

[47] Ruby on Rails, "Ruby on Rails," Ruby on Rails, April 2014. [Online]. Available: Ruby on Rails will ship with OS X 10.5 (Leopard). [Accessed April 2014].

[48] I. Žužak, "Improved pagination for the Repository Commits API," GitHub, 9 May 2014. [Online]. Available: https://developer.github.com/changes/2014-05-09-improved-pagination-for-the-repository-commits-api/. [Accessed 9 May 2014].

[49] Django Software Foundation , "The Web framework for perfectionists with deadlines | Django," Django Software Foundation , August 2014. [Online]. Available: https://www.djangoproject.com/. [Accessed July 2014].

[50] T. G. Consortium, "About," The Gephi Consortium, 2014. [Online]. Available: https://gephi.org/about/. [Accessed Februari 2014].

[51] NetworkX developer team, "Overview - Networkx," NetworkX , 2014. [Online]. Available: https://networkx.github.io/. [Accessed July 2014].

[52] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote Sensing of Environment,* vol. 62, no. 1, pp. 77-89, 1997.

[53] M. Kubat, R. C. Holte and S. Matwin, "Machine learning for the detection of oil spills in satellite radar images," *Machine Learning,* vol. 30, pp. 195-215, 1998.

[54] D. D. Brewer and C. M. Webster, "Forgetting of friends and its effects on measuring friendship networks.," *Social Networks,* vol. 21, pp. 361-373, 1999.

[55] R. Horn, "Educational Psychology 625: Intermediate Statistics," 2008. [Online]. Available: http://oak.ucc.nau.edu/rh232/courses/EPS625/Handouts/Nonparametric/The%20Friedman%20Test.pdf. [Accessed June 2014].

[56] P. Legendre, "Species Associations: The Kendall Coefficient of Concordance Revisited," *Journal of Agricultural, Biological, and Environmental Statistics,* vol. 10, no. 2, pp. 226-245, 2005.

[57] D. Dierckx, "Calculate a survey sample size (number of respondents needed)," Cherck Market, 2013. [Online]. Available: https://www.checkmarket.com/market-research-resources/sample-size-calculator/. [Accessed July 2014].

[58] T. Dingsøyr, T. Dyba and N. B. Moe, Agile Software Development, Current Research and Future Directions, Berlin: Springer, 2010.

[59] K. M. Lui, K. A. Barnes and C. Chan, "Pair Programming: Issues and Challenges," in *Agile Software Development*, Berlin, Springer, 2010, pp. 143-162.

[60] R. M. Stallman, "Words to Avoid," GNU, [Online]. Available: http://www.gnu.org/philosophy/words-to-avoid.html#Ecosystem. [Accessed June 2014].

[61] Žužak, Ivan, "Improved pagination for the Repository Commits API," GitHub, 9 May 2014. [Online]. Available: https://developer.github.com/changes/2014-05-09-improved-pagination-for-the-repository-commits-api/. [Accessed 9 May 2014].

[62] S. P. Borgatti and M. G. Everett, "A graph-theoretic perspective on centrality," *Social Networks*, pp. 466-484, 2006.

[63] L. C. Freeman, "The Gatekeeper, pair-dependency and structural centrality," *Quality and Quantity*, pp. 585-592, 1980.

[64] J. Luskovec, D. Huttenlocher and J. Kleinberg, "Signed Networks in Social Media," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, 2010.

[65] W. . W. Cohen, "Enron Email Dataset," 21 August 2009. [Online]. Available: http://www.cs.cmu.edu/~enron/. [Accessed 2013].

[66] A. Mislove , M. Marcon, K. P. Gummadi, P. Druschel and B. Bhattacharjee, "Measurement and Analysis of Online Social Networks," in *Proceedings of the 5th ACM/Usenix Internet Measurement Conference (IMC'07)*, San Diego, CA, 2007.

[67] D. J. Brass, "Being in the right place: a structural analysis of individual influence in an organization," *Administrative Science Quarterly*, pp. 518-39, 1984.

[68] M. De Choudhury, H. Sundaram, A. John and D. D. Seligmann, "Contextual Prediction of Communication Flow in Social Networks," *Web Intelligence*, pp. 57-65, 2007.

[69] J. McAuley and J. Leskovec, " Learning to Discover Social Circles in Ego Networks," *NIPS*, 2012.

[70] E. Gilbert, "Predicting Tie Strength in a New Medium," in *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, New York, 2012.

[71] M. E. J. Newman, "Analysis of weighted networks," *Physics Review*, 2004.

[72] X. Qi, E. Fuller, Q. Wu, Y. Wu and C.-Q. Zhang, "Laplacian centrality: A new centrality measure for weighted networks," *Information Sciences*, p. 240–253, 2012.

[73] D. Gruhl, R. Guha, D. L. Nowell and A. Tomkins, "Information Diffusion through Blogspace," in *Proceedings of the 13th international conference on World Wide Web*, 2004.

[74] P. De Meo, E. Ferrara, G. Fiumara and A. Ricciardello, "A Novel Measure of Edge Centrality in Social Networks," *Knowledge-Based Systems,* p. 136–150, 2012.

[75] X. Song, C.-Y. Lin, B. L. Tseng and M.-T. Sun, "Modeling and predicting personal information dissemination behavior," in *KDD '05 Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining,* 2005.

[76] D. Liben-Nowell and J. Kleinberg, "The Link Prediction Problem for Social Networks," in *Proceedings of the Twelfth Annual ACM International Conference on Information and Knowledge Management*, 2003.

[77] J. Leskovec, "SNAP," Oct 2013. [Online]. Available: https://snap.stanford.edu/.

[78] M. Granovetter, "Network Sampling: Some first steps," *American Journal of Sociology,* vol. 81, 1976.

[79] O. Frank, "Estimation of population totals by use of snowball samples," *Perspectives on Social Network Research,* 1979.

[80] J. K. Watters and P. Biernacki , "Targeted sampling: Options for the study of hidden populations," *Social Problems,* vol. 36, 1989.

[81] P. Erdos and A. Renyi, "On the evolution of random graphs," *Publications of the Mathematical Institute of the Hungarian Academy of Sciences,* vol. 5, pp. 17-61, 1960.

[82] Facebook, "Open Graph," Facebook, November 2013. [Online]. Available: http://developers.facebook.com/docs/opengraph/. [Accessed November 2013].

[83] D. Wynn Jr., Assessing the Health of an Open Source Ecosystem, IGI Publishing, 2007.

[84] D. Wahyudin, K. Mustofa, A. Schatten, S. Biffl and A. Min Tjo, "Monitoring the "health" status of open source web-engineering projects," *International Journal of Web Information Systems,* no. 3, pp. 116-139, 2007.

[85] Slashdot Media, "SourceForge - About," Slashdot Media, 2014. [Online]. Available: http://sourceforge.net/about. [Accessed Februari 2014].

[86] D. S. Foundation, "About the Django Software Foundation," Django Software Foundation, 2014. [Online]. Available: https://www.djangoproject.com/foundation/. [Accessed Februari 2014].

[87] E. D. Kolaczyk and P. N. Krivitsky, "On the Question of Effective Sample Size in Network Modeling," *arXiv:1112.0840,* 20012.

[88] M. Granovetter, "Network Sampling: Some First Steps," *American Journal of Sociology,* vol. 83, no. 3, 1977.

Appendix

Appendix A

The following consent statement was agreed to by all survey entry subjects, its details were partly communicated in the email send to subjects and of course in full on the actual survey webpage:

> **Background Information**
>
> Researcher: Arno Gregorian at Utrecht University
> Purpose of data collection: MBI Master Thesis
> Details of Participation: A request to complete online questionnaires concerning role of the participant as a contributor to a GitHub project.
>
> **Consent Statement**
>
> 1. I understand that my participation is voluntary and that I can withdraw unconditionally at any time from taking part in this online study.
>
> 2. I have been informed that a Debriefing Statement explaining the reasons for this study will be supplied via email if requested, following the completion of my participation.
>
> 3 My data are to be held confidentially and only the researcher and direct supervisors will have access to them.
>
> 4. My data will be pseudonymised, personally identifiable information will be transformed to meaningless keys with matching details in only one single data table only accessible to the researcher and direct supervisors.
>
> 5. My data will be kept in a locked cabinet for a period of at least five years after the appearance of any associated publications. Any aggregate data (e.g. spreadsheets) will be kept in electronic form for up to five years after which time they will be deleted.
>
> 6. In accordance with the requirements of some scientific journals and organizations, my coded and pseudonymised data may be shared with other competent researchers. My coded data may also be used in other related studies. My name and other identifying details will not be shared with anyone.
>
> 7. The overall findings may be submitted for publication in a scientific journal, or presented at scientific conferences.
>
> 8. This study will take approximately 2-3 minutes to complete.
>
> 9. I will be able to obtain general information about the results of this research from the researcher at their e-mail address a.gregorian@students.uu.nl
>
> I am giving my consent for data to be used for the outlined purposes of the present study

Appendix B

This appendix contains the more detailed results of the explorative analysis.

Appendix B1

The linear regression of nodes with a disproportionally high degree (hubs) shows that when the number of active nodes grows roughly 30, a new hub is probably amongst the new 30 nodes.

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-.291	.052		-5.562	.000
	Nodes	.031	.000	.951	68.415	.000

a. Dependent Variable: Nodes with a degree at least five times as high as the average degree

Table 31 | The linear regression of nodes with a high degree compared to the number of active nodes during one of the weeks of Rails.

Appendix B2

ANOVA[a]

	Model		Sum of Squares	df	Mean Square	F	Sig.
Edges between non Rails Members	1	Regression	.001	1	.001	.222	.638[b]
		Residual	99.330	25208	.004		
		Total	99.331	25209			
Edges between Rails Members	1	Regression	.331	1	.331	176.520	.000[b]
		Residual	1.438	766	.002		
		Total	1.769	767			

a. Dependent Variable: overlap

b. Predictors: (Constant), activity sum weight

Table 32 | The linear regression ANOVA comparing members and nonmembers and the relation between overlap and edge weight, based on the sum of activities of an edge.

Model Summary

	Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
Edges between non Rails Members	1	.003[a]	.000	.000	.0627728542515
Edges between Rails Members	1	.433[a]	.187	.186	.0433202452023

a. Predictors: (Constant), activity sum weight

Table 33 | The model summary of the linear regression comparing members and nonmembers and the relation between overlap and edge weight, based on the sum of activities of an edge.

Appendix B3

Some of the anonymized email responses received by the author after sending the survey request to Rails members.

1. *"I 'm really not active ror* [Ruby on Rails] *contributor. And my commits are almost random. And I have no time for contribution :(So your questions are unknown to me, sorry."*
2. *"I'd like to take the survey but didn't really feel like any of the options pertained to me. Since I feel like my situation is probably fairly common among the contributors I though I'd reply and explain why I contributed.*
3. *I contributed a tiny bit of code that fixed a longstanding bug in how rails rendered a small HTML tag. Nothing major, but it was something that I noticed didn't work quite right, so I took the time (probably around 2-3 hours) to dig into it and submit a fix. This is something that I've done on probably 10 or so different projects over the years and it's something that quite a few people do. Your survey doesn't really apply to folks like me who aren't really "contributors" since we don't really collaborate or anything with the other guys and will probably never submit any more code, but our contributions do add up to what makes open-source projects like Rails work."*
4. *"I filled out your survey, even though I think the extent of my contribution to Rails is one pull request a few years ago. So my answers to the first two questions are blank:*
5. *I don't recall collaborating with anyone, although probably there was someone who reviewed my PR. I don't keep track of who's contributing to Rails."*
6. *"Sorry, I only vaguely remember getting a commit accepted to Rails... can't even remember what it was. I haven't collaborated with any Rails devs directly."*
7. *"I tried to answer to your survey, but the questions target much more active contributors, than me. I have only 1 patch in Rails 4 years ago and I'm not following Rails community any more, so I can't answer your question about the most influential people. I wish I could help you with your research."*

Appendix B4

The relation between the number of nodes and the size and number of maximal cliques is investigated. The scatterplot is both fitted with a linear and quadratic function, the results show that the quadratic function is significantly more accurate compared to the linear function. This suggest indeed that as the number of nodes grows, after a certain threshold the size of the maximal cliques seem to stabilize.

Variables Entered/Removed[a]			
Model	Variables Entered	Variables Removed	Method
1	Max. clique size[b]		Enter
2	Max. clique size squared[b]		Enter
a. Dependent Variable: Nodes			
b. All requested variables entered.			

Table 34

Model Summary									
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics				
					R Square Change	F Change	df1	df2	Sig. F Change
1	.913[a]	.834	.834	35.655	.834	2474.302	1	493	.000
2	.921[b]	.849	.848	34.043	.015	48.786	1	492	.000
a. Predictors: (Constant), max_clique_size									
b. Predictors: (Constant), max_clique_size, max_clique_size_squared									

Table 35

Appendix B5

The degree descriptives of the network of 10 years of Ruby on Rails on Github, where edges are based on comment, issue, pull request and commit activities.

Degree Descriptives				
			Statistic	Std. Error
Degree	Mean		5.95	.412
	95% Confidence Interval for Mean	Lower Bound	5.14	
		Upper Bound	6.75	
	5% Trimmed Mean		3.01	
	Median		2.00	
	Variance		1476.902	
	Std. Deviation		38.430	
	Minimum		0	
	Maximum		1870	
	Range		1870	
	Interquartile Range		3	
	Skewness		27.688	.026
	Kurtosis		990.336	.052

Table 36

Appendix B5

A decision tree classification is created for all the survey results with a prediction class of either 0 or 1, where 1 is assigned to edges that are mentioned in the surveys and 0 is assigned to edges that are found in the complete adjacency matrix but are not mentioned by the survey subject. An extra attribute is added, partner degree, which is the degree of the partner which is the sum of other model edges connected to the partner node.

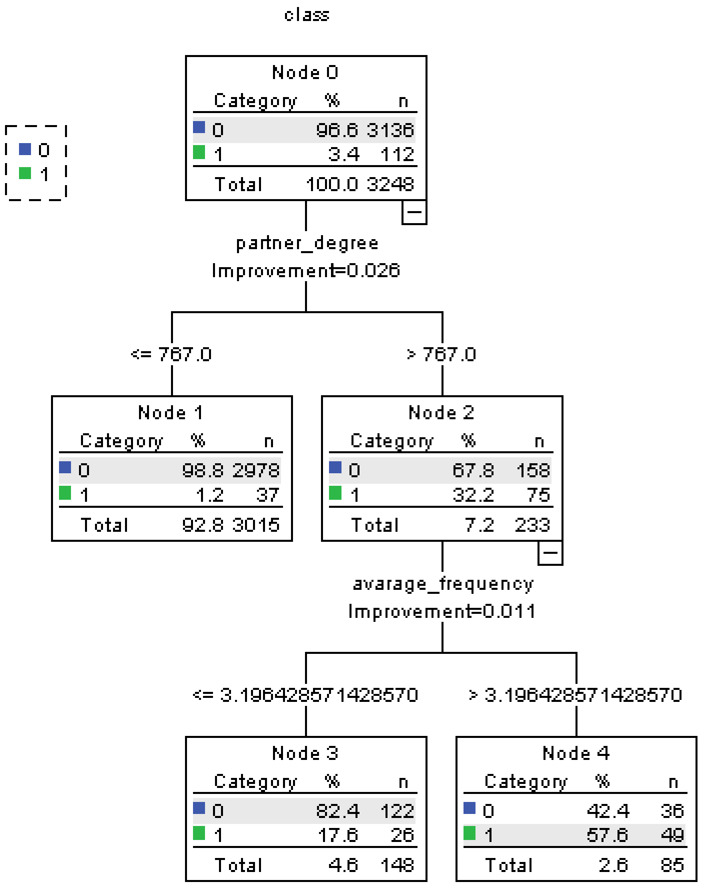


Figure 27

Risk

Estimate	Std. Error
.034	.003

Growing Method: CRT

Dependent Variable: class

Table 37

Classification

Observed	Predicted		
	0	1	Percent Correct
0	3100	36	98.9%
1	63	49	43.8%
Overall Percentage	97.4%	2.6%	97.0%

Growing Method: CRT

Dependent Variable: class

Table 38

Appendix B6

The graph in Figure 28 shows the model weighted network of all the Rails members. The graph is characterized by a high clustering coefficient: 0.742, high average degree: 17 and a small world network with a diameter of 4.

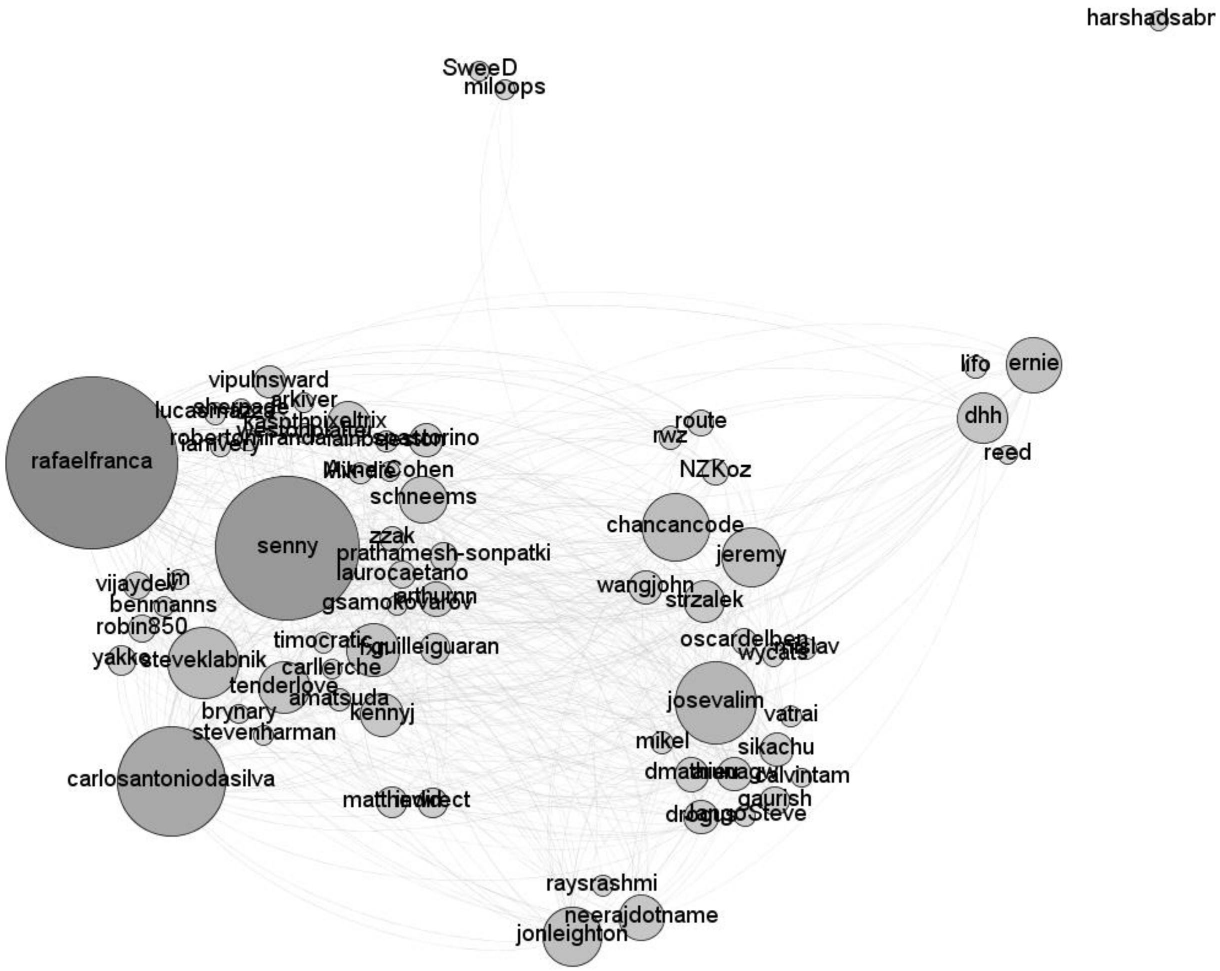


Figure 28 | The Ruby on Rails members

Appendix B7

The linear regression results of the comparison between k-clique-communities with k=4 and the number of active nodes during a single week.

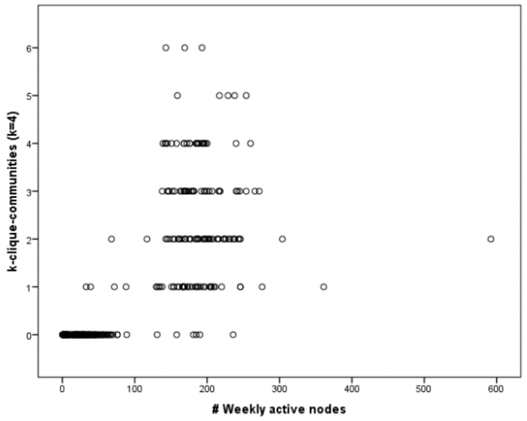


Figure 29

Descriptive Statistics			
	Mean	Std. Deviation	N
Nodes	74.04	87.386	495
K_4_clique_communities	.77	1.329	495

Table 39

Model Summary									
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics				
					R Square Change	F Change	df1	df2	Sig. F Change
1	.775[a]	.601	.600	55.254	.601	742.608	1	493	.000
a. Predictors: (Constant), K_4_clique_communities									

Table 40

ANOVA[a]						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	2267183.182	1	2267183.182	742.608	.000[b]
	Residual	1505130.163	493	3053.002		
	Total	3772313.345	494			
a. Dependent Variable: Nodes						
b. Predictors: (Constant), K_4_clique_communities						

Table 41

Coefficients[a]						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	34.598	2.874		12.037	.000
	K_4_clique_communities	50.971	1.870	.775	27.251	.000
a. Dependent Variable: Nodes						

Table 42

Appendix B8

The distributions of the survey results, even when logarithmically transformed do not show normal distributions and both the Kolmogorov-Smirnov and the Shapiro-Wilk tests are significant.

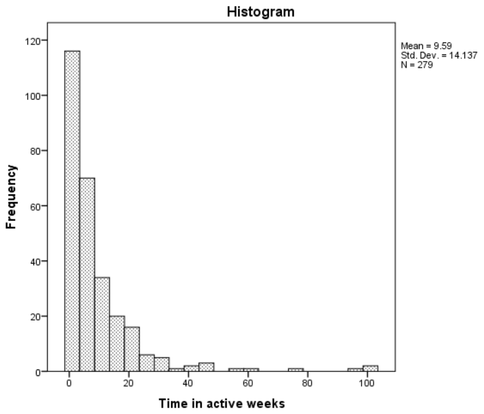


Figure 30

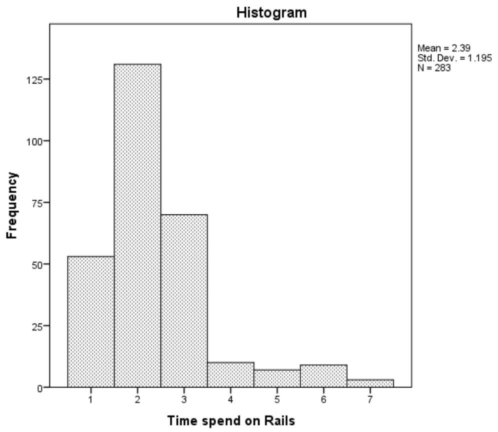


Figure 31

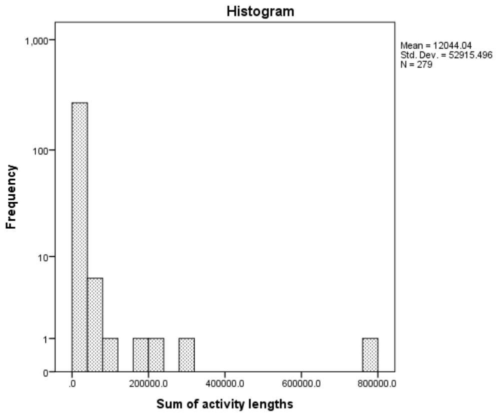


Figure 32

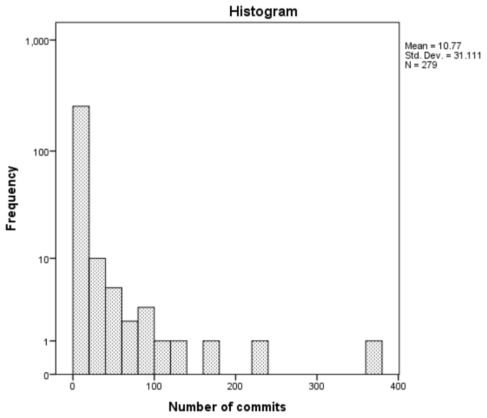


Figure 33

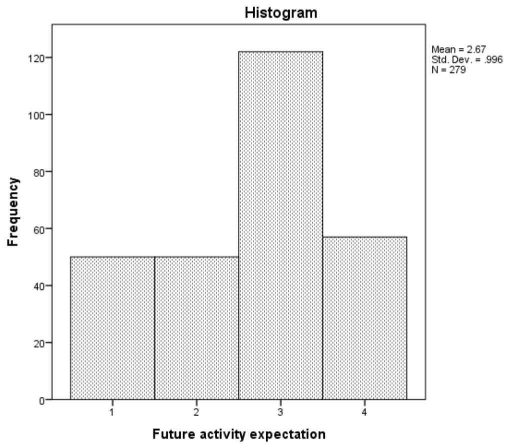


Figure 34

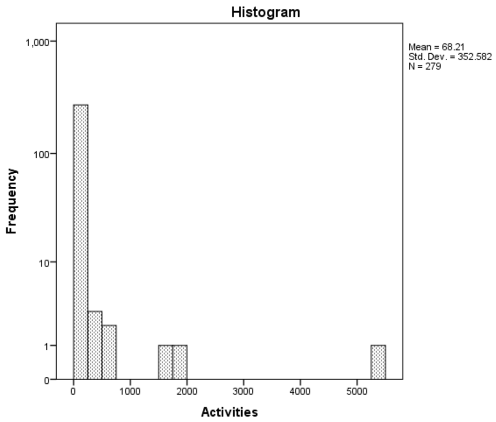


Figure 35

Descriptives of survey results

			Statistic	Std. Error
Time spend on Rails	Mean		2.39	.072
	95% Confidence Interval for Mean	Lower Bound	2.25	
		Upper Bound	2.53	
	5% Trimmed Mean		2.26	
	Median		2.00	
	Variance		1.447	
	Std. Deviation		1.203	
	Minimum		1	
	Maximum		7	
	Range		6	
	Interquartile Range		1	
	Skewness		1.567	.146
	Kurtosis		3.104	.291
Number of edges mentioned by survey actor	Mean		1.00	.072
	95% Confidence Interval for Mean	Lower Bound	.85	
		Upper Bound	1.14	
	5% Trimmed Mean		.88	
	Median		1.00	
	Variance		1.457	
	Std. Deviation		1.207	
	Minimum		0	
	Maximum		5	
	Range		5	
	Interquartile Range		2	
	Skewness		1.157	.146
	Kurtosis		.738	.291
Future activity expectation	Mean		2.67	.060
	95% Confidence Interval for Mean	Lower Bound	2.55	
		Upper Bound	2.78	
	5% Trimmed Mean		2.69	
	Median		3.00	
	Variance		.993	
	Std. Deviation		.996	
	Minimum		1	
	Maximum		4	
	Range		3	
	Interquartile Range		1	

	Skewness		-.391	.146
	Kurtosis		-.880	.291
Distinct weeks with at least one activity performed by the actor	Mean		9.59	.846
	95% Confidence Interval for Mean	Lower Bound	7.92	
		Upper Bound	11.25	
	5% Trimmed Mean		7.38	
	Median		5.00	
	Variance		199.862	
	Std. Deviation		14.137	
	Minimum		1	
	Maximum		103	
	Range		102	
	Interquartile Range		9	
	Skewness		3.878	.146
	Kurtosis		19.428	.291
Total number of activities	Mean		68.21	21.109
	95% Confidence Interval for Mean	Lower Bound	26.66	
		Upper Bound	109.76	
	5% Trimmed Mean		28.27	
	Median		16.00	
	Variance		124313.937	
	Std. Deviation		352.582	
	Minimum		1	
	Maximum		5346	
	Range		5345	
	Interquartile Range		37	
	Skewness		12.874	.146
	Kurtosis		184.819	.291
Sum of activity lengths	Mean		12044.039	3167.9679
	95% Confidence Interval for Mean	Lower Bound	5807.787	
		Upper Bound	18280.292	
	5% Trimmed Mean		5413.332	
	Median		2943.000	
	Variance		2800049741.117	
	Std. Deviation		52915.4962	
	Minimum		38.0	
	Maximum		772979.0	
	Range		772941.0	
	Interquartile Range		7938.0	

	Skewness		11.702	.146
	Kurtosis		158.361	.291
Number of commits	Mean		10.77	1.863
	95% Confidence Interval for Mean	Lower Bound	7.10	
		Upper Bound	14.43	
	5% Trimmed Mean		5.64	
	Median		3.00	
	Variance		967.884	
	Std. Deviation		31.111	
	Minimum		0	
	Maximum		366	
	Range		366	
	Interquartile Range		7	
	Skewness		7.587	.146
	Kurtosis		71.934	.291

Table 43

Appendix C

This appendix contains the extended results of the Friedman test performed to test the accuracy of the different networks. For each accuracy test the pairwise comparison is given. The pairwise comparison shows a single asterisk for a significant difference in mean between the pairs at the a=0.01 level, two asterisks are used for differences in mean at the a=0.05 level.

Accuracy

Ranks				
		N	Mean Rank	Sum of Ranks
sum_accuracy - model_accuracy	Negative Ranks	200[a]	103.94	20787.00
	Positive Ranks	4[b]	30.75	123.00
	Ties	79[c]		
	Total	283		
follow_accuracy - model_accuracy	Negative Ranks	254[d]	144.01	36579.50
	Positive Ranks	21[e]	65.26	1370.50
	Ties	8[f]		
	Total	283		
follow_follow_accuracy - model_accuracy	Negative Ranks	113[g]	118.32	13370.00
	Positive Ranks	127[h]	122.44	15550.00
	Ties	43[i]		
	Total	283		
follow_accuracy - sum_accuracy	Negative Ranks	242[j]	144.89	35063.00
	Positive Ranks	32[k]	81.63	2612.00
	Ties	9[l]		
	Total	283		
follow_follow_accuracy - sum_accuracy	Negative Ranks	77[m]	105.04	8088.00
	Positive Ranks	165[n]	129.18	21315.00
	Ties	41[o]		
	Total	283		
follow_follow_accuracy - follow_accuracy	Negative Ranks	0[p]	.00	.00
	Positive Ranks	268[q]	134.50	36046.00
	Ties	15[r]		
	Total	283		

a. sum_accuracy < model_accuracy

b. sum_accuracy > model_accuracy

c. sum_accuracy = model_accuracy

d. follow_accuracy < model_accuracy

e. follow_accuracy > model_accuracy

f. follow_accuracy = model_accuracy

g. follow_follow_accuracy < model_accuracy

h. follow_follow_accuracy > model_accuracy

i. follow_follow_accuracy = model_accuracy

j. follow_accuracy < sum_accuracy

k. follow_accuracy > sum_accuracy

l. follow_accuracy = sum_accuracy

m. follow_follow_accuracy < sum_accuracy

n. follow_follow_accuracy > sum_accuracy

o. follow_follow_accuracy = sum_accuracy

p. follow_follow_accuracy < follow_accuracy

q. follow_follow_accuracy > follow_accuracy

r. follow_follow_accuracy = follow_accuracy

Table 44

True Positive Rate

Ranks

		N	Mean Rank	Sum of Ranks
sum_true_positive_rate - model_true_positive_rate	Negative Ranks	0[a]	.00	.00
	Positive Ranks	28[b]	14.50	406.00
	Ties	255[c]		
	Total	283		
follow_true_positive_rate - model_true_positive_rate	Negative Ranks	64[d]	58.93	3771.50
	Positive Ranks	42[e]	45.23	1899.50
	Ties	177[f]		
	Total	283		
follow_follow_true_positive_rate - model_true_positive_rate	Negative Ranks	79[g]	46.80	3697.00
	Positive Ranks	12[h]	40.75	489.00
	Ties	192[i]		
	Total	283		
follow_true_positive_rate - sum_true_positive_rate	Negative Ranks	71[j]	57.44	4078.00
	Positive Ranks	32[k]	39.94	1278.00
	Ties	180[l]		
	Total	283		
	Negative Ranks	92[m]	54.00	4968.00

follow_follow_true_positive_rate - sum_true_positive_rate	Positive Ranks	11[n]	35.27	388.00
	Ties	180[o]		
	Total	283		
follow_follow_true_positive_rate - follow_true_positive_rate	Negative Ranks	47[p]	24.00	1128.00
	Positive Ranks	0[q]	.00	.00
	Ties	236[r]		
	Total	283		

a. sum_true_positive_rate < model_true_positive_rate

b. sum_true_positive_rate > model_true_positive_rate

c. sum_true_positive_rate = model_true_positive_rate

d. follow_true_positive_rate < model_true_positive_rate

e. follow_true_positive_rate > model_true_positive_rate

f. follow_true_positive_rate = model_true_positive_rate

g. follow_follow_true_positive_rate < model_true_positive_rate

h. follow_follow_true_positive_rate > model_true_positive_rate

i. follow_follow_true_positive_rate = model_true_positive_rate

j. follow_true_positive_rate < sum_true_positive_rate

k. follow_true_positive_rate > sum_true_positive_rate

l. follow_true_positive_rate = sum_true_positive_rate

m. follow_follow_true_positive_rate < sum_true_positive_rate

n. follow_follow_true_positive_rate > sum_true_positive_rate

o. follow_follow_true_positive_rate = sum_true_positive_rate

p. follow_follow_true_positive_rate < follow_true_positive_rate

q. follow_follow_true_positive_rate > follow_true_positive_rate

r. follow_follow_true_positive_rate = follow_true_positive_rate

Table 45

True Negative Rate

Ranks

		N	Mean Rank	Sum of Ranks
sum_true_negative_rate - model_true_negative_rate	Negative Ranks	203[a]	102.00	20706.00
	Positive Ranks	0[b]	.00	.00
	Ties	80[c]		
	Total	283		
follow_true_negative_rate - model_true_negative_rate	Negative Ranks	251[d]	144.07	36161.50
	Positive Ranks	23[e]	65.80	1513.50
	Ties	9[f]		
	Total	283		
follow_follow_true_negative_rate - model_true_negative_rate	Negative Ranks	105[g]	114.52	12025.00
	Positive Ranks	131[h]	121.69	15941.00

	Ties	47[i]		
	Total	283		
follow_true_negative_rate - sum_true_negative_rate	Negative Ranks	238[j]	144.57	34407.50
	Positive Ranks	34[k]	80.01	2720.50
	Ties	11[l]		
	Total	283		
follow_follow_true_negative_rate - sum_true_negative_rate	Negative Ranks	72[m]	104.51	7524.50
	Positive Ranks	171[n]	129.37	22121.50
	Ties	40[o]		
	Total	283		
follow_follow_true_negative_rate - follow_true_negative_rate	Negative Ranks	0[p]	.00	.00
	Positive Ranks	269[q]	135.00	36315.00
	Ties	14[r]		
	Total	283		

a. sum_true_negative_rate < model_true_negative_rate

b. sum_true_negative_rate > model_true_negative_rate

c. sum_true_negative_rate = model_true_negative_rate

d. follow_true_negative_rate < model_true_negative_rate

e. follow_true_negative_rate > model_true_negative_rate

f. follow_true_negative_rate = model_true_negative_rate

g. follow_follow_true_negative_rate < model_true_negative_rate

h. follow_follow_true_negative_rate > model_true_negative_rate

i. follow_follow_true_negative_rate = model_true_negative_rate

j. follow_true_negative_rate < sum_true_negative_rate

k. follow_true_negative_rate > sum_true_negative_rate

l. follow_true_negative_rate = sum_true_negative_rate

m. follow_follow_true_negative_rate < sum_true_negative_rate

n. follow_follow_true_negative_rate > sum_true_negative_rate

o. follow_follow_true_negative_rate = sum_true_negative_rate

p. follow_follow_true_negative_rate < follow_true_negative_rate

q. follow_follow_true_negative_rate > follow_true_negative_rate

r. follow_follow_true_negative_rate = follow_true_negative_rate

Table 46

Precision

Ranks

		N	Mean Rank	Sum of Ranks
sum_precision - model_precision	Negative Ranks	71[a]	44.84	3183.50
	Positive Ranks	20[b]	50.13	1002.50
	Ties	192[c]		

	Total	283		
follow_precision - model_precision	Negative Ranks	84[d]	77.21	6485.50
	Positive Ranks	38[e]	26.78	1017.50
	Ties	161[f]		
	Total	283		
follow_follow_precision - model_precision	Negative Ranks	80[g]	49.31	3945.00
	Positive Ranks	17[h]	47.53	808.00
	Ties	186[i]		
	Total	283		
follow_precision - sum_precision	Negative Ranks	95[j]	74.89	7114.50
	Positive Ranks	32[k]	31.67	1013.50
	Ties	156[l]		
	Total	283		
follow_follow_precision - sum_precision	Negative Ranks	90[m]	55.04	4954.00
	Positive Ranks	20[n]	57.55	1151.00
	Ties	173[o]		
	Total	283		
follow_follow_precision - follow_precision	Negative Ranks	40[p]	23.60	944.00
	Positive Ranks	24[q]	47.33	1136.00
	Ties	219[r]		
	Total	283		

a. sum_precision < model_precision

b. sum_precision > model_precision

c. sum_precision = model_precision

d. follow_precision < model_precision

e. follow_precision > model_precision

f. follow_precision = model_precision

g. follow_follow_precision < model_precision

h. follow_follow_precision > model_precision

i. follow_follow_precision = model_precision

j. follow_precision < sum_precision

k. follow_precision > sum_precision

l. follow_precision = sum_precision

m. follow_follow_precision < sum_precision

n. follow_follow_precision > sum_precision

o. follow_follow_precision = sum_precision

p. follow_follow_precision < follow_precision

q. follow_follow_precision > follow_precision

r. follow_follow_precision = follow_precision

Table 47

g-mean

Ranks				
		N	Mean Rank	Sum of Ranks
sum_g_mean - model_g_mean	Negative Ranks	67[a]	40.36	2704.00
	Positive Ranks	26[b]	64.12	1667.00
	Ties	190[c]		
	Total	283		
follow_g_mean - model_g_mean	Negative Ranks	83[d]	76.15	6320.50
	Positive Ranks	39[e]	30.32	1182.50
	Ties	161[f]		
	Total	283		
follow_follow_g_mean - model_g_mean	Negative Ranks	81[g]	51.63	4182.00
	Positive Ranks	17[h]	39.35	669.00
	Ties	185[i]		
	Total	283		
follow_g_mean - sum_precision	Negative Ranks	75[j]	70.67	5300.00
	Positive Ranks	52[k]	54.38	2828.00
	Ties	156[l]		
	Total	283		
follow_follow_g_mean - sum_g_mean	Negative Ranks	91[m]	56.96	5183.50
	Positive Ranks	18[n]	45.08	811.50
	Ties	174[o]		
	Total	283		
follow_follow_g_mean - follow_g_mean	Negative Ranks	42[p]	29.19	1226.00
	Positive Ranks	22[q]	38.82	854.00
	Ties	219[r]		
	Total	283		

a. sum_g_mean < model_g_mean

b. sum_g_mean > model_g_mean

c. sum_g_mean = model_g_mean

d. follow_g_mean < model_g_mean

e. follow_g_mean > model_g_mean

f. follow_g_mean = model_g_mean

g. follow_follow_g_mean < model_g_mean

h. follow_follow_g_mean > model_g_mean

i. follow_follow_g_mean = model_g_mean

j. follow_g_mean < sum_precision

k. follow_g_mean > sum_precision

l. follow_g_mean = sum_precision

m. follow_follow_g_mean < sum_g_mean

n. follow_follow_g_mean > sum_g_mean

o. follow_follow_g_mean = sum_g_mean

p. follow_follow_g_mean < follow_g_mean

q. follow_follow_g_mean > follow_g_mean

r. follow_follow_g_mean = follow_g_mean

Table 48

g-mean 2

Ranks

		N	Mean Rank	Sum of Ranks
sum_g_mean_2 - model_g_mean_2	Negative Ranks	65[a]	33.00	2145.00
	Positive Ranks	28[b]	79.50	2226.00
	Ties	190[c]		
	Total	283		
follow_g_mean_2 - model_g_mean_2	Negative Ranks	79[d]	66.96	5289.50
	Positive Ranks	43[e]	51.48	2213.50
	Ties	161[f]		
	Total	283		
follow_follow_g_mean_2 - model_g_mean_2	Negative Ranks	82[g]	52.12	4274.00
	Positive Ranks	16[h]	36.06	577.00
	Ties	185[i]		
	Total	283		
follow_g_mean_2 - sum_g_mean_2	Negative Ranks	91[j]	69.57	6330.50
	Positive Ranks	36[k]	49.93	1797.50
	Ties	156[l]		
	Total	283		

follow_follow_g_mean_2 - sum_g_mean_2	Negative Ranks	94[m]	59.65	5607.00
	Positive Ranks	16[n]	31.13	498.00
	Ties	173[o]		
	Total	283		
follow_follow_g_mean_2 - follow_g_mean_2	Negative Ranks	47[p]	41.00	1927.00
	Positive Ranks	17[q]	9.00	153.00
	Ties	219[r]		
	Total	283		

a. sum_g_mean_2 < model_g_mean_2

b. sum_g_mean_2 > model_g_mean_2

c. sum_g_mean_2 = model_g_mean_2

d. follow_g_mean_2 < model_g_mean_2

e. follow_g_mean_2 > model_g_mean_2

f. follow_g_mean_2 = model_g_mean_2

g. follow_follow_g_mean_2 < model_g_mean_2

h. follow_follow_g_mean_2 > model_g_mean_2

i. follow_follow_g_mean_2 = model_g_mean_2

j. follow_g_mean_2 < sum_g_mean_2

k. follow_g_mean_2 > sum_g_mean_2

l. follow_g_mean_2 = sum_g_mean_2

m. follow_follow_g_mean_2 < sum_g_mean_2

n. follow_follow_g_mean_2 > sum_g_mean_2

o. follow_follow_g_mean_2 = sum_g_mean_2

p. follow_follow_g_mean_2 < follow_g_mean_2

q. follow_follow_g_mean_2 > follow_g_mean_2

r. follow_follow_g_mean_2 = follow_g_mean_2

Table 49