The Structural Dimension of Cooperation

Cooperation Networks as Cohesive Small Worlds
PhD Thesis Defense

Jordi Torrents

Department of Sociology
University of Barcelona

June 4, 2017

Contents

Contents

Central Topic: Large Scale Cooperation

The last half of twentieth century has witnessed a key shift in knowledge intensive production processes.

The rising importance of collective research and cooperation (Wuchty et al., 2007)

- During most of the last century important papers or inventions were mostly developed by a single author
- Since 2000s top cited papers or inventions are developed by a team.

Central Topic: Large Scale Cooperation

The last half of twentieth century has witnessed a key shift in knowledge intensive production processes.

The rising importance of collective research and cooperation (Wuchty et al., 2007)

- During most of the last century important papers or inventions were mostly developed by a single author
- Since 2000s top cited papers or inventions are developed by a team.

Socialization concept drawn from the Marxian tradition (Adler, 2007)

- In Political Science: transfer of ownership from the private to the public sphere.
- In Psychology: process whereby people new to a culture internalize its knowledge, norms and values.
- Marx's use was broader than either and encompasses both:

  The [...] socialized (i.e. collective) labour come into being through cooperation, division of labour within the workshop, the use of machinery, and in general, the transformation of production by the conscious use of the sciences, of mechanics, chemistry, etc. for specific ends, technology, etc. and similarly, through the enormous increase of scale corresponding to such developments (for it is only socialized labour that is capable of applying the general products of human development, such as mathematics, to the immediate process of production; [...]). (Marx, 1990, 1024)

Theoretical Approaches to Cooperation

Macro level approach

Cooperation as a macro level phenomenon in which the center of analysis is the collective or group (Marx, 1990; Adler and Heckscher, 2006; Adler, 2015).

- Focus on large organizations and groups: Collaborative Communities
  - Shared values and goals
  - Generalized trust
  - Authority forms

Theoretical Approaches to Cooperation

Macro level approach

Cooperation as a macro level phenomenon in which the center of analysis is the collective or group (Marx, 1990; Adler and Heckscher, 2006; Adler, 2015).

- Focus on large organizations and groups: Collaborative Communities
  - Shared values and goals
  - Generalized trust
  - Authority forms

Micro level approach

Cooperation as a micro level phenomenon in which the center of analysis is the dyad (Axelrod and Hamilton, 1981; Watts, 1999; Eguíluz et al., 2005).

- Reductionist approach: Cooperation as an atomic process.
  - Strategic dyadic interactions.
  - Agent-based models.
  - Payoffs of different strategies.

Theoretical Approaches to Cooperation

Macro level approach

Cooperation as a macro level phenomenon in which the center of analysis is the collective or group (Marx, 1990; Adler and Heckscher, 2006; Adler, 2015).

- Focus on large organizations and groups: Collaborative Communities
  - Shared values and goals
  - Generalized trust
  - Authority forms

Micro level approach

Cooperation as a micro level phenomenon in which the center of analysis is the dyad (Axelrod and Hamilton, 1981; Watts, 1999; Eguíluz et al., 2005).

- Reductionist approach: Cooperation as an atomic process.
  - Strategic dyadic interactions.
  - Agent-based models.
  - Payoffs of different strategies.

A Meso level approach to Cooperation

Focus on Cooperation networks: patterns of relations that direct producers establish in the production process.

- Structural approach, that is, a network approach.
  - Sub-groups that are more connected internally than with the rest of the network.
  - Longitudinal analysis of the formation and dissolution of these groups.
  - Key mechanisms to explain and understand large scale cooperation.

Main Topic and Research Question

The central topic of this thesis is to understand and explain under which conditions and through which social mechanisms large scale cooperation operates.

Free and Open Source Software (FOSS) as a case study

Free Software is computer software that allows users to run, copy, distribute, study, change and improve it.

- Availability of detailed data about the production process.
- Impressive increment of scale in the last two decades.
- Allows to analyze the development methods of FOSS from the perspective of a knowledge based production process.
- focusing on the patterns of relations between direct producers we can analyze key mechanisms that enable and foster large scale cooperation.

Main Topic and Research Question

The central topic of this thesis is to understand and explain under which conditions and through which social mechanisms large scale cooperation operates.

Free and Open Source Software (FOSS) as a case study

Free Software is computer software that allows users to run, copy, distribute, study, change and improve it.

- Availability of detailed data about the production process.
- Impressive increment of scale in the last two decades.
- Allows to analyze the development methods of FOSS from the perspective of a knowledge based production process.
- focusing on the patterns of relations between direct producers we can analyze key mechanisms that enable and foster large scale cooperation.

Research question

How does large scale cooperation work in knowledge intensive and technically complex production processes developed in new organizational environments, such as Free and Open Source Software projects, where loosely coupled individuals that rarely meet face to face have to coordinate through internet in order to produce world class software products, without relying on hierarchy or market as coordinating mechanisms.

Literature on FOSS

The literature has focused on 4 main areas:

1. individual incentives.
2. innovation and intellectual property policy.
3. development methods (mainly from Computer Science).
4. organization and governance.

Literature on FOSS

The literature has focused on 4 main areas:

1. individual incentives.
2. innovation and intellectual property policy.
3. development methods (mainly from Computer Science).
4. organization and governance.

This research focuses on the intersection of areas 3 and 4

- analyze how individuals coordinate their actions in order to achieve collective outcomes.
- focus on the patterns of relations of direct producers to analyze meso level mechanisms that enable and foster large scale cooperation.
- FOSS projects as a case study; a proxy to analyze large scale cooperation in the context of new organizational forms.
- FOSS tenets are closely related to the ethos of science as described by Merton (1942), and thus analyzing FOSS we can generalize to other large scale cooperation social processes.

Contents

Group Cohesion in the sociological literature

Central concept that has a long and illustrious history in sociology. Its use in most sociological research has been ambiguous at best (Moody and White, 2003):

1. sloppy definitions of cohesion with lack of generality.
2. grounded mostly in intuition and common sense.

Group cohesion (Doreian and Fararo, 1998) can be divided analytically into:

Ideational component based on the members' identification with a collectivity.

Relational component based on the patterns of connections among members.

Group Cohesion in the sociological literature

Central concept that has a long and illustrious history in sociology. Its use in most sociological research has been ambiguous at best (Moody and White, 2003):

1. sloppy definitions of cohesion with lack of generality.
2. grounded mostly in intuition and common sense.

Group cohesion (Doreian and Fararo, 1998) can be divided analytically into:

Ideational component based on the members' identification with a collectivity.

Relational component based on the patterns of connections among members.

The relational component of group cohesion has been the focus of Social Network Analysis.

Network theory measures used to define group cohesion

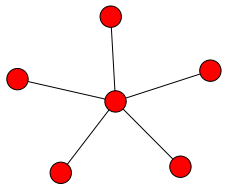
Classical measures cliques, clans, clubs, k-cores, lambda sets, ...

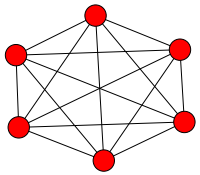
Community algorithms detect groups of nodes more densely connected among them than with the rest of the network.

Neither the classical approaches nor new developments in community analysis work well in empirical analysis of group cohesion.

Key properties that a cohesion measures should have

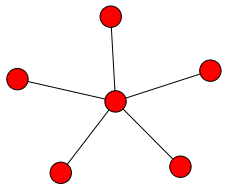
Robustness its qualification as a group should not be dependent on the actions of a single individual.

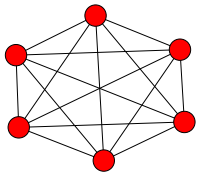




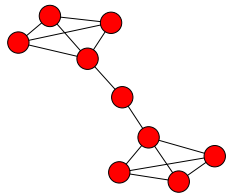
Key properties that a cohesion measures should have

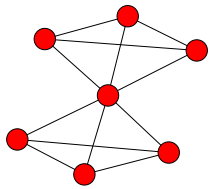
Robustness its qualification as a group should not be dependent on the actions of a single individual.





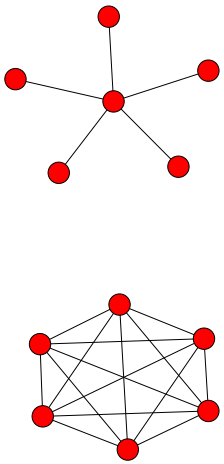
Overlap a cohesion measure should allow some actors to be part of more than one cohesive subgroup.



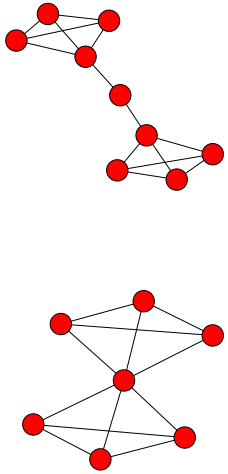


Key properties that a cohesion measures should have

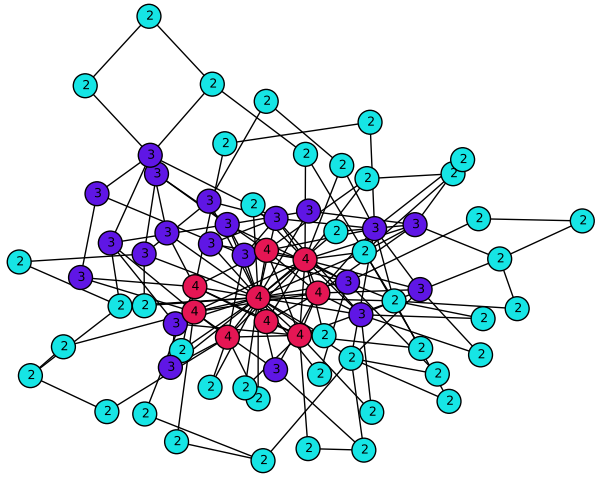
Robustness its qualification as a group should not be dependent on the actions of a single individual.



Overlap a cohesion measure should allow some actors to be part of more than one cohesive subgroup.



Hierarchical highly cohesive subgroups are nested inside less cohesive ones.



The structural cohesion model

The structural cohesion model (White and Harary, 2001; Moody and White, 2003) is based on two mathematically equivalent definitions of cohesion (by Menger's theorem).

Precise definition of group cohesion based on node connectivity

- a group's structural cohesion is equal to the minimum number of actors who, if removed from the group, would disconnect the group.
- a group's structural cohesion is equal to the minimum number of node independent paths linking each pair of actors in the group.

This equivalence relation has a deep sociological meaning because it allows to define structural cohesion in terms of:

- the difficulty to pull a group apart by removing actors.
- multiple relations between actors that keep a group together.

Components, node connectivity, and k-components

Component

A *component* of a graph G is a maximal connected subgraph, which means that there is at least one path between any two nodes in that subgraph.

Node connectivity $\kappa(G)$

The minimum number of nodes that must be removed in order to disconnect a graph G.

k-components

A k-component is a maximal subgraph that has, at least, node connectivity k: we need to remove at least k nodes to break it into more components.

Components, node connectivity, and k-components

Component

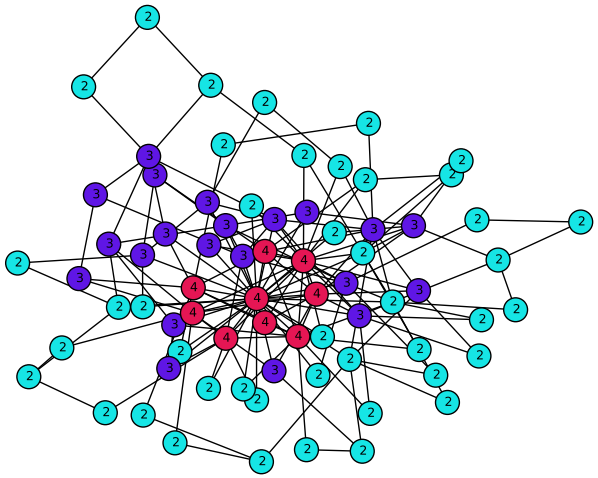
A *component* of a graph G is a maximal connected subgraph, which means that there is at least one path between any two nodes in that subgraph.

Node connectivity $\kappa(G)$

The minimum number of nodes that must be removed in order to disconnect a graph G.

k-components

A k-component is a maximal subgraph that has, at least, node connectivity k: we need to remove at least k nodes to break it into more components.



- **Red Nodes** form a 4-component: we need to remove 4 nodes to disconnect it.
- **Purple Nodes** form a 3-component along with red nodes: we need to remove 3 nodes to disconnect it.
- **Blue Nodes** form a 2-component along with red and purple nodes: we need to remove 2 nodes to disconnect it.

Structural cohesion in empirical analysis

Structural cohesion is a powerful and robust explanatory factor for a wide variety of interesting empirical social phenomena:

- likelihood of building alliances and partnerships among biotech firms (Powell et al., 2005)
- school attachment and academic performance of young people and the tendency of firms to enroll in similar political activity behaviors (Moody and White, 2003)
- emerging trust relations among neighborhood residents and the hiring relations among top level US graduate programs (Grannis, 2009)

Structural cohesion in empirical analysis

Structural cohesion is a powerful and robust explanatory factor for a wide variety of interesting empirical social phenomena:

- likelihood of building alliances and partnerships among biotech firms (Powell et al., 2005)
- school attachment and academic performance of young people and the tendency of firms to enroll in similar political activity behaviors (Moody and White, 2003)
- emerging trust relations among neighborhood residents and the hiring relations among top level US graduate programs (Grannis, 2009)

Some problems with the structural cohesion model

Despite all its merits, the structural cohesion model has not been widely applied to empirical analysis because:

- it is not practical to compute it for networks with more than a few hundreds nodes and edges due to its computational complexity.
- it was not implemented in most popular network analysis software packages.

Methodological Contributions: Algorithms and Heuristics

Implementation of connectivity measures

All contributed to NetworkX (Hagberg et al., 2008).

- Node and edge connectivity algorithms.
- Kanevsky (1993) algorithm for finding all minimum-size node cut-sets of a graph.
- Moody and White (2003) algorithm for identifying *k*-components.

Developed heuristics for fast computation

- Allow computing the approximate value of group cohesion for moderately large networks, one order of magnitude bigger than the ones manageable by the exact algorithm.
- The main logic of the heuristics is to repeatedly applying fast algorithms for *k*-cores and biconnected components in order to narrow down the number of pairs of nodes over which we have to compute node connectivity.

Methodological Contributions: Algorithms and Heuristics

Implementation of connectivity measures

All contributed to NetworkX (Hagberg et al., 2008).

- Node and edge connectivity algorithms.
- Kanevsky (1993) algorithm for finding all minimum-size node cut-sets of a graph.
- Moody and White (2003) algorithm for identifying *k*-components.

Developed heuristics for fast computation

- Allow computing the approximate value of group cohesion for moderately large networks, one order of magnitude bigger than the ones manageable by the exact algorithm.
- The main logic of the heuristics is to repeatedly applying fast algorithms for *k*-cores and biconnected components in order to narrow down the number of pairs of nodes over which we have to compute node connectivity.

Trade Accuracy for Speed

	Bipartite				Unipartite			
Network	# nodes	# edges	Av. degree	Time(s)	# nodes	# edges	Av. degree	Time(s)
Debian Lenny	13,121	20,220	3.08	1,105.2	1,383	5,216	7.54	204.7
High Energy (theory)	26,590	37,566	2.81	3,105.7	9,767	19,331	3.97	7,136.0
Nuclear Theory	10,371	15,969	3.08	1,205.2	4,827	14,488	6.00	3,934.1

Contents

Collaborative Communities

A new form of community, qualitatively different from the traditional *Gemeinschaft* and the modern *Gesellschaft* (Tönnies, 1974).

Collaborative Communities (Adler and Heckscher, 2006)

Novel organizational form —both inside and outside large capitalist corporations— strongly grounded on large scale cooperation which defy the traditional dichotomy between **hierarchy** and **market** as coordinating mechanisms.

- Generalized trust based on other's contributions towards a shared end.
- Conscious cooperation and high individual interdependence.
- Shared values and value-rational basis for legitimate authority.

Collaborative Communities

A new form of community, qualitatively different from the traditional *Gemeinschaft* and the modern *Gesellschaft* (Tönnies, 1974).

Collaborative Communities (Adler and Heckscher, 2006)

Novel organizational form —both inside and outside large capitalist corporations— strongly grounded on large scale cooperation which defy the traditional dichotomy between **hierarchy** and **market** as coordinating mechanisms.

- Generalized trust based on other's contributions towards a shared end.
- Conscious cooperation and high individual interdependence.
- Shared values and value-rational basis for legitimate authority.

So far scarce attention is given to the structural features of collaborative communities.

Question explore the network structure that lead to their emergence and effectiveness in the production and diffusion of knowledge.

Contribution I suggest that a unique network structure undergirds collaborative communities and build a model to understand its key mechanisms: the **cohesive small world** model.

knowledge-based production processes

Collaborative Communities are especially well suited to deal with the challenges of knowledge-based production processes because, hierarchy and market have proved ineffective, at best, at managing knowledge.

Hierarchy as a coordinating mechanism

- Knowledge is treated as a scare resource.
- Centralized at the higher levels where key decisions are taken.
- Rigidity prevents flexibility to deal with unanticipated problems.
- Difficulty to foster innovation and generation of new knowledge.

knowledge-based production processes

Collaborative Communities are especially well suited to deal with the challenges of knowledge-based production processes because, hierarchy and market have proved ineffective, at best, at managing knowledge.

Hierarchy as a coordinating mechanism

- Knowledge is treated as a scare resource.
- Centralized at the higher levels where key decisions are taken.
- Rigidity prevents flexibility to deal with unanticipated problems.
- Difficulty to foster innovation and generation of new knowledge.

Market as coordinating mechanism

- Fails to optimize at the same time production and allocation of knowledge.
- Knowledge is a public good that grow rather than diminish with use.
- Strong property rights incentive knowledge production, but
- they block socially optimal allocation (free access).

knowledge-based production processes

Collaborative Communities are especially well suited to deal with the challenges of knowledge-based production processes because, hierarchy and market have proved ineffective, at best, at managing knowledge.

Hierarchy as a coordinating mechanism

- Knowledge is treated as a scare resource.
- Centralized at the higher levels where key decisions are taken.
- Rigidity prevents flexibility to deal with unanticipated problems.
- Difficulty to foster innovation and generation of new knowledge.

Market as coordinating mechanism

- Fails to optimize at the same time production and allocation of knowledge.
- Knowledge is a public good that grow rather than diminish with use.
- Strong property rights incentive knowledge production, but
- they block socially optimal allocation (free access).

Collaborative Communities that excel at knowledge-based production

- Free and Open Source Software communities.
- Novel forms of professional work organization (Adler, Kwon, and Heckscher, 2008)
- Knowledge-intensive production processes in corporations (Adler and Heckscher, 2006)

Social Structure of Value Congruence and Generalized Trust

Importance of Authority

- Exclusivism and elitism are potential problems of Collaborative Communities. Effective authority is key to counter balance those downsides.
- The value-rational form of authority is essential to Collaborative Communities. It requires value congruence.
- Avoid the “iron cage” of instrumental-rational authority.

Social Structure of Value Congruence and Generalized Trust

Importance of Authority

- Exclusivism and elitism are potential problems of Collaborative Communities. Effective authority is key to counter balance those downsides.
- The value-rational form of authority is essential to Collaborative Communities. It requires value congruence.
- Avoid the “iron cage” of instrumental-rational authority.

Trust as the key mechanism

- Contribution-based trust as the primary social mechanism of collaborative communities.
- Reduces both transaction costs and agency risks.
- Mitigates coordination difficulties created by knowledge’s public good character: allows to combine different people with different sets of knowledge in order to solve complex problems.

Social Structure of Value Congruence and Generalized Trust

Importance of Authority

- Exclusivism and elitism are potential problems of Collaborative Communities. Effective authority is key to counter balance those downsides.
- The value-rational form of authority is essential to Collaborative Communities. It requires value congruence.
- Avoid the “iron cage” of instrumental-rational authority.

Trust as the key mechanism

- Contribution-based trust as the primary social mechanism of collaborative communities.
- Reduces both transaction costs and agency risks.
- Mitigates coordination difficulties created by knowledge’s public good character: allows to combine different people with different sets of knowledge in order to solve complex problems.

Social Structure as key element

- The literature on Collaborative Communities has focused on interdependence and common identities as the main elements that foster trust and value congruence.
- However we have ample evidence that the social structure plays a key role in enabling and fostering trust and value congruence (Granovetter, 1985; Coleman, 1988; Moody and White, 2003).
- I suggest that a unique social network structure undergirds collaborative communities, and facilitate the development of trust and value congruence.

A network model for Collaborative Communities

The Small World Model

Networks characterized by a high level of local density of social ties and short average distances among nodes.

- L: average number of intermediaries between any two nodes.
- CC: mean probability that two nodes that are neighbors of the same other node will themselves be neighbors.
- They foster the flow of information and ideas.

The Structural Cohesion Model

Networks characterized by the presence of increasingly cohesive groups nested inside each other.

- Cohesion definition based on the graph-theoretic property of connectivity.
- k-components as the key subgroups of the network.
- They foster the development of trust and social cohesion.

A network model for Collaborative Communities

The Small World Model

Networks characterized by a high level of local density of social ties and short average distances among nodes.

- L: average number of intermediaries between any two nodes.
- CC: mean probability that two nodes that are neighbors of the same other node will themselves be neighbors.
- They foster the flow of information and ideas.

The Structural Cohesion Model

Networks characterized by the presence of increasingly cohesive groups nested inside each other.

- Cohesion definition based on the graph-theoretic property of connectivity.
- k-components as the key subgroups of the network.
- They foster the development of trust and social cohesion.

These two models are not mutually exclusive

The networks that fit in the intersection of both models exhibit consistent structural patterns, which provide the structural scaffolding for collaborative communities.

The Cohesive Small World Model

Cohesive Small World Model

Network (structural) model for Collaborative Communities that help explain how trust, value congruence, and large scale cooperation are enabled and fostered.

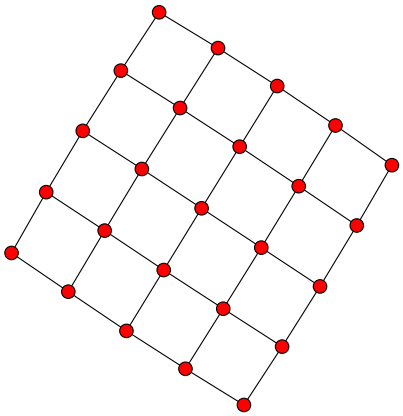
- Increasingly structurally cohesive groups nested inside each other.
  - Individuals embedded in them are able to compare independent perspectives on each other through a variety of relations that flow through distinct sets of intermediaries, which provides multiple independent sources of information about each other.
  - The perception of an individual of the other members of the group to whom she is not directly linked is filtered by the perception of a variety of others whom she trusts because is directly linked to them.
  - This mediated perception of the group generates trust and value congruence.
- Local clusters connected by short paths.
  - Allows successful cooperation among heterogeneous individuals with common interests and, at the same time, fosters the flow of information between these clusters preventing the local clusters to be trapped in echo chambers of like minded collaborators.
  - This structure prevents factionalism and fosters trust and value congruence,

Cohesive Small World

Pure Structural Cohesion

- Robust to node removal
- Not necessary short average distance
- Not necessary local clustering

Pure Structural Cohesion

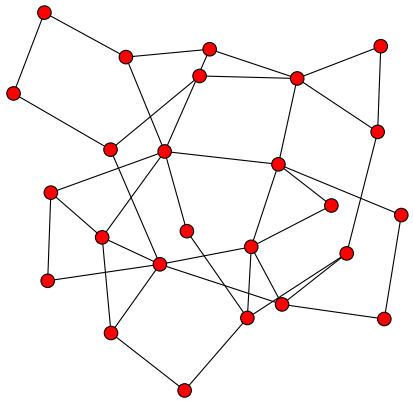


n=25, m=40, CC=0.00, APL=3.33, SWI=0.00, nodes in GBC=100%

Cohesive Small World

- Intersection of the two models
- Short average distance
- High local cluster coefficient
- Robust to node removal

Cohesive Small World

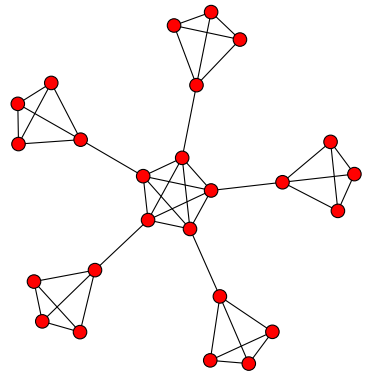


n=25, m=40, CC=0.19, APL=2.67, SWI=1.57, nodes in GBC=100%

Pure Small World

- Short average distance
- High local cluster coefficient
- Not necessary robust to node removal

Pure Small World



n=25, m=45, CC=0.82, APL=3.38, SWI=4.23, nodes in GBC=20%

Contents

Free and Open Source Projects: Debian and Python

Free Software, broadly defined, is computer software that allows users to run, copy, distribute, study, change and improve it.

The Debian Project: 1999-2012

- A free Operating System
- 392 developers in 1999, 1435 in 2012
- 2876 programs in 1999, 10469 in 2012
- Widely used in servers (google), desktops (Ubuntu) and embedded devices (raspberry pi)

The Python project: 1999-2014

- A free Programming Language
- 9 developers in 1999, 62 in 2014
- 1137 files in 1999, 2134 in 2014
- Widely used in web development (reddit, youtube) and scientific computing

Free and Open Source Projects: Debian and Python

Free Software, broadly defined, is computer software that allows users to run, copy, distribute, study, change and improve it.

The Debian Project: 1999-2012

- A free Operating System
- 392 developers in 1999, 1435 in 2012
- 2876 programs in 1999, 10469 in 2012
- Widely used in servers (google), desktops (Ubuntu) and embedded devices (raspberry pi)

The Python project: 1999-2014

- A free Programming Language
- 9 developers in 1999, 62 in 2014
- 1137 files in 1999, 2134 in 2014
- Widely used in web development (reddit, youtube) and scientific computing

The empirical analysis is a case study

- Its aim is to show that this two projects fit well the proposed Cohesive Small World model.
- This is not a empirical test of the model, it's rather an existence proof.

Cooperation Networks and Null Models

My modeling strategy to capture the patterns of relations among developers in these two projects is to focus on the actual contributions of each developer to the project.

Building Cooperation Networks

- Focus on the patterns of relations among developers in the productive process.
- Cooperation networks are bipartite or two mode; node sets are developers and programs/files and edges only link nodes from opposite sets.
- A developer is linked to the package (in Debian) or source code file (in Python) that she works on.
- We have complete electronic records of all package uplads to Debain and all sorce code file edits in Python.

Cooperation Networks and Null Models

My modeling strategy to capture the patterns of relations among developers in these two projects is to focus on the actual contributions of each developer to the project.

Building Cooperation Networks

- Focus on the patterns of relations among developers in the productive process.
- Cooperation networks are bipartite or two mode; node sets are developers and programs/files and edges only link nodes from opposite sets.
- A developer is linked to the package (in Debian) or source code file (in Python) that she works on.
- We have complete electronic records of all package uplads to Debain and all sorce code file edits in Python.

Building Suitable Null models

- We need to compare the statistical measures obtained from the actual networks with a suitable null model in order to assert that what we observe is not the result of pure chance.

Configuration Model assigns at random developers to packages, or developers to source code files, maintaining the concrete skewed distribution of packages by developer and files by developer observed in the actual networks. That is, the degree distribution.

The Cohesive Small World Model

The cooperation networks of both Debian and Python projects can be modeled using the proposed Cohesive Small World model; that is, they fit both the requirements of the Small World Model and the Structural Cohesion Model.

Small World Model

A network fits the small world model if it is more locally clustered than its random network null model but has approximately the same average distance between nodes.

Structural Cohesion Model

A network fits the structural cohesion model if it has more connectivity levels (k-components) than its random network null model and the percentage of nodes for $k \leq 2$ is comparable.

The Cohesive Small World Model

The cooperation networks of both Debian and Python projects can be modeled using the proposed Cohesive Small World model; that is, they fit both the requirements of the Small World Model and the Structural Cohesion Model.

Small World Model

A network fits the small world model if it is more locally clustered than its random network null model but has approximately the same average distance between nodes.

Structural Cohesion Model

A network fits the structural cohesion model if it has more connectivity levels (k-components) than its random network null model and the percentage of nodes for $k \leq 2$ is comparable.

But there are significative differences between Debian and Python.

Debian Cooperation Networks

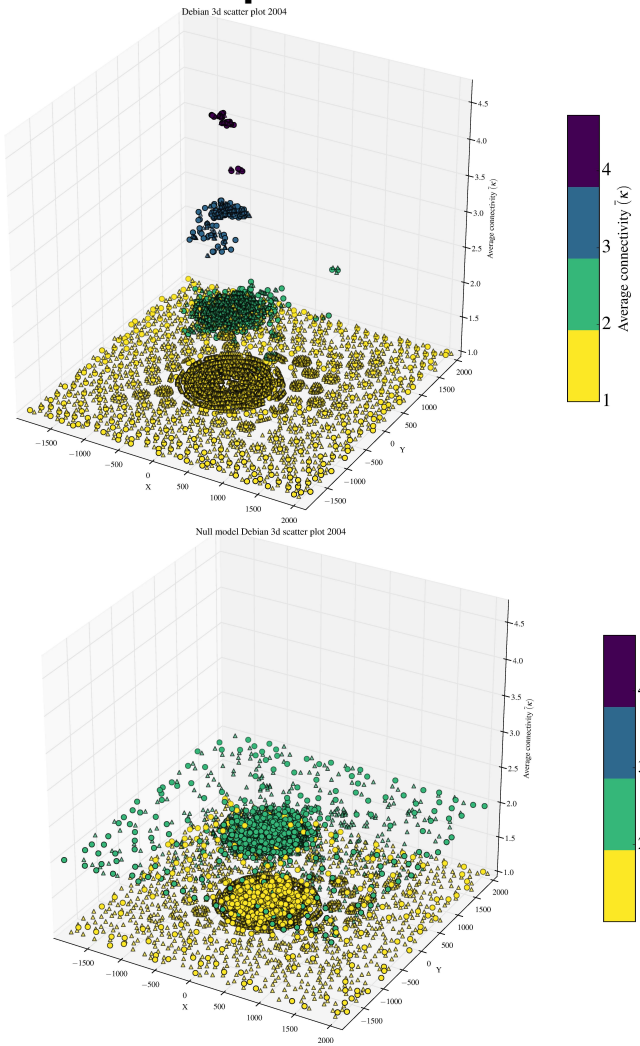
- Lean more towards the Small World Model.
- Subgroups work more independently from each other.
- High modularity.

Python Cooperation Networks

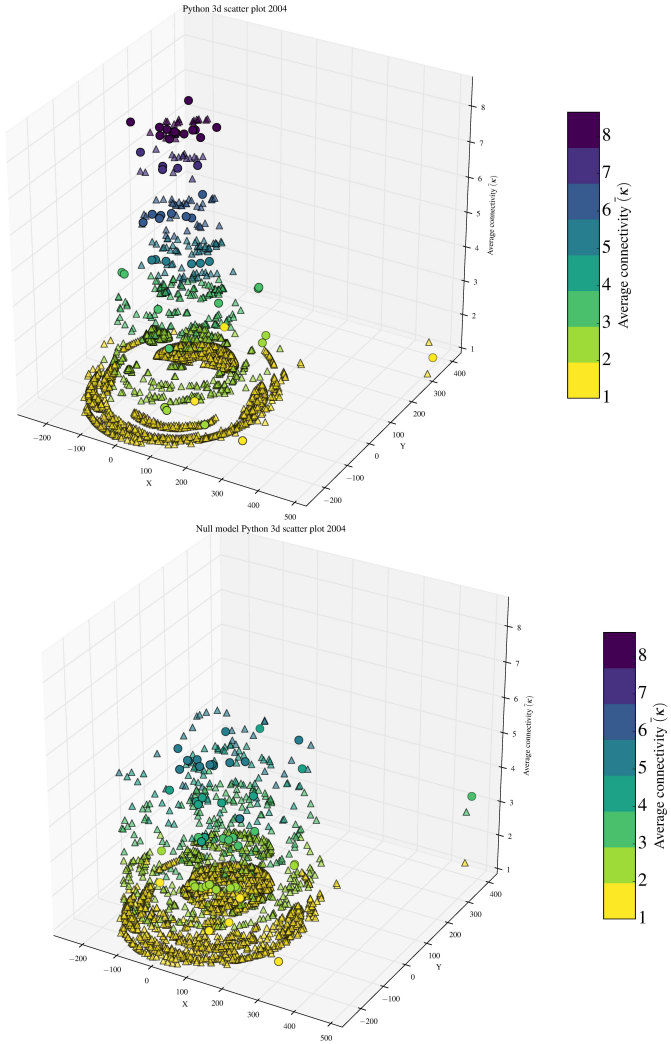
- Lean more towards the Structural Cohesion Model.
- Sharper and steep connectivity hierarchy.
- Less modularity.

Graphical Representation of the Structural Cohesion Analysis

Debian Cooperation Network 2004



Python Cooperation Network 2004



Contents

Dynamic Hierarchies as Open Elites

The analysis of the hierarchical structure of organizations has been a central topic on organizational research in the last decades.

Dynamic Analysis of Hierarchies

Focus the analysis on the ratio of renewal of the individuals in the positions defined by that hierarchy.

Open Elite "reconceptualization of the concept of elite, more as a fluidly reproduced ideal than as a stable demographic reality" (Padgett, 2010, 360).

Dynamic Hierarchies as Open Elites

The analysis of the hierarchical structure of organizations has been a central topic on organizational research in the last decades.

Dynamic Analysis of Hierarchies

Focus the analysis on the ratio of renewal of the individuals in the positions defined by that hierarchy.

Open Elite "reconceptualization of the concept of elite, more as a fluidly reproduced ideal than as a stable demographic reality" (Padgett, 2010, 360).

I propose that hierarchical structures can be classified in a continuum, the two extreme points of which are:

Static Hierarchy where when an individual is appointed in a position of the hierarchy, this position is for life.

Dynamic Hierarchy where the individuals occupying positions defined by the hierarchy have a very high pace of renewal.

The Hierarchical Structure of FOSS projects

Free and Open Source Software (FOSS) communities have attracted a lot of attention from researchers. Academic efforts took mainly two directions:

Reconcile with neoclassical economy

- Mainly focused on the individual motivations of the participants. What is usually referred as “microfundaments”.
- Trying to explain motivations in terms of of rational self-interested individuals to match the dominant economic accounts.

The Hierarchical Structure of FOSS projects

Free and Open Source Software (FOSS) communities have attracted a lot of attention from researchers. Academic efforts took mainly two directions:

Reconcile with neoclassical economy

- Mainly focused on the individual motivations of the participants. What is usually referred as “microfundaments”.
- Trying to explain motivations in terms of of rational self-interested individuals to match the dominant economic accounts.

Uncritically celebratory and a bit naive

- Supported the ethical stand that valued more cooperation and reciprocity than competition and self-interest.
- Assumed that FOSS communities are composed by loosely organized individuals with a very flat or nonexistent hierarchy among them.

The Hierarchical Structure of FOSS projects

Free and Open Source Software (FOSS) communities have attracted a lot of attention from researchers. Academic efforts took mainly two directions:

Reconcile with neoclassical economy

- Mainly focused on the individual motivations of the participants. What is usually referred as “microfundaments”.
- Trying to explain motivations in terms of of rational self-interested individuals to match the dominant economic accounts.

Uncritically celebratory and a bit naive

- Supported the ethical stand that valued more cooperation and reciprocity than competition and self-interest.
- Assumed that FOSS communities are composed by loosely organized individuals with a very flat or nonexistent hierarchy among them.

Well established empirical fact about FOSS projects

- Only few of the participants account for the lion’s share of the work done.

Empirical Analysis of Individual Contributions

Puzzle: FOSS projects as Oligarchies?

Michels (1915) “iron law of oligarchy” states that organizations tend towards oligarchy as they grow, even if democracy and participation are part of their core goals.

Empirical Analysis of Individual Contributions

Puzzle: FOSS projects as Oligarchies?

Michels (1915) “iron law of oligarchy” states that organizations tend towards oligarchy as they grow, even if democracy and participation are part of their core goals.

Hypothesis: FOSS projects as Open Elites

The continuous renewal of the people that does most of the work is a key mechanism to explain how FOSS projects can thrive and success through time.

Empirical Analysis of Individual Contributions

Puzzle: FOSS projects as Oligarchies?

Michels (1915) “iron law of oligarchy” states that organizations tend towards oligarchy as they grow, even if democracy and participation are part of their core goals.

Hypothesis: FOSS projects as Open Elites

The continuous renewal of the people that does most of the work is a key mechanism to explain how FOSS projects can thrive and success through time.

Longitudinal Analysis of the Social Structure of FOSS projects

If we analyze a FOSS project in a concrete point of time, only a very small fraction of the participants are the ones that actually do the lion’s share of contributions, as previous empirical research has shown.
However if we analyze the evolution of contributions longitudinally, we find that the persons that contribute the most change through time.

Empirical Analysis of Individual Contributions

Puzzle: FOSS projects as Oligarchies?

Michels (1915) “iron law of oligarchy” states that organizations tend towards oligarchy as they grow, even if democracy and participation are part of their core goals.

Hypothesis: FOSS projects as Open Elites

The continuous renewal of the people that does most of the work is a key mechanism to explain how FOSS projects can thrive and success through time.

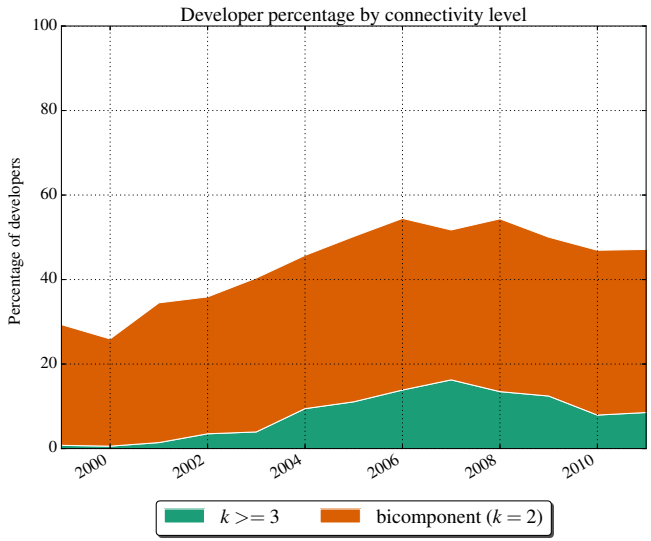
Longitudinal Analysis of the Social Structure of FOSS projects

If we analyze a FOSS project in a concrete point of time, only a very small fraction of the participants are the ones that actually do the lion’s share of contributions, as previous empirical research has shown.
However if we analyze the evolution of contributions longitudinally, we find that the persons that contribute the most change through time.

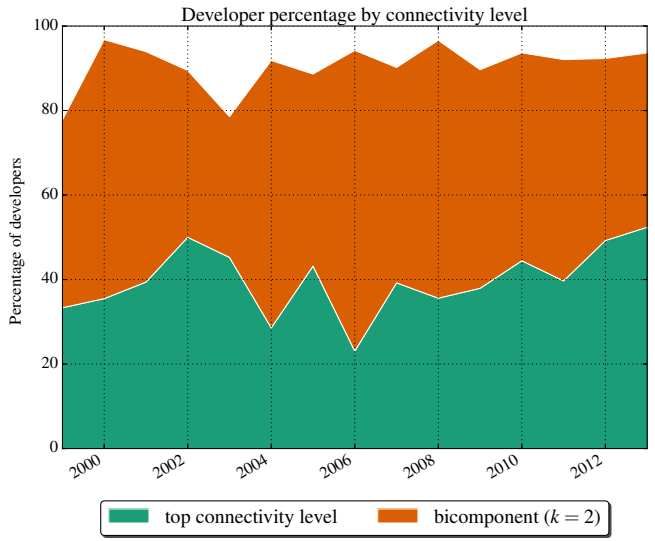
Main Empirical Results

- The developers that contribute the most are in the higher levels of the connectivity structure of the project’s cooperation networks.
- The ratio of renewal of individuals at these structural positions is quite fast, which characterizes FOSS communities as dynamic hierarchies and open elites.
- The position of an individual in the connectivity structure of the cooperation network also shapes the median active life of a developer in the project.

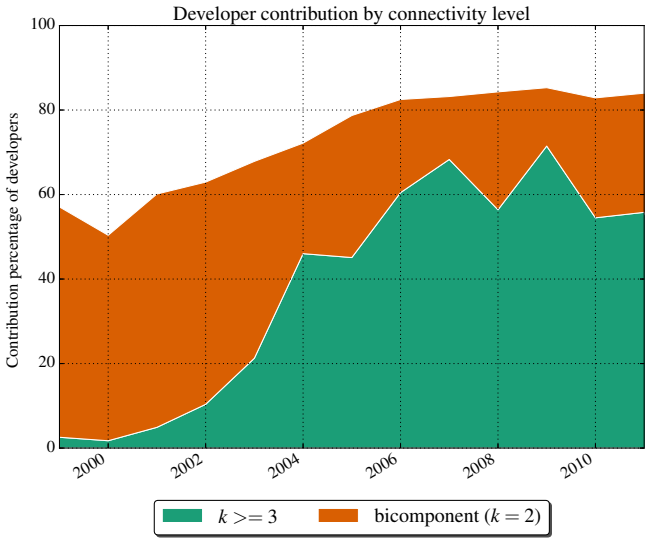
Individual contributions by connectivity level



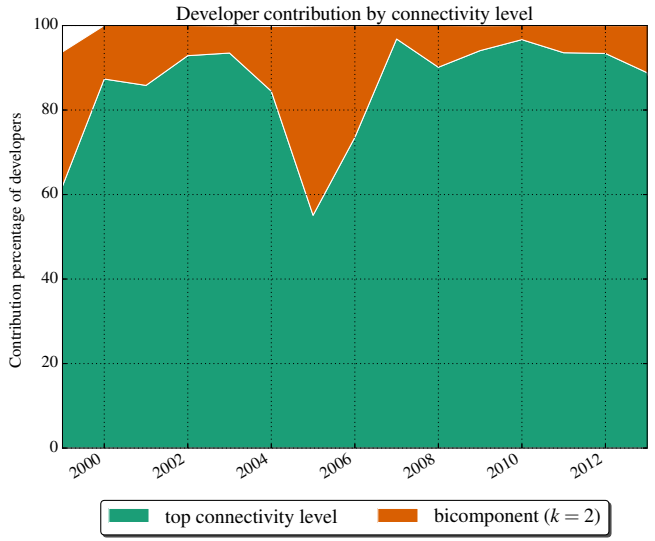
(a) Debian: developer % by connectivity level



(b) Python: developer % by connectivity level



(c) Debian: contributions by connectivity level



(d) Python: contributions by connectivity level

Analyzing Source Code Contributions and beyond

To further the analysis I used more sophisticated statistical modeling strategies to assess the impact of the connectivity structure in the individual contributions to the project.

- **Debian package uploads:** Modeled with a negative binomial regression. Over-dispersed and discrete dependent variable: # of uploads. The *k*-component number is the second most important independent variable only after Degree Centrality.
- **Added source code lines to Python:** Modeled with a panel regression with individual and year fixed effect. Dependent variable treated as a continuous variable. The *k*-component number is also the second most important independent variable only after Degree Centrality.

Analyzing Source Code Contributions and beyond

To further the analysis I used more sophisticated statistical modeling strategies to assess the impact of the connectivity structure in the individual contributions to the project.

- **Debian package uploads:** Modeled with a negative binomial regression. Over-dispersed and discrete dependent variable: # of uploads. The *k*-component number is the second most important independent variable only after Degree Centrality.
- **Added source code lines to Python:** Modeled with a panel regression with individual and year fixed effect. Dependent variable treated as a continuous variable. The *k*-component number is also the second most important independent variable only after Degree Centrality.

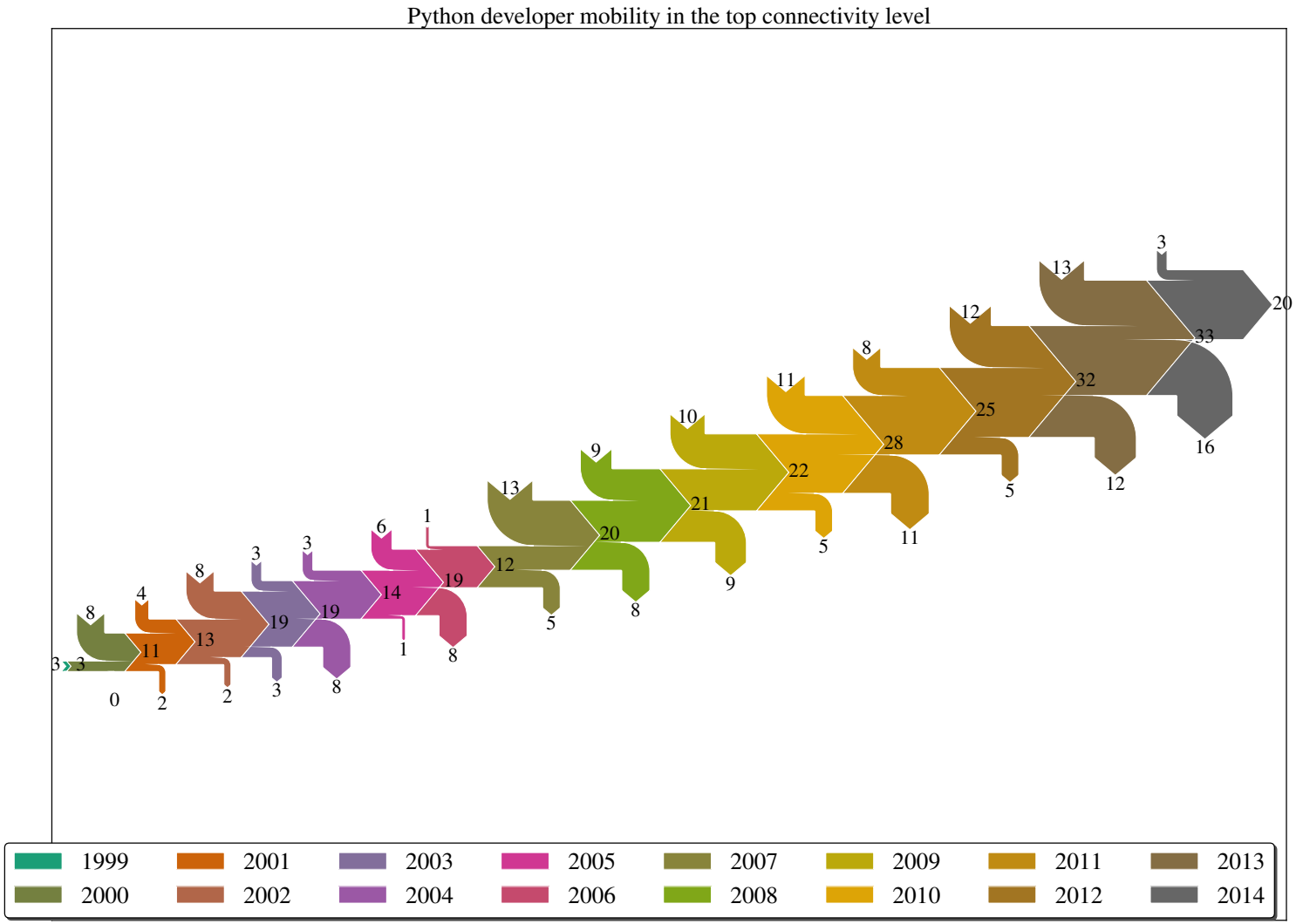
Potential endogenity problems of the previous models

There is a relation between the dependent variable and the network metrics (independent variables) because the cooperation networks are build precisely based on contributions to the projects.

- **Number of accepted *Python Enhancement Proposals* (PEP):** Modeled with a zero inflated negative binomial. PEPs are design documents that define the language; are thus contributions not directly related to the cooperation networks. Being part of the top connectivity level is also the second most important independent variable, this time only after developer's tenure (in years).

Cooperation Networks' Connectivity Hierarchies as Open Elites

The ratio of renewal of individuals at the top of the connectivity hierarchy is quite fast, which characterizes the CPython project as an open elite.



Modeling robustness of Cooperation Networks

Usual measures of robustness

- Failures: remove nodes at random and see how this affects the size of the giant component.
- Attacks: remove nodes incrementally starting for the ones with higher degree.

Modeling robustness of Cooperation Networks

Usual measures of robustness

- Failures: remove nodes at random and see how this affects the size of the giant component.
- Attacks: remove nodes incrementally starting for the ones with higher degree.

My approach: Survival analysis

- I model active life of a developer as the period that this developer is contributing to the project.
- I consider a developer “dead” when she no longer contributes to the project.

Modeling robustness of Cooperation Networks

Usual measures of robustness

- Failures: remove nodes at random and see how this affects the size of the giant component.
- Attacks: remove nodes incrementally starting for the ones with higher degree.

My approach: Survival analysis

- I model active life of a developer as the period that this developer is contributing to the project.
- I consider a developer “dead” when she no longer contributes to the project.

Cox proportional hazards model

- Time-dependent variables and right-censoring.
- Being part of the top connectivity level decreases the yearly hazard of leaving the project by 77%.
- Also an increment of one connectivity level decreases the yearly hazard of leaving the project by 26%.

Modeling robustness of Cooperation Networks

Usual measures of robustness

- Failures: remove nodes at random and see how this affects the size of the giant component.
- Attacks: remove nodes incrementally starting for the ones with higher degree.

My approach: Survival analysis

- I model active life of a developer as the period that this developer is contributing to the project.
- I consider a developer "dead" when she no longer contributes to the project.

Cox proportional hazards model

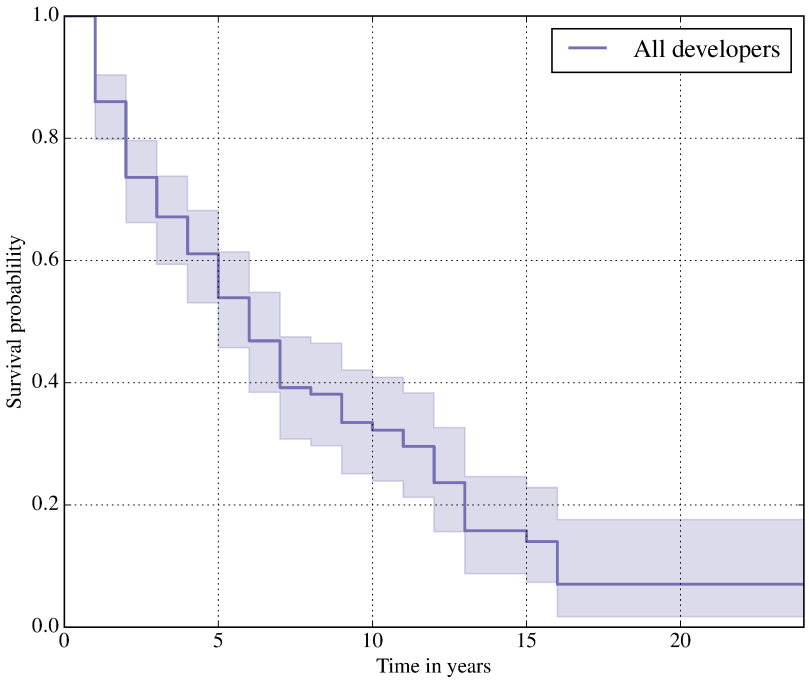
- Time-dependent variables and right-censoring.
- Being part of the top connectivity level decreases the yearly hazard of leaving the project by 77%.
- Also an increment of one connectivity level decreases the yearly hazard of leaving the project by 26%.

Estimating the survival function with the Kaplan-Meier estimator

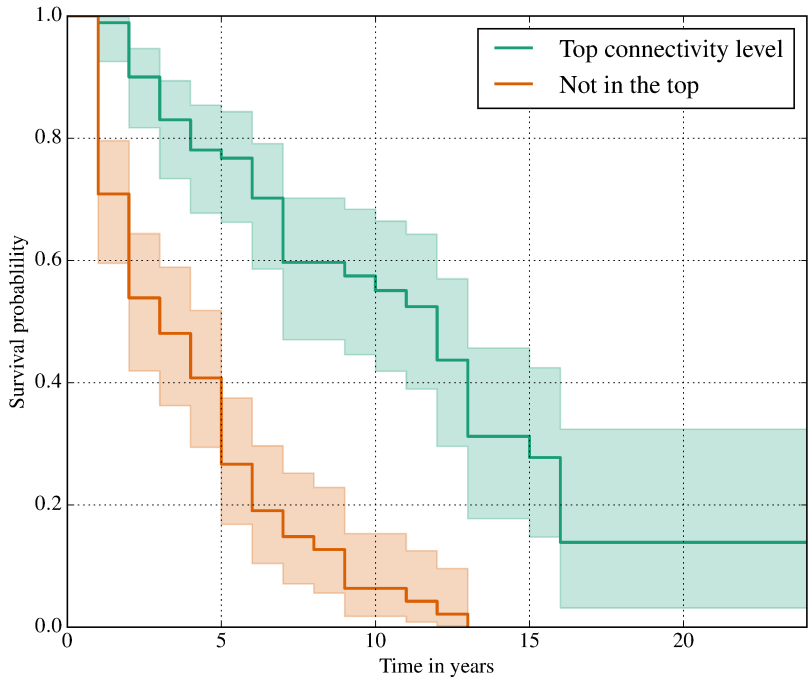
$$\hat{S}(t) = \prod_{t_i < t} \frac{n_i - d_i}{n_i}$$

where d_i are the number of "death events" at time t and n_i is the number of subjects at risk of death at time t.

Modeling robustness as median active live of individuals in the project



(a) Survival Function for all developers



(b) Survival Function for developers in the top connectivity level

Figure: Estimation of the survival function using the Kaplan-Meier estimate. The median survival time of a developer in the community, defined as the point in time where on average half of the population has abandoned the community, is 6 years if I consider all developers (left). But if I consider separately the developers in the top level of the connectivity hierarchy (right), their median survival time is 12 years; but only 3 years for the developers that are not on the top of the connectivity hierarchy.

Contents

Conclusions

- **Cooperation Networks:** A meso level approach to large scale cooperation focused on the patterns of relations that direct producers establish between them in knowledge intensive production processes.
- **The Cohesive Small World model**: The structural patterns defined by this model provide the scaffolding for enabling effective large scale cooperation.
  - **Structurally cohesive groups nested inside each other** that form a connectivity hierarchy. Provide a plausible mechanism of generation of trust and congruent values: individuals embedded in these structures are able to compare independent perspectives on each other through a variety of paths that flow through distinct sets of intermediaries.
  - **Dense local clusters connected by short paths** is a structural pattern that fosters information flow allowing successful cooperation among heterogeneous individuals with common interests.
- I **developed heuristics** to compute the *k*-components structure that allow for the computing of the approximate value of group cohesion for moderately large networks in a reasonable time frame.
- I **contributed an implementation** of these heuristics, and other related connectivity measures, to a popular Python software package for the analysis of complex networks: NetworkX (Hagberg et al., 2008).

Conclusions

- The **Cohesive Small World** model is a good fit to describe the cooperation networks of two big and mature FOSS projects: the Python programming language, and the Debian operating system.
- This model captures the **hierarchy in the patterns of relations** that individual contributors establish when working together.
  - reflects the empirically well established fact that in FOSS projects only a small fraction of the developers account for most of the contributions.
  - refutes the naive views of early academic accounts that characterized FOSS projects as a flat hierarchy of peers in which every individual does more or less the same.
- **Cooperation has a structural dimension** because membership in cohesive groups has an important and statistically significative impact on both the volume of individual contributions, and on the median active life of developers.
- The connectivity structure of these collaborative communities' cooperation networks can be characterized as an **open elite**, where the top levels of this hierarchy are filled with new individuals at a high pace.
- This is a **key mechanism** that allows FOSS communities to develop long term projects, with high individual turnover, and yet achieve high impact and coherent results as a result of large scale cooperation.
- **Socialization**, in Marxian terms, is a necessary condition for the **open elite dynamic**: replacing tacit knowledge generated in local contexts by knowledge that is explicitly codified and disseminated at a global level is what allows new individuals to join the top levels of the connectivity hierarchy at a high pace.

Limitations and Future Work

- It's not clear that the theoretical model that I proposed fits all cooperation networks of Collaborative Communities, in general, or even cooperation networks of FOSS projects, in particular.
- The empirical analysis presented in this thesis was aimed to develop a theoretical framework.
- The case study of two successful projects is therefore an existence proof, not an empirical test of the Cohesive Small World Model.
- The next steps in my research agenda will be to expand the kind of empirical analysis that I presented in this thesis to include:
  - Both successful and unsuccessful large scale cooperation projects. There is abundant data on failed FOSS projects.
  - Other knowledge intensive production processes beyond FOSS projects:
    - Scientific collaborations and coauthorship
    - State sponsored large scale scientific and technical projects (such as Space exploration)
    - Productions processes inside big capitalist corporations

References

Adler, P. (2007). The future of critical management studies: A paleo-marxist critique of labour process theory. *Organization Studies 28*(9), 1313.

Adler, P. and C. Heckscher (2006). *The Firm as a Collaborative Community: Reconstructing Trust in the Knowledge Economy*. Oxford University Press, Oxford, UK.

Adler, P., S. Kwon, and C. Heckscher (2008). Professional work: The emergence of collaborative community. *Organization Science 19*(2).

Adler, P. S. (2015). Community and innovation: from tönnies to marx. *Organization Studies 36*(4), 445–471.

Axelrod, R. and W. Hamilton (1981). The evolution of cooperation. *Science 211*(4489), 1390–1396.

Batagelj, V. and M. Zaveršnik (2011). Fast algorithms for determining (generalized) core groups in social networks. *Advances in Data Analysis and Classification 5*(2), 129–145.

Beineke, L., O. Oellermann, and R. Pippert (2002). The average connectivity of a graph. *Discrete mathematics 252*(1-3), 31–45.

Coleman, J. (1988). Social Capital in the Creation of Human Capital. *American Journal of Sociology*, 95–120.

Doreian, P. and T. Fararo (1998). *The problem of solidarity: theories and models*. Routledge.

Eguíluz, V., M. Zimmermann, C. Cela-Conde, and M. Miguel (2005). Cooperation and the Emergence of Role Differentiation in the Dynamics of Social Networks 1. *American journal of sociology 110*(4), 977–1008.

Grannis, R. (2009). Paths and semipaths: reconceptualizing structural cohesion in terms of directed relations. *Sociological Methodology 39*(1), 117–150.

Granovetter, M. (1985). Economic action and social structure: the problem of embeddedness. *American Journal of Sociology 91*(3), 481.

Hagberg, A., D. Schult, and P. Swart (2008, August). Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Pasadena, CA USA, pp. 11–15.

Kanevsky, A. (1993). Finding all minimum-size separating vertex sets in a graph. *Networks 23*(6), 533–541.

Marx, K. (1990). *Capital: a critique of political economy. Volume 1*. Penguin Books in association with New Left Review.

Merton, R. K. (1942). Science and technology in a democratic order. *Journal of Legal and Political Sociology 1*, 115–126.

Michels, R. (1915). *Political parties: A sociological study of the oligarchical tendencies of modern democracy*. Hearst's International Library Company.

Moody, J. and D. White (2003). Social cohesion and embeddedness: A hierarchical conception of social groups. *American Sociological Review 68*(1), 103–28.

Padgett, J. F. (2010). Open elite? social mobility, marriage, and family in florence, 1282–1494*. *Renaissance quarterly 63*(2), 357–411.

Powell, W., D. White, K. Koput, and J. Owen-Smith (2005). Network dynamics and field evolution: The growth of interorganizational collaboration in the life sciences. *American journal of sociology 110*(4), 1132–1205.

Seidman, S. (1983). Network structure and minimum degree. *Social networks 5*(3), 269–287.

Tarjan, R. (1972). Depth-first search and linear graph algorithms. In *Switching and Automata Theory, 1971., 12th Annual Symposium on*, pp. 114–121. IEEE.

Tönnies, F. (1974). *Community and association:(Gemeinschaft und Gesellschaft)*. Taylor & Francis.

Watts, D. (1999). *Small Worlds: The Dynamics of Networks Between Order and Randomness*. Princeton University Press.

White, D. and F. Harary (2001). The cohesiveness of blocks in social networks: Node connectivity and conditional density. *Sociological Methodology*, 305–359.

White, D. and M. Newman (2001). Fast approximation algorithms for finding node-independent paths in networks. *Santa Fe Institute Working Papers Series*.

Wuchty, S., B. Jones, and B. Uzzi (2007). The increasing dominance of teams in production of knowledge. *Science 316*(5827), 1036.

Contents

Appendix: Measures of structural cohesion

Node connectivity $\kappa(G)$

Local given two nodes u and v, $\kappa_G(u, v)$ is the minimum number of nodes that must be removed to destroy all paths that join u and v.

Global the minimum number of nodes that must be removed in order to disconnect a graph G.

$$\kappa = min\{\kappa_G(u, v) : u, v \in V(G)\}$$

Average node connectivity $\bar{\kappa}(G)$

the sum of local node connectivity between all pairs of different nodes of G divided by the number of distinct pairs of nodes. (Beineke, Oellermann, and Pippert, 2002).

$$\bar{\kappa}(G) = \frac{\sum_{u,v} \kappa_G(u, v)}{\binom{n}{2}}$$

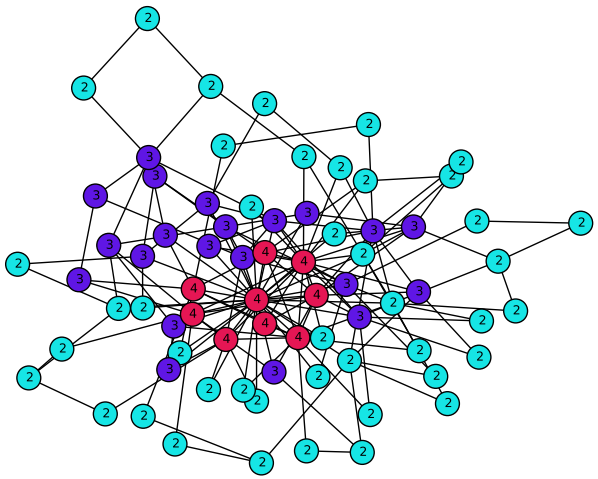
Appendix: Components and *k*-components

Component

A *component* of a graph *G* is a maximal connected subgraph, which means that there is at least one path between any two nodes in that subgraph.

k-components

A *k*-component is a maximal subgraph that has, at least, node connectivity *k*: we need to remove at least *k* nodes to break it into more components.



- **Red Nodes** form a 4-component: we need to remove 4 nodes to disconnect it.
- **Purple Nodes** form a 3-component along with red nodes: we need to remove 3 nodes to disconnect it.
- **Blue Nodes** form a 2-component along with red and purple nodes: we need to remove 2 nodes to disconnect it.

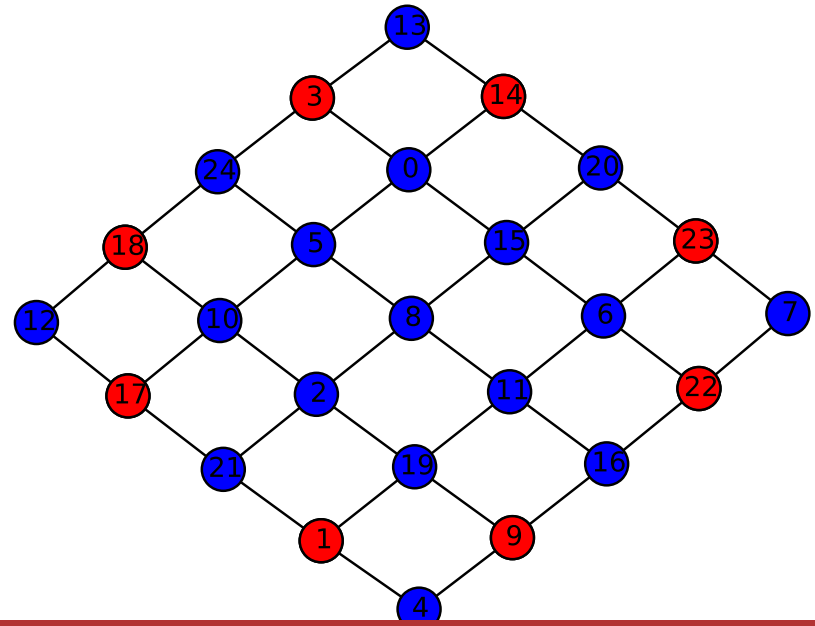
Appendix: Exact Algorithm

Moody and White (2003, appendix A) provide an algorithm for identifying k-components in a network, which is based on the Kanevsky (1993) algorithm for finding all minimum-size node cut-sets of a graph.

1. Identify the node connectivity, k, of the input graph.
2. Identify all k-cutsets at the current level of connectivity.
3. Generate new graph components based on the removal of these cutsets.
4. If the graph is neither complete nor trivial, return to 1; otherwise end.

```
import networkx as nx
from networkx import convert_node_lat

G = cnlti(nx.grid_2d_graph(5,5))
list(nx.all_node_cuts(G))
[{1, 9}, {22, 23}, {17, 18}, {3, 14}]
# To compute k-components
kcomponents = nx.k_components(G)
```



Appendix: Heuristics

The heuristics I developed are based on:

Approximation to local node connectivity

White and Newman (2001) fast approximation algorithm for finding good lower bounds of the number of node independent paths between two nodes.

```
import networkx as nx
from networkx.algorithms import approximation as apxa
# Exact computation of node connectivity
exact = nx.node_connectivity(G)
# Lower bound approximation
aprox = apxa.node_connectivity(G)
```

Whitney's theorem

inclusion relation among node connectivity $\kappa(G)$, edge connectivity $\lambda(G)$ and minimum degree $\delta(G)$ for any graph G:

$$\kappa(G) \leq \lambda(G) \leq \delta(G)$$

"k-cores can be regarded as seedbeds, within which we can expect highly cohesive subsets to be found" Seidman (1983, 281)

Appendix: Heuristics

Main logic of the heuristics

1. repeatedly applying fast algorithms for k-cores (Batagelj and Zaveršnik, 2011) and biconnected components (Tarjan, 1972) in order to narrow down the number of pairs of different nodes over which we have to compute their local node connectivity.
2. Build an auxiliary graph H in which two nodes are linked if they have at least k node independent paths connecting them.
3. Complete subgraphs in H have a one to one correspondence with subgraphs of G in which each node is connected to every other node in the subgraph for at least k node independent paths. That is, k-components.

Appendix: Heuristics

Main logic of the heuristics

1. repeatedly applying fast algorithms for k-cores (Batagelj and Zaveršnik, 2011) and biconnected components (Tarjan, 1972) in order to narrow down the number of pairs of different nodes over which we have to compute their local node connectivity.
2. Build an auxiliary graph H in which two nodes are linked if they have at least k node independent paths connecting them.
3. Complete subgraphs in H have a one to one correspondence with subgraphs of G in which each node is connected to every other node in the subgraph for at least k node independent paths. That is, k-components.

Trade Accuracy for Speed

	Bipartite				Unipartite			
Network	# nodes	# edges	Av. degree	Time(s)	# nodes	# edges	Av. degree	Time(s)
Debian Lenny	13,121	20,220	3.08	1,105.2	1,383	5,216	7.54	204.7
High Energy (theory)	26,590	37,566	2.81	3,105.7	9,767	19,331	3.97	7,136.0
Nuclear Theory	10,371	15,969	3.08	1,205.2	4,827	14,488	6.00	3,934.1

Small World Analysis

A network fits the small world model if it is more locally clustered (CC) than its random network counterpart but has approximately the same average distance (L) between nodes.

$$Q = \frac{CC_{ratio}}{L_{ratio}} \qquad \textit{Where:} \qquad CC_{ratio} = \frac{CC_{actual}}{CC_{random}} \qquad L_{ratio} = \frac{L_{actual}}{L_{random}} \tag{1}$$

If the Small World Index (Q) is greater than 1, then the network fits the model.

Small World Analysis

A network fits the small world model if it is more locally clustered (CC) than its random network counterpart but has approximately the same average distance (L) between nodes.

$$Q = \frac{CC_{ratio}}{L_{ratio}} \qquad \textit{Where:} \qquad CC_{ratio} = \frac{CC_{actual}}{CC_{random}} \qquad L_{ratio} = \frac{L_{actual}}{L_{random}} \tag{1}$$

If the Small World Index (Q) is greater than 1, then the network fits the model.

The Debian Project: 1999-2012

- $CC_{actual} \gg CC_{random}$ for all years.
- $L_{actual} > L_{random}$ for all years.
- $Q \gg 1$ for all years (21.4 in 2011 to 97 in year 2000).

The Python project: 1999-2014

- $CC_{actual} > CC_{random}$ for all years.
- $L_{actual} \approx L_{random}$ for all years.
- $Q > 1$ for all years (ranging from 3 in 1999 to 1.3 in 2011).

The low value of the Small World Index (Q) in the Python project, compared with the values of Q in the Debian project, can be attributed to the lower modularity of Python as a programming language compared with the inherent modularity of the Debian project as an operating system.

Small world metrics for debian networks

Years	Nodes	Developers	Packages	Edges	CC	random CC	APL	random APL	SWI (Q)
1999	3,259	392	2,867	3,253	0.128	0.002	9.4	8.6	73.8
2000	3,593	524	3,069	3,501	0.134	0.001	9.4	8.9	97.0
2001	5,943	777	5,166	6,241	0.049	0.002	8.4	7.8	28.1
2002	6,857	858	5,999	7,215	0.081	0.001	9.2	7.8	47.6
2003	7,276	914	6,362	7,892	0.101	0.001	9.1	7.6	57.9
2004	7,984	995	6,989	9,543	0.158	0.002	8.0	6.5	52.6
2005	8,328	1,048	7,280	10,373	0.166	0.003	7.5	6.2	43.5
2006	9,599	1,162	8,437	13,081	0.171	0.005	6.7	5.6	30.4
2007	9,471	1,181	8,290	13,023	0.148	0.005	6.8	5.6	26.2
2008	10,662	1,269	9,393	14,531	0.187	0.005	7.2	5.6	31.8
2009	11,336	1,343	9,993	15,842	0.227	0.006	7.0	5.3	28.6
2010	10,515	1,387	9,128	14,063	0.277	0.005	7.7	5.5	40.0
2011	12,362	1,430	10,932	16,265	0.143	0.005	7.5	5.5	21.4
2012	11,904	1,435	10,469	15,356	0.190	0.004	7.6	5.6	31.4

Table: Small world metrics for debian networks.

Small world metrics for python networks

Years	Nodes	Developers	Files	Edges	CC	random CC	APL	random APL	SWI (Q)
1999	1,146	9	1,137	1,236	0.102	0.039	3.1	3.5	3.0
2000	2,172	31	2,141	3,720	0.214	0.135	3.3	3.5	1.7
2001	2,511	33	2,478	4,507	0.205	0.129	3.4	3.6	1.7
2002	2,317	38	2,279	4,502	0.204	0.129	3.6	3.6	1.6
2003	1,805	42	1,763	3,192	0.153	0.112	3.5	3.6	1.4
2004	1,850	49	1,801	3,163	0.113	0.093	3.4	3.6	1.3
2005	1,007	44	963	1,759	0.129	0.079	3.7	3.7	1.7
2006	2,632	52	2,580	6,794	0.235	0.156	2.8	3.2	1.7
2007	3,359	51	3,308	7,790	0.223	0.177	2.9	3.3	1.4
2008	2,951	59	2,892	7,833	0.231	0.175	3.0	3.3	1.5
2009	2,219	58	2,161	4,708	0.228	0.142	3.1	3.4	1.7
2010	2,930	63	2,867	6,504	0.175	0.128	3.4	3.5	1.4
2011	2,174	63	2,111	4,459	0.145	0.114	3.5	3.6	1.3
2012	2,444	65	2,379	4,843	0.124	0.087	3.7	3.8	1.4
2013	2,285	63	2,222	4,743	0.147	0.099	3.6	3.7	1.5
2014	2,134	62	2,072	4,149	0.138	0.095	3.6	3.7	1.5

Table: Small world metrics for python networks.

Structural Cohesion Analysis

Analyze the percentage of nodes in k-components in actual networks and in their random counterparts.
For $k \leq 2$ the random null model is the upper bound of component size. Thus a network fits the model if it has more levels (k) than its random counterparts and the percentage of nodes for $k \leq 2$ is comparable.

Structural Cohesion Analysis

Analyze the percentage of nodes in k-components in actual networks and in their random counterparts.
For $k \leq 2$ the random null model is the upper bound of component size. Thus a network fits the model if it has more levels (k) than its random counterparts and the percentage of nodes for $k \leq 2$ is comparable.

The Debian Project: 1999-2012

- For 1-components $|V_{actual}| < |V_{random}|$
- For 2-components $|V_{actual}| < |V_{random}|$
- Maximum k:
  $3 \leq max(k_{actual}) \leq 5$ and
  $2 \leq max(k_{random}) \leq 3$, but always
  $max(k_{actual}) > max(k_{random})$.

The Python project: 1999-2014

- For 1-components $|V_{actual}| \approx |V_{random}|$
- For 2-components $|V_{actual}| \approx |V_{random}|$
- Maximum k:
  $5 \leq max(k_{actual}) \leq 10$ and
  $3 \leq max(k_{random}) \leq 7$, but always
  $max(k_{actual}) > max(k_{random})$.

The fact that cooperation networks in the Python project are more cohesive than the cooperation networks of the Debian project, can also be attributed in part to the lower modularity of Python as a programming language compared with the inherent modularity of the Debian project as an operating system.

Structural Cohesion metrics for debian networks

Years	Nodes	GC	Random GC	GBC	Random GBC	maximum k	Random max k
1999	3,259	66.6%	83.4%	9.4%	11.4%	3 (0.2%)	2 (11.4%)
2000	3,593	52.5%	77.8%	7.0%	10.7%	3 (0.2%)	2 (10.7%)
2001	5,943	71.6%	86.4%	13.9%	17.5%	3 (0.1%)	2 (17.5%)
2002	6,857	72.4%	88.1%	12.7%	17.0%	4 (0.2%)	2 (17.0%)
2003	7,276	75.6%	89.5%	14.8%	20.2%	5 (0.2%)	2 (20.2%)
2004	7,984	78.4%	94.4%	22.1%	27.7%	5 (0.2%)	2 (27.7%)
2005	8,328	83.8%	94.4%	26.1%	31.3%	4 (0.5%)	3 (4.5%)
2006	9,599	84.2%	96.7%	33.7%	39.0%	4 (0.6%)	3 (8.4%)
2007	9,471	86.5%	96.1%	35.6%	40.7%	4 (0.2%)	3 (8.6%)
2008	10,662	87.2%	96.4%	34.3%	40.3%	4 (0.6%)	3 (7.5%)
2009	11,336	89.4%	96.1%	35.7%	42.3%	5 (0.4%)	3 (8.2%)
2010	10,515	86.9%	95.5%	32.7%	39.8%	5 (0.2%)	3 (5.1%)
2011	12,362	87.7%	95.0%	30.6%	36.0%	5 (0.3%)	3 (5.3%)
2012	11,904	87.1%	95.0%	31.0%	36.7%	4 (0.1%)	3 (2.3%)

Table: Structural Cohesion metrics for debian networks.

Structural Cohesion metrics for python networks

Years	Nodes	GC	Random GC	GBC	Random GBC	maximum k	Random max k
1999	1146	66.0%	100.0%	6.7%	6.5%	3 (1.0%)	2 (6.5%)
2000	2172	96.5%	100.0%	33.4%	31.4%	8 (1.2%)	5 (3.1%)
2001	2511	97.3%	99.8%	34.1%	33.0%	9 (1.2%)	6 (2.4%)
2002	2317	100.0%	99.8%	38.1%	36.9%	9 (2.6%)	7 (1.9%)
2003	1805	100.0%	99.2%	34.8%	33.1%	7 (3.3%)	6 (2.4%)
2004	1850	99.8%	100.0%	39.7%	37.1%	7 (1.8%)	5 (2.5%)
2005	1007	99.8%	100.0%	45.7%	44.2%	5 (5.8%)	4 (7.6%)
2006	2632	100.0%	100.0%	74.2%	69.8%	9 (1.5%)	6 (3.9%)
2007	3359	100.0%	100.0%	58.6%	55.2%	9 (2.0%)	6 (2.2%)
2008	2951	100.0%	99.9%	64.5%	61.7%	10 (2.2%)	7 (2.5%)
2009	2219	100.0%	99.9%	51.0%	48.5%	7 (2.8%)	5 (5.6%)
2010	2930	100.0%	99.9%	48.7%	47.0%	9 (2.7%)	7 (2.4%)
2011	2174	100.0%	99.8%	47.7%	45.9%	8 (2.9%)	7 (1.7%)
2012	2444	99.8%	99.8%	41.1%	40.3%	8 (3.4%)	7 (3.4%)
2013	2285	99.9%	99.9%	51.6%	49.8%	7 (4.2%)	6 (4.0%)
2014	2134	100.0%	99.8%	44.6%	43.4%	7 (2.6%)	6 (3.1%)

Table: Structural Cohesion metrics for python networks.

Cooperation Networks' Connectivity Hierarchies as Open Elites

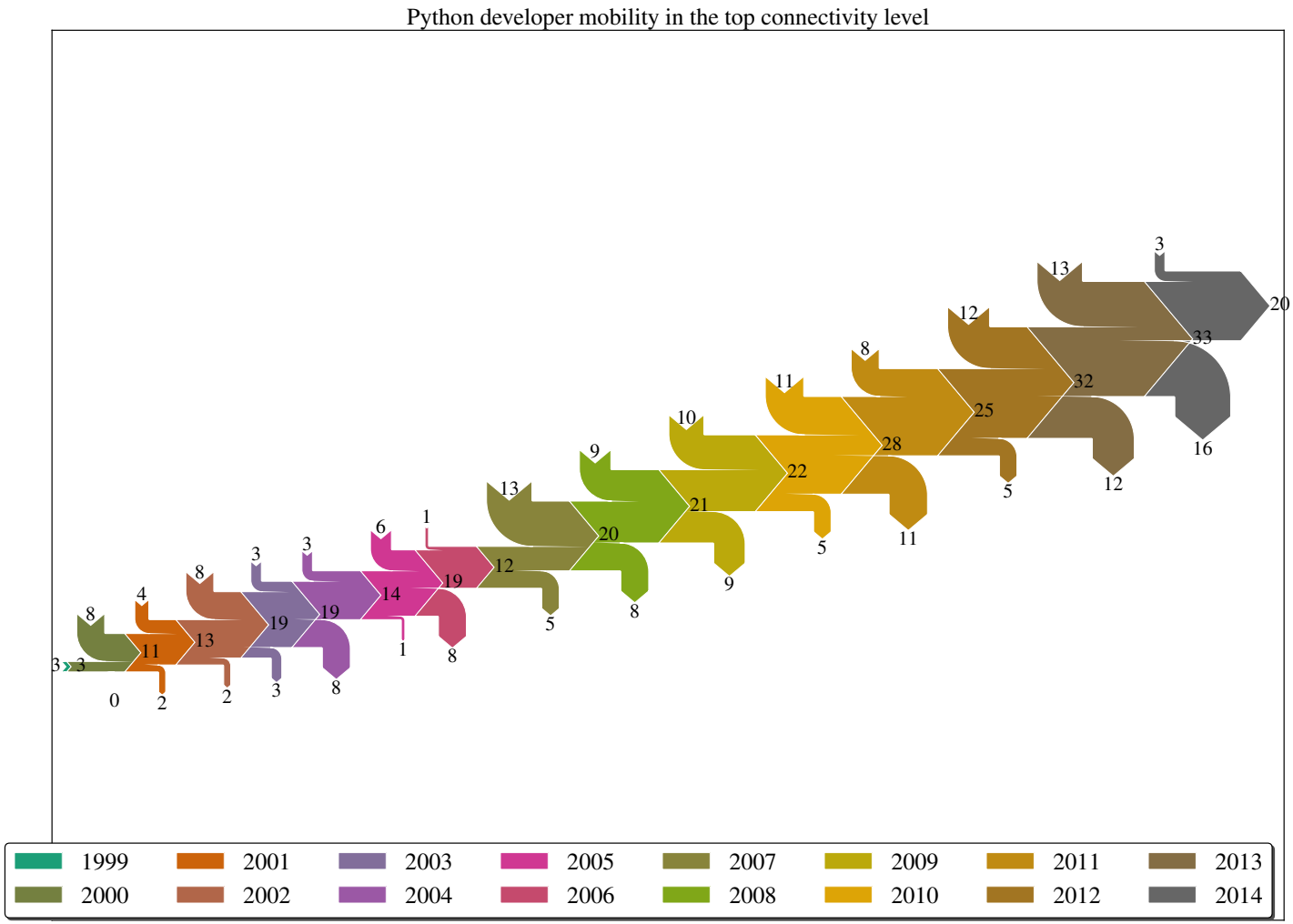


Figure: Sankey diagram of Python developer mobility at the top connectivity level.

Table: Developer mobility in the top connectivity level for the Python project.

Years	Top Developers	New Developers	Developers Out	Developers back
1999	3 (33.3%)	3 (100.0%)	0 (0.0%)	0 (0.0%)
2000	11 (35.5%)	8 (72.7%)	1 (9.1%)	0 (0.0%)
2001	13 (39.4%)	4 (30.8%)	1 (7.7%)	0 (0.0%)
2002	19 (50.0%)	8 (42.1%)	1 (5.3%)	0 (0.0%)
2003	19 (45.2%)	3 (15.8%)	5 (26.3%)	0 (0.0%)
2004	14 (28.6%)	3 (21.4%)	1 (7.1%)	0 (0.0%)
2005	19 (43.2%)	1 (5.3%)	4 (21.1%)	5 (26.3%)
2006	12 (23.1%)	0 (0.0%)	4 (33.3%)	1 (8.3%)
2007	20 (39.2%)	10 (50.0%)	7 (35.0%)	3 (15.0%)
2008	21 (35.6%)	6 (28.6%)	6 (28.6%)	3 (14.3%)
2009	22 (37.9%)	9 (40.9%)	3 (13.6%)	1 (4.5%)
2010	28 (44.4%)	9 (32.1%)	7 (25.0%)	2 (7.1%)
2011	25 (39.7%)	7 (28.0%)	2 (8.0%)	1 (4.0%)
2012	32 (49.2%)	8 (25.0%)	12 (37.5%)	4 (12.5%)
2013	33 (52.4%)	8 (24.2%)	16 (48.5%)	5 (15.2%)
2014	20 (32.3%)	3 (15.0%)	0 (0.0%)	0 (0.0%)