Cover Page

Title: Networks of Collaborations Among Government Organizations on Github

Journal Format: ACM Transactions on Knowledge Discovery from Data (TKDD)

Submitted as a partial fulfillment of the course,

08-801, Dynamic Network Analysis

Arun Kalyanasundaram

May 8th, 2015

Networks of Collaborations Among Government Organizations on Github

ARUN KALYANASUNDARAM, Carnegie Mellon University
KENNY JOSEPH, Carnegie Mellon University
KATHLEEN M. CARLEY, Carnegie Mellon University

There is an increasing trend among government organizations to open-source their software for their own use. Governments have always tried to involve people in creating policies and legislations, however, the right technology has been lacking. Recently, Github has emerged as a robust platform to allow people to collaborate on government projects both in building software and creating policies. Since Github also provides a rich set of social media features, these interactions create a complex network that provides insightful information on the way users collaborate. This network involves users interacting with projects and organizations in different ways and users forming ties among themselves using the *follow* feature. Therefore, in this paper, our goal is to study these networks of collaborations among these users and government organizations. Our primary hypothesis, informed from theory is to evaluate if members within an organization have more ties or not. We then build on this finding to better understand various aspects of collaboration such as reciprocity and homophily. Finally, we dissect the network into policy and code related collaborations and compare them on various dimensions, and validate our findings to already established theories.

Additional Key Words and Phrases: Social Network Analysis, Distributed Collaborations.

DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

HealthCare.gov is a health insurance exchange website operated by the US federal government designed to help US citizens find affordable health care. However, when it was launched in Oct 2013, a majority of users were experiencing technical difficulties. The site was almost unusable for most of the users. Several analysts felt this should not have happened but it is not uncommon. Bloomberg performed an analysis[1] and found that an open-source approach could have helped prevent the disaster. Open-source software is developed by volunteer users contributing and coordinating to iteratively build the software. Many open-source software products such as Linux, Apache have been extremely successful. The reason Bloomberg pointed this approach is because by open-sourcing a government project, allows ordinary citizens to help improve the code and they get a sense of satisfaction of contributing to the functioning of the government. Therefore, the suggestion from the analysts was that "people, including programmers are intrinsically interested in what the government doing often because their lives are affected directly" to "tap an army of interested coders ready to support official efforts".

This shows that two things – a) People are interested in the contributing to the functioning of the government, b) Tapping people's contributions can help function the government better. Therefore, if people can contribute to building software, they can also help write policies and legislations for the government. However, this also requires a platform that can help thousands of citizens collaborate.

[1] http://www.bloomberg.com/bw/articles/2013-10-16/open-source-everything-the-moral-of-the-healthcare-dot-gov-debacle#p2

One such initiative is called Government and Github (https://government.github.com/). It has several countries participating by uploading their code and policies on Github and soliciting contributions from ordinary citizens. Overall there are 40+ countries and around 600 organizations. However, we do not have a good understanding on who the contributors are, how they collaborate and how effective it is. Github has an underlying social network of users created by the follower-following relationship similar to the one on Twitter. Users can also belong to an organization, which creates a membership network. Therefore, the goal of this project is to use these networks of collaborations to understand how users collaborate.

Broadly, the data is organized as follows, there are several countries listed on the webpage - https://government.github.com/community/. Each country has a set of organizations. For this analysis we chose one country: U.S.Federal and study the collaboration of users and organizations of this country. Future studies can compare different countries. Each organization contains one or more projects (or *repositories* in Github's technical term). This gives us an Organization X Repository network. Users on Github can interact in three different ways with a repository, and therefore, each creates a link between a user and repository. As shown in Fig. 1, the three ways are - a) **Commit Changes**: A user makes changes to one or more files in a repository, b) **Fork**: A user clones a repository in order to work on it for her own use or submit the changes back to the repository later, c) **Watch**: A user is interested in the repository and may want updates about changes in the repository, in which case the user uses social

media features like watch or star a repository. The star is similar to the favorite feature on Twitter.

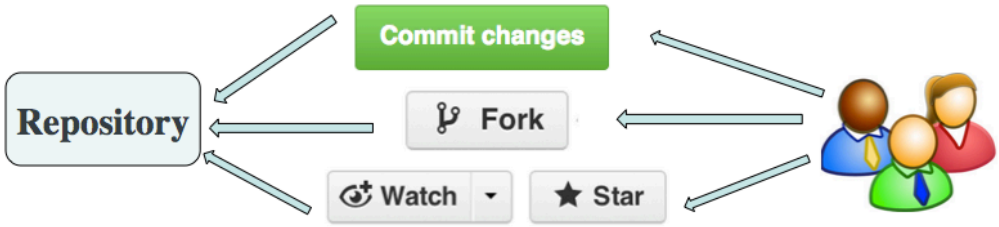


Figure 1: Three different ways of interactions between users and a repository.

Therefore, this gives us three different types of Repository X User networks. Since we have an Organization X Repository network, we can also get three Organization X User networks. From these meta-networks we can generate a single mode network for each of the node type by simple network algebra.

As we mentioned earlier, Github also offers users to follow each other and this gives an underlying social network among these users. We extract this as a directed User X User network, and use this along with out meta networks to address the overall goal of this paper – "Study the network of collaborations among government organizations and the social network of people involved." We propose the following research questions.

1. Do the different types of interactions create similar networks?
   - This implies if the different social features are used similarly.

2. Are there more social ties among members of an organization?
    - This informs higher group cohesion. We could also identify if there is lack of social ties across members of an organization.
3. Which came first, social ties or joining the organization?
    - This will tell us if the social ties are formed as a result of homophily. This means people in the same organization later formed social ties. If the other way is true, that would imply that people who knew each other before had joined the organization later.
4. Are the social ties reciprocal? To what extent do different types of users have more or less reciprocity?
    - Previous finding on Twitter and Github has shown very little reciprocity. This will tell us if this particular type of network we are studying exhibits same or different properties. For example, we would expect members of the same organization to have more reciprocity.
5. What are the differences between the collaboration network on Policies and Software?
    - Since the repositories can be identified based on whether or not they are meant for code, this can tell us if there is difference in the ways they collaborate on policies vs. software.

2. DATA COLLECTION

We pulled out the names of all organizations from https://government.github.com/community/ using a HTML parser written in Java. The purpose of this script is to extract the names and github IDs of these organizations. However, we are only interested in the list of organizations under U.S. Federal.

Since we are only interested in the set of organizations under U.S. Federal,

In order to extract the repositories and users, we did not use the Github API because of their strict rate limiting policies. Therefore, we used an offline archive of Github available at - ghtorrent.org/dblite/. The website organizes the Github data in SQL tables and offers an interface to invoke SQL commands to extract desired data without any rate limitation.

The SQL interface also allows us to load custom data onto the database in a separate schema. We loaded the list of organizations into a custom table. This allows us to write SQL queries to extract the three types of networks for any country. The script and SQL Queries and scripts are provided along with this paper. Each network is stored in a CSV file in a table of links format that ORA can read into. Fig. 2 shows a screen shot of the data from one of our networks. Each row has the name of the organization, repository and user and a two time stamps, one indicates the time of creation of the link between user and repository, and second indicates the time of creation of link between repository and organization.

	A	B	C	D	E	F	G	H
1	org_name	org_id	repo_name	repo_id	user_name	user_id	link_creation_date	repo_creation_date
2	NREL	402061	OpenStudio-	7838943	developerto	18	7/31/14 21:14	1/31/14 23:39
3	hpc	181	iptablesbuild	43	LANLgraham	182	8/1/12 22:53	8/1/12 15:19
4	BBGInnovate	503360	News-On-Loc	645533	ericpugh	479	8/31/12 20:35	8/31/12 18:26
5	BBGInnovate	503360	Ourblock-Sta	4471206	ericpugh	479	7/2/13 15:31	7/2/13 15:31
6	BBGInnovate	503360	LSAP	5540020	ericpugh	479	9/5/13 20:41	9/5/13 20:31

Figure 2: Snapshot of the meta network data

The following table gives a summary of the size of each of the meta network.

Table 1: Meta networks size

Meta network	**#Organizations**	**#Repositories**	**#Users**	**#Edges – User X Repository**
Commit	102	1622	2186	5544
Fork	83	894	2821	4462
Watch / Star	86	844	6052	8738

As we can see, the number of users is the highest in the watch network because a lot of people could be interested in a repository or organization, whereas if someone forks a repository, they are not only interested but also willing to make changes to files in the repository.

We used the same GHTorrent interface to extract the underlying social network of users in each of the three meta networks. For each user in each of the meta network, we extracted their followers and stored the creation date of the link in a csv format. The following table gives the User X User directed network for each type of meta network. The filtered edges represent only those edges where both users have interacted with at-least one repository.

Table 2: User X User networks size

Meta Network	**#Users**	**#Edges**	**#Edges Filtered**
Commit	2186	14341	1166

Fork	2821	42248	1795
Watch / Star	6052	110575	10139

3. BACKGROUND

Broadly, we divide our literature into the following three areas – a) Policy Netwoks, b) Networks of Open source software projects and c) Social coding on Github. The reason why our paper depends on these areas is because the goal of our paper is to study collaborations across government organizations on building software and policies on Github. Therefore, this involves first understanding how networks among policy decision makers and government organizations is formed and studied. Second, since a large part of the collaboration is also creating software, we review the related literature of collaboration in open source software. Finally, the organizations collaborate on Github, which is essentially an online distributed software development community with a rich set of social media features. Therefore, we briefly touch upon the relevant literature of studies of networks on Github.

Policy networks are the network of relationships that are created among actors involved in policy decision making (Dowding, 1995). It may appear similar to any other social network, however, there are some key differences. First, policy networks are formed by interactions over several years or sometimes even decades (Henry et.al, 2010). Therefore, one theory is that such ties are formed only when the policy related ideologies of actors match (Sabatier, Jenkinks-Simth, 1993). Ideologies are basically strong beliefs on certain issue, for example a policy maker who is in favor of environmentalism may have strong

belief about conserving the environment. Previous literature has established that these networks form based on belief homophily and advocacy coalition framework is a theory to study the creation of policy networks. An empirical study of real world policy networks showed that there is also social capital involved in the formation of these networks (Henry et.al, 2010). In our paper, we study a network of collaboration where some of the nodes are involved in policy decision making, however, it is unclear if these well understood theories of network formation would still hold in our policy networks.

Collaboration among software developers is another related area of research. We are interested in the network of collaboration in open source software projects. In our paper we treat projects (or repositories) as nodes and members working on the same project as connected members. This type of analysis has been done with data from SourceForge, where Madey et. al, 2002 analyzed the network between 39000 open source projects and 33000 developers. They found a power law distribution for the number of projects per developer, which we expect to find in our data as well. They performed a clustering analysis and found that there was one cluster with more than 6500 nodes and the second cluster was just 55 nodes. This is something worth investigating in our paper, to study the size and meaning of clusters. The key result of their paper is that networks of open source networks are formed as a result of preferential attachment, that is the effect of "rich-get-richer". This is also the reason for power law distribution. In our paper we will study if this phenomenon holds for our data.

However, not all members within an open source software project have the same role or duties. Although there is no explicit hierarchy as in a

formal organization, there are roles generally identified by the community such as core developers, project leaders, contributors and users. Xu et. al, (2006) studied the networks of these different types of members rather than the whole network. They found that project leaders are highly disconnected, perhaps they lack time to join other projects. They also found a scale free degree distribution as shown by previous studies. However, they found a small diameter and a high clustering coefficient, which are properties of small world networks.

The networks in open source software have not only been studied based on membership on roles but also on the tasks and interactions around these tasks. One of the common activities that cuts across all open source software projects is the task of fixing bugs. Crowston and Howison (2003), studied interactions among open source software developers on bug reports in 122 large and active projects. They studied the social structure of teams using these network of interactions. They hypothesized a network with core developers being central actors then the contributors and finally users on the peripheral of the networks. However, they found that some projects were highly decentralized, so, even the peripheral users were key players in the network. This was a result of the norms of participation and contribution in those projects.

Therefore, these networks in OSS are also influenced by the policies and norms enforced by the project or community. Sagers, (2004) studied the effects of governance on the structure of open source software project networks. They found that restricting membership access greatly influences the formation of networks. However, such policies also have a positive effect, they found that those projects with restricted membership access were more successful than others.

However, they also found that increased exchange of interactions among members in fact decreases the ability to resolve bugs. In our paper, we aim to study networks formed by both membership in projects and interactions among contributors.

Although most of empirical studies on networks in open source software have been about projects on Sourceforge, there has been a recent increase in OSS projects on a platform called Github, which is essentially a website to host source code but with all social media functionalities. One of earliest works to study Github was by Dabbish et. al, (2012), who considered the impact of various social media features on the way people collaborate. The network structure of this social coding process was studied by Thung et. al, (2013) who used centrality measures to find the most influential projects and developers, and the strength of relationship among developers and projects. In our paper we are interested in understanding this network of relationships between developers on Github. We are also interested in how projects disseminate in the Github's social graph. Jang et.al (2013) studied the time taken for developers to participate after a project is created. They found that there is very low reciprocity, that is a lot of social links are directed – very few developers follow their followers back. It would be interesting to see if this holds in our network of developers and policy makers of government organizations. We are also interested in using location attributes of developers to study the effect of colocation on the creation of ties. A general study of location of github developers for different programming languages has been studied Rusk et. al, (2014).

The study of government organizations collaborating on Github has been not been well studied, although there is this one paper where an

exploratory analysis of the developers and their affiliation was done. The social network was constructed based on interaction among government Github developers. They found that forking is a very common practice, where a project reuses the code of another project, and they used a forking collaboration network to identify the central projects. Although there is a lot of forking, there is very less contribution to existing projects, probably indicating that forking is done to reuse and modify code for their own use.

Finally, we review the related literature on the methodologies used in our paper. Our main hypotheses are informed from theory. Contractor, Wasserman and Faust (2006) proposed eight hypotheses to study organizational networks. One of the goals is to understand cohesion among members of an organization and this informed from theory (Contractor, Wasserman, Faust, 2006) – a) Actors who belong to same organization are likely to have ties with one another. Therefore, we test this hypothesis in our research question using block modeling of a set of organizations. We assign each user with the organization (the first organization they joined) as an attribute and this allows us to easily compute the block model using the follower network. We also detect changes in the network over time using techniques proposed by McCulloh and Carley, 2011, we find if a user joined an organization before they created a social tie, indicating a homophily effect. Our approach in creating the single mode networks from multi-mode meta-networks is based on the standard approaches in (Carley 1999), who describe how agents interacting with physical world create relations with physical objects (repositories in our case) and which in turn can be used to create a network among the agents or physical objects.

4. ANALYSIS AND RESULTS

We first provide high-level network metrics for each of the single mode network obtained after folding. We get three organization networks and three repository networks. Table 3 gives the details of all the organization networks and Table 4 gives the details of the repository networks.

Table 3: Organization Networks after folding Organization X User meta-networks.

Network	# Nodes / Edges	Density	Clustering Coefficient	Characteristic path length	Components of 4 or more	Isolates
Commit Organization Network	102 / 206	0.039	0.516	2.7	2	27
Fork Organization Network	83 / 426	.122	.58	2.2	1	20
Watch / Start Organization Network	86 / 960	.257	.772	1.78	1	12

Table 4: Repository Networks after folding Repository X User meta-networks.

Network	# Nodes	Density	Clustering Coefficient	Characteristic path length	Components of 4 or more	Isolates

	/ Edges					
Commit Repository Network	1622 / 21583	0.016	0.874	3.6	45	66
Fork Repository Network	894 / 7105	0.018	.718	3.4	11	162
Watch Repository Network	841 / 22212	0.063	.73	2.5	7	111

There is an interesting trend noticeable in the above tables. The clustering coefficient for organization networks increases in the order Commit, Fork, Watch; whereas decreases for repository networks in the same order. This is because there are more nodes in the repository networks and the folding creates more clustering. The characteristic path length is calculated after make the links binary in the folded networks. It is interesting to note that the organization networks only have one or two components that are four or more nodes. Looking at the number of isolates give us more information on the exact size of these components considering that there are very few diads and triads.

4.1 High Level Description of the Network

Before we proceed, we would like to provide some qualitative information about the network. To get a better sense we list the top

organizations with the highest number of users in each type of meta network.

Table 5: Top organizations in Commit Network

Organization	Description	# Users
NCIP	National Cancer Informatics Program	239
Kbase	The Department of Energy Systems Biology Knowledgebase	230
MOSES	Located in Orlando, FL, manages set of tools for cloud grid based software.	168
CHAOS	A development team in Livermore, CA who have several tools for network infrastructure	82
NASA	National Aeronautics and Space Administration	74

Table 6: Top organizations in Fork Network

Organization	Decription	#Users
WhiteHouse	The federal government's repositories on code and data – it has the 2016 budget data (https://github.com/WhiteHouse/2016-budget-data)	584

Project-Open-Data	Mostly planning data from several cities shared openly.	243
Adlnet	Advanced Distributed Learning	173
NASA	National Aeronautics and Space Administration	154
FCC	The Federal Communications Commission	127

Table 7: Top organizations in Watch / Star Network

Organization	Description	#Users
WhiteHouse	The federal government's repositories on code and data – it has the 2016 budget data (https://github.com/WhiteHouse/2016-budget-data)	1311
Project-Open-Data	Mostly planning data from several cities shared openly.	499
NASA	National Aeronautics and Space Administration.	492
CFPB	Consumer Financial Protection Bureau	373
GSA	General Services Administration	288

Surprisingly, there is not a lot of common organizations between the top ones in Commit and Fork meta networks, except NASA appears in both. Also, the number of users in the top five organizations in Commit network is lowest, followed by Fork and then Watch / Star network.

However, there is a considerable overlap between Fork and Watch networks in the top organizations. These show that people who fork may not necessarily commit, however, people who fork also watch. For a better understanding of this phenomenon, we will use MR-QAP to compare the organization and repository networks of each type to determine the correlation among these activities.

We also found that these organizations that have the most number of users are not always the key entities in terms of centrality. For example, we found that in the commit network – the organizations 18F and GSA had the highest total degree centrality and ego-betweenness centrality, but neither of them were in the top five number of users.

4.2 Comparing the three types of networks using MR-QAP

In order to compare the networks using MR-QAP we first find the intersection of nodes all the networks. However, if we filter nodes based on taking the common nodes across all three networks, then we might lose valuable data and misinterpret the data. Therefore, we show the general network metrics of the three networks by only considering the organizations and repositories that are common across all three meta-networks. This gave us 72 Organizations and 534 Repositories common across all three meta-networks. In Tables 3 and 4 above, we listed the network metrics of the organization and repository networks for each type of meta-network. In the following figures 3 and 4 we show the same network metrics but along with the metrics of filtered network below the original network. We find that by comparing the network metrics before and after filtering, there is no significant change in these values indicating a relatively less loss of information as a result of filtering.

ORG Network	Density	Clustering Coeff	Characteristic path length	#Components > 4 nodes
Commit	.039	.516	2.7	2
Fork	.122	.58	2.2	1
Watch	.257	.772	1.78	1

ORG Network	Density	Clustering Coeff	Characteristic path length	#Components > 4 nodes
Commit	.029	.184	2.02	1
Fork	.11	.379	1.92	1
Watch	.28	.57	1.72	1

Figure 3: Comparing network metrics of Organization networks before and after filtering for MR-QAP.

REPO Network	Density	Clustering Coeff	Characteristic path length	#Components > 4 nodes
Commit	.016	.874	3.6	45
Fork	.018	.718	3.4	11
Watch	.063	.73	2.5	7

REPO Network	Density	Clustering Coeff	Characteristic path length	#Components > 4 nodes
Commit	.024	.585	3.3	21
Fork	.023	.4	3.03	7
Watch	.09	.57	2.3	2.

Figure 4: Comparing network metrics of Repository networks before and after filtering for MR-QAP

Based on the above metrics we see that the density and characteristic path length are similar even after filtering. The clustering coefficient follows a similar trend even after filtering, although lower.

We performed MR-QAP with the commit network as the dependent network and the fork and watch as the independent network. Figure 5 gives the results of the MR-QAP for the Organization networks. The organization network is generated by folding Organization X User network in each of the three meta-networks

Correlation Results

Network	Correlation	Significance	Hamming Distance	Euclidean Distance
Fork : Organization x Organization - Fork	0.343	0	502	22.405
Watcher : Organization x Organization - Watcher	0.191	0	1350	36.742

Regression Results

R-Squared: 0.118927811104

Variable	Coef	Std.Coef	Sig.Y-Perm	Sig.Dekker
Constant	0.005		0	
Fork : Organization x Organization - Fork	0.173	0.323	0	0
Watcher : Organization x Organization - Watcher	0.016	0.043	0.220	0.130

Figure 5: MR-QAP results of the three Organization networks

We find that the Fork is statistically significantly correlated with the Commit network, whereas the watch network is not. The correlation coefficient is not too high, but indicates the Fork and Commit organization networks are somewhat correlated. The following Figure 6, gives the same MR-QAP results but for the repository network. This network is generated by folding Repository X User networks.

Correlation Results

Network	Correlation	Significance	Hamming Distance	Euclidean Distance
Fork : Knowledge x Knowledge - Fork	0.375	0	8080	89.889
Watcher : Knowledge x Knowledge - Watcher	0.164	0	26990	164.286

Regression Results

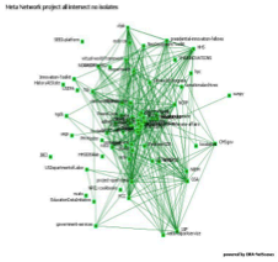
R-Squared: 0.145742528623

Variable	Coef	Std.Coef	Sig.Y-Perm	Sig.Dekker
Constant	0.012		0	
Fork : Knowledge x Knowledge - Fork	0.361	0.356	0	0
Watcher : Knowledge x Knowledge - Watcher	0.040	0.075	0	0

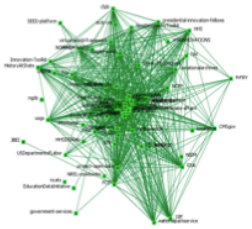
Figure 6: MR-QAP results of the three Repository networks

In the repository networks we see statistically significant correlation for both Fork and Watch networks. However, the correlation coefficient

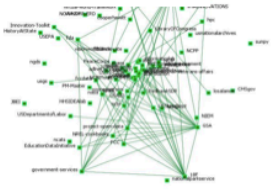
values are similar to the results of the organization networks. The Fork repository network is reasonably correlated with the Commit repository networks. This shows that the collaboration network formed by users who commit and fork operation is similar whereas the one formed by the watch operation is not different. This is also intuitive because a broad spectrum of people are interested in the projects and may choose to star / watch them but only a fraction of them are interested in contributing and they will fork the repositories. This shows that the pattern of forks almost imitates the commit, whereas the pattern of watch activity is somewhat unrelated. The fact that the watch / star feature results in a different kind of network bolsters the case for having this feature on Github. If our findings showed that they were used similarly, then there is no need for this feature. On the other hand, Fork and Commit are designed to be used in tandem, for example most repositories don't even allow to commit unless you fork the repository. Following figure gives a visual representation of the organization networks of the three types of meta-networks after filtering.



Fork



Watch



Commit

4.3 Block Modeling to identify group cohesion

In this section our goal is to determine if the social ties (follower network) among users within a group is stronger than inter group ties. To answer this a block modeling approach is an ideal technique. For each user we assign the organization as an attribute, if a user is part of more than one organization then we choose the organization that he first interacted with.

Since we have more than 70 organizations the block modeling can only be useful if we have less than five groups, therefore, in each type of meta network we identify a set of five organizations that have the most number of users. We then combine all users of these five organizations and build the adjacency matrix of connections between these users from the follower network. We used a python script (attached) to compute this adjacency matrix and the within group and inter-group densities. In the following tables the diagonal cells indicate the densities of users within an organization. All other cells represent the density of connections going from an organization in the row to an organization in the column of the table. Finally, we compare each cell with the overall density and assign a 0 or 1 based on whether it is less than or greater than the overall density. This allows us to draw a network of relations among the organizations.

Table 8: Block Modeling of users in five different organizations in the Commit Activity Meta-Network (Total density: $1.6*10^{-4}$)

densities	NCIP	KBase	MOSES	Chaos	NASA
NCIP	$1.1 * 10^{-4}$	0	0	0	0
KBase	0	$1.2*10^{-3}$	0	0	0
MOSES	0	0	$3*10^{-4}$	~0	0

Chaos	0	0	0	$6 * 10^{-4}$	0
NASA	0	0	0	0	$3.5*10^{-2}$

The following figure gives the network of organizations of users involved in the commit activity as a result of block modeling. We find that there are no inter-connections between organizations, which means users who made changes (commit) to these five organizations had almost no connections with others in other organizations. However, there is a relatively strong social network among users who commit to the organizations. These organizations are isolated with each other because there is not much overlap in terms of what they do and as a result people do not collaborate across these organizations.

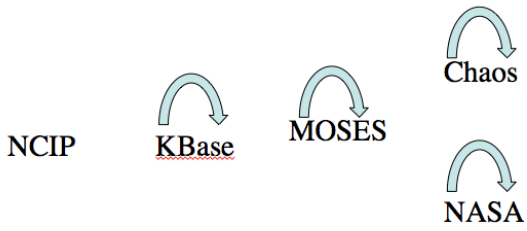


Figure 7: Connections among organizations formed by block model from the commit meta-network

We perform the block modeling for users involved in the Fork activity. This gives a slightly denser network than Commit. However, the top five organizations are different from Commit than Fork. Using block modeling we derive the network of organizations as shown in Figure 8.

We wanted to see how the follower network of users in these five organizations looked like when we colored by the name of the organization. In the figure below, one on the left shows nodes colored by organization name, and one on the right shows nodes colored by newman grouping. We do see a good amount of similarity in the way newmann groups and the grouping of nodes based on attribute.

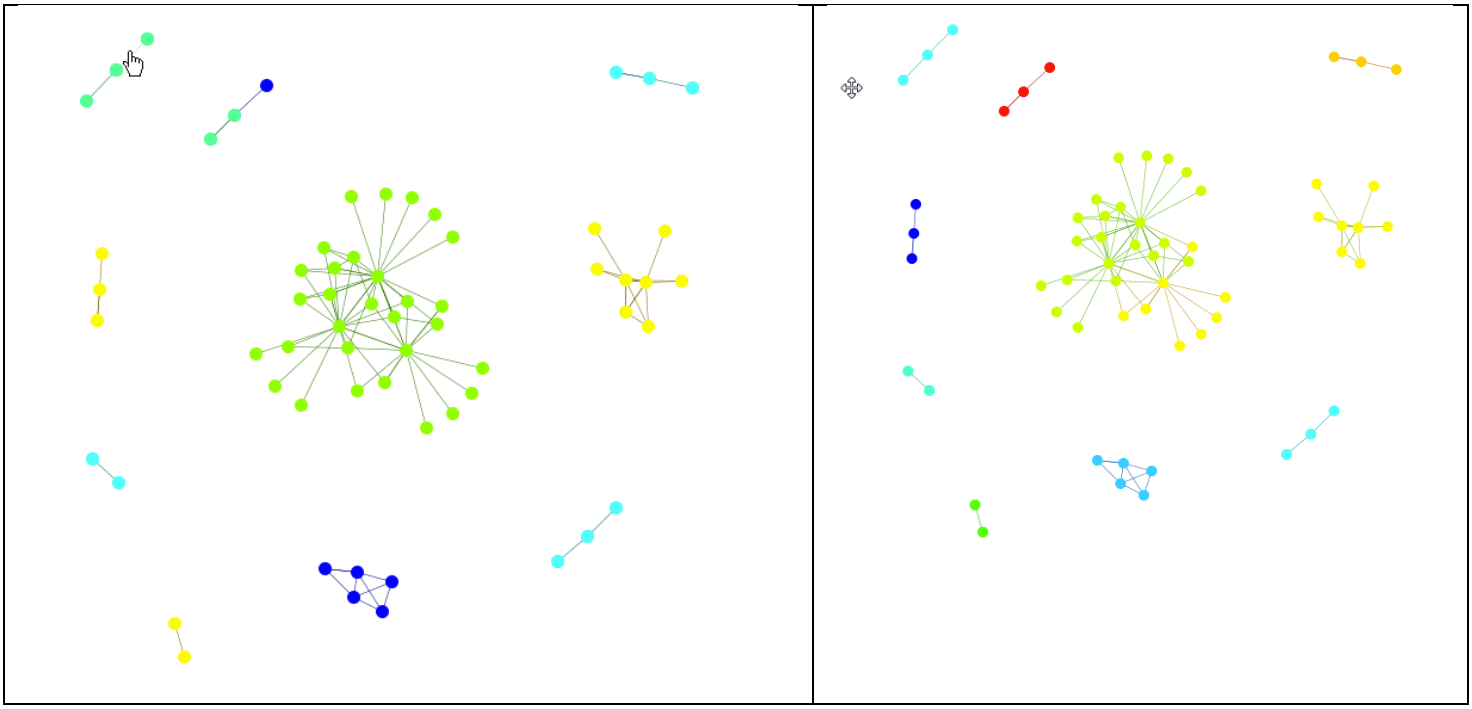


Table 9: Block Modeling of users in five different organizations in the Fork Activity Meta-Network (Total density: $2.6*10^{-4}$)

densities	White House	Project-Open-Data	ADLNET	NASA	FCC

White House	$2*10^{-4}$	$3*10^{-4}$	~0	~0	$1*10^{-4}$
Project-Open-Data	$2.2*10^{-4}$	$2*10^{-3}$	~0	~0	$7.7*10^{-4}$
ADLNET	~0	~0	$1.7*10^{-3}$	0	0
NASA	~0	~0	0	$7.6*10^{-4}$	0
FCC	$2.1*10^{-4}$	$6.8*10^{-4}$	0	~0	$4.3*10^{-4}$

We see that the there is a relatively strong intra-organization connections among members of all five organizations except White house. However, we also see a reciprocal connection between FCC and project-open-data, which indicates that users among these two organizations could have reciprocal ties. We explore the aspect of reciprocity in our next section. **It is interesting to note that the directed link from white house to project-open-data exists indicating that quite a few users who fork project-open-data follow users in white house.** It will be interesting to explore whether it is the content of the organization or nature of users in White house creates this relationship with project-open-data.

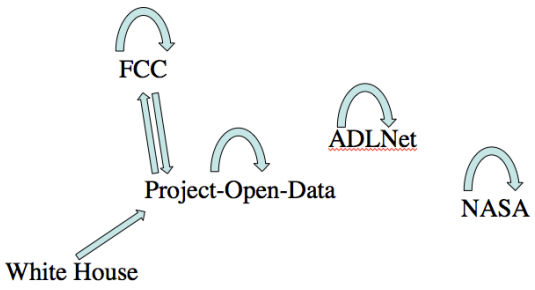


Figure 8: Connections among organizations formed by block model from the Fork meta-network

Finally, we perform block modeling of users involved in the watch activity for the five organizations selected. Surprisingly the density of the user follower network is almost the same (slightly less) as Fork , as one would expect a higher density. This shows that even though there are more people who are involved in watching activity than Fork, the density of inter-connections among the users is almost the same. **In other words, people who watch / star don't necessarily follow more people (create more social ties) than people who Fork**.

In the figure below, the one on the left shows nodes colored by organization name, and one on the right shows nodes colored by newman grouping. We do see a good amount of similarity in the way newmann groups and the grouping of nodes within organizations.

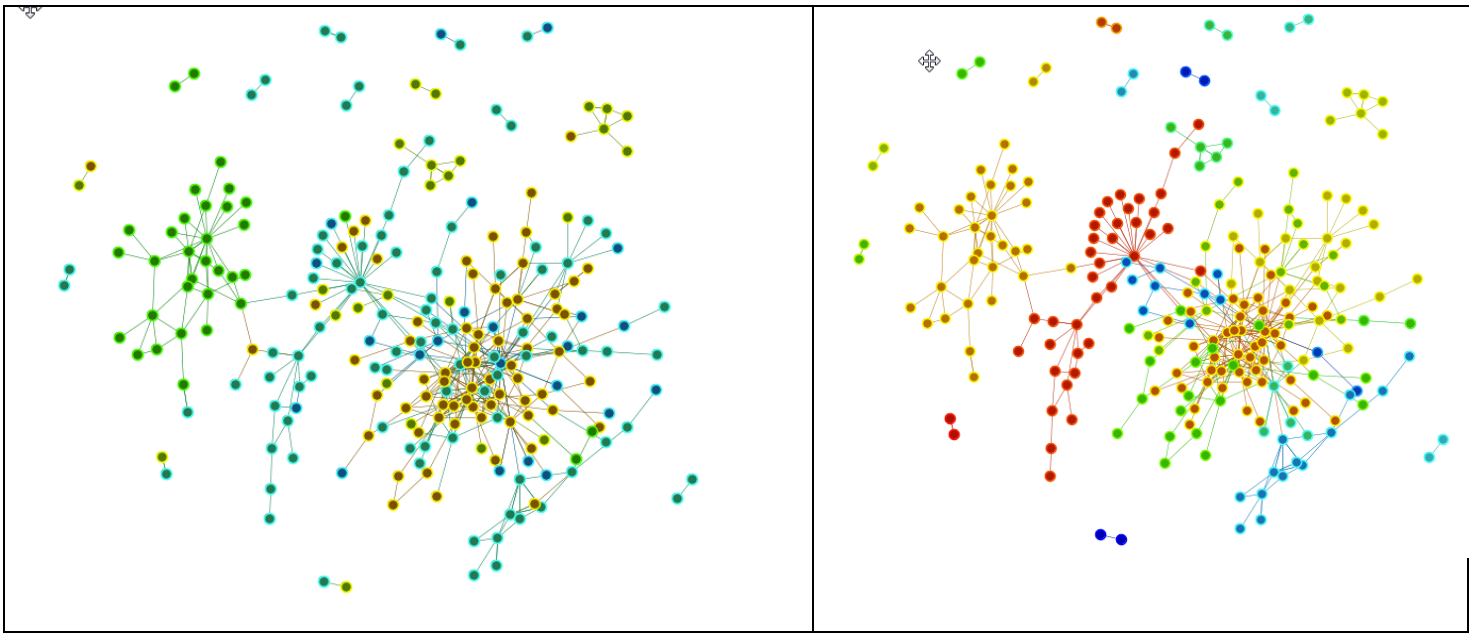


Table 9: Block Modeling of users in five different organizations in the Watch activity Meta-Network (Total density: $2.5*10^{-4}$)

densities	White House	Project-Open-Data	NASA	CFPB	GSA
White House	$2.7*10^{-4}$	$2.2*10^{-4}$	$1.5*10^{-4}$	$2.3*10^{-4}$	$2.9*10^{-4}$
Project-Open-Data	$1.8*10^{-4}$	$4*10^{-4}$	$1.2*10^{-4}$	$3*10^{-4}$	$4*10^{-4}$
NASA	$1*10^{-4}$	$8*10^{-5}$	$3*10^{-4}$	$1*10^{-4}$	$2*10^{-4}$
CFPB	$2*10^{-4}$	$2.7*10^{-4}$	$3*10^{-4}$	1.4*10-3	$5*10^{-4}$
GSA	$2*10^{-4}$	$2.8*10^{-4}$	$1.3*10^{-4}$	$4*10^{-4}$	$5.7*10^{-4}$

Figure 9 gives the network of organizations formed as a result of block modeling. We see all organizations have relatively strong inter-connections, an indication of strong group cohesion among people involved in Watch activity within each of these organizations. However, it is interesting that project-open-data, CFPB and GSA form a triad and each has reciprocal links with each other**. This shows that people in these organizations could potentially closely collaborate with each other if needed since they have reasonable number ties among them**. Contrary to the Fork activity network, we see that White house has a directed link to GSA instead of the project-open-data. Although there is not link from white-house to

project-open-data as we saw in Fork activity network, there is a reasonable density between the two groups but lower than the overall density.

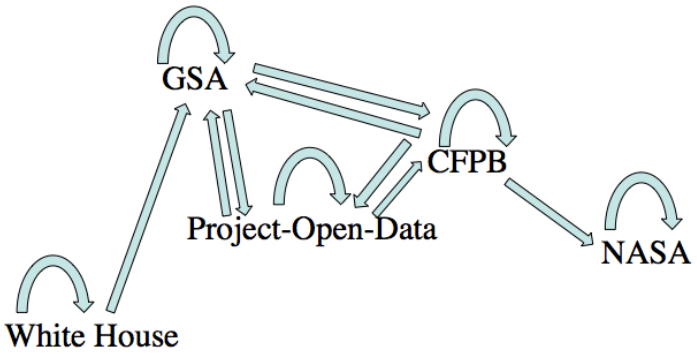


Figure 9: Connections among organizations formed by block model from the Watch meta-network

4.4 Cause of Tie formation – Homophily vs. Prior Connections

So far, we have found that there are intra-group connections and some groups have inter group connections. However, we do not know if the connections were created first among the users as a result of pre-existing ties or due to homophily, i.e they created ties after joining the organization. In this section we would like to explore if the cause of ties formed among the users is because they joined the same organization or they already had prior connections before they joined the organization. We cannot really test this difference statistically, but we can give a general sense based on the time when each of the link was created. We find for each of the top five organizations, the percentage of social ties created before joining the organization and after joining the organization. The following three tables shows the actual number and the percentage of links in each organization for the three activities.

We also show the average in each table, which indicates the overall for all organizations the number of links that were created before and after joining the organization.

Table: Percentage of social ties (follower relationships) created before and after joining the organization in the Commit activity.

Organization	Total Links among users	Links created before joining org	Links Created after joining org	Links created before joining org (%)	Links Created after joining org (%)
NCIP	6	3	3	50%	50%
KBase	63	50	13	79%	21%
MOSES	9	8	1	88%	11%
Chaos	4	2	2	50%	50%
NASA	19	11	8	57%	42%
Average	**101**	**74**	**27**	**73%**	**27%**

Table: Percentage of social ties (follower relationships) created before and after joining the organization in the Fork activity.

Organization	Total Links among users	Links created before joining org	Links Created after joining org	Links created before joining org (%)	Links Created after joining org (%)
White House	71	49	22	64%	36%
Project-Open-Data	114	73	41	64%	36%

ADLNET	50	32	18	64%	36%
NASA	18	9	9	50%	50%
FCC	7	5	2	71%	29%
Average	**260**	**168**	**92**	**62%**	**36%**

Table: Percentage of social ties (follower relationships) created before and after joining the organization in the Watch / Star activity.

Organization	Total Links among users	Links created before joining org	Links Created after joining org	Links created before joining org (%)	Links Created after joining org (%)
White House	468	385	83	82%	18%
Project-Open-Data	114	73	41	64%	36%
NASA	73	50	23	68%	32%
CFPB	199	162	37	81%	19%
GSA	47	32	15	68%	32%
Average	**884**	**705**	**179**	**80%**	**20%**

We notice that for all organizations the majority of links are created among the users before they joined the organization. **One possible explanation is that most members of these organizations already had a working relationship with each other, because the github organization is nothing but a digital place for the actual organization.** However, the users who created connections after joining the organization will be interesting to study because this will indicate if Github facilitates new collaborations of people who have

never worked together before. It is interesting that users who were involved in watch / star activities were least likely to form social ties after interacting with the organization as opposed to Fork followed by commit activities. The average of users having ties before joining the organization in the commit activity is above 70% indicating a lot of users who make changes already knew each other. Also, White house has consistently higher percentage of users who had ties before joining the organization, indicating there is scope to foster more collaborations and social ties after joining the organization.

4.5 Reciprocity

So far we have considered follower and following relationship interchangeably, however, a person following another person may not necessarily know the person, but if that person also follows back there is social capital and a good chance the relationship is stronger. In this section we explore the general reciprocity in these networks, and find which organizations have highest reciprocity, and what kinds of users have high or low reciprocity. We compute reciprocity by the ratio of number of reciprocal links to the total links. For example if there are three nodes A,B,C; and nodes A-B and B-C have reciprocal links, and there are total 5 links, then reciprocity is (1+1)*2/5 = 0.8. Note each reciprocal link is counted twice, because if all links are reciprocal then we should get a reciprocity of 1. We could use ORA to find the reciprocity of the overall network but to find reciprocity within organizations we used custom scripts written in Python. The following table gives the overall reciprocity of all the three networks

Activity	Total	Total Links	Reciprocal	Reciprocity

(Meta network Type)	users		Links	
Commit	2186	1166	206	0.35
Fork	2821	1795	249	0.27
Watch / Star	6052	10139	1158	0.23

We see that there is a reasonable amount of reciprocity in these networks, on an average above 25% reciprocity is considered reasonable for this type of social network, such as Twitter or Github. This could mean two things, a) users know each other and therefore follow each other creating a reciprocal link, b) since users are in the same organizations there could be more pressure to follow back because you are collaborating with that person and it is a way to build the social capital.

To further explore this cause we show if users have more than average reciprocal ties within organization. We pick top five organizations in each type of network. We find that in the commit activity network NASA had the highest reciprocity of 0.42, the other organizations had below overall average reciprocity. This indicates that users in NASA know each other better and this could improve the efficiency of the organization. This is a scope of future work where we can compare the reciprocity with the efficiency, which in turn can be measured by different metrics such as average lines of code per day.

In the fork network we found NASA again had a high reciprocity of 0.44, followed by project open-data which had 0.33, however, in white-house we found relatively low reciprocity of 0.2. Even though the number of links was higher (71) in White-house, there were very few

reciprocal links, indicating people may not have working relationship. Comparing reciprocity with group cohesion, we see that white house also has a very low group cohesion in general and that bolsters our finding, however, whitehouse had a high percentage of links formed before joining the organization, which could indicate we rule out homophily and that people already knew each other but low reciprocity could refute that. Therefore, there is a complex process of link creation among the members of this particular organization and it will be interesting to take a qualitative look at how these ties are formed and their effect on the efficiency. Finally, we look at the watch / star activity network and find that NASA again had the highest reciprocity at 0.54, consistently indicating that members of NASA know each other very well and have a strong social capital. Whitehouse had a similar to Fork activity reciprocity of 0.21.

We looked at reciprocity at the level of individual organizations, however, there could be certain characteristics that each individual user might posses that could lead to higher or lower reciprocity. The general notion is that users who are most and least influential may actually have low reciprocity. For example, a celebrity might have several followers but may not follow any one, on the other hand a new business man might want to follow a lot of other users but may not have any followers. In this network we found that users with the highest followers were *not* the ones who had the highest reciprocity. For example, a user equus12 had very few links to the other members of organizations, but had 18 out of 24 links reciprocal. On the other hand another user, Tom Macwrithg (tmcw) had about 125 followers in the other organizations but only about 25 were reciprocal. We used a

script to sort the users with highest out degree and their reciprocity, however, more analysis is required to understand what kinds of users have high and low reciprocity.

4.6 Policy Networks – Understanding the Collaboration Network on Non-Code Repositories

We know that users in these organizations collaborate both on Code and non-code related stuff such as data, policy and legislation. Github associates a programming language with each repository automatically based on the contents of the repository. If the contents of a repository does not contain any code, it leaves the language as blank. We use this parameter to differentiate between repositories that are code related and those that are not. Sometimes a repository might not be code but could be html pages of policy documents, in which case our technique will treat them as code repositories. In future we could come up with more accurate metrics to detect if a repository is code or non-code. Therefore, the goal is to understand the difference in the collaboration networks of repositories on Code and Non-Code. Our assumption is that these non-code repositories will be on discussing policy changes in the government. For example the 2016 Budget repository under the White House organization does not have a language and is clearly a policy related collaboration. (https://github.com/WhiteHouse/2016-budget-data)

What are Policy Networks? Reiterating our description from the background section – "Policy networks refer to the structure of relationships a limited set of organizational actors establish among themselves during policy formulation and implementation." Previous research has shown belief homophily and Social Capital as the driving

factors of creation of policy networks. In our data we are interested in measuring social capital using reciprocity and comparing if it is different for actors in policy networks vs. for actors in code networks. We can also determine if the intra-organization density as an indicator of social capital in this context, since users don't necessarily have to create connections. So, our hypothesis is that if they collaborate on policies then their intra-group density should be higher. To measure the effect of homophily we simply use the percentage of links created after joining the organization. To summarize, we test the following hypotheses to evaluate the differences between policy and code networks.

a. Reciprocity among members who collaborate on policies (and data) is higher than people who collaborate on code (software).
b. The density of follower network of members within organizations is higher for policy networks than for code networks.
c. The fraction of links created among members after they join the organization (or repository in this case) is higher for policy related repositories than code related repositories.

Before we evaluate these hypotheses we give a high level overview of the two networks. We see that in each type of meta network the number of links on policy related is about one third of the links in meta networks on code related. This although is small is a reasonable size to make comparisons between the two.

Table: The meta networks only comprising of **policy** (non-code) related repositories.

Meta Network Type (Activity)	#Organizations	#Repositories	#Users	#User X Repository Links
Commit	81	475	792	1381
Fork	56	279	794	1130
Watch	64	256	1817	2174

Table: The meta networks only comprising of **code** related repositories.

Meta Network Type (Activity)	#Organizations	#Repositories	#Users	#User X Repository Links
Commit	91	1183	1778	4163
Fork	74	634	2292	3312
Watch	76	608	4760	6564

The following two tables we compare the reciprocity of the members involved in policy and code related repositories. Overall the policy related members have a high reciprocity, except on the commit activity the members involved in code have a high reciprocity. This indicates that members who work together on making code changes tend to have closer relationship. We can neither refute nor support whether the members of policy networks have higher reciprocity compared to members in code networks, this can be tested once these collaborations mature after a few years.

Table: Reciprocity of members involved in Policy networks

Activity	#Users	#Links	#Reciprocal Links	Reciprocity - Non-Code Network
Commit	792	598	102	0.34
Fork	794	559	87	0.31
Watch	1817	1977	249	0.25

Table: Reciprocity of members on Code related networks

Activity	#Users	#Links	#Reciprocal Links	Reciprocity - Code Network
Commit	1778	1004	186	0.37
Fork	2292	1453	197	0.27
Watch	4760	7463	868	0.23

The following tables we show the densities of members involved in a set of five organizations. In the first table we show only those members who collaborate on policy, whereas in the second table we show the members who collaborate on code. We find a higher intra-organization density in every organization for policy related collaborations. This implies more ties among members who collaborate on policies than those who collaborate on code. This could also be due to a tendency of non-code related members follow more members than those who are more technical might use the Github follow feature sparingly.

Table: Showing the densities of social network of members in different organizations for policy related repositories. (Note: Organizations chosen from a mixture of all three activities)

Organization	#Users	Intra-Org density – policy related repositories
WhiteHouse	574	$8*10^{-4}$
NASA	113	0.018
KBase	68	$3*10^{-3}$
AdlNet	109	$3.4*10^{-3}$

Table: Showing the densities of social network of members in different organizations for code related repositories. (Note: Organizations chosen from a mixture of all three activities)

Organization	#Users	Intra-Org density – code related repositories
WhiteHouse	817	$3*10^{-4}$
NASA	283	$3.2*10^{-4}$
KBase	213	$1.2*10^{-3}$
AdlNet	88	$1.5*10^{-3}$

The following tables we indirectly measure the effect of homophily, and we find that users who collaborate on policy had fewer fraction of links created after they joined the organizations. This indicates that in order to collaborate on policy you need to have more prior ties than if you were to collaborate on code. It however, says that there is a less effect of homophily on policy related collaborations. People who work on policies may not form ties very easily after they join organizations as

compared to those who collaborate on code. This might be due to the characteristics of users, generally people who collaborate on code might be young and more receptive to form ties after they think they have worked together. Whereas, members who work on policies might expect prior relationship even before they are willing to collaborate.

Table: Showing the fraction of links created among users after joining the organization for the policy and code related networks.

Activity	% Links created after joining org - Policy	%Links created before joining org - Code
Commit	12%	28%
Fork	23%	35%
Watch	18%	20%

5. CONCLUSION

We first showed that the networks created by the Fork and Commit interactions are reasonably correlated whereas the Watch / Star interaction was not correlated. This showed us two things, the Watch and Star features on Github are necessary since they are not used the same way as commit or fork. Second, the users who do Commit have to Fork. We also showed that the social network created by follower-following relationship is denser for members within an organization. Although we found a few members that have ties across organizations, which is a good characteristic in open collaborations such as these.

Github could use our findings to create features that provide suggestions to users on who to follow and why they should follow someone. Our results showed that there is a need to foster collaborations across organizations and this feature could help facilitate that. Our results also showed that most members of an organization could already have had working relationships with other members and that the Github as a platform did not foster new collaborations. Therefore, we could use this analysis to check if the newer connections are formed after people join organizations in future. Reciprocity is another measure we used to evaluate the effectiveness of these collaborations. Surprisingly, we found a high reciprocity compared to previous research on similar platforms. More study is required to understand why these networks have higher reciprocity than the average on github. Finally we showed that the networks formed by members who collaborate on policies had higher average reciprocity and more dense connections among members within their organization. These findings support the general theories on tie formation in policy networks. However, we did not find evidence of tie formation as a result of homophily in these policy related networks. Since there is an increasing use of Github apart from code, we suggest developing newer features on Github that facilitate collaboration on non-code related tasks and also improve mechanisms to automatically these projects from code related projects.

REFERENCES

Dowding, Keith. "Model or metaphor? A critical review of the policy network approach." Political studies 1995

Sabatier, Paul A. and Hank C. Jenkins-Smith, eds. 1993. Policy change and learning: An advocacy coalition approach. Boulder, CO: Westview.

Henry, A.D., Lubell, M., and McCoy, M. Belief Systems and Social Capital as Drivers of Policy Network Structure: The Case of California Regional Planning. Journal of Public Administration Research and Theory , (2010)

Carley, K. 2002. Smart agents and organizations of the future. In L. Lievrouw & S. Livingstone (Eds.), Handbook of new media: 206 –220. London: Sage.

I. McCulloh and K. M. Carley, “Detecting change in longitudinal social networks,” Journal of Social Structure, vol. 12, no. 3, pp. 1–37, 2011.

J. Xu, S. Christley, and Greg Madey, "Application of Social Network Analysis to the Study of Open Source Software", in The Economics of Open Source Software Development, J. Bitzer and P.J.H. Schröder eds., Elsevier Press, 2006.

G. Madey, V. Freeh, and R. Tynan, "The open source software development phenomenon: An analysis based on social network theory," Americas Conference on Information Systems, 2002.

Crowston, K., & Howison, J. (2005). The social structure of free and open source software development. First Monday, 10(2). doi:10.5210/fm.v10i2.1207

Sagers, Glen, "The Influence of Network Governance Factors on Success in Open Source Software Development Projects" (2004). ICIS 2004 Proceedings. Paper 34.

Ferdian Thung, Tegawende F. Bissyande, David Lo, and Lingxiao Jiang. 2013. Network Structure of Social Coding in GitHub. In Proceedings of the 2013 17th European Conference on Software Maintenance and Reengineering (CSMR '13). IEEE Computer Society, Washington, DC, USA, 323-326.

Jing Jiang; Li Zhang; Lei Li, "Understanding project dissemination on a social coding site," Reverse Engineering (WCRE), 2013 20th Working Conference on , vol., no., pp.132,141, 14-17 Oct. 2013

Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. 2012. Social coding in GitHub: transparency and collaboration in an open software repository. In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work (CSCW '12). ACM, New York, NY, USA, 1277-1286.

Mergel, Ines, Introducing Open Collaboration in the Public Sector: The Case of Social Coding on Github (September 16, 2014). Available at SSRN: http://ssrn.com/abstract=2497204 or http://dx.doi.org/10.2139/ssrn.2497204

Karen Mossberger, Yonghong Wu, Jared Crawford, Connecting citizens and local governments? Social media and interactivity in major U.S. cities, Government Information Quarterly, Volume 30, Issue 4, October 2013, Pages 351-358, ISSN 0740-624X

Contractor, N., Wasserman, S., Faust, K., 2006. Testing multi-theoretical multilevel hypotheses about organizational networks: an analytic framework and empirical example. Academy of Management Journal 31, 681–703.

L. Tang, H. Liu, J. Zhang, and Z. Nazeri. Community evolution in dynamic multi-mode networks. In Proc. 14th ACM SIGKDD international conference on Knowledge Discovery and Data mining, pages 677–685. ACM, 2008