Exploring the Patterns of Social Behavior in GitHub

Yue Yu, Gang Yin, Huaimin Wang, Tao Wang
National Laboratory for Parallel and Distributed Processing
School of Computer Science, National University of Defense Technology, Changsha, 410073, China
{yuyue, yingang, hmwang, taowang2005}@nudt.edu.cn

ABSTRACT

Social coding paradigm is reshaping the distributed software development with a surprising speed in recent years. Github, a remarkable social coding community, attracts a huge number of developers in a short time. Various kinds of social networks are formed based on social activities among developers. Why this new paradigm can achieve such a great success in attracting external developers, and how they are connected in such a massive community, are interesting questions for revealing power of social coding paradigm. In this paper, we firstly compare the growth curves of project and user in GitHub with three traditional open source software communities to explore differences of their growth modes. We find an explosive growth of the users in GitHub and introduce the *Diffusion of Innovation* theory to illustrate intrinsic sociological basis of this phenomenon. Secondly, we construct *follow-networks* according to the *follow* behaviors among developers in GitHub. Finally, we present four typical social behavior patterns by mining *follow-networks* containing *independence-pattern*, *group-pattern*, *star-pattern* and *hub-pattern*. This study can provide several instructions of crowd collaboration to newcomers. According to the typical behavior patterns, the community manager could design corresponding assistive tools for developers.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics - *Process metrics*; D.2.9 [**Software Engineering**]: Management - *Programming teams*

General Terms

Human Factors, Measurement, Management

Keywords

Behavior pattern, Social network, Social coding, Distributed software development

CrowdSoft'14, November 17, 2014, Hong Kong, China
Copyright 2014 ACM 978-1-4503-3224-8/14/11...$15.00
http://dx.doi.org/10.1145/2666539.2666571

1. INTRODUCTION

In recent years, social coding paradigm has been brought into focus in distributed software development for the developers from all over the world. Various kinds of social media [10, 11] are employed in software development, which help building social ties among developers and form different types of social networks. Such social mechanisms can achieve transparency [5] within social coding ecosystem and improve the degree of collaboration in software development.

GitHub[1], a typical social coding community, attracts a large number of users and projects in a short period of time. When launched in 2008, there were only four users [2]. But it seems to rise to fame overnight and increases to more than 3.5 million developers now. GitHub employs several social media such as *follow*, *watch* and *fork*. The developers can track the activities of others and be aware of changes in project using these tools in the community. Many interesting social networks of developers can be constructed. For example, the *follow* relation is created when a developer click the "*follow*" button in the profile of another developer, and then the *follow* relations among developers can form a social network which is called *follow-network* in this paper. Why this new paradigm can achieve such a great success in attracting a large number of developers, and how they are connected in such a massive community, are important questions for understanding such a new paradigm. Many researches are conducted on analyzing the influence of social network in Open Source Software (OSS) communities (see Section 7). However, these work study the network structure [13] of collaboration-oriented social network and collaboration pattern [12] in traditional OSS communities. However, none of them has explored the growth modes of communities and social behavior patterns of developers.

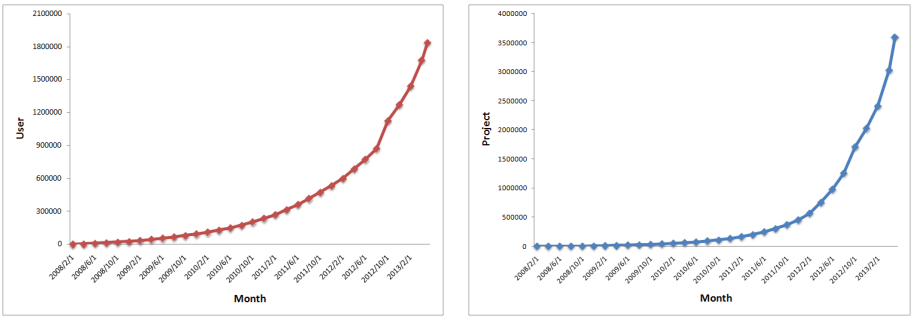
In this paper, we firstly explore the growth curves of GitHub compared to three traditional OSS communities. Then, we construct *follow-networks* from the follow behaviors among developers, which is a typical interest-oriented social network. Finally, we analyze the social behavior patterns among developers by mining the *follow-networks*.

In summary, the following research questions would be answered in this paper:

RQ1: What are the differences between the growth modes of GitHub and traditional OSS communities, and is there any sociological theory that supports the special growth mode of GitHub?

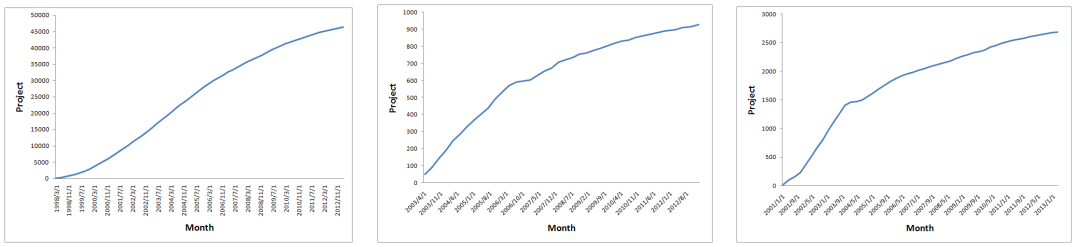
RQ2: Whether or not the social connections among developers form some distinctive behavior patterns in GitHub,

[1]https://github.com



(a) User-growth curve (b) Project-growth curve

Figure 1: The growth figure of Github



(a) Freecode (b) Alioth (c) Savannah

Figure 2: The growth figure of tranditional OSS community

and if it is true, what are these patterns?

To the best of our knowledge, this is the first research combining the high level analysis of the growth mode and the specific level of pattern mining in GitHub. We try to make some interesting observations at these two levels. The remainder of this paper is structured as follows. Section 2 introduces the statistics of raw data entities in our dataset. Section 3 analyzes the growth mode of GitHub and Section 4 illustrates the method of constructing the *follow-networks*. Section 5 depicts the typical behavior patterns hidden in the *follow-networks*. We discuss threats in Section 6 and related work in Section 7. Finally, we draw our conclusions in Section 8.

2. DATASET

Our study is based on the data from the GHTorrent project [8, 6], which keeps on creating a scalable off-line mirror of event streams and provides persistent data of GitHub for research. We use the Mysql dump update until 2013-05-29, which contains detailed information of social coding activities about 1,838,805 users. Among all these users, about 55.46% of them (1,019,839 users), joined this community during a short period of time from 2012-07-01 to 2013-05-29. Since then, the number of users keeps on growing by over fifty thousand per month.

3. GROWTH MODE

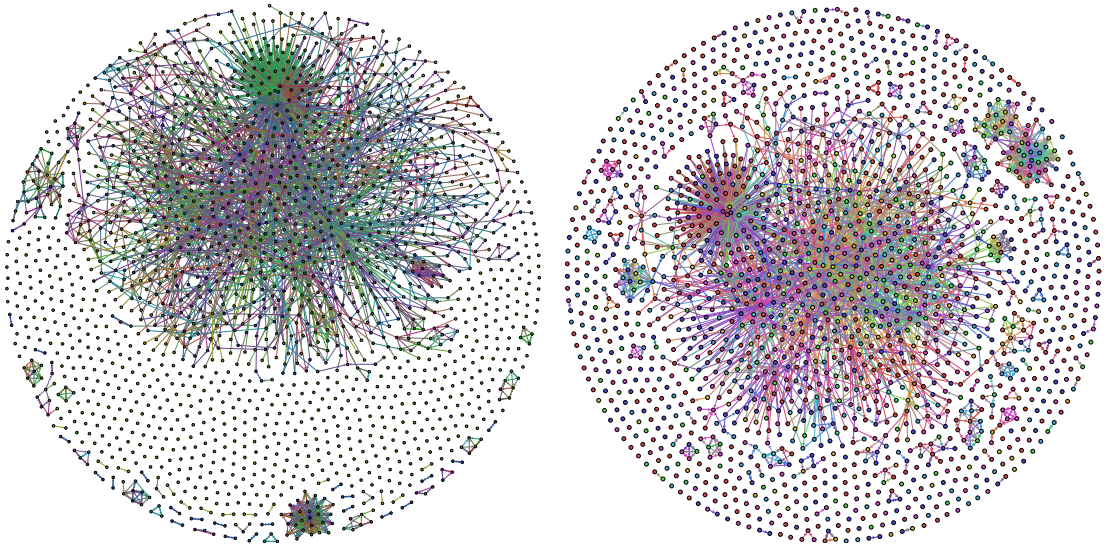
As a popular social coding community, GitHub draws widespread attention from all over the world hosting a huge number of software projects. However, the growth mode of GitHub has a huge difference during two periods of time.

Figure 1 shows the monthly growth trajectory of user and project in GitHub. As can be seen from this chart, after a relatively long time of accumulation till the early 2012, the number of users and projects experienced a big leap in a short time, which seems to make GitHub rise to fame overnight. In this paper, we defined the explosive growth mode of GitHub as "*outburst-type*". The outburst-type is quite different from the growth mode of the traditional OSS communities, such as Freecode[2], Alioth[3] and Savannah[4]. As shown in Figure 2, the traditional OSS communities often grow smoothly and stably. After a period of rising, the growth curves gradually slow down. We use *Gini coefficient* to measure the skewness of outburst-type. The Gini index of three traditional OSS communities is on average 24.5%. By contrast, the Gini index of GitHub is over 58.1%. It means that the growth of GitHub is too imbalanced that the majority of developers join this community in a short period of time.

The core service of these three traditional OSS communities is to support project (code) hosting. In these communities, the main services such as *version control system*, *bug tracking* and *release management* are strongly related to project management. Around the main service, there are some classic communication tools such as *mailing lists* and *forum* used to assist developers in distributed development. Thus, users do not have direct experiences and strong feelings about its strengths.

However, the *human factor* is the core factor in the social coding paradigm. The innovative services in GitHub, such as the follow-based social networking, fork-based sharing system and the pull-based software development model [7], catapult users into a new software develop experiences. According to the *Diffusion of Innovations* theory[9], if there were 2.5% *innovators* and 13.5% *early adopters* hosting their projects on GitHub and promoting to others, the "tipping point" would be achieved. Then, the majority customers

[2]http://freecode.com
[3]http://alioth.debian.org
[4]http://savannah.gnu.org



(a) 2012-08 subset (b) 2012-09 subset

Figure 3: Two examples of follow-network

would join GitHub community. When GitHub leaps the chasm, it grows dramatically. Thus, GitHub grows as the outburst-type.

Furthermore, we hold two viewpoints of the reason why majorities are involved in GitHub.

Effect of leader: There is a part of developers enjoying a high reputation such as *Linus Torvalds*[5] who have been followed by 13,267 users on GitHub. Similarly, some outstanding projects have a lot of eyes on them such as Rails[6] stared by 19,915 users. When these people or projects are active in GitHub, a lot of developers are involved because they want to join the projects or study with the experts.

Herd behavior: A large number of users join GitHub just for the reason that he find so many developers around him talk about GitHub frequently. However, for himself, he may not know the advantages of GitHub clearly.

4. FOLLOW-NETWORK

We aim to understand the social behavior patterns of the developers who join GitHub during the outburst period. We firstly construct *follow-networks* from the follow behaviors among developers, which can directly reflect users' relationships in social activities.

If a user U_1 has followed U_2, we consider that the collaboration activities of U_1 would be influenced by U_2's. The *follow-network* can be defined as a directed graph $G_{fn} = \langle V, E \rangle$. The set of vertices is all users in our dataset denoted by V. The set of edges in G_{fn} denoted by E is a set of node pairs $E(V) = \{(u, v)|u, v \in V\}$. If the node v_j is followed by v_i, then there is a edge from v_i to v_j. For a node v_i, the number of edges pointing to it is called the *indegree* $deg^-(v_i)$ and the number of edges starting from it represents its *outdegree* $deg^+(v_i)$. And the *degree* is the sum of *indegree* and *outdegree* $deg(v_i) = deg^-(v_i) + deg^+(v_i)$.

In this paper, we focus on the social behavior of developers who join GitHub during the period of fast growing. Thus, we divide the dataset into several monthly subsets according to developers' registration time, and then construct the *follow-networks* separately. Table 1 lists the monthly statistics of corresponding *follow-networks*. There are over 85,000 of new users join GitHub each month.

Table 1: Statistics of Dataset

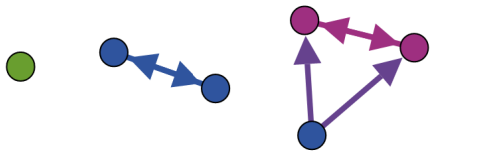
Month	#User	#Node	#Edge	Average Degree
2012-08	**150,851**	32,796	31,677	0.966
2012-09	**102,056**	40,401	40,793	1.010
2012-11	88,857	28,665	26,232	0.915
2013-01	89,004	23,463	19,562	0.834
2013-02	**142,358**	27,064	21,970	0.812
2013-03	95,087	23,650	19,161	0.810
2013-05	90,413	13,160	9,704	0.737

5. SOCIAL BEHAVIOR PATTERNS

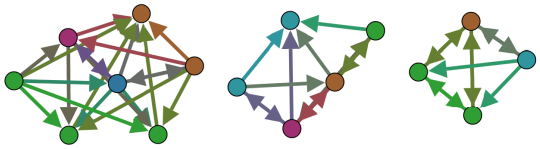
The quantity of registered users is over one hundred thousand in 2012-08 and 2012-09 subsets. Those developers have formed rich social relations after a period of time. Therefore, we choose these two subsets to demonstrate the *follow-networks*. The *follow-network* is so complex that we delete the nodes whose degrees are less than 5. There are nearly 90% useless links that can be filtered. It means that most of users program in GitHub without the help of follow-based social service. Thus, it is possible to show that a large number of developers are involved in GitHub because of *Herd behavior*. In the Figure 3, we show the preprocessed *follow-networks* visualized by Gephi [1]. In general, the *follow-networks* can be divided into two parts, i.e. isolated part and interlaced part.

In isolated part, we can find two typical patterns, containing the *independence-pattern* and the *group-pattern*. Figure 4 shows some typical examples of them. The *independence-pattern* indicates that a developer use Github as a traditional way and he always only link up with acquaintances. He just hosts his code or watches an interesting project but rarely makes a contribution to it. According to our statistics, in the 2012-08 subset, 30.33% nodes are isolated and 13.80%

[5]https://github.com/torvalds
[6]https://github.com/rails/rails



(a) Independence-pattern



(b) Group-pattern

Figure 4: The typical patterns in the isolated part

nodes only connect with one node. The *group-pattern* is often formed by a group developers who collaborate with each other to develop the same project.

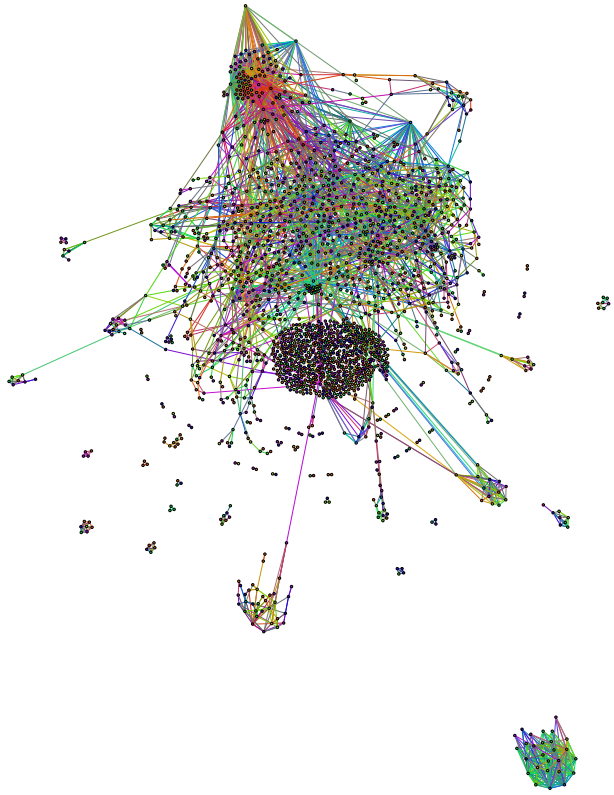
To show the features of the *group-pattern* more clearly, we use two different force-directed graph drawing algorithms to redraw the *follow* network, as shown in Figure 5. We present three observations as follows:

- **Observation 1:** For a given group, the number of links between this group and the interlaced part presents the degree of social collaboration among the group's members and community. If a group has few link with the core network, the projects developed by them would be hard to attract public attentions. Besides, the distance between the group and the centre of the network reflects the degree of correlation between them. For example, the group in the bottom right corner, which is far from the core network, hosts a industrial design project on GitHub. This project has no correlation to software development.

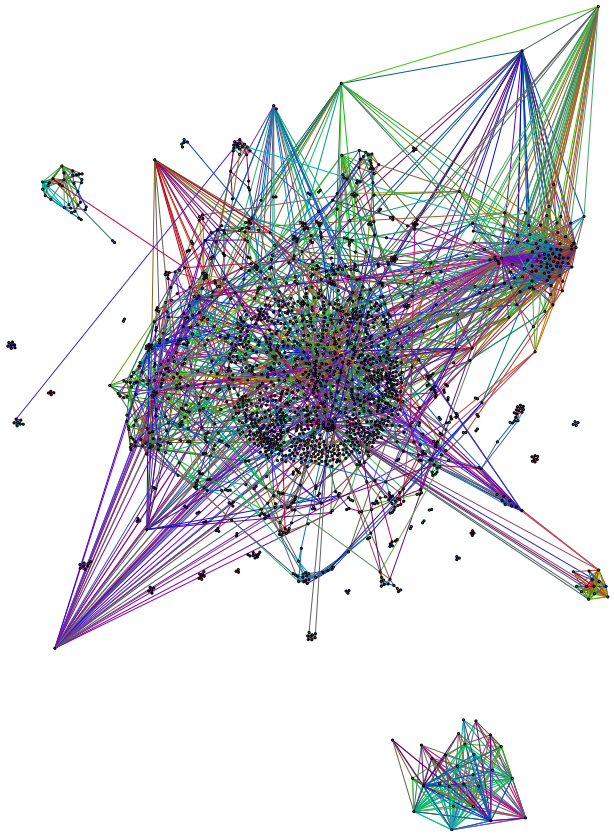
- **Observation 2:** In a group, there is a relatively small number of users who follow the external developers and there are not many internal users following them as well. In addition, these developers always follow the external developers with high indegree who are the leaders of a well-known project. Thus, they are not the core programmer of their project, but they import some novel idea from the community into the group.

- **Observation 3:** In general, the more developers are followed by external users, the faster their project growing. When their project is popular enough in the specific domain, the *group-pattern* would be merged into the interlaced part of *follow-network*, because more and more developers follow the group's members and contribute to their project.

In the interlaced part of *follow-networks*, we extract the community structures using a popular algorithm of community detection purposed by Blondel et al. [4]. As shown in Figure 6, there are 4 large communities in the network which have been painted in different colors. The size of a node represent its indegree. We can find that different communities represent different groups of developers who focus on different kinds of projects. There is a leader in each community. For example, the pink community is about *Ruby on Rails* development and the orange community is related to



(a) The Network redrawn by *Force Atlas* algorithm



(b) The Network redrawn by *Force Atlas 2* algorithm

Figure 5: The redrawn follow-network of 2012-08 subset

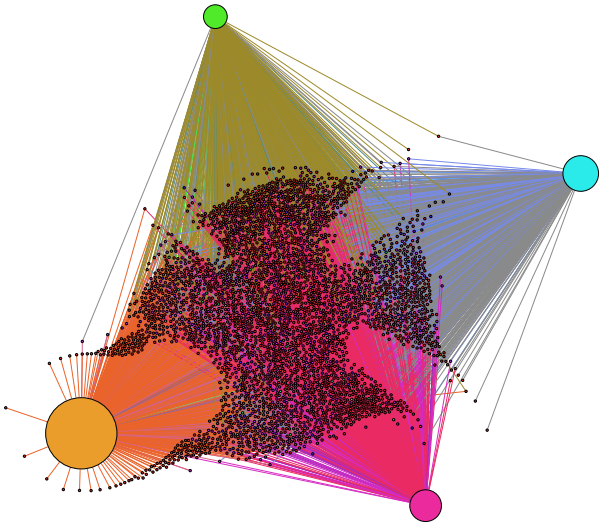
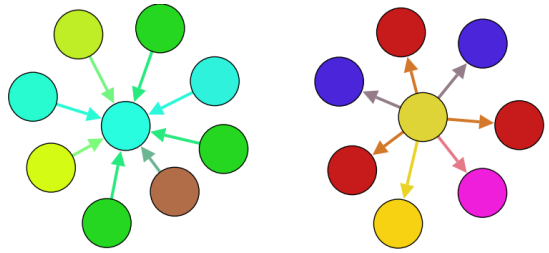


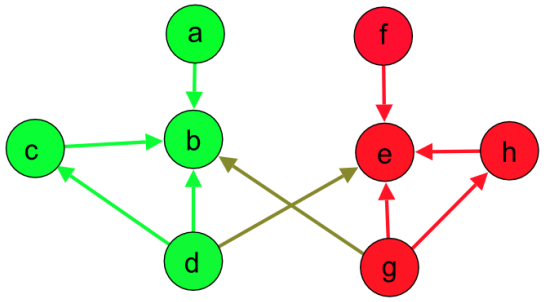
Figure 6: Community structures in the *follow-network* of 2012-08 subset

Linux project. Furthermore, we extract two typical social behavior patterns from the interlaced part of *follow-network*, including *star-pattern* and *hub-pattern*.

As shown in figure 7(a), there are two distinct structures of the *star-pattern*. The first structure indicates that a famous man (or a team) is followed by a large number of users but he almost never pay any attention to others, which exactly reflects the influence about the *Effect of leader* described in Section 3. The other one indicates that a user follow many irrelevant developers but almost never be followed by others. This kind of structure can be used to find crawler's IDs or advertiser's IDs. For example, we find *KBishop*[7] is a advertiser's account in GitHub.



(a) Star-pattern



(b) Hub-pattern

Figure 7: The typical patterns in the interlaced part

To depict the *hub-pattern* clearly, Figure 7(b) show a typical example of the *hub-pattern* with eight labeled nodes. Each node represents a developer of GitHub. The eight developers develop their projects in two different communities. There is a core developer in the corresponding community, such as the developer ***b*** in the green community and ***e*** in the red community. The core developers just have connections with internal users. The *hub-node*, such as ***d*** and ***g***, not only follow internal users, but also make a connection with other communities. In this pattern, we find that the projects developed by different communities always have something in common. For example, they use the same programming language or frameworks. The quantity of hub-nodes is highly related to the commonality and similarity of the projects.

[7]https://github.com/KbishopSTC

6. THREATS TO VALIDITY

In this section, we discuss some threats to validity which may affect the results of our observations. Firstly, the number of projects hosted in GitHub is still growing fast, so GitHub may be still in the early or middle phases of growth. Thus, we cannot ensure the majority of users joint GitHub during one outburst period. That is to say, GitHub may have two or three outburst periods of growth. However, as the market becomes saturated, the growth curve would be slow down. Secondly, a part of users have been included to compensate for users committing to Github without having a GitHub account or shared an account with other developers. In this paper, we do not take these developers into consideration in the preprocessing stage. Thirdly, the follow relations of some users are dynamic. They would follow an expert at the beginning. However, they disengage from the follow relationship at some time for personal reasons.

7. RELATED WORK

With the development of social coding paradigm, many studies have been conducted on analyzing the mechanisms and the value of social network in software development. Begel et al. [2] conduct semistructured interviews with the leader of GitHub to understand the role social network plays in the software development process. Dabbish et al. [5] explore the value of the social media in GitHub and found that the transparency in collaboration brought in by such mechanisms can support innovation, knowledge sharing and community building. Tsay et al. [14] further above study to evaluate the influence of social signals. They find that developers use both technical and social information when evaluating potential contributions to open source software projects.

In addition, collaboration network in social coding has attracted many interests among researchers. Thung et al. [13] investigate the developer-developer and project-project networks in Github. They use PageRank to identify the most influential developers and projects by exploring these two types of network. Surian et al. [12] employ a novel combination of graph mining and graph matching to discover the collaboration patterns in SourceForge. Begel et al. [3] present a framework of social network for connecting developers and their work artifacts together. By analyzing the social network, software engineers can keep track on activities of colleagues and developing status of work artifacts. Vasilescu et al. [15] analyze the interplay between Stackoverflow activities and the commit behaviors in Github, and they find that the developers' activities in the two platforms

are positively associated.

Different from above researches, our work focus on the *follow-network* and analyze the social behavior patterns of crowd developers using sociological theory, which is a brand new perspective.

8. CONCLUSION AND FUTURE WORK

Social coding paradigm exert a tremendous impact on the software engineering activities. In the current, hosting more than 5 million software repositories and attracting over 2 million users, GitHub is one of the most significant open source software communities which is fundamentally changing the traditional paradigms of distributed software development.

In this paper, we analyze the growth curves of Github compared with the curves of traditional OSS communities, we answer the research question that why does GitHub grow in a explosive way. We draw an important conclusion that the *Effect of leader* and *Herd Behavior* are the intrinsic sociological basis of this phenomenon. Furthermore, by mining the *follow-network* of the developers who get GitHub account during the rapid growth period, we illustrate four typical social behavior patterns.

In the future, we plan to study more social behavior patterns about *fork-network*, *pull request-network* and *watch-network* of GitHub. Based on these social behavior patterns, we can develop some novel collaboration tools integrated with the social mechanisms. For example, we can design a recommender system which can push the most relevant projects to users. In addition, we also plan to combine social behavior patterns with our previous work [16, 17] of social software feature mining. According to the social features, we can choose the corresponding collaboration patterns to design prototype system.

9. ACKNOWLEDGEMENT

This research is supported by the National High Technology Research and Development Program of China (Grant No.2012AA011201), the National Science Foundation of China (Grant No.61432020 and No.61472430) and the Postgraduate Innovation Fund of University of Defense Technology (Grant No.B130607).

10. REFERENCES

[1] M. Bastian, S. Heymann, and M. Jacomy. Gephi: an open source software for exploring and manipulating networks. In *ICWSM*, 2009.

[2] A. Begel, J. Bosch, and M.-A. Storey. Social networking meets software development: Perspectives from github, msdn, stack exchange, and topcoder. *IEEE Software*, 30(1):52–66, 2013.

[3] A. Begel, Y. P. Khoo, and T. Zimmermann. Codebook: Discovering and exploiting relationships in software repositories. In *Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering - Volume 1*, ICSE '10, pages 125–134, 2010.

[4] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[5] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb. 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, pages 1277–1286, 2012.

[6] G. Gousios. The ghtorent dataset and tool suite. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, MSR '13, pages 233–236, Piscataway, NJ, USA, 2013. IEEE Press.

[7] G. Gousios, M. Pinzger, and A. v. Deursen. An exploratory study of the pull-based software development model. In *Proceedings of the 36th International Conference on Software Engineering*, ICSE 2014, pages 345–355, 2014.

[8] G. Gousios and D. Spinellis. Ghtorrent: Github's data from a firehose. In *Mining Software Repositories (MSR), 2012 9th IEEE Working Conference on*, pages 12–21, June 2012.

[9] E. M. Rogers. *Diffusion of innovations*. Simon and Schuster, 2010.

[10] L. Singer and K. Schneider. Influencing the adoption of software engineering methods using social software. In *ICSE*, pages 1325–1328, 2012.

[11] M.-A. Storey, C. Treude, A. van Deursen, and L.-T. Cheng. The impact of social media on software engineering practices and tools. In *Proceedings of the FSE/SDP workshop on Future of software engineering research*, FoSER '10, pages 359–364, New York, NY, USA, 2010. ACM.

[12] D. Surian, D. Lo, and E.-P. Lim. Mining collaboration patterns from a large developer network. In *Reverse Engineering (WCRE), 2010 17th Working Conference on*, pages 269–273. IEEE, 2010.

[13] F. Thung, T. F. Bissyande, D. Lo, and L. Jiang. Network structure of social coding in github. In *Proceedings of the 2013 17th European Conference on Software Maintenance and Reengineering*, CSMR '13, pages 323–326, Washington, DC, USA, 2013. IEEE Computer Society.

[14] J. Tsay, L. Dabbish, and J. Herbsleb. Influence of social and technical factors for evaluating contribution in github. In *Proceedings of the 36th International Conference on Software Engineering*, ICSE '14, pages 356–366, 2014.

[15] B. Vasilescu, V. Filkov, and A. Serebrenik. Stackoverflow and github: Associations between software development and crowdsourced knowledge. In *Proceedings of the 2013 International Conference on Social Computing*, SOCIALCOM '13, pages 188–195, 2013.

[16] Y. Yu, H. Wang, G. Yin, X. Li, and C. Yang. Hesa: The construction and evaluation of hierarchical software feature repository. In *SEKE*, pages 624–631, 2013.

[17] Y. Yu, H. Wang, G. Yin, and B. Liu. Mining and recommending software features across multiple web repositories. In *Proceedings of the 5th Asia-Pacific Symposium on Internetware*, Internetware '13, pages 9:1–9:9, 2013.