Winter 12-2019

Rockstar Effect in Distributed Project Management on GitHub Social Networks

Yifan Huang

Wingyan Chung

Rockstar Effect in Distributed Project Management on GitHub Social Networks

Research-in-Progress

[1] Yifan Huang and [2] Wingyan Chung
[1-2] School of Modeling, Simulation & Training, University of Central Florida
[2] Department of Computer Science, The University of Hong Kong
[1] yifan@knights.ucf.edu, [2] wchung@ucf.edu

Abstract

The internet has become increasingly social, opening up new space for online collaboration and distributed project management. Decentralized management techniques such as open-source software, distributed development, and software-as-a-service allow software developers to easily connect online and to solve complex problems collaboratively. Online rockstars, who are well-respected in a community and are followed by numerous other users, often influence the decisions of project managers and clients in software development. Understanding the effects of these rockstars can greatly facilitate technology development and adoption in distributed project management. This paper presents a study of the GitHub social network to understand rockstar effect in distributed project management. In GitHub, developers often collaborate in distributed teams and interact in their online social networks, which evolve with popularity of software repositories and actions of rockstars. To understand how rockstars influence the popularity of software repositories, this research constructed temporal social networks during 2015 to 2017 between 13.5 million software repositories and 2.6 million GitHub users and examined the evolvement of behavior of 245,501 rockstar followers. The results show that the more followers a rockstar has, the more triadic events there are in his/her participated repository. And the difference of number of events between top rockstar and other rockstars is much higher in participative events than in contributive events, indicating higher triadic influence from top rockstar in those events for technology development in distributed project management.

Keywords

Popularity, GitHub, Rockstars, Bipartite Social Networks, Repositories, Network topology.

Introduction

Traditionally, project management involves planning and supervision of homogenous teams to reach predictability on project progresses and budgets. Due to lacking resources and intense competition, companies had been seeking allies to help deliver business value in a more cost-effective way. However, hiring a large group of professional people with a set of required skills can be too costly for most companies. As the Internet loosened the restrictions from the geographic barrier, more and more project management tasks shifted from local to global (Bhagwati, 2004). In today's the world, people are connected more closely as technology has accelerated the spread of information and made it possible for remote and asynchronous collaboration (Friedman, 2007). Distributed project management is becoming a trend to advance how people interact, communicate and achieve project goals across countries and communities. Software development follows the same patterns, companies outsource these tasks to service providers, typically in less expensive areas within the country or other emerging nations. Comparing to contracting the entire project to one or more parties, distributed software development has significant portions of individual contributors that come from different regions and communities. This situation gave rise to new challenges in culture, communication and synchronization (Herbsleb, 2007). Online social media, which greatly facilitates knowledge sharing and virtual interaction (Chung, 2016), is a powerful tool to address these challenges.

Because of the importance of GitHub in modern open-source software development, researchers started to study the popularity of GitHub repositories. For example, the standard folders, such as document, testing, and examples, are not only among the most frequently used, but their presence in a project is associated with increased chances that a project's code will be forked. Popular projects tend to attract more documentation collaborators (Zhu, Zhou, & Mockus, 2014). The factors that impact the popularity of GitHub repositories include programming language, application domain, repository owner (user or organization), age, temporal dynamics (Chung, Vora, et al., 2019), and release frequency (Aggarwal, Hindle, & Stroulia, 2014). Repositories owned by organizations are more popular than the ones owned by individuals and there is a strong correlation between following and participation, a weak correlation between following and revision. The number of times the repositories of the developer have been forked is an indicator of the value of the content produced by the developer and their influences are spread across different programming languages (Dubois & Gaffney, 2014). A software project's success is influenced by the visibility of its developers' activities through motivating others (Dabbish, Stuart, Tsay, & Herbsleb, 2012). Developers choose projects to which they contribute by considering the projects with frequent contributions or with contributions from developers with high status (Storey, Singer, Cleary, Figueira, & Zagalsky, 2014; Treude & Storey, 2010). And transparency of the contributions enables users to form impressions of one's work (Marlow, Dabbish, & Herbsleb, 2013).

People can interact with each other and repositories, forming a complex social network. GitHub also supports social coding features (Borges, Hora, & Valente, 2016b), which have been shown to increase activity and collaboration among users (Zhao & Rosson, 2009). As in any other social media, the "Follow" feature allows users to receive update notification from those they shared common interests, making it possible for a user who has a large number of followers to become a rockstar with tremendous public attentions (Kumar, Gupta, Rai, & Sinha, 2013). Rockstars are influencing their followers' perception and understanding in the coding world. As a cognitive process that occurs in distributed project management, people's new behaviors can be learnt via observation and imitation from rockstars (Bandura & Walters, 1977). It is of great importance to understand how these influences are generated and propagated. In distributed project management platforms like GitHub, developers can present their interests to a repository (i.e. folder of a project containing all of the files of the project and each file's revision history) owned by rockstars by the Watch functionality, which essentially plays the same role as the "Subscription" function in other social networks. Therefore, the number of Watch events of a repository works like an easily accessible and reliable proxy to its popularity. And by definition we measure the number of followers each user had and select the target rockstars for this study. It is challenging to understand how rockstars influences the popularity of repositories.

In distributed project management platforms like GitHub, interactions exist between repositories and users in the form of a bipartite network (Breiger, 1974). Bipartite networks are well studied in recommender systems: the historical record of users' activities are usually represented by the connections in a user-item bipartite graph. In terms of computation speed improvement, with the use of a two-step diffusion in user-item bipartite networks, computation time is reduced while the recommendation accuracy doesn't deteriorate comparing to the standard collaborative filtering (Zhou, Ren, Medo, & Zhang, 2007). For evaluation, repository popularity was measured by the number of stars and popularity prediction is based on a few growth trends identified by KSC clustering algorithm (Borges, Hora, & Valente, 2016a) (e.g., rapid growth or slow growth). And rank of repositories is also taken into consideration. Factors that impact the repository popularity were studied using top 2500 public repositories. Similarly, the number of stars is used to measure popularity difference for a variety of programming languages, application domains, and repository owners. Whether age, number of commits, number of contributors, and number of forks affect a repository's popularity were investigated. And the gains of popularity are concentrated in specific phases of a repository's lifetime (Lee et al., 2013).

The impact of releasing new repository on popularity is another interesting aspect to study. Influence in GitHub was measured by number of followers and stars and generally data is collected from an API GHTorrent (Dubois & Gaffney, 2014). Common features include user mentioned count, user quoted count and user up voted count. Common metrics for bipartite network include degree centrality, clustering coefficient, closeness centrality, diameter, etc. These metrics were also applied to studies of specific programming languages such as Python (Weber & Luo, 2014), where features were measured on the relative occurrence of various Python abstract syntax tree (AST) nodes. On the other hand, some less commonly used features are specific to certain domains. These features include but not limited to project volume,

documentation volume, presence of supporting files, code volume, standard library usage, and popularity velocity.

A Study of Rockstar Effects on Distributed Project Management

While existing research mainly conducts surveys and machine learning methods to study this phenomenon, there is a need to gain deeper understanding of how rockstar influences the social network by examining link formation, network topology and its effect on user behavior such as triadic influence, transitive effect, and importance of interaction among network nodes (Chung, Rao, & Wang, 2019). The goal of this study is to examine link formation in the presence of rockstars in online social networks used for distributed project management. As a research test bed, GitHub can be modeled as an undirected bipartite network (graph), whose nodes (vertices) are divided into two disjoint and independent sets: *Repository* and *User*, and that has each link (edge) connecting one node in *Repository* to one node in *User*. To understand how rockstars influence the popularity of distributed project management, we raised three research questions:

RQ #1: Do rockstars influence the popularity of repositories? If so, what's the level of influence and response time?

RQ #2: Do rockstars have equal influence on popularity of all types of events in repositories? If not, how different types of events are influenced by rockstars? (contributive and participative events)

RQ #3: Do rockstars' activities on a repository trigger activity from their followers on the same repository? If so, how does it affect network formation over time?

In GitHub, people follow those who are well-respected in a community that influence the decisions of project managers and clients in software development. To study this phenomenon, first we selected rockstar as users having the highest number of followers. Popular repositories were identified by calculating the overall "Watch" events a repository got throughout the entire dataset. Second, we constructed daily bipartite networks between repositories and users and examined the rockstar effect by observing the evolvement of followers' behavior in a time-sensitive manner.

Dataset

The dataset used in this paper spans from 2015-01-01 to 2017-08-31 with 13.5 million software repositories, 2.6 million users and 245,501 rockstar followers. The entire graph of our data has 52,260,372 nodes and 870,532,947 edges. Top 100 repositories were examined. There are two major reasons to limit the number of repositories: 1. To focus on the characteristics of the highly popular GitHub repositories. 2. Since this research is to investigate the impact of rockstar on repository's popularity, which demands a manual observation and classification on some parts of the experiments, limiting the number of repo enables more fine-grained manual observation and investigation of Rockstar impact. All data was obtained using the GitHub API (Gousios & Spinellis, 2012), which provides services to search public repositories and to retrieve specific data about them (e.g., number of follows, number of code updates, posted issues, and project participations). To retrieve the number of "Watch" events on different dates for each repository, we used service from the API (Gousios & Spinellis, 2012) to get all the events of our target repository.

Results and Implication

We implemented different network-based approaches to model and experiment the dynamics between repo popularities and networks. And we examine how the heavy-tailed degree distribution, user interactions, small-world property and rockstar's roles impact the popularity of repositories. For preliminary results, rank 1 rockstar has 52,722 followers and rank 2 rockstars have 30,161 and 25,827 followers respectively. All the top ten rockstars have at least 14,000 followers. Figure 1 shows the Rockstar-participated-repository's contributive and participative popularities respectively spanning from 2015-01-01 to 2018-01-31. For contributive events, users generally are developers and can modify and update the repository; while for participative events, users typically receive update information and notifications passively. The x-axis represents the order of the months from 2015-01-01 to 2018-01-31 and the y-axis shows the montly contributive event counts and participative event counts respectively. From our empirical observations, rockstars that have more followers tend to have more participative events in their participated repositories.

And the same holds true for contributive events. Although the overall number of events correlates to the rockstar's popularity. The number of contributive events peaked at 1st month with over 5,000 events and the number of participative events peaked at 6th month with over 25,000 events. The #1 rockstar has much higher influence in participative events than in contributive events in the early stages while others have normal fluctuations over the 3 years. Figure 2 (left) illustrates the rockstar triadic influence on follower. When rockstar commits an activity in a repository (e.g., WatchEvent, ForkEvent, PushEvent or CreateEvent), it triggers his/her follower to commit activities in the same repository (e.g., WatchEvent, ForkEvent, IssueEvent, IssueCommentEvent, PushEvent, PullRequestEvent or CommitCommentEvent). From Figure 1 we can see that rockstars with more followers has stronger triadic influence on his/her participated repository (with more overall monthly events). The difference of number of events between top rockstar and other rockstars is much higher in participative events than in contributive events, indicating higher triadic influence from top rockstar in those events for technology development in distributed management system (Bandura & Walters, 1977). Apart from the triadic influence, there are more network dynamics that is worth to study such as subgraphs of two-star (e.g., number of rockstar with FollowEvent from two users), three-star (e.g., number of repositories with WatchEvent from three users), triangle (e.g., three users form a following chain), and clique (e.g., four users that have events in every pairs of them) for social network link forming probability prediction (Chandrasekhar & Jackson, 2016). Oftentimes only final network is observed but the formation process that reveals the online social dynamics is neglected, hence the social network link formation is an important aspect for this study.

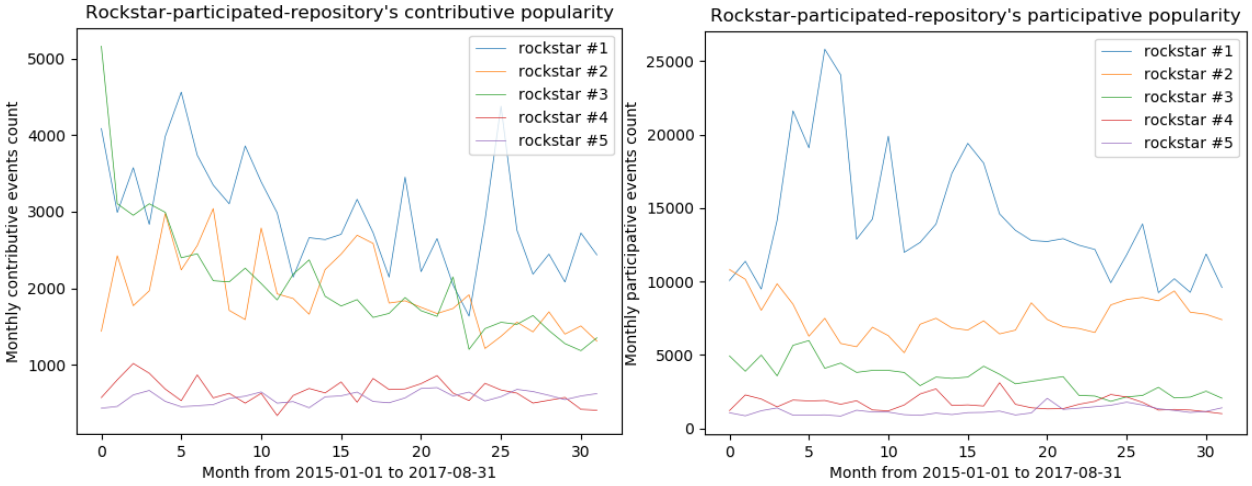

Figure 1. Rockstar-participated-repository's contributive and participative popularity

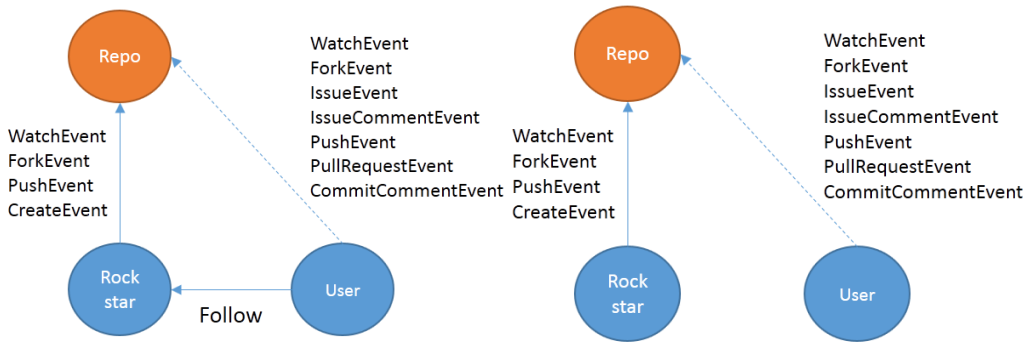

Figure 2. Rockstar Triadic Influence on Follower

Conclusions

We presented a network-based approach in bipartite social network to understand rockstars influence in distributed project management system. GitHub, as an important and popular online software development platform, provides features for developers to collaborate in distributed teams and interact with community members. As popular repositories come out or influential users have activities, dynamics of the social network evolve, as quantified by the network metrics. In this work, we constructed daily bipartite social networks during 2015 to 2017 between 13.5 million software repositories and 2.6 million GitHub users and observe the evolvement of behavior of 245,501 rockstar followers. From the interactions between influential and normal users, we inspected and studied the rockstar effect in complex online social networks. Empirical results show that rockstars have diverse effects on participative and contributive events. The more followers a rockstar has, the more triadic events there are in his/her participated repository. Meanwhile, the difference of number of events between top rockstar and other rockstars is much higher in participative events than in contributive events, indicating higher triadic influence from top rockstar in those events for technology development in distributed project management. In future work, we will keep exploring the rockstar triadic effect by investigating the behaviors of the users that are not rockstar's followers (Figure 2 right), and how subgraphs of two-star, three-star, triangle and clique affect the formation of the social network.

Acknowledgements

This research was supported in part by the U.S. Defense Advanced Research Projects Agency (contract FA8650-18-C-7824). The authors thank all project members, the conference editor and reviewers.

References

Aggarwal, K., Hindle, A., & Stroulia, E. (2014). *Co-evolution of project documentation and popularity within GitHub*. Paper presented at the Proceedings of the 11th Working Conference on Mining Software Repositories.

Bandura, A., & Walters, R. H. (1977). *Social learning theory* (Vol. 1): Prentice-hall Englewood Cliffs, NJ.

Bhagwati, J. (2004). *In defense of globalization: With a new afterword*: Oxford University Press.

Borges, H., Hora, A., & Valente, M. T. (2016a). *Predicting the popularity of GitHub repositories*. Paper presented at the Proceedings of the The 12th International Conference on Predictive Models and Data Analytics in Software Engineering.

Borges, H., Hora, A., & Valente, M. T. (2016b). *Understanding the factors that impact the popularity of GitHub repositories*. Paper presented at the Software Maintenance and Evolution (ICSME), 2016 IEEE International Conference on.

Breiger, R. L. (1974). The duality of persons and groups. *Social forces, 53*(2), 181-190.
Chandrasekhar, A. G., & Jackson, M. O. (2016). A network formation model based on subgraphs. *Available at SSRN 2660381*.
Chung, W. (2016). Social media analytics: Security and privacy issues. *Journal Of Information Privacy & Security, 12*(3), 105.
Chung, W., Rao, B., & Wang, L. (2019). Interaction Models for Detecting Nodal Activities in Temporal Social Media Networks. *ACM Transactions on Management Information Systems*.
Chung, W., Vora, M., Liu, J., Huang, Y., Mustaine, E., & Lai, V. S. (2019). Simulating Temporal Dynamics in Cryptocurrency Software Social Networks.
Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012). *Social coding in GitHub: transparency and collaboration in an open software repository*. Paper presented at the Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work.
Dubois, E., & Gaffney, D. (2014). The multiple facets of influence: Identifying political influentials and opinion leaders on Twitter. *American Behavioral Scientist, 58*(10), 1260-1277.
Friedman, M. (2007). The social responsibility of business is to increase its profits. In *Corporate ethics and corporate governance* (pp. 173-178): Springer.
Gousios, G., & Spinellis, D. (2012). *GHTorrent: GitHub's data from a firehose*. Paper presented at the 2012 9th IEEE Working Conference on Mining Software Repositories (MSR).
Herbsleb, J. D. (2007). *Global software engineering: The future of socio-technical coordination*. Paper presented at the Future of Software Engineering (FOSE'07).
Kumar, A., Gupta, S. K., Rai, A. K., & Sinha, S. (2013). Social networking sites and their security issues. *International Journal of Scientific and Research Publications, 3*(4), 1-5.
Lee, M. J., Ferwerda, B., Choi, J., Hahn, J., Moon, J. Y., & Kim, J. (2013). *GitHub developers use rockstars to overcome overflow of news*. Paper presented at the CHI'13 Extended Abstracts on Human Factors in Computing Systems.
Marlow, J., Dabbish, L., & Herbsleb, J. (2013). *Impression formation in online peer production: activity traces and personal profiles in github*. Paper presented at the Proceedings of the 2013 conference on Computer supported cooperative work.
Storey, A., Singer, L., Cleary, B., Figueira, F., & Zagalsky, A. (2014). *The (r) evolution of social media in software engineering*. Paper presented at the Proceedings of the on Future of Software Engineering.
Treude, C., & Storey, M.-A. (2010). *Awareness 2.0: staying aware of projects, developers and tasks using dashboards and feeds*. Paper presented at the Software Engineering, 2010 ACM/IEEE 32nd International Conference on.
Weber, S., & Luo, J. (2014). *What makes an open source code popular on git hub?* Paper presented at the Data Mining Workshop (ICDMW), 2014 IEEE International Conference on.
Zhao, D., & Rosson, M. B. (2009). *How and why people Twitter: the role that micro-blogging plays in informal communication at work*. Paper presented at the Proceedings of the ACM 2009 international conference on Supporting group work.
Zhou, T., Ren, J., Medo, M., & Zhang, Y.-C. (2007). Bipartite network projection and personal recommendation. *Physical Review E, 76*(4), 046115.
Zhu, J., Zhou, M., & Mockus, A. (2014). *Patterns of folder use and project popularity: A case study of GitHub repositories*. Paper presented at the Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement.