CS224W Reaction Paper and Proposal

Tiffany Low (tlow), Joseph Marrama (jmarrama)

October 21, 2012

Code is about the people writing it. We focus on lowering the barriers of collaboration by building powerful features into our products that make it easier to contribute. (**GitHub motto**)

1 Introduction

In an age of social media, source control management is evolving to take on a more collaborative flair. Communities like GitHub, SourceForge, BitBucket and Redmine are all examples of online communities centered around collaborative coding. These systems typically include expected features such as version control, branching, and merging, and also incorporate social media features such as commenting, followers and groups. We want to explore the nature of communities and collaboration in one of the largest and most popular websites, GitHub, to see if it has lived up to its promise of "social coding".

During the course, we will learn about community discovery algorithms and modeling community growth and interaction. We selected papers that presented algorithms for community discovery and that had a focus on evaluating the success of their algorithms. We first examined Newman and Girvan's approach to discovering communities in a social network. Although we found their methodology sound, their model lacked sophistication and left out important considerations about the quality of interaction between users [5]. The work by Brandes et al. presented a powerful model of collaborative structures within Wikipedia and is the motivation for our research on the GitHub community [1]. We intend to borrow from their approach but use a different model than the one proposed in their paper. Finally, the paper by Jin et al. attempted to combine the work of a more sophisticated user interaction model with a community detection algorithm. We were interested by their definition of modularity and betweenness given this new model, and hope to improve upon their work by applying a more robust algorithm.

Based on our analysis of these papers, we propose our approach to determining the collaboration structure of the GitHub social graph.

2 Summary

a) Finding and evaluating community structure in networks

Newman and Girvan chose a divisive approach to finding communities in a given network, citing failings of agglomeration methods. It applies measures such as betweenness (shortest geodesic path), random-walk betweenness and current-flow betweenness to examine their effectiveness in detecting communities.

In their proposed algorithm, the betweenness value of all edges in the network are computed. Then, the edge of least betweenness is removed. These steps are repeated to produce a hierarchical division of the network. However, based on this hierarchical grouping, many divisions could be made.

To determine the best division, they introduced the concept of modularity, a factor based on the fraction of outgoing edges of the community to other communities and the fraction of edges within the community. A high modularity score indicates a strong community network. For each proposed division, the modularity score of the resultant graph is returned and the best division occurs at the peak modularity value.

b) Network Analysis of Collaboration Structure in Wikipedia

This paper primarily focused on analyzing the Wikipedia edit network through a new measure *bipolarity*, which estimates to what extent there are two opposing communities of editors. They build upon the Wikipedia edit network, which contains every addition, deletion, and other modifications to wikipedia pages. Using this data, they develop an expressive network model containing many different measures of user engagement and activity. To evaluate *bipolarity* and other measures in their network, they compare the measures on two sets of wikipedia pages: those that are labelled as 'controversial' pages, and those that aren't controversial but are highly active. They find that controversial wiki pages actually have higher bipolarity measures on average, as they predict.

c) A Center-based Community Detection Method In Weighted Networks

The authors propose an algorithm that begins with finding the top k nodes that will be the centers of communities within the network. These centers are found using a variant of the PageRank algorithm to score the centrality of nodes in the graph. Given these centers, the groups within the network can then be agglomerated. To improve the result, an additional adjustment step can be done, similar to the recursive step in the k-means algorithm. Because they are finding communities in a weighted network, the authors had to define a new measure to replace modularity and betweenness in the original unweighted variant of the problem. They define a measure of *connectivity strength* describing how close any two nodes in the graph are to each other, and assign nodes to the group that maximizes the connectivity strength between the node and the group.

3 Critique

a) Finding and evaluating community structure in networks

Newman and Girvan demonstrated the algorithm across several datasets and showed clearly how to find the peak modularity values given a hierarchical division of nodes. It also explained limitations of earlier methods and the improved performance of the proposed algorithm. They provided multiple measures of betweenness and evaluating them to determine there was little difference in results, and that a simplification could be made to simplify runtime (using shortest path with $O(mn)$ runtime).

However, the graph edges are unweighted. All interactions are modelled to be of the same weight and this is an assumption that leaves out the intensity and nature of this interaction between nodes in the graph. This model is applied to multiple datasets without consideration of domain specific constraints. The main weakness of the paper was the lack of ground truth with which to compare the algorithm's results. Thus, the utility of the proposed divisions could not be assessed. In addition, the paper made the constraint of having nodes classified into exactly one community. However, real world datasets often come with some ambiguity and overlap between groups. The model was not expressive enough to represent such overlaps in membership.

Newman and Girvan's introduction of modularity was not given sufficient treatment to justify why such a measure was meaningful or important with respect to community identification. In Fortunato and

Barthlemy's paper[2], we see that the measure has a resolution limit for which communities below a scale proportional to the connectivity of the network would not be identified. More work could have been done to explain and justify their choice of measure.

Finally, their model treats graph edges as having unit cost, ignoring interactions that are better modelled using weighted graphs or graphs with positive or negative values. For example, in Wikipedia, the contribution of an edit versus a delete/revert action is substantially different.

b) Network Analysis of Collaboration Structure in Wikipedia

The paper didn't spend much time or effort on evaluating the various measures it proposed. The brief time they spent analyzing their results was primarily spent making conjectures about user behavior that supported their results, rather than proposing and running more experiments to support their claims. For example, they guess that authors of controversial articles "try to include both points of view" to explain a counter intuitive result. Rather than test this hypothesis, which they could likely have easily done with their model, they simply say that this has to be validated in future research.

Another potential shortcoming of the paper was that it didn't explore dividing the edit network into multiple classes, instead of just two. They could have easily extended their bipolarity measure to produce multiple "n-polarity" measures, which could have improved their results in cases where there are actually more than 2 competing groups of editors.

c) A Center-based Community Detection Method In Weighted Networks

The authors propose an efficient algorithm that considers the weight of edges when detecting communities in a network. They also define a new optimization criteria, identifying the shortcomings of modularity and betweenness measures.The algorithm is benchmarked against known datasets and the strengths and weaknesses of their proposed algorithm are explained.

The biggest weakness to their approach is the need to provide the number of communities in the network. The proposed algorithm includes a post-grouping adjustment step to optimize the centers of the detected community groupings, which makes their algorithm resemble a modified version of k-means (that completes in a single iteration). The algorithm relies on weights when selecting the centers of communities and the authors do not explain the best way to tune these parameters in a given problem instance. Additionally, the approach is agglomerative and does not explain how to address the problem identified by Newman and Girvan of grouping peripheral vertices.

Their paper could have been more effective if they had applied their algorithm in the unweighted case (a special instance of the weighted problem where all edge weights are unit weight), to demonstrate the difference in modularity and their proposed replacement of connectivity strength. The inclusion of such a comparison would have made their results more convincing.

4 Assessment

Newman and Girvans work gives a good starting point for future work on community detection. It defines an important measure used in future works to determine the effectiveness of a particular partition of nodes in a network. The paper references several datasets used as benchmarks to demonstrate the results of the algorithm. We can apply the same methodology as a comparison of our approach against their results. In particular, we feel that there is a strong need to extend their basic model to incorporate further information

about the dataset. Community detection in a weighted representation of the network would allow for more insights into the range of user collaboration within the community and demonstrate the shortcomings of assumptions of betweenness used in the algorithm.

Brandes et al.s paper could provide valuable insight for modeling GitHub on a 'micro' basis; that is, modeling collaboration in a single GitHub repository with a single network. The paper leverages the rich dataset of Wikipedia edits to construct a rich model at the 'micro' level. The dataset offered by repository revision history is very similar to that of individual Wikipedia pages; there are records of every 'edit' done to the source code and their size, what they changed, etc. However, there is one crucial distinction that may detract from the relevancy of Brandes et al.s paper to our project: Wikipedia pages generally have many 'negative' edits where editors delete the work of others', but souce code repositories likely have very few 'negative' edges. This paper primarily tries to find how opposed editors are in a Wikipedia page, but that opposition may not exist in a GitHub repository. It would be very interesting to try and validate this hypothesis, or to look at different directions we can take modeling GitHub repositories on a 'micro' basis.

Jin et al. provides a theoretical framework for community detection given a weighted graph. However, their implementation could be compared more robustly with benchmark algorithms for community detection in the unweighted case. Their results are promising and we feel that further work should attempt to generalize the unweighted network problem to an instance of community detection in a weighted graph. Another question to explore is that of creating a new measure to evalute solutions in the weighted scenario, as the typical measure of optimality, modularity, ignores the weighted contribution of edges.

From the papers analyzed, we see the need to model a social graph with more sophisticated interactions and to provide an analysis of how such modeling will improve community detection. While Newman and Girvan demonstrated a general framework in which to address the community detection problem, in Brandes et al.s paper it was shown that incorporating signed interaction weights revealed the kind of interactions between users that could result in further stratification. These two papers thus address the problem of community detection on different fronts.

The work by Jin et al. can be seen as an attempt to incorporate both ideas into a single work. However, there are some flaws in the algorithm and approach taken by the authors. Even so, they observed improved accuracy in community detection on common datasets using the additional weight information. We see the importance of further work in applying such ideas to real world datasets.

5 Proposal

a) Problem Statement

We intend to evaluate the breadth and depth of teamwork across GitHubs community by modelling the social graph in GitHub as an instance of the community detection problem. GitHub's motto is 'Social Coding' and we are interested to know to what extent that goal has been achieved. Although much work on collaboration has been done on services like Wikipedia, less attention has been paid to the similar problem of collaboration in open source development communities. Taking into consideration the work of Brandes et al. and Jin et al., we intend to detect communities in the graph of users and repositories.

b) Dataset

GitHub publishes their data publicly and it is indexed by Google BigQuery. The dataset contains a list of every public event (including commits, comments and pull requests) within the year. BigQuery supports SQL-like requests to return a subset of events given a time range condition or some other criteria. A count

of all the actions in the database reveals 29 million commits and 9 million repositories created in the total dataset.

We intend to limit our dataset to a range of 3 months, ending on Oct 1 2012. The dataset will be based on commits within that time period and contain information about the type of event, the event creator and the repository affected. The public dataset includes various other events such as comment creation, issue creation and repository creation, but our focus is on commits as a form of meaningful contribution from a user to a repository. We thus focus on a subset of all the events available in the full dataset.

Event type	Count
Push	11218575
Create	3217963
Watch	2142209
Fork	826028
Pull Request	760442
Follow	605770
Delete	138015

Table 1: Count of event types between July 2012 and Oct 2012

To evaluate the results from our community detection algorithms, we intend to do a comparison of the collaboration score between users (or user and repository) against the watcher and follower data available on GitHub. Users can choose to follow other users or repositories of interest and receive notifications of any activity in those streams. We will need additional calls to the GitHub API to pull the current number of followers for users and watchers for repositories.

Repository	Commits by kozo2	Commits by kaizu
ecell/reaction_reader	138	183
ecell/ecell3	8	2
kozo2/conf	82	0
cytoscape/kgmlreader	12	0
ecell/ecell	12	0
kozo2/kozo2.github.com	12	0
ecell/newio	9	0
ecell/pd_visualizer	4	0
ecell/epdp	0	35
ecell/ecell3-spatiocyte	0	2

Table 2: Commit history for two users, kozo2 and kaizu

c) Model

We plan to model GitHub on a macro scale as a weighted, undirected graph where the nodes represent users, and each edge contains the collaboration score $collab(u_i, u_j)$ between two users. Using this graph, we will be able to run community detection algorithms to find communities of collaborators on GitHub.

The challenging part of constructing our model lies in computing the collaboration score, $collab(u_i, u_j)$, for all pairs of users. We plan to calculate this score as the total amount of collaboration that two users

participate in across all repositories, where collaboration is defined as the minimum number of commits that each user has made to a repository. To aid in this, we will construct another graph of GitHub, this time as a heterogeneous weighted graph where the nodes represent users and repositories. There exists an undirected edge between a user u and a repository r of weight $w_{u,r}$, where $w_{u,r}$ is the count of commit events found in the dataset. The collaboration score, $collab(u_i, u_j)$, will be defined as $\sum_{r \in R} min(w_{u_i,r}, w_{u_j,r})$. This general formulation of the collaboration score will likely form the basis of our model. We provide an example dataset between two users. In the case of kozo2 and kaizu, we will have a collaboration score of 138+2 = 140. We may consider a logarithmic weighting instead of linearly weighting the collaboration score.

Depending on how this formulation of the collaboration score performs, we may change the weighting scheme $w_{u,r}$. We make a strong assumption here that the larger the number of contributions to a repository, the more of a contribution a user has made. To model the quality of submissions, however, we could take into account the size of the commit, or the number of lines added in a commit. We may consider such additional detail in our project for a subset of users in the network.

d) Algorithm

In order to partition our model into disjoint communities of users, we plan to try a variety of community detection methods related to the methods found in Jin et al. and Newman and Girvan. A simple approach we plan to implement first is the algorithm found in Newman and Girvan. This would provide a nice starting point to evaluate our network. However, this doesnt account for the weights on each edge, so a good next step would be to formulate a different version of the betweenness and modularity measures that will take into account the weights of each edge.

We also plan to use the methods contained in Jin et al.. While one of the main limitations of the algorithm presented is that it uses a fixed number of communities from the start, we can sidestep this by using the number of communities found by the methods in Newman and Girvan. Also, we plan to investigate making modifications to the metrics contained in Jin et al. in hopes of applying them to the divisive clustering algorithm in Newman and Girvan. Specifically, we want to modify the *CSw* metric, which measures connectivity strength between nodes, to measure the connectivity between different communities instead. This could serve as another good measure of modularity of the detected communities. Also, we want to modify the *centrality* metric, which quantifies the centrality of a given node, to instead measure betweenness. Without running experiments on actual data it is hard to tell if these directions will be fruitful, but we were impressed with the metrics contained in Jin et al. and with the elegance of the divisive clustering algorithm in Newman and Girvan, so we would like to try and combine the two. As a sanity check to ensure that any new methods we develop actually work, we will evaluate the resulting detected communities with established metrics of modularity and examine a subset of the results by hand. We can also evaluate it against other standard datasets commonly used in evaluating

e) Evaluation

In GitHub, users can follow other users or watch repositories to track activity on a per-user or a per-repository basis. Also, many GitHub users are part of organizations on GitHub that correspond to real world companies and groups. We can pull this data from the GitHub dataset and use it to create additional networks of the social structure of GitHub. These graphs will serve as our evaluation data, so that we can compare our computed collaboration scores and communities to see if our model correlates with real world information.

As we are proposing a new algorithm for community detection in a weighted network, we will test our algorithm against our dataset as well as some other well-known baseline datasets for community detection. We will also investigate the difference between applying popular community detection algorithms for unweighted graphs and our algorithm when applied to an unweighted instance of the social graph.

f) Deliverables

We will construct a dataset of graph information given public GitHub data and provide graphs of the distribution of communities by connectivity and by size. Given our evaluation set, we will examine the proposed communities as found by our algorithm and assess the relationship between watch/follow relations and collaboration scores in our results.

References

[1] Ulrik Brandes, Patrick Kenis, Jürgen Lerner, and Denise van Raaij. Network analysis of collaboration structure in wikipedia. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 731–740, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-487-4. doi: 10.1145/1526709.1526808. URL `http://doi.acm.org/10.1145/1526709.1526808`.

[2] Santo Fortunato and Marc Barthlemy. Resolution limit in community detection. *Proceedings of the National Acadamy of Sciences of the United States of America (PNAS)*, 104(1):36–41, 2007. URL `http://www.pnas.org/content/104/1/36.abstract`.

[3] Jie Jin, Lei Pan, Chongjun Wang, and Junyuan Xie. A center-based community detection method in weighted networks. In *Proceedings of the 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, ICTAI '11, pages 513–518, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-0-7695-4596-7. doi: 10.1109/ICTAI.2011.83. URL `http://dx.doi.org/10.1109/ICTAI.2011.83`.

[4] Jie Jin, Lei Pan, Chongjun Wang, and Junyuan Xie. A center-based community detection method in weighted networks. In *Proceedings of the 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, ICTAI '11, pages 513–518, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-0-7695-4596-7. doi: 10.1109/ICTAI.2011.83. URL `http://dx.doi.org/10.1109/ICTAI.2011.83`.

[5] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review*, E 69(026113), 2004.