Predicting Social Network Evolution from Community Data Partitions

Neda Hajiakhoond Bidoki, Madeline Schiappa,
Gita Sukthankar, and Ivan Garibay

University of Central Florida, Orlando, FL USA

Abstract. Social media users exhibit repetitive behavior patterns that can be leveraged to predict trends in network evolution. Our hypothesis is that these patterns exhibit greater consistency within a single community of users; hence global data distributions can be more accurately modeled by composing community data distributions. This paper compares two different strategies for predicting social media usage with sampled historical data on Reddit and GitHub. We demonstrate that our community-based model outperforms the global one at predicting population, user, and content activity, along with network topology over three different datasets.

Keywords: social network evolution · community structure · prediction

1 Introduction

Our empirical understanding of social media campaigns and influence operations greatly outpaces our theoretical knowledge of virality, gatekeepers, and social influence [1–3]. The aim of our research is to accurately simulate the dynamics of cross-platform information propagation and network evolution at multiple levels of granularity: user, community, and population. Instead of employing a purely machine learning prediction approach [4], our team has created a high-fidelity agent-based simulation of user activity. To generalize across platforms, our models are initialized with data distributions and network structure extracted from different social media sources.

Our agent-based models rely on a combination of evolutionary model discovery [5], data-driven user archetypes [6] and parameter fitting. However, our tests have revealed the truth of the saying "the world is its own best model"[1] and thus our simulation directly employs sampled historical data to generate user background activity. Events generated by the information propagation models are mixed with sampled historical data to predict the final population level event sequences.

Rather than treating the population level patterns as being the aggregate result of individual activities, we propose that many of these social media platforms can be modeled as collections of virtual organizations. For instance, social coding platforms such as GitHub are designed to empower teamwork and

[1] Rod Brooks's assertion about the role of sensors in robotics

collaboration between software developers [7]. Subreddits facilitate rapid news exchange among user communities about niche topics.

Inspired by Katz and Tushman [8] who note that "organizations can be viewed as social groupings with relatively stable patterns of interaction over time", this paper introduces a community-based data mixture model for simulating the background activity of social media users. Our tests show that the community-based model outperforms our population level data model for platforms such as GitHub and Reddit, where users exhibit repetitive usage patterns. The next section describes how we preprocess the data to create time series data and sequential network snapshots, extract communities and use their features to predict the network evolution.

2 Method

Using historical data of user interactions on a social media platform, our aim is to predict its future nodes and edges, along with edge types and the time of edge formation. Given a large bipartite network of $G = (U, V, E, T)$ in which U, V, E and T are two set of nodes, edges and edge formation time, we want to predict $g = (u, v, e, t)$ for a specific period of time. Depending on the social media platform, edges can have different types; for instance, Reddit has post and comment interactions, whereas GitHub has events related to code version control such as push and pull.

2.1 Data Description

Our cross-platform dataset contains one year of communications and events related to three categories of software projects with existing communities and the strong potential for cross-platform propagation: 1) cryptocurrency (crypto), 2) cyber-threats (cyber), and 3) common vulnerabilities and exposures (cve).

Our experiments were conducted on data extracted from two large online social media platforms, GitHub and Reddit. GitHub is a social coding platform that is employed by open source developers for source code management and issue tracking. Reddit is an social news aggregation website where participants can submit, rate, and vote on content. The data was subdivided into periods of one week; ideally the time period would be fitted to the periodicity of the data but weekly segments capture the rhythms of many professional organizations. These segments are then used to create the sequential networks; Table 1 summarizes the weekly statistics for the datasets used in our cross-platform propagation experiments.

2.2 Community Feature Representation

Communities comprise a large portion of social networks; therefore, the networks dynamics are strongly affected by the dynamics of individual communities. Our dataset was partitioned into communities based on topic and location; However,

Platform	Domain	Avg. Users	Avg. Content	Avg. New users	Avg. Edges	# Edge types
GitHub	Crypto	3581	1102	1113	15898	14
	CVE	40034	5500	5505	225423	
	Cyber	297106	95402	95606	1837565	
Reddit	Crypto	7072	37652	37715	37661	2
	CVE	1171	2457	2454	2510	
	Cyber	71883	233424	232584	233464	

Table 1. Weekly statistical data for the datasets

there are many alternative network-based community detection approaches that could be employed (see [9] or [10] for an overview). Examples of topic based communities include a programming language called Scala and Android. Reddit's communities are defined by a subreddit. The term community and subreddit may be used interchangeably when discussing this platform. Each community was represented by a set of features characterizing its dynamics: 1) burstiness, 2) modularity, and 3) number of contributing users. Burstiness is the intermittent increase or decrease in activity, or "bursts" in activity as a function of time; modularity is a measure of community structure that measures the proportion of intra/inter community edges.

After measuring all the features of interest for detected communities, a feature vector $v_{c_i} = [m_1, ..., m_l]$ was created for each. The feature vector of the network associated with each week w_i is represented as $V_{w_i} = [v_{c_1}, v_{c_2}, ..., v_{c_n}]$ in which n and l are the number of communities and features respectively.

These feature vectors are used to compute similarity between community data partitions. In order to predict the network of the target week, we consider the week immediately before our target, calculate its feature vector, and retrieve the community data partitions associated with the closest feature vectors by cosine distance.

Our assumption is that communities exhibit more stable user behavior patterns, thus if the features are similar for one week, the dynamics of the subsequent week may be similar. Figure 1 illustrates the community feature extraction process.

2.3 Node Arrival Analysis

One of the key aspects of modeling network evolution is to accurately forecast growth dynamics. Both of our datasets can be modeled as bipartite networks with user and content nodes. In GitHub, software repositories can be treated as content nodes, whereas in Reddit content nodes are the posts. Nodes in our target network can be divided into categories as follows:

- U_o and C_o: Users/Content who have been active/in attention before the current time step respectively
- U_r and C_r: Users/Content who are active/have been read during the current time step respectively
- U_n and C_n: Users/Content who joined/have been created recently (new users/content)

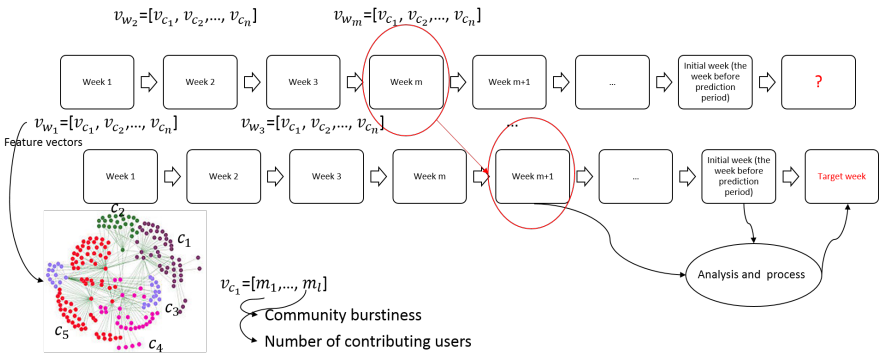

Fig. 1. Community feature extraction

To model node arrival and departure, we utilize these six statistics. Figure 2 shows that there is a huge influx of users that join the network every week in the GitHub crypto dataset, due to the excitement surrounding cryptocurrency. Our preliminary analysis reveals that the cumulative number of all nodes per node type over time is a reasonable choice for estimating the number of new nodes.

Figure 3 shows the best fit curve for the weekly number of new users and recently created content nodes. The number of new user/content nodes in each week w_i can be calculated by $F_{U_n}(w_i) - F_{U_n}(w_{i-1})$ and $F_{C_n}(w_i) - F_{C_n}(w_{i-1})$; this function is estimated for both new user nodes and content nodes in order to forecast next week's network. The total number of nodes per type for the next time step is predicted using time series analysis.

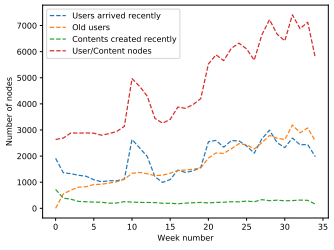

Fig. 2. GitHub/crypto node category distribution by week

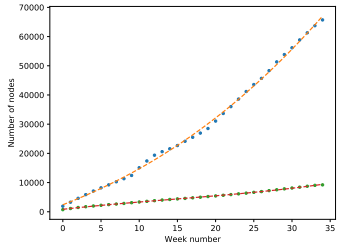

Fig. 3. GitHub/crypto cumulative user/content nodes and model fit

2.4 Modeling Recently Active Users

Even after selecting the closest data partition based on community features and injecting the new content and user nodes, our method often generates fewer nodes in the target week than are required to match the time series prediction. To address this problem, we supplement the generated data with samples of recently active nodes. The Jaccard index is used to calculate the node overlap between sequential weeks:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

This analysis shows that a significant number of user and content nodes are common between two sequential weeks and confirms that augmenting the dataset with active users or recently read content is a reasonable strategy. Users and content are added to the sample with a probability proportional to their activity level. Conversely if the initial estimate is higher than the time series prediction, less active nodes are dropped from the sample in order to create the final set of recent users, U_r, as well as recent content, C_r. At the conclusion of the process, our target set contains a complete set of node categories described in 2.3.

2.5 Edge Creation

After the set of nodes has been finalized, our model still needs to simulate the formation of edges. In the case of existing nodes, we simply replay events from the previous week during the new time period. This captures both the distribution of events and their occurrence time. However, this strategy cannot be applied to simulating the new nodes that have just joined the network. Their corresponding event distribution is different and also they lack historical data samples. To generate these edges, we rely on the distribution statistics of node degrees as well as the edge type. Figure 5 shows the edge type distribution of new users for the GitHub/crypto dataset. We see that most new users commence their usage of the GitHub platform by creating an edge of type *watch* on a repository. To create the number of edges for each node and choose edge targets, preferential attachment is applied as follows:

- New content is:
  - connected to the number of users equal to the average degree of new content nodes
  - connected to the users in a way that is proportional to their activity level; these users might be replaced by a new user as described later.
- A new user is:
  - connected to the number of content nodes equal to the average degree of new users
  - the activity type and time of occurrence is sampled from the history to be proportional to the corresponding distribution

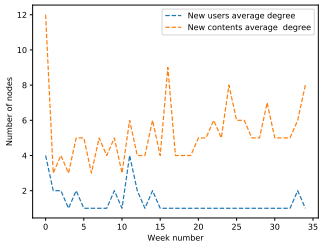

Fig. 4. GitHub/crypto new user/content node average degree

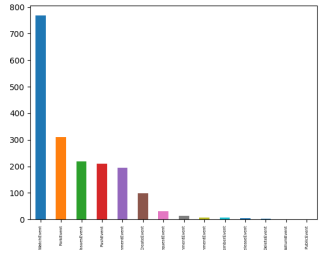

Fig. 5. GitHub/crypto new user event distribution

- a content node is selected from the set of content including both old and new
- content nodes are selected proportionally to the attention they have received; if new content is selected, the new user is replaced with another user to prevent exceeding its average degree.

By the end of this phase we have the set of nodes, edges, edge type, along with the edge formation time. We can then generate the predicted network for our target week using the following algorithm:

Step 1: Create an empty set of nodes and select a time step granularity
Step 2: Partition the data into sequential networks for each time step
Step 3: Detect network communities
Step 4: Generate the weekly community feature vector for each network
Step 5: Select the closest feature vector for the timestep preceding the target
Step 6: Add the nodes of the selected data partition to the node set
Step 7: Create the expected number of new nodes
Step 8: Estimate total number of nodes for the target network using the time series
Step 9: Add the expected number of recently active nodes or drop the low activity nodes as needed
Step 10: Replay selected nodes and community activities with the same time occurrence distribution as in the history
Step 11: Connect new nodes to the others proportional to their estimated degrees and distributions using preferential attachment or edge type considerations

3 Evaluation

A combination of different metrics and measurements were used to compare the simulated output with the ground truth in a hierarchical granularity. The metrics are grouped into categories relating to population, content, user, and network.

3.1 Measurements

Content based measurements primarily focus on information spread. Content is defined as posts and replies in Reddit or comments in Github. They are computed via aggregation of user interaction with content, e.g. with a comment or post. An example of a content based measurement is the number of hourly unique users who comment on a post as a function of time for Reddit and the number of daily unique contributors to a repo as a function of time for Github.

User level measurements are similar measurements to content but focused on user nodes. These measurements are aggregates of user activity. For example, the daily contribution counts of the user over time.

Population measurements are defined by the entire network of either Github or Reddit. They includes the same information spread measurements and aggregate characteristics of events and users on the platform.

Additionally some of the measurements compute **network** attributes including the number of nodes, number of edges, density, average shortest path length, assortativity coefficient, number of connected components, average clustering coefficient, maximum and average node degree, degree distribution, and community structure computed by the Clauset-Newman-Moore modularity measuring how strongly the network resolves into communities or modules.

3.2 Benchmarks

The evaluation process begins with computing the measurements at all resolutions for the community features model (CFM), sampled historical data (SHD) and the ground truth (GT). SHD is our benchmark model that sampled edges from the networks in the recent history before the target time period without considering community features. The sample size is equal to the predicted size. SHD simulates the network under the assumption that many users perform activities similar to their previous patterns. This model was executed 10 times for each week then averaged over all the measurements for each week to provide with one set of measurements. After the measurements are computed, metrics are calculated between CFM and GT which are then compared to the benchmark results of metrics between SHD and GT. Depending on the measurement type, different metrics may be used. Metrics used include: Jensen-Shannon Divergence, KolmogorovSmirnov test, and absolute percentage error.

4 Results

Overall Performance To compare the approaches over all metrics and measurements, we normalized the results by measurement group, metric type and platform. We then averaged them across the measurements and their respective metrics for the different resolutions mentioned in Section 3. This resulted in the metrics ranging from 0 to 1 in which lower means better performance. Figures 6 and 7 show the overall performance of the models on GitHub and Reddit where CFM shows an improvement over the benchmark in all resolutions.

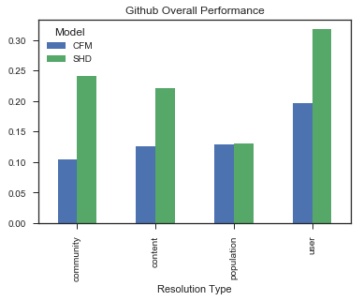

Fig. 6. Overall performance on GitHub

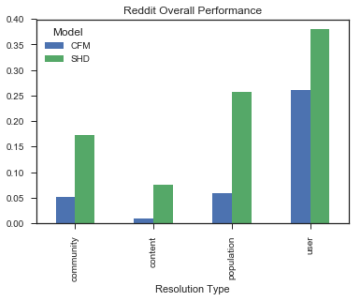

Fig. 7. Overall performance on Reddit

Performance at Different Resolutions We compared CFM to SHD at the population level for Reddit using measurements related to network attributes. Figure 8 visualizes the results of this analysis in which CFM performs better on all based on the absolute percentage error. The community focus of CFM was visible by the significant improvement in the number of connected components representing communities and the assortativity coefficient measuring the degree between pairs of linked nodes.

We compared CFM to SHD on content and user levels for both platforms. To better compare these measurements, we normalized them over the metric type, the platform, and the resolution. Figure 9 and 10 shows that CFM performs better across all measurements at the content level and figure 11 and 12 similarly show this for measurements at the user level.

One caveat is that CFM does not consistently improve performance on other social media platforms that we experimented with, such as Twitter, or on community level measurements; for these measurement groups, SHD and CFM are tied.

5 Conclusion and Future Work

This paper introduces a versatile community-based data mixture model for predicting social network evolution. Community features are used to retrieve the most similar historical data partition; new nodes and edges are added to the data based on a combination of intelligent sampling and preferential attachment. Our proposed model convincingly outperforms our population level data model at forecasting population, content, user, and network metrics on three GitHub and Reddit datasets. In future work, we are exploring the benefits of modeling influence dynamics, exogenous data sources, and cross-platform propagation at the community level rather than at the platform level.

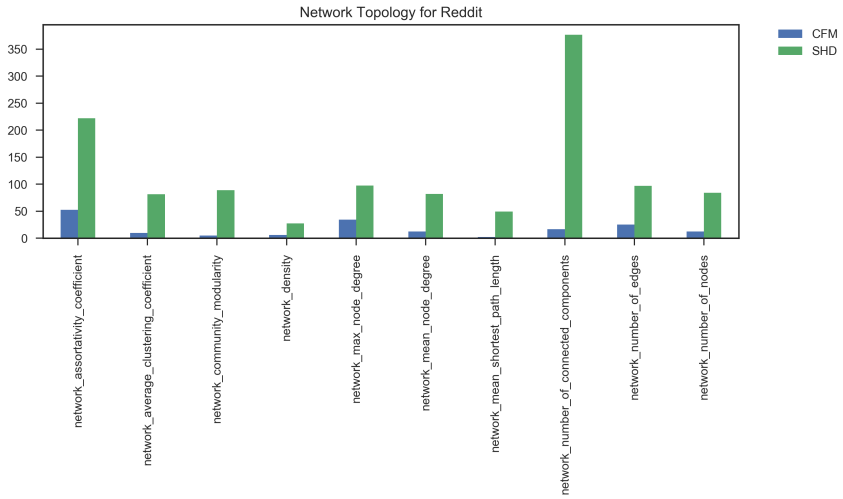

Fig. 8. Network topology for Reddit.

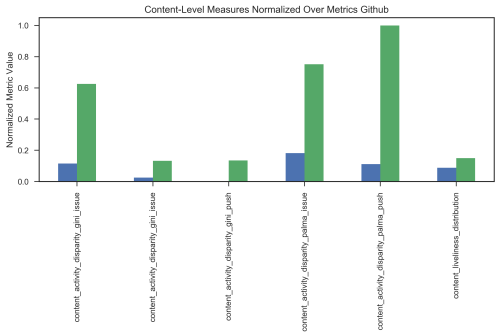

Fig. 9. Content-level Measurements on GitHub

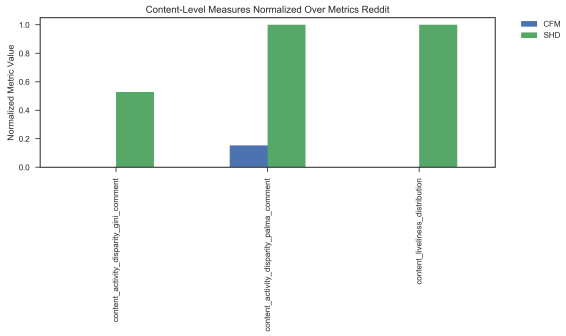

Fig. 10. Content-level Measurements on Reddit

Acknowledgments

This work was partially supported by grant FA8650-18-C-7823 from the Defense Advanced Research Projects Agency (DARPA). The views and opinions contained in this article are the authors and should not be construed as official or as reflecting the views of the University of Central Florida, DARPA, or the U.S. Department of Defense.

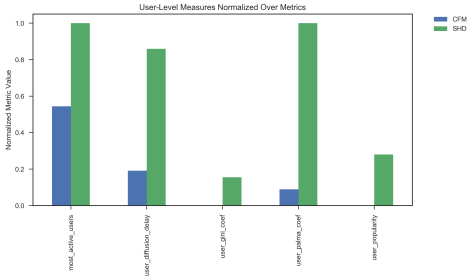

Fig. 11. User-level Measurements on GitHub

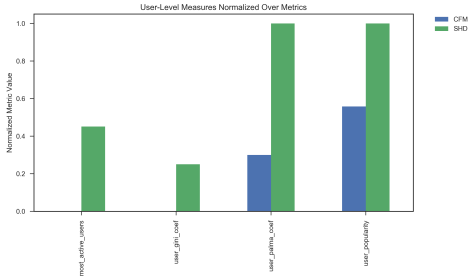

Fig. 12. User-level Measurements on Reddit

References

1. J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2009, pp. 807–816.
2. R. M. Bond, C. J. Fariss, J. J. Jones, A. D. Kramer, C. Marlow, J. E. Settle, and J. H. Fowler, "A 61-million-person experiment in social influence and political mobilization," *Nature*, vol. 489, no. 7415, p. 295, 2012.
3. N. H. Bidoki, G. Sukthankar, H. Keathley, and I. Garibay, "A cross-repository model for predicting popularity in github," *arXiv preprint arXiv:1902.05216*, 2019.
4. J. Qiu, J. Tang, H. Ma, Y. Dong, K. Wang, and J. Tang, "Deepinf: Social influence prediction with deep learning," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 2110–2119.
5. I. Garibay, M. Schiappa, G. C. Anagnostopoulos, C. Bouwens, S. Fiore, H. Keathley, A. Mantzaris, J. T. Murphy, W. Rand, A. Salter, M. Stanfill, G. Sukthankar, N. Baral, A. Berea, C. Gunaratne, N. B. Hajiakhoond, G. Fair, J. Jasser, C. Jayalath, E. Mutlu, O. Newton, A. Rajabi, S. Saadat, S. R. Scheinert, C. Senevirathna, and X. Winter, Rachel Zhang, "Deep agent: Computational social science centered simulation of online information environments," in *Proceedings of International Conference on Computational Social Science*, 2019.
6. C. Gunaratne and I. Garibay, "Evolutionary model discovery of factors for farm selection by the artificial anasazi," *CoRR*, vol. abs/1802.00435, 2018. [Online]. Available: http://arxiv.org/abs/1802.00435
7. 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 Conference on Computer Supported Cooperative Work*. ACM, 2012, pp. 1277–1286.
8. R. Katz and M. L. Tushman, "Communication patterns, project performance and task characteristics: An empirical evaluation and integration in an r&d setting," *Organizational Behavior and Human Performance*, vol. 23, no. 2, pp. 139–162, 1979.
9. J. Kim and J.-G. Lee, "Community detection in multi-layer graphs: A survey," *ACM SIGMOD Record*, vol. 44, no. 3, pp. 37–48, 2015.
10. C. Wang, W. Tang, B. Sun, J. Fang, and Y. Wang, "Review on community detection algorithms in social networks," in *Progress in Informatics and Computing (PIC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 551–555.