2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

A Local Measure of Community Change in Dynamic Graphs

Anita Zakrzewska, Eisha Nathan, James Fairbanks, and David A Bader
School of Computational Science and Engineering
Georgia Institute of Technology, Atlanta, Georgia
Email: {azakrzewska3,enathan3,james.fairbanks,bader}@gatech.edu

***Abstract*—In this work we present a new local, vertex-level measure of community change. Our measure detects vertices that change community membership due to the actions (edges) of a vertex itself and not only due to global community shifts. The local nature of our measure is important for analyzing real graphs because communities may change to a large degree from one snapshot in time to the next. Using both real and synthetic graphs, we compare our measure to an alternative, global approach. Both approaches detect community switching vertices in a synthetic example with little overall community change. However, when communities do not evolve smoothly over time, the global approach flags a very large number of vertices, while our local method does not.**

I. Introduction

Graphs are used to represent relationships between entities, whether in web traffic, financial transactions, computer networks, or society, and often contain dense subsets of highly interacting vertices called communities. Many real-world networks are constantly evolving, requiring a measure applicable for dynamic graphs. In a dynamic graph, nodes may move between communities. Here, we focus on finding vertices that experience a change in their local community behavior, and call these vertices allegiance switching. For example, in a co-authorship network, we want to find researchers who have moved from one lab or department to another or changed their field of publication.

The contribution of this paper is a new local measure of community change. Our measure has the following properties: (1) sensitivity: it detects vertices that have a change in their community and (2) stability: the community change detected is related to the actions (edges) of the vertex and not only caused by global community shifts. We also compare an alternative approach, show that it detects a different set of vertices, and explain why it is insufficient for our goals.

A. Related Work

Popular community detection algorithms include greedy modularity maximization, spectral partitioning, label propagation, and clique percolation [8]. In our experiments, we use the Louvain method of community detection, which iteratively greedily optimizes modularity by making local swaps [5]. However, our method can use any algorithm that labels communities.

As many real world networks are constantly evolving, there is a large body of work in the dynamic community detection area [3]. Some approaches use the entire history of temporal data to identify communities with smooth evolutions over time [14]. However, many applications require an online approach – identifying communities at regular intervals in time without knowledge of future data [7]. This can be done by maximizing community quality as well as minimizing transition cost from the previous community decomposition [6]. However, this smooth transition approach is often computationally expensive and limits the methods available. The result of communities found at the previous timestep can also be used as a starting point for detection at the next timestep. Examples include an incremental version of the Louvain algorithm [4] and incremental spectral clustering [12].

In addition to detecting the communities themselves in dynamic networks, the notion of tracking community behavior over time has been studied. Tracking the overlap between communities from one timestep to another is essential to detect continuing, merging, splitting, and newly forming communities [13]. Tracking vertex behavior with respect to their communities is also a question of interest. Asur *et al.* look at vertices that appear, disappear, join, and leave a community and how this behavior influences the communities they are a part of [2]. Our work focuses on this area, but differs from [2] because it uses a local measure.

II. Allegiance Changing Vertices

A. Definitions

Let $G = (V, E)$ be a graph, where E is the set of edges and V the set of vertices. A dynamic graph changes over time due to activity such as edge deletions, additions, and weight changes, as well as vertices appearing and disappearing. As a graph changes, we can take snapshots of its current state at any time. We denote the snapshot of the dynamic graph G at time t by $G_t = (V_t, E_t)$. In this work we use a sliding window on the stream of edge insertions to create graph snapshots over time, though other approaches could be used. For example, with a window size of 10, $G_{t_{10}}$ would consist of all edges present in the stream from time t_1 to time t_{10}. $G_{t_{15}}$ would then partially overlap with $G_{t_{10}}$ by consisting of all edges from time t_5 to t_{15}. We refer to the community of vertex v at time t by $C_t(v)$.

B. Motivation

In this work we present a local, vertex-level measure of community change. We seek to detect vertices that have

IEEE/ACM ASONAM 2016, August 18-21, 2016, San Francisco, CA, USA
978-1-5090-2846-7/16/$31.00

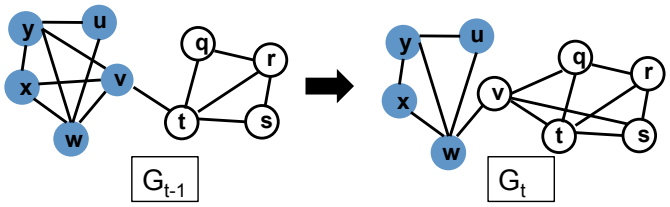


Fig. 1: As the graph changes over time, vertex v has a high local measure of community change. Its neighbor set is $N_t(v) = \{y, x, w, q, s, t\}$. The set of neighbors that leave v's community is $L_t(v) = \{y, x, w\}$. The set of neighbors that join v's community is $J_t(v) = \{q, s, t\}$. The set of neighbors that stay in the same community is $S_t(v) = \emptyset$.

changed their community membership and refer to them as allegiance switching. Recall the coauthorship graph where vertices represent researchers and an edge indicates that two researchers have coauthored a paper within the time period under consideration. In such a graph, we may wish to detect researchers who have moved to a different lab or university or changed the field in which they publish.

The detected community change of a vertex v should be local: caused by a change in v's actions (its edges) and not only a global shift. For example, suppose vertex v has edges only to vertices w and u. If w and u both move to a new community, v will likely move along with them and exhibit a global change in community membership. However, the behavior of v will not have changed (as it is still connected to the same two neighbors) and it will remain in the same community as its neighbors. Therefore, we would not want to mark v as allegiance changing.

C. Proposed Method

Given a dynamic graph, at each timepoint t, we create a snapshot G_t, and detect communities C_t. Using this community decomposition we define three sets for each vertex $v \in V_t$. $S_t(v)$ is the set of neighbors of v that were in the same community as v at the previous snapshot G_{t-1} and stay in the same community as v in G_t. $L_t(v)$ is the set of neighbors of v that were in the same community in G_{t-1}, but are no longer in G_t. $J_t(v)$ is the set of neighbors of v that were not in the same community as v in G_{t-1}, but are in G_t. Formally these are defined in Equations 1, 2, and 3 respectively.

We define the set of neighboring vertices of v at time t by $N_t(v)$ as all vertices w that have an edge to v in either G_t or G_{t-1}. Note that $N_t(v)$ contains vertices adjacent to v at the current and previous snapshot, instead of just the current snapshot, because the set of neighbors of v may change significantly.

$$S_t(v) = \{w | w \in N_t(v) \wedge C_{t-1}(v) = C_{t-1}(w) \wedge C_t(v) = C_t(w)\} \quad (1)$$

$$L_t(v) = \{w | w \in N_t(v) \wedge C_{t-1}(v) = C_{t-1}(w) \wedge C_t(v) \neq C_t(w)\} \quad (2)$$

$$J_t(v) = \{w | w \in N_t(v) \wedge C_{t-1}(v) \neq C_{t-1}(w) \wedge C_t(v) = C_t(w)\} \quad (3)$$

A large change in v's neighborhood at time t occurs when (1) the set $L_t(v)$ is large, indicating that many vertices with which v interacted and which were in the same community as v are no longer in the same community as v, (2) the set $J_t(v)$ is large, indicating that v is now connected to vertices which were not previously in its community, and (3) the set $S_t(v)$ is relatively small, so that a low percentage of v's neighborhood was and continues to be in the same community.

Using $S_t(v)$, $L_t(v)$, and $J_t(v)$, we define $\alpha_t(v)$ and $\beta_t(v)$ below.

$$\alpha_t(v) = \frac{|L_t(v)|}{|S_t(v) \cup L_t(v)|} \quad (4)$$

$$\beta_t(v) = \frac{|J_t(v)|}{|S_t(v) \cup J_t(v)|} \quad (5)$$

Figure 1 shows an example. Intuitively, a high value of $\alpha_t(v)$ indicates that a large percentage of neighbors of v have left v's community (or v has left theirs). Similarly, a high value of $\beta_t(v)$ means that a large percentage of v's neighbors were not in the same community as v at time $t-1$, but joined v's community at time t (or v joined their community). Therefore, vertices v with high values $\alpha_t(v)$ and $\beta_t(v)$ have experienced a large change in their local neighborhood.

The focus is on the community membership of a vertex v relative to its neighbor set $N_t(v)$ because the neighbors represent other entities v has directly interacted with. The one hop neighborhood of a vertex v represents all other entities with which v has interacted. With a larger number of hops, any relationship becomes much weaker and more difficult to characterize. v may be in the same community as many vertices with which it does not interact or which are not even in a two hop neighborhood. Therefore, our measure of community change is local.

D. Setting a Threshold

In order to mark a vertex as allegiance switching, it needs to have both high values of $\alpha_t(v)$ and $\beta_t(v)$. In this section we discuss how to combine the two scores and set a threshold.

To combine the two values, both of which fall between 0 and 1, we can draw from work from the field of fuzzy logic, where the logical conjunction of two values ("x and y") is represented by taking the minimum ($min(x, y)$), using Gödel's t-norm, or by taking the product ($x * y$), using the Product t-norm [10] [11]. In section III, we use both the minimum and the product of $\alpha_t(v)$ and $\beta_t(v)$ to demonstrate two different approaches of combining the two values, although other methods could be used as well.

As in any extreme value problem, there can be various ways of setting a threshold, depending on the application. In section III, we use a fixed threshold for $min(x, y)$ and the top 1 score percentile for $x * y$. A fixed threshold is easily interpreted and can be set by the end user. For example, for a given application, it may be determined that both $\alpha_t(v)$ and

$\beta_t(v)$ need to be above 0.8. On the other hand, choosing only vertices with a score greater than or equal to a top percentile score allows the threshold to be adjusted to the distribution of a particular dataset.

E. Alternative Approaches

In this section we consider two alternative approaches to detecting allegiance changing vertices and discuss why our metric better captures the desired behavior. The first is a global approach, which matches communities in consecutive snapshots. For each pair of consecutive graph snapshots G_{t-1} and G_t, we compute communities using the Louvain algorithm [5]. Two communities, $C_j \in G_{t-t}$ and $C_i \in G_t$, are matched as equal if the Jaccard index $|C_j \cap C_i|/|C_j \cup C_i| \geq 0.75$, as a split if $|C_j \cap C_i|/|C_i|) \geq 0.75$, and as a merge if $|C_j \cap C_i|/|C_j|) \geq 0.75$. A vertex is marked as allegiance switching if $C_{t-1}(v)$ and $C_t(v)$ are not matched as the equal, merging, or splitting. In section III, we compare the vertices marked with our approach to those marked with this global approach, which is similar to [2] [9].

However, using such a global approach has several disadvantages. A vertex may be marked as allegiance changing due to purely global changes. A re-arrangement of community composition may prevent a high enough overlap between $C_{t-1}(v)$ and $C_t(v)$ to allow the clusters to match, and v would then be marked as allegiance changing even if its own behavior has not changed. In a dataset with a very smooth community evolution, a global approach may work well; in practice however, a graph may greatly change between two snapshots, causing the communities to change drastically as well. In section III we show that this global approach identifies a very large percent of vertices as community switching because communities change a lot between consecutive snapshots. The success of the global approach may be increased by using a community detection algorithm that tries to detect communities with smooth transitions by taking into account historical data [6] [14]; however, such algorithms can be more computationally expensive.

Another alternative approach is to compare the neighbors of a vertex v at time t to those at time $t-1$ and count how many have changed. However, doing so would not capture community allegiance changing behavior. For example, consider the co-authorship network example mentioned in Section II-B. From year to year, a researcher may change who he co-authors papers with without changing his field or even his work group. In the time represented by G_{t-1} he may co-author with colleagues w and u and during the time represented by G_t co-author with colleagues y and z. In this case, the set of neighboring vertices would change completely with no overlap. However, if w, u, y, and z all belong to the same lab or department and work together, then we would not consider v to have changed behavior because he would be working within the same group. Therefore, simply looking at the change in vertex's neighbors will not capture the behavior we would like to find and we must consider behavior relative to the community.

III. Experiments

A. Synthetic Graphs

In this section we compare our local approach with the global alternative described in section II-E using synthetic graphs built with a stochastic block model. In an assortative stochastic block model, vertices within the same community are connected by an edge with probability p, while vertices in different communities are connected with probability q, where $p > q$. We generate a graph using 2 communities, 1000 vertices, $p = 0.3$, and $q = 0.01$. Then we randomly choose 5 vertices and change their community assignment. Both the global approach and our local measure are able to identify all of the vertices that moved between communities. After 20 such runs, $\alpha_t(v) * \beta_t(v)$ values of the selected vertices ranged from 0.93 to 1, while their $min(\alpha_t(v), \beta_t(v))$ values ranged from 0.96 to 1. In this example, vertices cleanly move from one community to another and both the global and local methods can detect the change.

However, in graphs from real data, communities do not always persist over time, but may may break apart and rearrange. We simulate such change using a hierarchical block model. Each community is composed of multiple, more tightly connected, sub-communities. Vertices within the same sub-community are connected with probability p_1, vertices within different sub-communities, but the same community, have an edge with probability p_2, and all others are connected with probability q, where $p_1 > p_2 > q$. Community change is then created by rearranging the sub-communities into different communities, as shown in Figure 2. With 2 sub-communities per community, the overlap between communities of the two snapshots is 50% using the Jaccard index. Therefore, no community will be matched and the global method will mark every vertex as community changing, despite the fact that the sub-communities remain the same. The local metric scores $\alpha_t(v)$ and $\beta_t(v)$, however, remain low for all vertices. For example, after 20 runs using 2 communities, 2 sub-groups per community, $p_1 = 0.6$, $p_2 = 0.15$, and $q = 0.01$, the largest value of $min(\alpha_t(v), \beta_t(v))$ of any vertex in any run is 0.21 and the largest value of $\alpha_t(v) * \beta_t(v)$ is 0.04. In this synthetic example, while the communities are globally unstable from one snapshot to the next, large portions of the communities remain the same. The global approach only detects this global change, while the local measure detects that each vertex experienced little change in its own community behavior.

B. Real Datasets

We evaluate our measure on three dynamic graphs from the KONECT collection [1], listed in Table I. We create snapshots G_t with the window size and overlap listed. A window size of 6 months with 4 month overlap means that each snapshot contains edges from a 6 month period and consecutive snapshots overlap by 4 months. In the experiments below, we find communities using the popular Louvain algorithm [5], although any algorithm can be used, as our approach is agnostic to it.

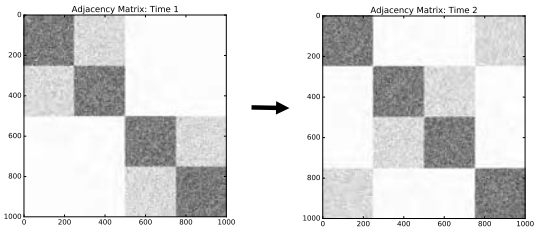


Fig. 2: The rearrangement of communities in a hierarchical stochastic block model is shown.

Graph	Vertices	Edges	Snapshot Window	Window Overlap
DBLP	1,314,050	18,986,618	6 years	4 years
YouTube	3,223,589	9,375,374	3 months	2 months
Facebook	46,952	876,993	6 months	4 months

TABLE I: A description of the graphs and sliding window used to build snapshots is shown.

By running the Louvain algorithm on each snapshot and computing the overlap of communities in consecutive snapshots, as described in section II-E, we can see how smoothly the detected communities change. Table II shows the average percentage of communities with size greater than one that are matched with a community in the following snapshot, either as continuing, merging, or splitting. While a large number of communities are matched, many are not. This does not mean that all non-matched communities necessarily dissolve completely. Rearranged communities, such as in the stochastic block model example, will also fail to match. Communities were matched using only vertices with non-zero degree in both consecutive snapshots. Using all vertices produced even fewer matches. This suggests that relying on a global matching of communities is unreliable. The smoothness of community evolution will of course depend on both the algorithm and sliding window used. However, our results suggest that we cannot rely on the dataset in question to have smooth transitions between communities of different snapshots, making the global metric sensitive to many factors. Our measure does not require stable clusters between snapshots because it only considers the community of a vertex relative to that of its neighbors. Clusters can be locally stable, without being globally stable.

Figure 3 shows distributions of the scores for $\alpha_t(v)$ and $\beta_t(v)$ and the corresponding histograms of $\alpha_t(v) * \beta_t(v)$ and $min(\alpha_t(v), \beta_t(v))$ for one snapshot of the DBLP graph. While we do see evidence of outliers using just the raw $\alpha_t(v)$ and

Graph	% Continuing	% Splitting	% Merging
DBLP	63.3	2.0	18.7
YouTube	54.0	5.8	17.6
Facebook	51.5	6.4	28.8

TABLE II: The average percentage of communities matched between snapshots is shown.

Graph	Global %	$min(\alpha_t(v), \beta_t(v))$ %	$\alpha_t(v) * \beta_t(v)$ %
DBLP	64.6	2.5	2.5
YouTube	51.4	4.1	4.0
Facebook	77.5	6.3	6.2

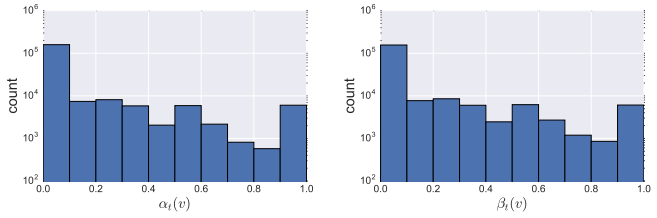
TABLE III: The average percentage of vertices flagged is shown for all three datasets.

Overlap	Global %	$min(\alpha_t(v), \beta_t(v))$ %	$\alpha_t(v) * \beta_t(v)$ %
0 years	77.0	19.5	18.7
2 years	67.2	7.0	6.7
4 years	64.6	2.5	2.5
5.5 years	56.8	0.9	0.9

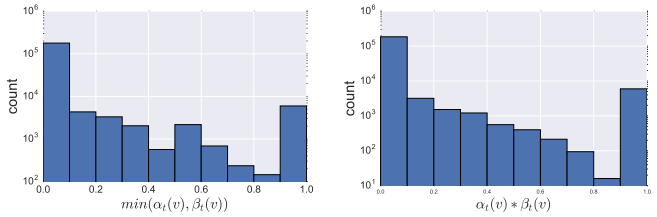
TABLE IV: The average percentage of vertices flagged using varying overlaps of the DBLP dataset is shown. Each snapshot contains data from 6 years and consecutive snapshots overlap by the amount shown.

$\beta_t(v)$ scores in figures 3a and 3b, the outliers are much clearer by looking at the combined metrics in Figures 3c and 3d. Figure 3d especially shows a clear trend of a decreasing number of vertices as the score increases, and then a spike in outlying vertices with a very high score.

Table III shows the average (over all snapshots) percentages of vertices flagged by our local measure using both the minimum and product to combine $\alpha_t(v)$ and $\beta_t(v)$. Minimum has a fixed threshold of 0.8 and product uses the top 1% score. The percentage flagged by the alternative global metric is also shown. For all datasets, we see that the global metric flags many more vertices than either local metric. We also found that almost all vertices flagged by the local method were also flagged by the global approach. Therefore, for analysts wishing to tag allegiance changing vertices, the local metrics provide a much smaller set of candidates than their global counterpart that are likely to be the vertices of interest. The



(a) Histogram of the $\alpha_t(v)$ scores. (b) Histogram of the $\beta_t(v)$ scores.



(c) Histogram of the $min(\alpha_t(v), \beta_t(v))$ scores. (d) Histogram of the $\alpha_t(v) * \beta_t(v)$ scores.

Fig. 3: Histograms of individual and combined metrics for one snapshot of the DBLP graph.

global approach identifies such a large number of vertices because many communities not are matched between consecutive snapshots G_{t-1} and G_t. Because the communities do not evolve smoothly, for a very large percentage of vertices v, $C_{t-1}(v)$ and $C_t(v)$ do not match, causing v to be marked as switching.

Table IV shows the percentage of vertices flagged as allegiance changing using both the thresholding approaches of $min(\alpha_t(v), \beta_t(v)$ and $\alpha_t(v) * \beta_t(v)$ for various overlap intervals for the DBLP dataset. We see that as we increase how much consecutive snapshots overlap, the percentage of vertices flagged as allegiance changing decreases. This is expected because as the overlap between two consecutive snapshots increases, fewer community changes occur.

IV. Conclusion

In this paper we have presented a local metric for identifying vertex-level community changes. Using synthetic graphs, we find that when communities change very little, both the global and our local measures correctly detect vertices that switch communities. However, when the communities of a graph change between snapshots, the global method is unreliable, flagging a majority of vertices, while our local method does not. On graphs from real social networks, we find that the local approach presented flags far fewer vertices of interest compared to the global alternative. While there is an increasing interest in dynamic networks and dynamic community detection, finding community allegiance changing vertices is a relatively new area. Measuring vertex level community-oriented changes pinpoints vertices with interesting behavior, allowing for further investigation. Our results suggest that the global approach is flawed when communities do not evolve smoothly between snapshots. Further work will investigate this phenomenon in more detail and include more experimental validation of our measure.

Acknowledgments

This work was partially sponsored by Defense Advanced Research Projects Agency (DARPA) under agreement #HR0011-13-2-0001 (DARPA PERFECT). The content, views and conclusions presented in this document do not necessarily reflect the position or the policy of DARPA or the U.S. Government, no official endorsement should be inferred.

References

[1] The koblenz network collection KONECT, April 2016.

[2] Sitaram Asur, Srinivasan Parthasarathy, and Duygu Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(4):16, 2009.

[3] Thomas Aynaud, Eric Fleury, Jean-Loup Guillaume, and Qinna Wang. Communities in evolving networks: Definitions, detection, and analysis techniques. In *Dynamics On and Of Complex Networks, Volume 2*, pages 159–200. Springer, 2013.

[4] Thomas Aynaud and Jean-Loup Guillaume. Static community detection algorithms for evolving networks. In *WiOpt'10: Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pages 508–514, 2010.

[5] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[6] Deepayan Chakrabarti, Ravi Kumar, and Andrew Tomkins. Evolutionary clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 554–560. ACM, 2006.

[7] David Ediger, Jason Riedy, David A Bader, and Henning Meyerhenke. Tracking structure of streaming social networks. In *5th Workshop on Multithreaded Architectures and Applications (MTAAP)*, May 2011.

[8] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[9] Derek Greene, Donal Doyle, and Padraig Cunningham. Tracking the evolution of communities in dynamic social networks. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 176–183. IEEE, 2010.

[10] Petr Hájek. Basic fuzzy logic and bl-algebras. *Soft computing*, 2(3):124–128, 1998.

[11] Petr Hájek, Lluís Godo, and Francesc Esteva. A complete many-valued logic with product-conjunction. *Archive for mathematical logic*, 35(3):191–208, 1996.

[12] Huazhong Ning, Wei Xu, Yun Chi, Yihong Gong, and Thomas S Huang. Incremental spectral clustering by efficiently updating the eigen-system. *Pattern Recognition*, 43(1):113–127, 2010.

[13] Myra Spiliopoulou. Evolution in social networks: A survey. In *Social Network Data Analytics*, pages 149–175. Springer, 2011.

[14] Chayant Tantipathananandh, Tanya Berger-Wolf, and David Kempe. A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 717–726. ACM, 2007.