Network pruning for scaling dynamic community detection

Gaurav Ghosh
University at Albany, State University of New York, gaurav.ghosh05@gmail.com

The University at Albany community has made this article openly available. Please share how this access benefits you.

Follow this and additional works at: https://scholarsarchive.library.albany.edu/legacy-etd

Part of the Computer Sciences Commons

Recommended Citation
https://scholarsarchive.library.albany.edu/legacy-etd/1616

This Master's Thesis is brought to you for free and open access by the The Graduate School at Scholars Archive. It has been accepted for inclusion in Legacy Theses & Dissertations (2009 - 2024) by an authorized administrator of Scholars Archive. Please see Terms of Use. For more information, please contact scholarsarchive@albany.edu.
Local community detection is an important tool for the analysis of networks of different genres. The goal is to identify only the best communities in a network instance as opposed to computing a partitioning of the whole network. The majority of the work on local community detection has focused on static networks with less attention on networks that evolve over time. Given a trace of temporal interaction among nodes in a network, how can we detect a period of high interaction for a specific group of nodes? To help temporal community detection with the need to search in the time domain in addition to the graph structure a time period based pruning solution is presented.

By using spectral graph theory we calculate bounds which can tell us about the quality of the community within certain time periods. To get an idea of what a good community would look like in a given data set, a quick solution is calculated based on thresholding the top percentage of active participants. This solution is then used to compare with the bounds we have calculated and then the potentially more interesting time periods can be highlighted. An interval based bound calculation is then implemented which further improves scalability of the solution.
LIST OF FIGURES

3.1 The structure of the consecutive time based intervals . . . . . . . . . . . 9
4.1 Bounds of Varying Length And Threshold Solution . . . . . . . . . 11
4.2 The Percentage Pruned for Every Time Period . . . . . . . . . . . . 12
4.3 Pruning Power with Different Burst Strength . . . . . . . . . . . . . . 12
4.4 Performance with All Bounds and Composite Bounds . . . . . . . . . 13
CHAPTER 1

Introduction

Given a large network in which entities communicate at different instances of time, how can one detect communities whose interactions intensify in a given period? Our social lives are comprised of a mixture of diverse communities whose in-community interactions intensify at different points in time. For example, email interactions among co-authors increase in periods before a paper deadline [1], while communication among family members increases before family reunions. Vocal fans of sporting teams will be much more involved on social media just before crucial matches. Detecting and tracking such bursty interaction behavior among a subset of people is essential in decoupling the various local communities in which one is embedded and for predicting participants of future bursts by correlating their activation with a given temporal context (i.e. conference deadlines, Thanksgiving, etc.). This allows us to find out about how people can be better informed about major happening events, what kind of people are the ideal market for such events and who are not. There are also direct anti-terrorism use cases where patterns of high activity between specific suspects can be analyzed and used for prediction and prevention of their plans.

Beyond social networks, bursty localized interactions are exhibited in brain networks during cognitive tasks such as motion planning, in which coordination among visual and sensori-motor brain regions become more prominent. Tracking such temporal coordinated bursts using fMRI data can elucidate brain region “communities” involved in complex cognitive tasks and identify anomalies due to neurological diseases.

In this paper we consider a generalization of local community detection for the setting of dynamic networks. While there are many community detection algorithms for looking at networks in single time values, these struggle with obvious problem of working over large time periods, where a naive implementation would require them to have to look at all possible time bounds for detection of interesting communities
that might span arbitrary time lengths.

By focusing on finding out where the possible bursts of interactions are in the overall graph, a pruning-based approach is used which is designed to work with a community detection algorithm and scale up its possible effectiveness on much larger data sets than that would be possible otherwise. Performance is further improved by using a sub-interval based bounding process that significantly improves scaling power at a relatively small cost of quality. For more complex networks community detection algorithms and dense graph mining methods can be used for further results [2, 3, 4].
CHAPTER 2
Problem Definition

Within a network there exists potentially more interesting groups of nodes that have the qualities that we are looking for. We measure this quality with conductance.

Given a weighted graph $G(V,E,W)$, let the $S \in V$ be a subset of the nodes. Then the conductance of the subset is defined as

$$\phi(S) = \frac{\delta(S)}{\min(\text{vol}(S), \text{vol}(\bar{S}))},$$

where $\delta(S) = \sum_{(i,j) \in E, i \in S, j \in \bar{S}} w_{i,j}$ is the volume of the cut and $\text{vol}(S) = \sum_{i \in S} \sum_{j \in N(i)} w_{i,j}$ is the volume of the nodes in $S$. The conductance of the graph $\phi(G)$ is the minimum possible conductance over all sets $S \in V$:

$$\phi(G) = \min_{S \in V} \phi(S).$$

Interval conductance is not quite sufficient for our purposes, however, since a very short interval around a small set of interactions can produce very low conductance values. Since communities that last longer are generally of greater interest anyway, our measure of what constitutes a ”good” temporal community must incorporate the length of the window in which the community exists.

Given the above, we define the temporal conductance of a temporal community, $\phi_t^{(k)}(C) = \frac{\phi_t(C)}{(t_2-t_1)^k}$. The parameter $k$ allows the impact of interval length to be tuned; we found a value of $k = 1/3$ worked well in our experiments, and that value is used throughout this paper—as such, we will simply write $\phi_t(C)$.

The problem we seek to solve, then, is this: Given a temporal network $G(V,E)$, find the time periods $(t_1, t_2), (t_3, t_4) \ldots (t_n, t_m)$ which have communities with low conductance.

The problem of finding the subset of lowest conductance on a non-temporal
graph was proven to be NP-hard by Šíma and Schaeffer[5]. A temporal graph in which all timestamps are the same is equivalent to a non-temporal graph; thus, by reduction, our problem is also NP-hard.
CHAPTER 3

Bounds

To find out where there might exist low conductance subgraphs, the conductance of the graph can be bounded using the spectral properties of the graph. Let $A$ be the adjacency matrix with elements $A_{i,j} = w_{i,j}$ containing the weights of edges and $D$ be the diagonal volume matrix with elements $D_{i,i} = \sum_{j \in N(i)} w_{i,j}$ and 0 in all off-diagonal elements. The matrix $L = D - A$ is called the unnormalized graph Laplacian, while the matrix $N = D^{-1/2}LD^{-1/2}$ is called the normalized Laplacian of the graph [6].

The elements of the normalized graph Laplacian $N$ are as follows:

$$N_{i,j} = \begin{cases} 
1 & \text{if } i = j, \text{vol}(i) > 0, \\
-\frac{w_{i,j}}{\sqrt{\text{vol}(i)\text{vol}(j)}} & \text{if } i \neq j, (i,j) \in E, \\
0 & \text{otherwise}.
\end{cases}$$

Let $0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{|V|}$ be the eigenvalues of $N$. Then, based on Cheeger’s inequality [6] one can show that $\phi(G)$ can be bounded as follows:

$$\frac{\lambda_2}{2} \leq \phi(G) \leq \sqrt{2\lambda_2}$$

The lower bound can be used to discard time periods where we believe that bursty communities do not exist. But to do so we have to come up with what a bursty community looks like in the graph.
3.1 Thresholding

For getting a value to use for pruning away time periods, it is more important that we can get it fast than that we get the subgraph with the lowest possible conductance in the whole temporal graph. If it takes us too long to get this solution, we might as well have run a more complex community detection algorithm in the first place. Even sacrificing some quality, we should be able to prune significant time periods.

Algorithm 1 Fast Thresholding

Input: A temporal network $G(V,E,C)$, parameters $k$

Output: conductance value

begin
    
    \textbf{Edge Rank Pruning}
    
    for every time period $t_n \in \{V \times T\}$ do
    
    Sort and rank all edges at this time period based on weight.
    
    Keep top $k$ percentage edges and prune out the rest.
    
    end
    
    end

    \textbf{Connect and Calculate}
    
    for every time period $t_n \in \{V \times T\}$ do
    
    for every node in thresholded network do
    
    While common nodes exist in time period, traverse to next time period
    
    Every time a node is added, calculate conductance in the original network.
    
    end
    
    end
    
    end

return Lowest recorded conductance

By calculating the top $k$ percentage of active edges at each time period, we can form a much smaller temporal graph consisting of only these top edges. Then as we traverse the graph we can calculate conductance for all possible time values and nodes which are connected across time periods, storing the smallest possible value that we find.
3.2 Composite Bounds

Calculating these conductance bounds can be expensive and would scale poorly when computing all possible time periods. For example, for a graph with total 100 time periods, there are 5050 total possible bounds that are need to be calculated. Thus to reduce the required number of these expensive matrix calculations, re-using already computed eigenvalues would be very useful. Specifically, we are looking to find a bound for a period of time by dividing it into smaller intervals of pre-calculated values.

To derive a bound on the conductance of aggregated graphs for all super-intervals of a given interval:

**Theorem 3.1** Let $\{[l_1, r_1], [l_2, r_2]...[l_k, r_k]\}$ be $k$ consecutive non-overlapping time intervals, such that $r_i = l_{i+1} - 1, \forall i \in [1, k)$ with corresponding aggregated normalized graph Laplacians $\{N_1, N_2...N_k\}$ and degree matrices $\{D_1, D_2...D_k\}$. Then,

$$\lambda_2(\hat{N}) \geq \sum_{i=1}^{k} \min_j \frac{D_{ij}}{D_{jj}} \lambda_2(N_i), \quad (3.1)$$

where $\lambda_2()$ is the second smallest eigenvalue of the corresponding Laplacian, $\hat{N}$ and $\hat{D}$ are the Laplacian and degree matrices of the aggregated graph in the whole span of intervals $[l_1, r_k]$ and $tr()$ denotes the trace a matrix.

**Proof:**

According to the Rayleigh-Ritz theorem the second eigenvalue of the Laplacian can be characterized as the minimum of its Rayleigh quotient:

$$\lambda_2(\hat{N}) = \min_{g \perp \hat{d}^{1/2}} \frac{\hat{g}' \hat{N} g}{\hat{g}' \hat{g}},$$

where $\hat{d}$ is a vector node volumes in $[l_1, r_k]$. 
\[ \lambda_2(\tilde{N}) = \min_{g \perp \hat{d}/2} \frac{g'\tilde{N}g}{g'g} \]

\[ = \min_{f \perp \hat{d}} \frac{f'\hat{L}f}{f'\hat{D}f} \]

\[ = \min_{f \perp \hat{d}} \sum_{i=1}^{k} \frac{f'L_if}{f'D_if} \]

\[ = \min_{f \perp \hat{d}} \sum_{i=1}^{k} \frac{f'D_if}{f'\hat{D}f} \lambda_2(N_i) \]

\[ \geq \min_{f \perp \hat{d}} \sum_{i=1}^{k} \frac{g'(D_i\hat{D}^{-1/2})g\lambda_2(N_i)}{\sum_{j=1}^{k} \min_{j} \frac{D_{ij}}{D_{jj}}} \lambda_2(N_i) \]

We have first changed variables \( g = \hat{D}^{1/2}f \) and used the fact that the combinatorial Laplacian can be expressed as a sum of Laplacians when considering aggregations of weighted graphs. The fifth equality follows from the inequality from the Rayleigh-Ritz theorem for the Rayleigh quotients of the individual intervals \( \lambda_2(N_i) = \min_{f \perp d_i} \frac{f'L_if}{f'D_if} \).

The bound in Theorem 3.1 is tight. To see this, consider the case of equally weighted subgraphs in the consecutive intervals. In this case the spectra of the normalized Laplacians in the subintervals as well as in the aggregated interval will be identical.

While we can use any subinterval structure to store our pre-computed intervals, there is a trade-off of performance and quality of result. If we only store values for small time periods, the number of calculations will be less but due to increased number of subintervals for larger time periods, accuracy of bounds will be lower. A reasonable middle ground can be reached with a 2 power based binary structure. Here, we calculate all consecutive bounds of size 1, 2, 4, 8 and so on till the graph size.
Figure 3.1: The structure of the consecutive time based intervals

As per Fig 3.1, if we want to compute bounds for the time period $\hat{\mathit{N}} t_3$ to $t_5$, we would need to use subinterval of size 1 at $t_3$ and size 2 at $t_4 - t_5$. As larger and more numerous time periods are processed, this allows a very significant performance increase as we do not have to calculate all possible bounds.
CHAPTER 4

Experiments

To test the algorithm, synthetic graphs are created with the required data. We began by creating a scale-free network (average degree = 20) using the well-known preferential attachment approach of Barabási and Albert[7]. We then added a set of burst nodes $B$, added edges internal to the burst to guarantee connectivity and a certain density, and then added a number of edges from the burst to the rest of the graph to provide a desired underlying conductance value, $\phi_B$. By default we used an overall network size of 1000 nodes, 20 of which were burst nodes. The burst typically had an underlying density of 0.6 and an underlying conductance of 0.4; variations from these values are noted in the results for that experiment.

Once the underlying graph was built, we selected an overall length of the timeline and start and stop times for the injected burst. We then added timestamps to each edge using an exponential distribution, truncating fractions. The mean time between interactions on an edge was controlled by one of three parameters: $\tau_i$, for interactions on internal burst edges during the burst interval; $\tau_c$, for interactions on cut edges (from burst nodes to non-burst nodes) during the burst interval; and $\tau_n$, for all other edge/time combinations. Unless otherwise indicated, we used values of $\tau_i = 0.5$, $\tau_c = 6.0$, and $\tau_n = 2.0$ in our synthetic graphs. All bursts have been inserted between t20 and t30 in all synthetic graphs.
4.1 Results

First, as per Fig 4.1, it is observed how the bounds of the intervals of different length react to the burst, which is between 20-30, the dip in the bounds starts as it approaches the burst. The bounds of the time periods that are significantly bigger than the burst itself are too polluted by the background to realize the burst. The static line represents the conductance value found by thresholding, the bounds that are above it are going to be successfully pruned away.

In Fig 4.2, we see the total percentage pruned for each time period using the actual burst conductance and the conductance found through thresholding. As the threshold is not as low as the burst, it prunes less effectively. After a point when the time periods become too big, both start pruning away all remaining sizes as they have become too long to react to the size 10 burst. This confirms that our method is able to pick up the provided bursts for the relevant time periods.

This is the total percentage pruned in all time periods in Fig 4.3. We see the effect of changing the burst’s internal connectivity, making it harder or weaker to find among the background. Also added are the results of using composite bounds as well as computing all possible bounds. It is observed that composite bounds do a fairly
good job of pruning except in the case of the weak internal burst. The main reason for this is the relatively higher conductance solution found with thresholding combined with the less tight bounds of the composite method leading to very ineffective pruning.

Finally we see the significant performance benefits of the composite bounds method in Fig 4.4. As total time of the graph increases computing all the combinations of time become exponentially more expensive in comparison the logarithmic increase of composite calculations.
Figure 4.4: Performance with All Bounds and Composite Bounds
CHAPTER 5
Conclusion and Future Work

By taking advantage of the useful properties of normalized Laplacian matrices we are able to get a good idea of what kind of communities exists in various time periods. Then, we come up with a very fast solution of a subgraph with low conductance which we are able to use as a cut off point for the bounds. Moreover, we are able to significantly improve performance of graphs with larger time periods by using the interval based composite bounds. With the given results we have been able to prove the effectiveness of our methods on synthetic data. Moving on, implementation with multi-burst and real world data is going to be the priority. By integrating with a more powerful community detection algorithm that can take advantage of the performance benefits of these methods will also be very useful.
BIBLIOGRAPHY


