Community Cores: Removing Size Bias From Community Detection

Isaac Jones[1]**, Ran Wang**[1]**, Jiawei Han**[2]**, and Huan Liu**[1]
[1]Arizona State University [2]University of Illinois at Urbana-Champaign
[1]{ipjones, rwang32, huan.liu}@asu.edu [2]hanj@illinois.edu

Abstract

Community discovery in social networks has received a significant amount of attention in the social media research community. The techniques developed by the community have become quite adept at identifying the large communities in a network, but often neglect smaller communities. Evaluation techniques also show this bias, as the resolution limit problem in modularity indicates. Small communities, however, account for a higher proportion of a social network's community membership and reveal important information about the members of these communities. In this work, we introduce a re-weighting method to improve both the overall performance of community detection algorithms and performance on small community detection.

Introduction

In real social networks, community sizes are widely distributed. For example, in Facebook[1], there are huge university communities with tens of thousands of people including students, professors, alumni, etc. However, these networks also contain small communities of club sports, research groups, and classes, among others, that are much smaller. Prior work has shown that community sizes are distributed according to a power law in social networks (Tang and Liu 2010). This means that networks will always have far more small communities than big communities.

In addition to this superset and subset relationships within communities, different communities that share similar interests may also have more connections to each other. For example, members of two karate clubs that come from different universities may become friends. Because of the tight connection between the two communities, the number of inter-community edges increases, and current algorithms may merge the two communities together.

These phenomena can combine to cause useful communities to be completely "hidden" in larger communities, which makes community detection extremely difficult. However, in real world applications, we are usually more interested in the small groups. The difficulty of detecting these groups make it an area of interest for many organizations.

In this paper, we introduce a method that can discover the small, denser communities "hidden" inside the larger communities. Instead of developing an entirely new method, we introduce a technique for reweighting the network.

Background and Problems

As discussed, small communities can easily hide in large communities. In our investigation of this phenomenon, we noticed that a large class of existing algorithms depend heavily on edge information: Infomap (Rosvall and Bergstrom 2008) relies on the fact that a random walker tends stay inside communities but will eventually leave and SLPA (Xie, Szymanski, and Liu 2011) propagates labels from one node to its adjacent neighbors by the same probability that random walks use. Thus, the edge weights play a large role in the performance of the algorithm. If we can find a set of intra-community edges and assign them higher weights, the walks or labels will stay in smaller communities.

Community Detection Formulation

In this paper, we are interested in applying our method to undirected networks representing social networks. The following description of the community detection problem is oriented toward networks with no edge weights, but the method described can be generalized to weighted networks.

Some notation is as follows: Let $G = (V, E)$ be the graph associated with the network, with a node set V and an edge set E. n, m are the number of nodes and edges in G, respectively. A is the adjacency matrix of G, where $A = (A_{ij})$:

$$A_{ij} = \begin{cases} 1 & ij \in G \\ 0 & ij \notin G \end{cases}$$

It is worth noting that A is assumed to be a sparse matrix since G is a representation of a social network.

We wish to assign weights to the edges in E, such that inter-community edges receive higher weights, while intra-community edges receive lower weights. Furthermore, we expect that the application of these weights will improve the precision of current community detection algorithms without affecting complexity.

Discovering Community Cores

We already claimed different weights can play an important role in community detection. In this section, we address the

[1] www.facebook.com

two problems proposed earlier.

Choosing an appropriate reweighting method is not easy or obvious. Edge betweenness centrality is a strong candidate since Girvan and Newman already use it as a technique for detecting communities in their classical algorithm.

Selecting edge betweenness gives us two subsequent problems to solve. First, edge betweenness measures inter-community edges with higher weight, while we need these edges to be lower weight. Second, the complexity for calculating betweenness is $O(n^3)$, which is infeasible. The first problem is simpler to solve, and is addressed later. We solve the second problem by approximating current-flow betweenness, as introduced by (Newman and Girvan 2004).

Betweenness Centrality

Approximating current-flow betweenness can also be done by approximating spanning-tree betweenness. Spanning-tree betweenness is a measure of edge betweenness that measures the number of spanning trees on which each edge lies. This spanning-tree betweenness is equivalent to current-flow betweenness according to (Mavroforakis et al. 2015) and demonstrated in (Bollobás 1998). In addition to reiterating this equivalence, the former also describes a efficient way to compute this betweenness. For brevity, we omit the exact algorithm and discussion of the algorithm's working. In summary, the algorithm uses a sampling technique to approximate current-flow betweenness and a fast, approximate linear equation solver to solve Kirchoff's Laws. These two approximations reduce the computation's complexity from $O(n^3)$ to $\tilde{O}(m \log^2 n \log\left(\frac{1}{\epsilon}\right))$, where ϵ is the error of both approximations.

Measuring Intimacy

Using these edge centrality metrics, we can then move on to actually reweighting the network. We define the measure that we will use to reweight the network as Intimacy:

Definition 1 (Intimacy) *In a network, the intimacy measure for each edge $ij \in E(G)$ is a real number I_{ij} that is inversely proportional to its betweenness measure c_{ij}:*

$$I_{ij} := \max_{i,j} c_{ij} + \min_{i,j} c_{ij} - c_{ij}$$

To calculate intimacy, we first calculate the betweenness measure over the entire network, which can be done quickly thanks to the work in the previous section. From the description above, Algorithm 1 formalizes the procedure.

In the final step of this algorithm, we add an additional ϵ term to ensure that every edge in the network has at least some intimacy.

Intimacy Verification

In order to demonstrate that intimacy does, in fact, distinguish between inter- and intra-community edges, we performed two test of intimacy values. The first, a Mann-Whitney-Wilcoxon (Mann, Whitney, and others 1947) test, demonstrated that inter- and intra- community edges had substantially different distributions with a maximum p-value of 3.54×10^{-8}. The second, a pair comparison test, varied the level of community overlap and showed that intra-community edges had greater values than inter-community edge with greater than 50% probability until more than 80% of the community overlapped. From these tests, we can conclude that intimacy does, in fact, distinguish edge types.

```
**Input:** Original network adjacency matrix $A$
**Output:** Intimacy matrix I
Choose a set $T$ of $k$ samples of positive pole and negative poles
**for** *each* $s \in T$ **do**
    Solve $v$ in linear equation $(\mathbf{D} - \mathbf{A})\mathbf{v}^{(st)} = \mathbf{b}^{(st)}$.
    Calculate current-flow betweenness $c_{ij}$ for all edges $ij$
**end**
Find $c_{\min} = \min(c_{ij}),\ c_{\max} = \max c_{ij}$.
Calculate $S = c_{\max} + c_{\min}$.
**for** *each edge* $ij \in E$ **do**
    Compute $I_{ij} = S - c_{ij}$
**end**
**for** *each pair* $(i, j)$ *with* $A_{ij} = 1$ **do**
    $I_{ij} = I_{ij} + \epsilon$
**end**
```

Algorithm 1: Intimacy Calculation

Incorporating Intimacy

Since Intimacy can be computed quickly and differentiates edge types, we can integrate the intimacy measure into common community detection algorithms.

1. Infomap (Rosvall and Bergstrom 2008). Using the properties of Random Walks, this algorithm generates a number of modules and then optimizes communities by combining and separating these modules in such a way that minimizes the map equation.
2. Speaker-listener Label Propagation Algorithm (SLPA) (Xie, Szymanski, and Liu 2011). Proceeding iteratively, SLPA gives all nodes labels a gives a node's label to each of its neighbors and then repeats. After all iterations are complete, labels above a given threshold are kept.
3. Louvain's algorithm (Blondel et al. 2008). A very popular modularity optimization method, Louvain's algorithm is an agglomerative heuristic algorithm.

To incorporate our method into these algorithms, we simply apply the original method on the reweighted network.

Experimental Results

Datasets

The data we use for our experiments are as follows, with statistics given in Table 1.

- LFR Benchmark network (Lancichinetti, Fortunato, and Radicchi 2008). The LFR benchmark is not a single data set, but an algorithm for generating datasets. This algorithm for generating datasets allows the user to control a large number of features of the output graph.

Network	Nodes	Edges	Communities
Amazon	334,863	925,872	271,270
Youtube	1,134,890	2,987,624	8,385
DBLP	317,080	1,049,866	13,477

Table 1: Real-world dataset statistics

- Amazon Co-Purchasing Network (Leskovec and Krevl 2014): The Amazon Co-Purchasing Network is based on the 'Customers Who Bought This Item Also Bought' feature. Two products, nodes, are linked if they are frequently purchased together.
- YouTube (Leskovec and Krevl 2014): The YouTube network is based on user-defined groups on the YouTube[2] video sharing site. User groups form the ground-truth community assignments.
- DBLP (Leskovec and Krevl 2014): The DBLP network is based on collaboration between researchers in Computer Science. In this network, the ground-truth communities are determined by publication venue.

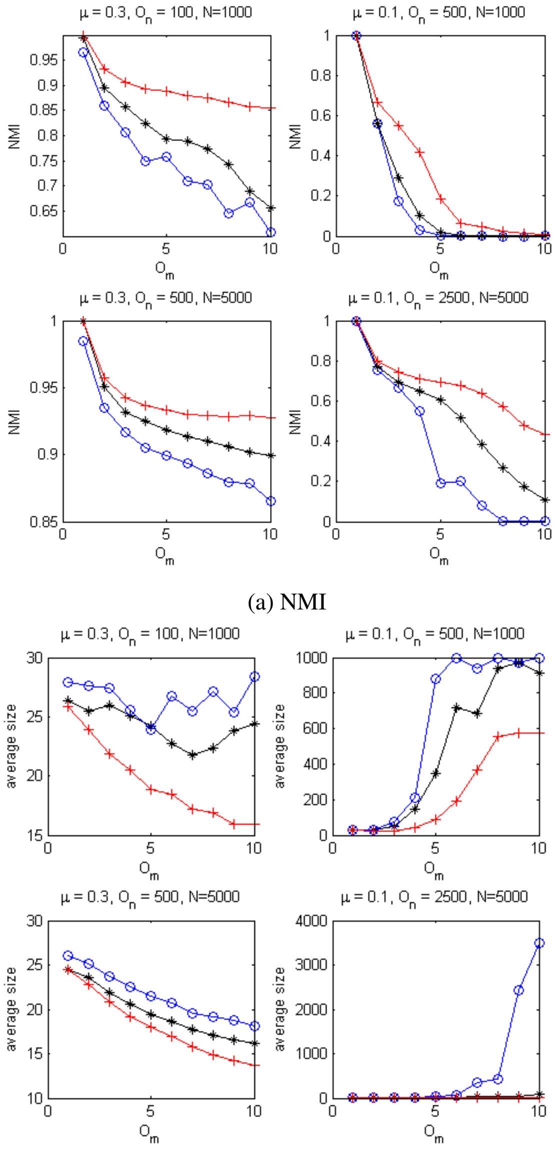
  The sizes of these datasets can be found in Table 1.

Results

Previously, we demonstrated that our intimacy measure effectively distinguishes edge type by their values. We will next demonstrate the performance-increasing potential of the intimacy reweighting technique. To judge the level of improvement, we will be evaluating community partitions using Generalized Normalized Mutual Information (GNMI) (Lancichinetti, Fortunato, and Radicchi 2008). We chose GNMI since standard NMI does not properly handle overlapping communities. An implementation of this method can be found online[3].

Our evaluation depicted in Figure 1 took place on synthetic networks generated using the LFR benchmark. In these synthetic networks, we varied a number of parameters, including the network size (N), number of nodes in overlapping communities (O_n), number of communities to which nodes with overlap belong (O_m), and the topological mixing parameter (μ). We use 2 for the power-law degree constants for both community sizes and node degree distributions and 25 for the average node degree in all instances. To ensure that our results are consistent we use 30 network realizations and plot the average.

Figure 1 shows the results of combining our reweighting technique with the SLPA algorithm. In all subgraphs, we show the base algorithm's performance (blue), the base algorithm with intimacy reweighting (red), and the base algorithm with random reweighting (black). Random reweighting occurred by selecting edge weights randomly from the uniform distribution of $[1, 5]$. Figure 1(a) clearly shows that the NMI of community partitions increases when the network is reweighted with intimacy. Figure 1(b) additionally shows that average community sizes decrease or remain approximately the same when the network is reweighted. These together show that the communities detected after the network is reweighted are smaller on average and that these smaller communities are more accurate to the ground truth.



(a) NMI

(b) Community average size

Figure 1: Synthetic results: SLPA

In addition to reweighting on SLPA, we also performed experiments with Infomap, which uses some global information. Infomap maximizes the map equation to find its final partitions, which uses global information. Under these conditions, reweighting with intimacy does not provide as strong a result as it does with SLPA.

Since our method shows promising results on these synthetic networks, we next look to the real-world data sets. Table 2 shows the results of testing these combinations. Our

[2] www.youtube.com

[3] https://sites.google.com/site/andrealancichinetti/mutual

Network	SLPA(non)	SLPA(int)	SLPA(rnd)	IMP(non)	IMP(int)	IMP(rnd)	LVA(non)	LVA(int)	LVA(rnd)
Amazon	0.242	**0.273**	0.250	**0.226**	**0.226**	0.224	0.280	**0.285**	0.263
Youtube	**0.024**	0.018	0.013	0.004	**0.006**	0.005	**0.027**	0.026	0.018
DBLP	0.130	**0.143**	0.132	**0.096**	**0.096**	0.091	**0.150**	0.149	0.136

Table 2: NMI results for Infomap (IMP), SLPA, and Louvain's algorithm (LVA) on real-world networks with no reweighting (non), random reweighting (rnd), and intimacy reweighting (int).

results here are similar to the ones obtained on synthetic networks with two exceptions. The first, SLPA and intimacy on the YouTube network is interesting, as any reweighting degrades performance. This pattern is not repeated, so it is unclear why SLPA performs so well. The second anomaly exists with Louvain's algorithm on the YouTube and DBLP networks. In this case, we see performance degradations when reweighting. We believe that this is because the final step of Louvain's algorithm uses modularity to determine the optimal partition. Scaling up the network to these sizes likely caused modularity to combine some of the new, smaller communities together due to its resolution limit, resulting in a lower NMI. Since the communities in our synthetic networks were large relative to the size of the network, this did not occur in our synthetic results.

These results indicate that our intimacy reweighting technique does provide strong, positive results across a wide variety of network topologies, though the technique works better as the size and complexity of the network increase. In addition, the technique is more effective when used alongside algorithms that do not take global information into account, like SLPA and Louvain's algorithm. Algorithms that use global information like Infomap can benefit, although not as much.

Guidelines

Effectively applying this intimacy reweighting technique can be situational. Since our technique operates faster than the base community detection algorithms, it can be applied without affecting computational complexity. Our technique works best when the algorithm of choice does not use global information. In addition, using a modularity-based technique may hamper performance on some networks. That said, our method does result in smaller detected communities across detection methods and network types, so it may be valuable if small communities are the goal.

Conclusion

This paper develops a new reweighting measure to improve the performance of existing community detection methods without compromising complexity. The reweighting technique finds global information in near linear time, and has a wide range of applications. Applying this technique shows gains in local-information based methods, like SLPA.

Our work also shifts the focus of common community detection algorithms from large communities to small communities. Handling small communities may have impacts outside of improving community detection results and matching community size distributions. One emerging problem in community detection is Evolving Community Detection. Additional ability to to detect communities earlier in their course of course may assist with the task of detecting these evolving, highly dynamic communities.

Acknowledgments

The authors would like to acknowledge the contributions of all of the members of ASU's DMML. Research was sponsored in part by the U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA).

References

Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R.; and Lefebvre, E. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008(10):P10008.

Bollobás, B. 1998. *Modern graph theory*, volume 184. Springer Science & Business Media.

Lancichinetti, A.; Fortunato, S.; and Radicchi, F. 2008. Benchmark graphs for testing community detection algorithms. *Physical Review E* 78(4):046110.

Leskovec, J., and Krevl, A. 2014. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data.

Mann, H. B.; Whitney, D. R.; et al. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics* 18(1):50–60.

Mavroforakis, C.; Garcia-Lebron, R.; Koutis, I.; and Terzi, E. 2015. Spanning edge centrality: Large-scale computation and applications. In *Proceedings of the 24th International Conference on World Wide Web*, 732–742. International World Wide Web Conferences Steering Committee.

Newman, M. E., and Girvan, M. 2004. Finding and evaluating community structure in networks. *Physical review E* 69(2):026113.

Rosvall, M., and Bergstrom, C. T. 2008. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences* 105(4):1118–1123.

Tang, L., and Liu, H. 2010. Community detection and mining in social media. *Synthesis Lectures on Data Mining and Knowledge Discovery* 2(1):1–137.

Xie, J.; Szymanski, B. K.; and Liu, X. 2011. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, 344–349. IEEE.