A study of algorithms for detecting community in networks

Abstract

Detecting community structure in networks is highly desirable in many application domains, such as finding proteins with similar functionality in a biological pathway or automatically grouping relevant people in a social network. However, this is still a daunting task due to the network sizes, as well as the complicated relations between entities. This paper provides a study on algorithms to find communities in a network using edge betweenness centrality. We have implemented the modularity in order to compute the suitable network structure on weighted undirected networks. We also discuss the pros and cons of existing techniques in detecting community structures. To highlight the benefits of the selected techniques, we demonstrate their applications on various datasets, including Victor Hugo's Les Misérables, the movies's network of actors, the author's collaboration network in visualization publications, and the protein interaction network.

1 Introduction

Network is an important representation in data visualization and visual analytics. A network consists of vertices and edges representing individual entities and relationships between them. Dense connections between related nodes forms cliques/communities. However, visually detecting such communities is challenging even with a small network. Many solutions [1, 8, 9] have been proposed to automatically detect community structure in networks. One effective technique is based on edge betweenness centrality [8], the sum of the fraction of all-pairs' shortest paths that pass through a given edge. However, this is still an on-going challenge due to the increasing amount of data, especially the complicated relationships between entities. One instance of such real-world data is the biological network of protein interactions which may contain millions of proteins and billions of connections between them [3]. Additionally, edges may have different weights to indicate the levels of correlations between related entities (and usually shown by the thicknesses of edges in the graph). These are weighted networks.

The research presented in this paper focuses on some available techniques on edge betweenness centrality algorithm, focusing on the undirected weighted network. The main contributions of this paper include:

- We integrate virtual nodes method into Newman edge betweenness algorithm [8]. We then apply a sampling technique [7] on edge betweennness centrality which significantly reduces the computing time.
- We demonstrate the usefulness of our revised technique on various real-world data. The data and community detection techniques (implemented in forms of javascripts library) are freely provided to research community.
- We conduct a quantitative study on networks with different features. The results from this study can serve as a guideline for selecting the community detection techniques for a given network.

2 Method

We implement algorithms in JavaScript for the original weighted networks. Wen use Newman algorithm [8] for community detection which is based on edge betweenness centrality. Three approaches to calculate the edge betweenness centrality are conducted. The first approach uses Brandes' algorithm, the second approach converts the weighted network into unweighted network by adding virtual nodes (thereafter, called VS) and use Breath First Search for computing edge beweenness centrality. The final approach is the Random

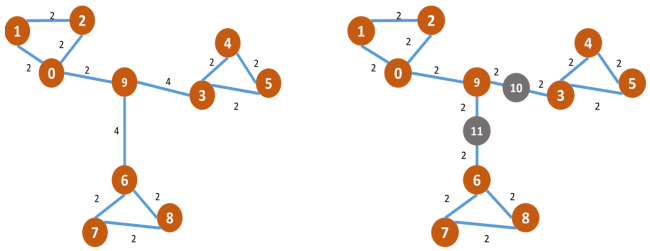

Figure 1: An example of VS: (a) Original network with 10 nodes, (b) New network with two more virtual nodes. Orange are actual nodes while gray are virtual nodes.

sampling method [7] which estimates the betweenness centrality based on vertex-diameter (and independent from the network size). First, the sample size is calculated as follows:

$$r = \frac{c}{\varepsilon^2}\left(\lfloor \log_2(VD(G)-2)\rfloor + 1 + \ln\frac{1}{\delta}\right)$$

where the two parameters $\varepsilon, \delta \in (0,1)$, and c is an universal positive constant. $VD(G)$ is the sum of the length of two shortest paths with maximum size from v to two other distinct vertices u and w. Then the algorithm runs r times to accumulate the betweenness centrality.

We use Modularity Q as a measure to evaluate the quality of the community which is calculated based on the following equation:

$$Q = \sum_i (e_{ii} - a_i^2) \tag{1}$$

where i and j are communities, e_{ij} is the fraction of edges with one end vertices in community i and the other in community j, and a_i is the fraction of ends of edges that are attached to vertices in community i.

3 Evaluation and Finding

We study the running time of community detection algorithms on four weighted real-world networks of various sizes, namely: Victor Hugo's Les Misérables dataset [6], the movie's network of actors available on the IMDB website [4], the author's collaboration network in Visualization publications [5], and the protein interaction network [2]. In the last three datasets, we extracted the sub-networks (from the original data) of 250, 500, 750, and 1,000 highest-degree vertices for testing purposes. Table 1 displays prominent features of networks in our study. The last two columns are colored differently since these attributes are specific for VS and sampling techniques.

Table 1: Prominent features of datasets in our study.

Dataset	n	m	$\bar{w}$	$\bar{D}$	Q	#VS	#sample
Victor Hugo	77	254	3.21	6.6	0.51	170	215
IndexCards	250	579	1.07	4.6	0.83	295	265
	500	1,108	1.01	4.4	0.81	562	115
	750	1,433	1.04	3.8	0.91	816	315
	1,000	1,761	1.04	3.5	0.92	1,074	315
VisPublication	250	663	1.88	5.3	0.85	457	265
	500	1,436	1.73	5.7	0.72	838	265
	750	2,178	1.59	5.8	0.90	1,226	265
	1,000	2,894	1.53	5.7	0.93	1,483	315
IMDB	250	721	2.62	5.7	0.55	371	215
	500	1,126	2.49	4.5	0.69	638	265
	750	1,461	2.40	3.9	0.72	888	265
	1,000	1,772	2.35	3.5	0.76	1,146	265

In Figure 2, the top left panel shows modularity graph by the number of clusters (which guides the selection of number of clusters using a slider) and the network dendrogram is below. The current selection (the black dotted line) of cluster numbers is also reflected on the dendrogram. By default, we set the number of clusters at the maximum Modularity Q. As the number of communities changes, the graph layout gets updated dynamically. Observations results:

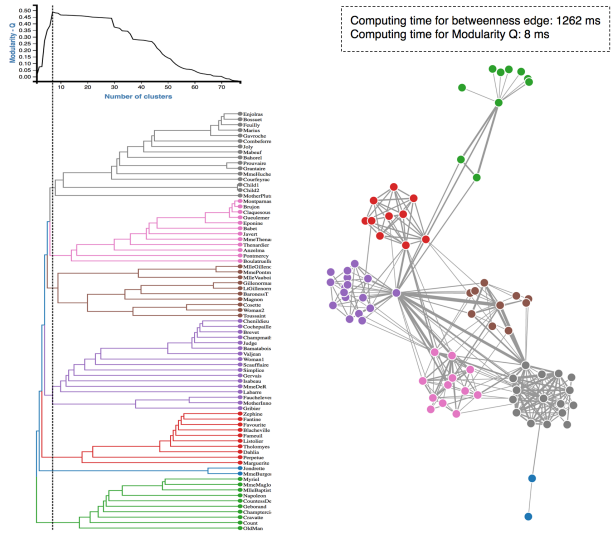

Figure 2: Interface for the Victor Hugo dataset: number cluster selected by a slider (left) and the associated network visualization (right) where nodes are colored by cluster.

- Brandes and VS running times exponentially increase with network sizes but random sampling technique is almost linear.
- As depicted in Figure 3 (a) at the network size of 250, the sampling method (green) is slower than other two algorithms. This rare case occurs on smaller networks with higher vertex-diameter. The number of iterations (or sample size of the first row for the Indexcards dataset in Table 1) to calculate edge betweenness centrality is greater than network size (265>250).

For small and highly weighted networks, we suggest using Brandes' algorithm due to high accuracy and smaller computational time. But for larger networks, VS with random sampling technique reduces running time significantly.

4 Conclusion

In this paper, we have applied VS and random sampling method for faster community detection on weighted networks. The running times of different algorithms are compared in Section 3. As depicted in Figure 2, Brandes' algorithm [8] does not scale well with the network size. VS [10] is even worse in most cases since it

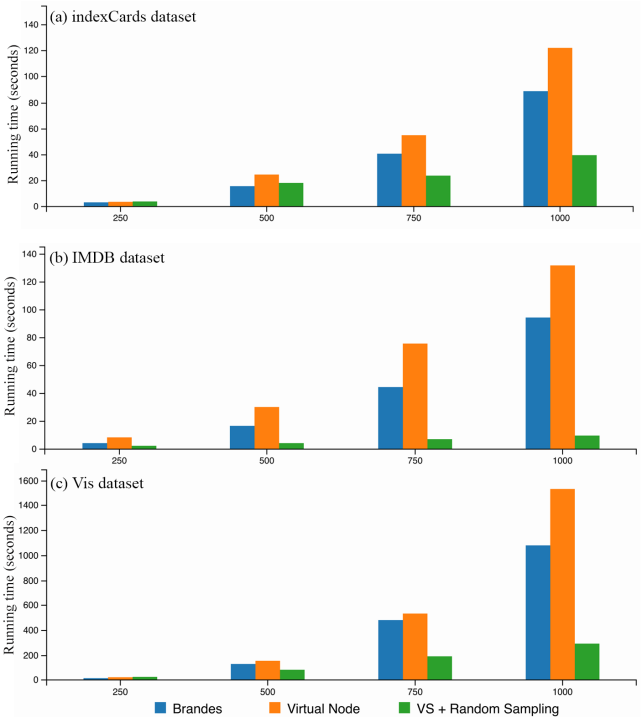

Figure 3: Computational time on different network sizes: blue for Brandes' algorithms, orange for VS, green for random sampling.

introduces many virtual nodes into the existing network. However, VS helps to convert weighted networks into unweighted ones for which we can apply random sampling [7] to reduce the running times. Notice that the random sampling method is independent from network size but depends on vertex-diameter of the given network. Although our empirical study is limited on a single machine (can not handle large networks), Figure 2 clearly depicts that sampling technique combined with VS scales are better, compared the Brandes' algorithm.

References

[1] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163177, 2001.

[2] E. G. Cerami, B. E. Gross, E. Demir, I. Rodchenkov, Ö. Babur, N. Anwar, N. Schultz, G. D. Bader, and C. Sander. Pathway commons, a web resource for biological pathway data. *Nucleic acids research*, 39(suppl 1):D685–D690, 2011.

[3] T. N. Dang, P. Murray, and A. G. Forbes. PathwayMatrix: Visualizing binary relationships between proteins in biological pathways. *BMC Proceedings*, 9(6):S3, 2015. 10.1186/1753-6561-9-S6-S3.

[4] IMDB. Imdb support community. Dataset: http://www.imdb.com/interfaces, accessed Jan, 2017.

[5] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. Stolper, M. Sedlmair, J. Chen, T. Möller, and J. Stasko. Visualization publication dataset. Dataset: http://vispubdata.org/, 2015. Published Jun. 2015.

[6] D. E. Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*, vol. 37. Addison-Wesley Reading, 1993.

[7] E. M. K. Matteo Riondato. Fast approximation of betweenness centrality through sampling. *Data Min Knowl Disc*, 30:438–475, 2015.

[8] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Preprint cond-mat/0308217*, 2003.

[9] P. Pons and M. Latapy. Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications*, 10(4):191–218, 2006.

[10] J. Yang and Y. Chen. Fast computing betweenness centrality with virtual nodes on large sparse networks. *PLoS ONE*, p. 6:e22557, 2011.