Community Detection by Local Influence

Nuno Cravino and Álvaro Figueira

CRACS/INESC TEC, Faculdade de Ciências, Universidade do Porto
{nuno.cravino, arf}@dcc.fc.up.pt

Abstract. We present a new algorithm to discover overlapping communities in networks with a scale free structure. This algorithm is based on a node evaluation function that scores the local influence of a node based on its degree and neighbourhood, allowing for the identification of hubs within a network. Using this function we are able to identify communities, and also to attribute meaningful titles to the communities that are discovered. Our novel methodology is assessed using LFR benchmark for networks with overlapping community structure and the generalized normalized mutual information (NMI) measure. We show that the evaluation function described is able to detect influential nodes in a network, and also that it is possible to build a well performing community detection algorithm based on this function.

Keywords: graph theory, link analysis, centrality, community detection, overlapping communities

1 Introduction

Nature and human derived complex networks follow certain patterns in their structure and development[2]. Social networks, computer networks, protein interaction networks, among others, tend to follow fat tailed distributions of node degree at least asymptotically. Many of these networks also display a community structure revealed by the existence of groups with highly interconnected nodes, with low connectivity to other groups. In most cases maintaining a low average path length between any node in the network, whether or not in the same group. Community detection algorithms are an attempt to retrieve these groups. The most currently used algorithms retrieve disjunct clusters from the networks. Recent developments have shown that many networks display overlapping clusters or an hierarchical disposition of clusters[13, 10, 8, 11], which creates a pressing need for newer techniques that should be not only able to retrieve these communities, but that should also be capable of doing so in the large scale networks.

1.1 Previous Developments in Overlapping Community Detection

The Clique Percolation Method (CPM)[10] is a popular method for overlapping community detection which assumes that communities arise as densely connected

sub-graphs. The search for communities is done by the identification of all cliques of a certain size. After this step it generates a new graph containing all identified cliques as nodes that are considered adjacent if they share most of their elements (clique size minus one). The communities are then found by retrieving the connected components in this final graph. Speaker-listener Label Propagation Algorithm (SLPA)[12] is another algorithm that uses a variant of the label propagation algorithm to construct the communities. In this algorithm labels are shared between neighbour nodes, and afterwards in the distribution of these labels is processed in order to retrieve the communities.

Our approach differs from these algorithms by using a local evaluation function that discovers the structure of the network around a certain node, and uses this information to guide the search to a local maxima of the function. Clusters are then formed by nodes around the local maxima. Our algorithm is relatively stable since its results will only be affected by the order of evaluation, and if the order remains the same the results will be the same.

2 Community Detection by Local Influence

This research was developed in order to retrieve socially relevant information from a tag co-occurrence network, to be used to socially influence[4] the classification of documents in a news related social network[1]. Stability and performance were paramount to this research in order to maintain user acceptable results and delays.

2.1 The Local Influence Score

We developed a new local scoring function, the local influence, in order to retrieve hub candidates from the neighbourhood of a node. This function is built upon the properties of networks having a community structure, and of scale free networks, and is a local measure of the influence of a node in a network. Informally the local influence of a node can be defined as a score that measures the importance of a given node to the overall structure of a network. Nodes that poorly affect the structural properties of the network will have a low score, while high scoring nodes will have a significant impact. Higher influence nodes not only connect to most nodes within a certain range (i.e. in the same community), but they also provide connectivity to other sets of well connected nodes farther within the network, by forming bridges or providing increased connectivity to nodes that do so. Due to this, their removal would increase the average shortest path length between nodes within the same community, and also decrease connectivity to other communities in the network.

Scale free networks[3] have hubs which are high degree nodes that connect a large set of nodes, where the removal of just one of them can result in a significant increase in network diameter. The overall influence of these nodes over the

structure of the network is high, since they ensure the low average shortest path between all nodes in the network when compared to random graphs. Based on this structural importance of hubs in scale free networks, a node influence on the network structure needs, by our informal definition, to be proportional to its degree.

Communities[5], in the context of community detection, are loosely defined as groups of highly interconnected nodes, with significantly lower connectivity to other communities. Given this definition we add to our previous hypothesis that a node influence is higher if it connects with other high degree nodes, as would be the case of nodes within communities, and even more if they also connect to other communities.

Finally, we add that in a weighted network the neighbours importance is proportional to the weight of the edge that connects with them. Based on this set of hypothesis we construct the local influence scoring function as

$$score(n) = degree(n) \times \sum_{v \sim n} (degree(v) \times weight(n, v)) \tag{1}$$

where n is the node under evaluation, v a neighbour node of n, and $\sim$ the adjacency relation of the network. This function can be used to score nodes proportionally to the chance of being a hub, where its value will decrease for lesser connected nodes, and increase for nodes that follow our informal definition of local influence. This measure can also be viewed as an extension of the notion of degree centrality, taking into account the neighbourhood of the node being scored. The Hyperlink-Induced Topic Search (HITS)[6] algorithm also enables the identification of network hubs, using iterative improvement to compute its authority and hub scores. Unlike the local influence function here described, HITS takes a global approach, and therefore presents a higher complexity.

In figure 1 we show the nodes' local influence on the Zachary's Karate Club network[14]. It is visible that the hubs are the two nodes that originated the split, which consequentially have an high local influence score using our metric. There are other nodes that can be identified as influential given their high degree centrality and connections with other influential nodes.

3 The Community Detection Algorithm

In algorithm 1 we present the pseudo code for our algorithm. We have excluded the initialization and post-processing phases for conciseness. The initialization phase is simply the construction of a table of all node scores in order to be possible to access them in constant time.

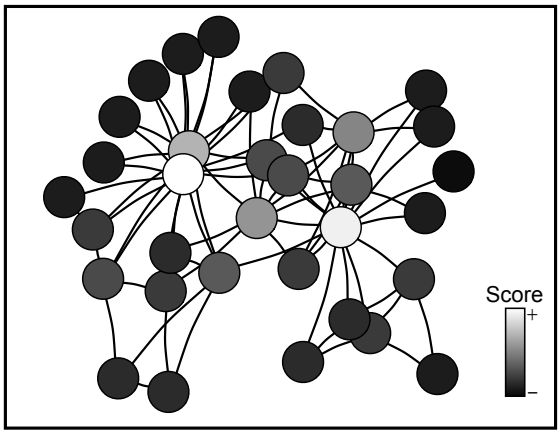

Fig. 1. Zachary Karate Club network local influence scores

Algorithm 1 Community Detection by Local Influence

```
1: function FindCommunities(VertexList)
2:     Counts ← (empty sparse matrix)
3:     maxHop ← ⌊ln(ln(|VertexList|))⌋
4:     for all candidate ∈ VertexList do
5:         maxScore ← −∞
6:         hub ← nil
7:         S ← ∅
8:         while (candidate ≠ hub) ∧ (hop < maxHop) do
9:             hub ← candidate
10:            hop ← hop + 1
11:            for all v ∈ hub.Adjacency do
12:                if (score(v) × weight(hub, v)) > maxScore then
13:                    maxScore ← score(v)
14:                    candidate ← v
15:                S ← S ∪ v
16:        for all u ∈ S do
17:            Counts[hub][u] ← Counts[hub][u] + 1
18:    return process(Counts)
```

3.1 The Community Detection Algorithm

The detection phase uses a guided search strategy to find a single path from each node to the nearest local hub, storing the set of the evaluated nodes. This path length is limited by an approximation of the average path length, since for nodes within the same community the path length will be less than the average. We use $\lfloor \ln(\ln(|N|)) \rfloor$ as the approximation, building upon the result by Cohen *et al.* in [3], about the order of the distance in scale free networks. This limit enables us to reduce the overall complexity of the algorithm, given the slow growth of $\ln(\ln(|N|))$.

The communities are formed by processing the node occurrence counts in the paths leading to each hub. Given $Counts[h]$ the count vector of all nodes found in the paths to hub h, we retrieve $Com[h]$, the community formed around h, as follows

$$Com[h] = \{v \in Counts[h] : Counts[h][v] \geq \frac{\sum_{u \in Counts[h]}(Counts[h][u])}{|Counts[h]|}\} \quad (2)$$

being that v and u are nodes. By joining together only those nodes with a significant occurrence count, equal to or greater than the arithmetic mean, we are able to reduce the presence of non related nodes in the same cluster.

The retrieved communities can be optionally post-processed to join similar communities and attribute titles. We cycle through all clusters and use the top n scoring nodes in each cluster to assign a title, merging all clusters sharing the exact same title. This allows us to reduce the number of clusters with very similar node sets without the need for costly comparisons. Also in word based networks we are able to obtain meaningful titles that allow to better identify the underlying context of the cluster.

3.2 Complexity

Since the scores are not updated during the detection phase, we can generate the table of all scores prior to that phase so we can have constant access time to the scores. In order to do this all the neighbours for each node must be processed, which takes a total of $\sum_{n \in N} degree(n)$ steps. This value can be simplified to a bound of $|N|d$ where d is the ceiling of the average node degree. Also, given that the node degree distribution follows a power law for scale free networks, it holds that the average node degree is significantly smaller than the number of nodes. For the detection phase we will once again have to process all the neighbours for each node, but in addition we have a cycle with at worst $\ln(\ln(|N|))$ time complexity. The complexity for processing nodes was shown before to be $|N|d$, but with the additional cycle it becomes $|N|d\ln(\ln(|N|))$. The count processing operation, after the inner loops, takes time linear on the size of the smallest set, so its complexity is bounded by the complexity of the inner loops. This is because at most the number of nodes to add to the count will be the same as the number of steps in the inner loops, therefore it only increases the complexity by a constant factor, and so it can be discarded. Therefore the overall complexity upper bound for the detection phase will be $|N|d\ln(\ln(|N|))$, being below quadratic in relation to $|N|$ for most non complete graphs, and well below that value for scale free networks.

In the count processing phase we process for each cluster i all of its nodes $|C_i|$. Since the mean calculation for each cluster only increases complexity by a constant factor, the complexity for this phase is bounded by $\sum_{i \in clusters} |C_i|$, the

sum of the sizes of all clusters. This sum cannot ever exceed the bound of the detection phase, since all the cluster elements are retrieved in that phase, therefore this sum of the cardinalities of the clusters will be at most the same as the number of steps of the detection phase. Thus the complexity for the processing phase is the same as that of the detection phase.

The overall complexity of the algorithm is bounded by $|N|d + 2|N|d\ln(\ln(|N|))$, that can be simplified to a time complexity bound of $|N|d\ln(\ln(|N|))$. In practice the average shortest path of most real complex networks will remain mostly constant and it can also be assumed that the degree will always be significantly smaller than the number of nodes in the network. Therefore, this algorithm presents near linear complexity over the number of nodes, and edges, in these specific networks.

The algorithm we present is also inherently parallelizable, since the computation of the path to the hub of a node needs not to update any information relevant to any computation for any other node. Therefore, with little modification, these paths can theoretically be computed concurrently, followed by the processing of clusters.

4 Evaluation

4.1 Benchmark and Evaluation Metric

In order to evaluate the our algorithm we use the LFR model, by Lancichinetti *et al.* l[9, 7], for weighted undirected graphs with overlapping structure to generate the graphs with a ground truth. LFR has various parameters, but we allow variation in only a few. The number of nodes ($|N|$) is set to $\{5000, 50000\}$ in order to test the algorithm in different scale graphs. The number of overlapping nodes (O_n) was set to $\{1\%|N|, 10\%|N|\}$ in order to evaluate the algorithm in networks with different overlapping structural tendencies. The maximum membership size for overlapping nodes (O_m) for $\{2, 6\}$ in order to assess the response of the algorithm to higher overlaps. The mixing parameter for topology (μ_t) allows, for higher values, changes on the structure of the network in order to simulate more dense networks with less defined community structures and was set to $\{0.1, 0.2, 0.3\}$. The rest of the parameters are set as Xie *et al.*[11] in the case of small communities, with $\mu_w = \mu_t$, community sizes between $[20, 50]$, average degree of 10 and maximum degree of 50. We used the generalized NMI[8] for overlapping clusters to evaluate the results obtained

4.2 Results

Our results follow the general trend shown by others in previous comparisons[11] using the same benchmark, though showing a particular sensitivity to changes in the topology of the network and increases in the total number of overlapping

Table 1. The results of our tests for $|N| = 5000$

Om	2			6		
μ_t	0.1	0.2	0.3	0.1	0.2	0.3
NMI(1%On)	0.7534	0.6148	0.4476	0.7315	0.5818	0.4260
NMI(10%On)	0.6170	0.5035	0.3335	0.4473	0.3566	0.2333

Table 2. The results of our tests for $|N| = 50000$

Om	2			6		
μ_t	0.1	0.2	0.3	0.1	0.2	0.3
NMI(1%On)	0.7734	0.6030	0.4249	0.7345	0.5863	0.4087
NMI(10%On)	0.6242	0.4821	0.3455	0.4528	0.3656	0.2728

nodes. This may be due to the restrictions imposed onto the path finding process, that may not be entirely compatible with the structural changes resulting from changes in μ. Overall, and given its low complexity, it performs well on networks with a well defined community structure as shown on tables 1 and 2.

5 Conclusion

We presented a new technique for community detection, performing as well as some other techniques previously reported while remaining mostly deterministic and presenting a good complexity profile. These characteristics are indispensable in order to provide users with a real time response and a consistent experience, necessary features of most real time non-technical systems. The scoring function here presented can be used independently from the algorithm, as a measure of influence, in order to analyse the network and identify potential hubs. It can also be used to assign meaningful titles to groups or clusters of nodes based on their scores. This algorithm can be further enhanced with optimized cluster generation, to better take into account the distribution nodes into communities.

6 Acknowledgements

This work is financed by National Funds through the FCT Fundao para a Cincia e a Tecnologia (Portuguese Foundation for Science and Technology) within project PEst-C/EEI/LA0014/2011.

References

1. Álvaro Figueira et al. Breadcrumbs: A social network based on the relations established by collections of fragments taken from online news. *Retrieved January 19, 2012, from http://breadcrumbs.up.pt.*
2. Albert-László Barabási. Scale-free networks: a decade and beyond. *Science*, 325(5939):412–413, 2009.

3. Reuven Cohen, Shlomo Havlin, and D Ben-Avraham. Structural properties of scale free networks, 2002.
4. Nuno Cravino, José Devezas, and Álvaro Figueira. Using the Overlapping Community Structure of a Network of Tags to Improve Text Clustering. In *Proceedings of the 23rd ACM Conference on Hypertext and Social Media HT 2012*, 2012.
5. M Girvan and M E J Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826, 2002.
6. Jon M Kleinberg. Hubs, authorities, and communities. *ACM Computing Surveys*, 31(4es):5–es, 1999.
7. Andrea Lancichinetti and Santo Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):9, 2009.
8. Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.
9. Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E - Statistical, Nonlinear and Soft Matter Physics*, 78(4 Pt 2):6, 2008.
10. Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.
11. Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. Overlapping Community Detection in Networks : the State of the Art and Comparative Study. *Arxiv preprint arXiv11105813*, V(November):1–30, 2011.
12. Jierui Xie, Boleslaw K Szymanski, and Xiaoming Liu. SLPA: Uncovering Overlapping Communities in Social Networks via a Speaker-Listener Interaction Dynamic Process, 2011.
13. Jaewon Yang and Jure Leskovec. Structure and Overlaps of Communities in Networks. *Arxiv preprint arXiv12056228*, 2012.
14. Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.