Towards Democratic Group Detection in Complex Networks

Michele Coscia[1], Fosca Giannotti[2], and Dino Pedreschi[1]

[1] Computer Science Dep., University of Pisa, Italy
{coscia,pedre}@di.unipi.it
[2] ISTI - CNR, Area della Ricerca di Pisa, Italy
fosca.giannotti@isti.cnr.it

Abstract. To detect groups in networks is an interesting problem with applications in social and security analysis. Many large networks lack a global community organization. In these cases, traditional partitioning algorithms fail to detect a hidden modular structure, assuming a global modular organization. We define a prototype for a simple local-first approach to community discovery, namely the democratic vote of each node for the communities in its ego neighborhood. We create a preliminary test of this intuition against the state-of-the-art community discovery methods, and find that our new method outperforms them in the quality of the obtained groups, evaluated using metadata of two real world networks. We give also the intuition of the incremental nature and the limited time complexity of the proposed algorithm.

1 Introduction

Complex network analysis has emerged as a popular domain of data analysis over the last decade and especially its community discovery (CD) sub field has been proven useful for many applications, related also to crime prevention. The concept of a "community" in a network is intuitively a set of individuals that are very similar to each other, more than to anybody else outside the community [2]. In network terms, sets of nodes densely connected to each other and sparsely connected with the rest of the network. To efficiently detect these structures is very useful for a number of applications, we recall here information spreading [5] and crime understanding and prevention [11].

The classical problem definition of community discovery finds a very intuitive counterpart for small networks, but for medium and large scale networks the CD problem becomes much harder. At the global level, very little can be said about the modular structure of the network: the organization of the system becomes simply too complex. The graph of Facebook includes more than 800 millions nodes[1], but even in less than 0.002% of the total network no evident organization can be identified easily (Figure 1 on the left), resulting in a structureless hairball. In these cases generic community discovery algorithms tend to fail trying to

[1] http://www.facebook.com/press/info.php?statistics

S.J. Yang, A.M. Greenberg, and M. Endsley (Eds.): SBP 2012, LNCS 7227, pp. 105–113, 2012.

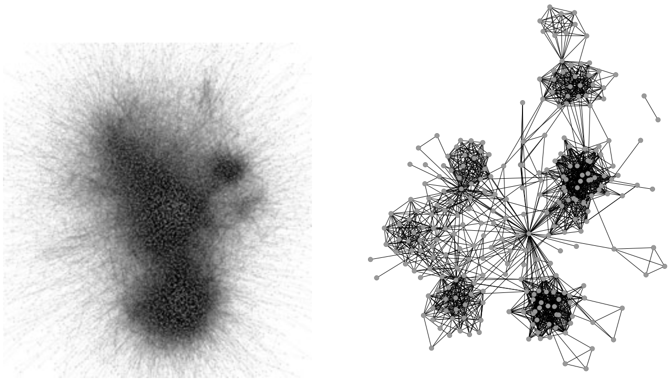


Fig. 1. The real world example of the "local vs global" structure intuition

cluster the whole structure and return some huge communities and a long list of small branches. This approach generally fails for large networks due to the difference in structural organization at global and local scale.

To cope with this difficulty, we propose a change of mentality: since our community definition works perfectly in the small scale, then it should be applied only at this small scale. The structure of cohesive groups of nodes emerges considering a *local* fragment of an otherwise big network. But what does *local* mean? Common sense goes that people are good at identifying the reasons why they know the people they know: each node has presumably an incomplete, yet clear, vision of the social communities that surrounds it. In Figure 1 on the right we chose one node from the example and extracted its ego-network and some groups can be easily spotted. The ego is part of all these communities and knows that particular subsets of its neighborhood are part of these communities too. Different egos have different perspectives over the same neighbors and it is the union of all these perspectives that creates an optimal partition of the network. This is achieved by a *democratic* bottom-up approach: in turn, each node gives the perspective of the communities surrounding it and then all the different perspectives are merged together in an overlapping structure.

We name our algorithm **D**emocratic **E**stimate of the **M**odular **O**rganization of a **N**etwork, or *DEMON*: we extract the ego network of each node and apply a Label Propagation CD algorithm [8] on this structure, ignoring the presence of the ego itself, that will be judged by its peers neighbors. We then combine, with equity, the vote of everyone in the network. The result of this combination is a set of (overlapping) modules, the guess of the real communities in the global system, made not by an external observer, but by the actors of the network itself. *DEMON* is *incremental*, allowing to recompute the communities only for the newly incoming nodes and edges. Nevertheless, *DEMON* has a low theoretical time complexity, and our experiments show that its performance is superior to that of fastest competitor methods, both overlapping and non overlapping. Online social networks have proved that individuals are part of many different communities and groups of interest, therefore the overlapping partition of groups is a crucial feature. The properties of *DEMON* support its use in massive real world scenarios, for example in tasks of group and threat detection.

The paper is organized as follows. In Section 2 we present the related work in the community discovery literature. In Section 3 we have the problem definition. Section 4 describe the algorithm structure. Our experiments are presented in Section 5, and finally Section 6 concludes the paper.

2 Related Work

The problem of finding groups in complex networks has been tackled by an impressive number of valid works. Traditionally, a community is defined as a dense subgraph, in which the number of edges among the members of the community is significantly higher than the outgoing edges. This definition does not cover many real world scenarios, and in the years many different solutions started to explore alternative definitions of communities in complex networks [2].

A variety of CD methods are based on the *modularity* concept, a quality function of a partition proposed by Newman [7]. Modularity scores high values for partitions in which the internal cluster density is higher than the external density. Besides modularity, a particular important field is the application of information theory techniques. In particular, Infomap has been proven to be one amongst the best performing non overlapping algorithms [2]. Modularity approaches are affected by known issues, namely the resolution problem and the degeneracies of good solutions [4]. A very important property for community discovery is the ability to return overlapping partitions [10], i.e., the possibility of a node to be part of more than one community. Specific algorithms developed over this property are Hierarchical Link Clustering [1], HCDF [6] and k-clique percolation [3]. Finally, an important approach is Label Propagation [8]: in this work authors detect communities by spreading labels through the edges of the graph and then labeling nodes according to the majority of the labels attached to their neighbors, until a general consensus is reached. This algorithm is extremely fast and provides a reasonable good quality of the partition.

To find groups in networks and to extract useful knowledge from their modular structure is a prolific track of research with a number of applications. We recall the GuruMine framework, whose aim is to identify leaders in information spread and to detect groups of users that are usually influenced by the same leaders [5]. Many other works investigate the possibility of applying network analysis for studying, for instance, the dynamics of crime related behaviors [11].

3 Networks and Communities

We represent a network as an undirected, unlabeled and unweighted simple graph, denoted by $\mathcal{G} = (V, E)$ where V is a set of nodes and E is a set of edges, i.e., pairs (u, v) representing the fact that there is a link in the network connecting nodes u and v. Our problem definition is to find groups in complex networks. However, this is an ambiguous goal, as the definition itself of "community" in a complex network is not unique [2]. Furthermore, if we want to develop an efficient instrument for criminal profiling and prevention, an analyst may want to cluster many different kinds of groups of suspects for many different reasons. Therefore, we need to narrow down our problem definition as follows.

We define a basic graph operation, namely the *EgoMinusEgo* operation: it consists in extracting the ego network of a node, i.e. the collection of a node and all its direct neighbors and all the edges between them, eliminating the ego itself. Given a graph $\mathcal{G}$ and a node $v \in V$, the set of *local communities* $\mathcal{C}(v)$ of node v is a set of (possibly overlapping) sets of nodes in $EgoMinusEgo(v, \mathcal{G})$, where each set $c \in \mathcal{C}(v)$ is a community: each node in c is closer to any other node in c than to any other node in $c' \in \mathcal{C}(v)$, with $c \neq c'$. We refer here to the topological distance between nodes in a graph, namely the length of the shortest path connecting any two nodes. Finally, we define the set of *global communities*, or simply communities, of a graph $\mathcal{G}$ as $\mathcal{C} = Max(\bigcup_{v \in V} \mathcal{C}(v))$, where, given a set of sets $\mathcal{S}$, $Max(\mathcal{S})$ denotes the subset of $\mathcal{S}$ formed by its maximal sets only; namely, every set $s \in \mathcal{S}$ such that there is no other set $s' \in \mathcal{S}$ with $s \subset s'$. In other words, we generalize from local to global communities by selecting the maximal local communities that cover the entire collection of local communities, each found in the *EgoMinusEgo* network of each individual node.

4 The Algorithm

In this section we informally describe our proposed solution to the community discovery problem. *DEMON* cycles over each individual node, it applies the $EgoMinusEgo(v, \mathcal{G})$ operation (we cannot simply extract the ego network, because the ego node is directly linked to all nodes leading to noise). The next step is to compute the communities contained in $EgoMinusEgo(v, \mathcal{G})$. We chose to perform this step by using a community discovery algorithm borrowed from the literature: the Label Propagation (LP) algorithm [8]. This choice has been made for the following reasons: (1) LP shares with this work the definition of what is a community; (2) LP is known as the least complex algorithm in the literature; (3) LP will return results of a quality comparable to more complex algorithms [2]. Reason #2 is particularly important, since this step needs to be performed once for every node of the network and we cannot spend a superlinear time for each node at this stage, if we want to scale up to millions of nodes.

The result of this step of the algorithm is a set of the local communities, from node v point of view. These local communities are the node social identity, and they are then used to get a global perspective of the network, not of the single node per se. These communities are likely to be incomplete and should be used to enrich what *DEMON* already discovered so far. Thus, the next step is to merge each local community of $\mathcal{C}(v)$ into the result set $\mathcal{C}$. The *Merge* operation is here defined:

$$Merge(c, \mathcal{C}) = \begin{cases} \mathcal{C} & \exists c' \in \mathcal{C} : c \subseteq c' \\ \{c\} \cup \{c' \in \mathcal{C} \mid c' \nsubseteq c\} & \text{otherwise} \end{cases}$$

In other words, the community c is added to the collection $\mathcal{C}$ only if it is not covered by any community already in $\mathcal{C}$; in this case, all communities in $\mathcal{C}$ covered by c, if any, are removed. This procedure guarantees that the following property holds.

Property 1. **Incrementality.** Given a graph $\mathcal{G}$, an initial set of communities $\mathcal{C}$ and an incremental update $\Delta\mathcal{G}$ consisting of new nodes and new edges added to $\mathcal{G}$, where $\Delta\mathcal{G}$ contains the entire ego networks of all new nodes and of all the preexisting nodes reached by new links, then

$$DEMON(\mathcal{G} \cup \Delta\mathcal{G}, \mathcal{C}) = DEMON(\Delta\mathcal{G}, DEMON(\mathcal{G}, \mathcal{C})) \qquad (1)$$

This is a consequence of the fact that only the local communities of nodes in $\mathcal{G}$ affected by new links need to be reexamined, so we can run *DEMON* on $\Delta\mathcal{G}$ only, avoiding to run it from scratch on $\mathcal{G} \cup \Delta\mathcal{G}$.

Intuitively, *DEMON* algorithm presents also the properties of determinacy, order insensitivity and compositionality, but we leave the proof of there properties for future work. The incrementality property entails that *DEMON* can efficiently run in a streamed fashion, considering incremental updates of the graph as they arrive in subsequent batches; essentially, incrementality means that it is not necessary to run *DEMON* from scratch as batches of new nodes and new links arrive: the new communities can be found by considering only the ego networks of the nodes affected by the updates (both new nodes and old nodes reached by new links.)

As for the time complexity of *DEMON*, we note that it is based on the Label Propagation algorithm, whose complexity is $\mathcal{O}(n+m)$ [8], where n is the number of nodes and m is the number of edges. LP is performed once for each node, thus a rough estimate of worst case complexity of *DEMON* is $\mathcal{O}(n^2 + nm)$. However, LP is not applied to the network as a whole, but only to small ego networks: this bound would be tight if and only if each node has n neighbors, i.e., the entire graph is a clique. A better estimate is $\mathcal{O}(n\bar{k}^2)$, where $\bar{k}$ is the average degree of the network, since it is expected that each ego network will contain $\bar{k} - 1$ nodes and $(\bar{k} - 1)^2$ edges on the average worst case.

5 Experiments

We now present our experimental findings. We considered two networks, a general overview about the statistics of these networks can be found in Table 1. The two networks are:

Table 1. Basic statistics of the studied networks. $|V|$ is the number of nodes, $|E|$ is the number of edges and $\bar{k}$ is the average degree of the network.

Network	$\|V\|$	$\|E\|$	$\bar{k}$
Congress	526	14,198	53.98
IMDb	56,542	185,347	6.55

Congress[2], the network of US representatives of the House and the Senate during the 111st US congress (2009-2011): the bills are usually co-sponsored by many politicians and we connected politicians if they have at least 75 co-sponsorships (deleting the connections created only by bills with more than 10 co-sponsors). The set of subjects a politician frequently worked on is the *qualitative attribute* of this network, the *quantitative attributes* are derived from the size of the set of subjects of each politician. **IMDb**[3], the network of actors connected if they appear together in at least two movies from 2001 to 2010. The *qualitative attributes* are the user assigned keywords summarizing the movies each actor has been part of. The total number of movies in which an actor appeared is instead the basis of the *quantitative attributes*. In both networks we expect to find a rich overlap being politicians usually involved in many different topics and actors present in many different crews.

We now evaluate the quality of a set of communities discovered in these datasets, by performing a direct comparison between the discovered communities and the qualitative and quantitative attributes attached to the nodes. We then evaluate our algorithm against two state-of-the-art community discovery methods using the proposed quality measures, namely the Hierarchical Link Clustering [1] as state-of-the-art for overlapping; and Infomap [9], as state-of-the-art of non-overlapping algorithms. Besides output quality, we also compare computational performances. Finally, we present some examples of web related knowledge that we are able to extract with *DEMON* algorithm. The experiments were performed on a Dual Core Intel i7 64 bits @ 2.8 GHz, equipped with 8 GB of RAM and with a kernel Linux 2.6.35-22-generic.

We test the quality of the community sets returned by each algorithm against four main quality functions. The chosen functions do not consider only the network structure, like Modularity [7]. By looking only at the topology of the network we cannot evaluate communities according to their meaning in the real world: using a semantics-based set of quality measures is particularly critical in the case of social and web networks. The evaluation measures used in this paper are directly inherited from [1]. We present their formulation and we point to that reference for further details. A key concept of these evaluation measures is the definition of a non-trivial community, i.e. a community with at least three nodes.

Community Quality. The networks studied here possess *qualitative attributes* that attaches a small set of annotations or tags to each node. We state that "similar" nodes share more *qualitative attributes* than dissimilar nodes, quantifying this intuition by evaluating how much higher are on average the Jaccard coefficient of the set of *qualitative attributes* for pair of nodes inside the communities over the average of the entire network.

Overlap Quality. We can use the *quantitative attributes* to quantify how many different communities a node should be part of. We define the Overlap Quality as follows: $OQ(X; MD) = \sum_{y \in MD} \sum_{x \in X} p(x, y) \log \frac{p(x, y)}{p(x)\, p(y)}$, where

[2] Data downloaded from `http://govtrack.us/`

[3] Data downloaded from `http://www.imdb.com/`

X is the vector that assigns to each node the number of nontrivial communities extracted by the algorithm, MD is the vector of the quantitative attributes, and $p(x, y)$ is the probability of each variable value co-occurrence.

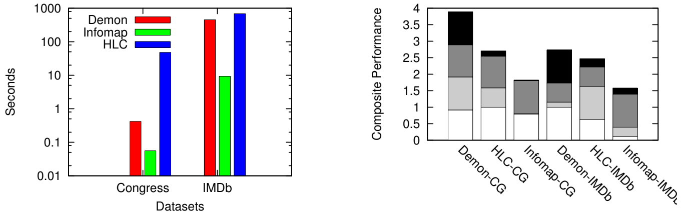
Community Coverage. To measure community coverage, we simply count the fraction of nodes that belong to nontrivial community.

Overlap Coverage. We count the average number of memberships in nontrivial communities for each node. This measure shows how much information is extracted from that portion of the network that the particular algorithm was able to analyze.

5.1 Evaluation

In Figure 2a we report the runtimes of the tested algorithms on the networks we analyzed. One important caveat regards the Infomap, that is very dependent on random walks and greedy heuristics, therefore Infomap needs to be performed several times to get reasonable results. In Figure 2 we report the cost of one single iteration, but for Infomap the experiments need to run at least 50 or 100 iterations, making the total running time of Infomap in the same order of magnitude of *DEMON*. Also, Infomap is not incremental as *DEMON*. On the other hand, in general HLC is 2x to 100x slower than *DEMON*.

In Figure 2b we report the scores of the combination algorithm-network using the evaluation measures we introduced in this section. We stacked the score of each measure in one single bar and we normalized the values of each measure (since some of them do not take values from 0 to 1): from bottom to top they are Community Quality (white), Overlap Quality (light gray), Community Coverage (dark gray) and Overlap Coverage (black). The normalization was done by dividing each score of the algorithm by the maximum score registered among *DEMON*, HLC and Infomap. *DEMON* algorithm performs better according to the combined score over both datasets. In general, Community Quality is always very good, while a complete constant is better scores in the Overlap Coverage. Also the Overlap Quality is generally better except in IMDb. Overlap is a crucial aspect of real world data. From this analysis we can conclude that not only



(a) The runtimes of the tested algorithms on our networks.

(b) The composite performances of the tested algorithms on our networks.

Fig. 2. Comparison between *DEMON*, HLC and Infomap

DEMON is in general a better algorithm, but it is especially the optimal choice if we are particular interested in a better description of the overlapping and complex reality.

5.2 Case Study

In this Section we present a brief case study using the communities extracted from the IMDb network. In Figure 3 left we represent one chosen community, IMDb #276. All actors are related with the Star Wars saga. Using the power of overlapping we can jump from this community to the related communities, i.e. the other communities of its members.

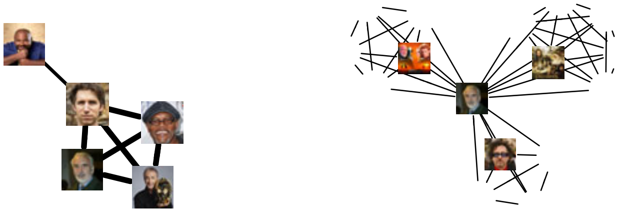


Fig. 3. Representations of some communities extracted from the IMDb network

In Figure 3 right we chose to visualize some of the surrounding communities of the British actor Christopher Lee: he is part also of a wider and more complete community regarding the whole new Star Wars trilogy, besides the communities regarding the trilogy of Lord of the Rings. Another interesting galaxy of communities is composed by different groups of actors that usually work together with director Tim Burton. The picture we get from Figure 3 right is a decent summary of the movie acting career of sir Christopher Lee for the past decade. If we have reliable data about criminal activities, collaborations and affiliations, a similar set of results can be obtained to profile criminals and to obtain a similarity measure for criminal groups, therefore providing a fundamental tool to law enforcement and crime prevention.

6 Conclusion and Future Works

In this paper we proposed a new method for solving the problem of finding significant groups in complex networks, aiming at real world applications such as crime prevention. We propose a democratic approach, where the peer nodes judge where their neighbors should be clustered together. This approach is fast and incremental. We have shown in the experimental section that this method allows a discovery of communities in different real world networks: the quality of the overlapping partition is improved w.r.t state-of-the-art algorithms.

Many lines of research remain open for future work. We want to prove other fundamental properties of our algorithm, namely determinacy, order insensitivity and compositionality: these properties can be exploited to apply *DEMON* on huge networks. A more comprehensive experimental section, with more and larger datasets, is also in the roadmap to test *DEMON* in a wider set of domains.

Acknowledgments. Michele Coscia is a recipient of the Google Europe Fellowship in Social Computing, and this research is supported in part by this Google Fellowship.

References

1. Ahn, Y.-Y., Bagrow, J.P., Lehmann, S.: Link communities reveal multiscale complexity in networks. Nature 466(7307), 761–764 (2010)
2. Coscia, M., Giannotti, F., Pedreschi, D.: A classification for community discovery methods in complex networks. SAM 4(5), 512–546 (2011)
3. Derényi, I., Palla, G., Vicsek, T.: Clique Percolation in Random Networks. Physical Review Letters 94(16), 160202 (2005)
4. Fortunato, S., Barthélemy, M.: Resolution limit in community detection. Proceedings of the National Academy of Sciences 104(1), 36–41 (2007)
5. Goyal, A., On, B.-W., Bonchi, F., Lakshmanan, L.V.S.: Gurumine: A pattern mining system for discovering leaders and tribes. In: International Conference on Data Engineering, pp. 1471–1474 (2009)
6. Henderson, K., Eliassi-Rad, T., Papadimitriou, S., Faloutsos, C.: Hcdf: A hybrid community discovery framework. In: SDM, pp. 754–765 (2010)
7. Newman, M.E.J.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences 103(23), 8577–8582 (2006)
8. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Physical Review E (2007)
9. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. PNAS 105(4), 1118–1123 (2008)
10. Shen, H.-W., Cheng, X.-Q., Guo, J.-F.: Quantifying and identifying the overlapping community structure in networks. J. Stat. Mech. (2009)
11. Yonas, M.A., Borrebach, J.D., Burke, J.G., Brown, S.T., Philp, K.D., Burke, D.S., Grefenstette, J.J.: Dynamic Simulation of Community Crime and Crime-Reporting Behavior. In: Salerno, J., Yang, S.J., Nau, D., Chai, S.-K. (eds.) SBP 2011. LNCS, vol. 6589, pp. 97–104. Springer, Heidelberg (2011)