Node-Centric Detection of Overlapping Communities in Social Networks

Yehonatan Cohen, Danny Hendler, Amir Rubin
Computer Science Department, Ben-Gurion University of the Negev, Be'er-Sheva, Israel
{yehonatc,hendlerd,amirrub}@cs.bgu.ac.il

***Abstract*—We present NECTAR, a community detection algorithm that generalizes Louvain method's local search heuristic for overlapping community structures. NECTAR chooses dynamically which objective function to optimize based on the network on which it is invoked. Our experimental evaluation on both synthetic benchmark graphs and real-world networks, based on ground-truth communities, shows that NECTAR provides excellent results as compared with state of the art community detection algorithms.**

***Index Terms*—Community detection, overlapping communities, objective function, modularity, Louvain method.**

I. Introduction

Revealing the community structure [1] underlying complex networks in general, and social networks in particular, is a key problem with many applications that is the focus of intense research. While research focus was initially on detecting *disjoint communities*, in recent years there is growing interest in the detection of *overlapping communities*, where a node may belong to several communities.

Many community detection algorithms are guided by an *objective function* that provides a quality measure of the clusterings they examine in the course of their execution. A key example is Blondel et al.'s widely-used algorithm [2], also known by the name "Louvain method" (LM). It aims to maximize the modularity objective function [3]. Underlying the algorithm is a greedy local search heuristic that iterates over all nodes, assigning each node to the community it fits most (as quantified by modularity) and seeking a local optimum. Unfortunately, the applicability of LM is limited to disjoint community detection. In this work we show that LM's simple local search heuristic can be generalized in a natural manner to obtain a highly effective detector for overlapping communities.

We present NECTAR, a Node-centric ovErlapping Community deTection AlgoRithm. NECTAR generalizes the node-centric local search heuristic of the Louvain method [2] so that it can be applied also for networks possessing overlapping community structure.

Modularity (used by LM) assumes disjoint communities. Which objective functions should be used for overlapping community detection? Yang and Leskovec [4] evaluated several objective functions and showed that which is most appropriate depends on the network at hand. They observe that objective functions that are based on triadic closure provide the best results when there is significant overlap between communities. Weighted Community Clustering (WCC) [5] is such an objective function but is defined only for disjoint community structures.

We present Weighted Overlapping Community Clustering (WOCC), a generalization of WCC that may be applied for overlapping community detection. Another objective function that fits the overlapping setting is Q^E - an extension of modularity for overlapping communities [6] that, as indicated by the results of our experiments, is more adequate for graphs with relatively small inter-community overlap.

A unique feature of NECTAR is that it chooses dynamically whether to use WOCC or Q^E, depending on the structure of the graph at hand. This allows NECTAR to provide good results on graphs with both high and low community overlaps. To the best of our knowledge, NECTAR is the first community-detection algorithm that selects dynamically which objective function to use based on the graph on which it is invoked. NECTAR is *node-centric*, as it employs a heuristic that iteratively considers each node and adds it to those communities that are "best" in terms of the objective function.

II. NECTAR: Detailed Description

The high-level pseudo-code of NECTAR is given by Algorithm 1. Its input is a graph $G =< V, E >$ and an algorithm parameter $\beta \geq 1$ used to determined the number of communities to which a node should belong.

`NECTAR` proceeds in *external iterations* (lines **12–28**). In each external iteration, the algorithm performs *internal iterations*, in which it iterates over all nodes $v \in V$ (in some random order), attempting to determine the set of communities to which node v belongs such that the objective function is maximized. NECTAR selects dynamically whether to use WOCC or Q^E, depending on the rate of closed triangles in the graph on which it is invoked. (lines **5–8**). We use *trRate* $= 5$, as this provides a good separation between communities with high overlap (on which WOCC is superior) and low overlap (on which extended modularity is superior).

In each internal iteration (lines **14–23**), `NECTAR` first computes the set C_v of communities to which node v currently

Partially supported by the Lynne and William Frankel Center for Computing Science at Ben-Gurion University.
IEEE/ACM ASONAM 2016, August 18-21 2016, San Francisco, CA, USA
978-1-5090-2846-7/16/$31.00 ©2016 IEEE

Figure 1: NECTAR algorithm pseudo-code.

```
1  const maxIter ← 20                                   /* max iterations */
2  const α ← 0.8                                        /* merge threshold */
3  const trRate ← 5                                     /* WOCC threshold */
4  Procedure NECTAR(G=<V,E>, β){
5  if triangles(G)/|V| ≥ trRate then
6  |   use WOCC                                   /* use WOCC obj. function */
7  else
8  |   use Q^E                                    /* use Q^E obj. function */
9  end
10 Initialize communities
11 i ← 0                                     /* number of extern. iterations */
12 repeat
13 |   s ← 0                                       /* number of stable nodes */
14 |   forall v ∈ V do
15 |   |   C_v ← communities to which v belongs
16 |   |   Remove v from all the communities of C_v
17 |   |   S_v ← {C ∈ 𝒞 | ∃u : u ∈ C ∧ (v, u) ∈ E}
18 |   |   D ← {Δ(v, C) | C ∈ S_v}
19 |   |   C'_v ← {C ∈ S_v | Δ(v, C) · β ≥ max(D)}
20 |   |   Add v to all the communities of C'_v
21 |   |   if C'_v = C_v then
22 |   |   |   s++
23 |   end
24 |   merge(α)                                       /* merge communities */
25 |   if merge reduced number of communities then
26 |   |   s ← 0
27 |   i++
28 until (s = |V|) ∨ (i = maxIter)
```

belongs (line 15). Then, v is removed from all these communities (line 16). Next, the set S_v of v's neighboring communities is computed in line 17. Then, the gain in the objective function value that would result from adding v to each neighboring community (relative to the current set of communities $\mathcal{C}$) is computed in line 18. Node v is then added to the community maximizing the gain in objective function and to any community for which the gain is at least a fraction of $1/\beta$ of that maximum (lines 19–20).[1] Thus, the number of communities to which a node belongs may change dynamically throughout the computation, as does the set of communities $\mathcal{C}$.

If the internal iteration did not change the set of communities to which v belongs, then v is a *stable node* of the current external iteration and the number of stable nodes (which is initialized to 0 in line 13) is incremented (lines 21–22). After all nodes have been considered, the possibly-new set of communities is checked in order to prevent the emergence of different communities that are too similar to one another. This is accomplished by the `merge` procedure (whose code is not shown), called in line 24. It receives as its single parameter a value α and merges any two communities whose relative overlap is α or more. We use $\alpha = 0.8$, as this is the value that gave the best results (line 2). If the number of communities was reduced by `merge`, the counter of stable nodes is reset to 0 (lines 25–26).

The computation proceeds until either the last external iteration does not cause any changes (hence the number of stable nodes equals $|V|$) or until the maximum number of iterations is reached (line 28), whichever occurs first.

III. Experimental Evaluation

We conducted extensive competitive analysis of NECTAR (using a node-centric approach) and six other state-of-the-art overlapping community detection algorithms. Our evaluation was done using both synthetic graphs and real-world networks with ground-truth communities. We quantify the quality of the cover computed by the algorithms by employing three widely-used measures: *Normalized Mutual Information* (NMI) [7], *Omega-index* [8], and *Average F1 score* [9].

NECTAR outperformed all other algorithms in terms of average detection quality and was best or second-best for almost all networks. For lack of space, we only present here the evaluation results for Amazon's product co-purchasing network. For more details, please refer to our full paper [10].

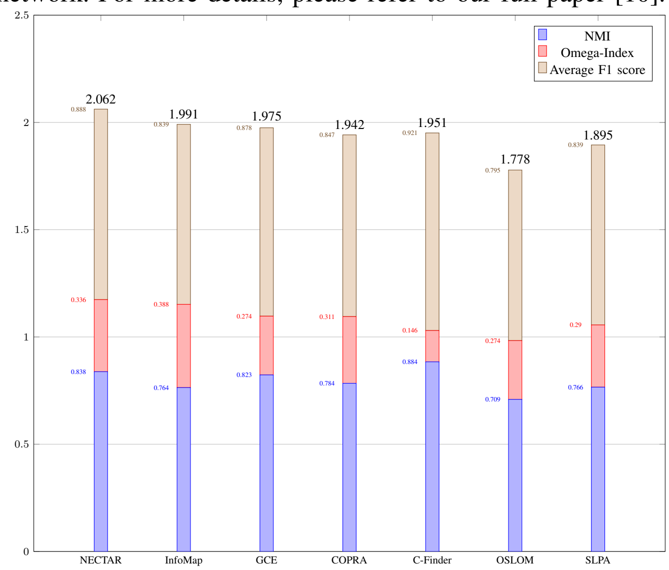

Fig. 1: Amazon competitive analysis

References

[1] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3, pp. 75–174, 2010.

[2] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[3] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[4] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *Knowledge and Information Systems*, vol. 42, no. 1, pp. 181–213, 2015.

[5] A. Prat-Pérez, D. Dominguez-Sal, J. M. Brunat, and J.-L. Larriba-Pey, "Shaping communities out of triangles," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 1677–1681.

[6] M. Chen, K. Kuzmin, and B. K. Szymanski, "Extension of modularity density for overlapping community structure," in *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*. IEEE, 2014, pp. 856–863.

[7] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, vol. 11, no. 3, p. 033015, 2009.

[8] L. M. Collins and C. W. Dent, "Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions," *Multivariate Behavioral Research*, vol. 23, no. 2, pp. 231–242, 1988.

[9] J. Yang and J. Leskovec, "Community-affiliation graph model for overlapping network community detection," in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE, 2012, pp. 1170–1175.

[10] Y. Cohen, D. Hendler, and A. Rubin, "Node-Centric Detection of Overlapping Communities in Social Networks," *ArXiv e-prints*, Jul. 2016.

[1] If no gain is positive, v remains as a singleton.