Chapter 1
Node-Centric Detection of Overlapping Communities in Social Networks

Yehonatan Cohen, Danny Hendler, and Amir Rubin

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.

1.1 Introduction

Social networks tend to exhibit community structure [1], that is, they may be partitioned to sets of nodes called *communities* (a.k.a. *clusters*), each of which relatively densely-interconnected, with relatively few connections between different communities. Revealing the community structure underlying complex networks in general, and social networks in particular, is a key problem with many applications (see e.g. [2, 3]) that is the focus of intense research. Numerous community detection algorithms were proposed (see e.g. [4–14]). 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. Since exhaustive-search optimization of these functions is generally intractable (see e.g. [15, 16]), existing methods settle for an approximation of the optimum and employ heuristic search strategies.

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

A key example is Blondel et al.'s algorithm [8], also known by the name "Louvain method" (LM). The algorithm is fast and relatively simple to understand and use and has been successfully applied for detecting communities in numerous networks. It aims to maximize the modularity objective function [9]. Underlying it 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.

Our Contributions: We present NECTAR, a Node-centric ovErlapping Community deTection AlgoRithm. NECTAR generalizes the node-centric local search heuristic of the Louvain algorithm so that it can be applied also to networks possessing overlapping community structure. Several algorithmic issues have to be dealt with in order to allow the LM heuristic to support multiple community-memberships per node. First, rather than adding a node v to the *single* community maximizing an objective function, v may have to be added to several such communities. However, since the "correct" number of communities to which v should belong is not a-priori known to the algorithm, it must be chosen dynamically.

A second issue that arises from multiple community-memberships is that different communities with large overlaps may emerge during the algorithm's execution and must be merged. We describe the new algorithm and how it resolves these issues in Section 1.2.

Modularity (used by LM) assumes disjoint communities. Which objective functions should be used for overlapping community detection? Yang and Leskovec [17] 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) [18] is such an objective function but is defined only for disjoint community structures.

We define Weighted Overlapping Community Clustering (WOCC), a generalization of WCC that may be applied for overlapping community detection. More details can be found in our technical report [19]. Another objective function that fits the overlapping setting is Q^E ([20]) - an extension of modularity for overlapping communities.

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 it to provide good results on graphs with both high and low community overlaps. NECTAR is the first community-detection algorithm that selects dynamically which objective function to use based on the graph on which it is invoked.

Local search heuristics guided by an objective function may be categorized as either *node-centric* or *community-centric*. Node-centric heuristics iterate over nodes. For each node, communities are considered and it is added to those of them that are "best" in terms of the objective function. Community-centric heuristics do the opposite: they iterate over communities. For each community, nodes are considered and the "best" nodes are added to it. In order to investigate which of these approaches is superior in the context of social networks, we implemented both a node-centric and a community-centric versions of NECTAR and compared the two implementations using both the WOCC and the Q^E metrics. As can be seen in our technical report [19], the node-centric approach was significantly superior for both metrics used.

We conducted extensive competitive analysis of NECTAR (using a node-centric approach) and nine 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, based on several commonly-used metrics. NECTAR outperformed all other algorithms in terms of average detection quality and was best or second-best for almost all networks. Our code is publicly available for download.[1]

Background: We now briefly describe a few key notions directly related to our work. Louvain method [8] is a widely-used disjoint community detection algorithm, based on a simple node-centric search heuristic that seeks to maximize the *modularity* [9] objective function. Chen et al. extended the definition of modularity to the overlapping setting [20]. For a collection of sets of nodes $\mathscr{C}$, their *extended modularity* definition, denoted $Q^E(\mathscr{C})$, is given by:

$$Q^E(\mathscr{C}) = \frac{1}{2|E|} \sum_{C \in \mathscr{C}} \sum_{i,j \in C} \left[A_{ij} - \frac{k_i k_j}{2|E|} \right] \frac{1}{O_i O_j}, \tag{1.1}$$

where A is the adjacency matrix, k_i is the degree of node i, and O_i is the number of communities i is a member of. If $\mathscr{C}$ is a partition of network nodes, Q^E reduces to (regular) modularity.

Yang and Leskovec [17] conducted a comparative analysis of 13 objective functions in order to determine which captures better the community structure of a network. They show that which function is best depends on the network at hand. They also 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) [18] is such an objective function. It is based on the observation that triangle structures are much more likely to exist within communities than across them. This observation is leveraged for quantifying the quality of graph partitions (that is, non-overlapping communities). It is formally defined as follows. For a set of nodes S and a node v, let $t(v,S)$ denote the number of triangles that v closes with nodes of S. Also, let $vt(v,S)$ denote the number of nodes of S that form at least one triangle with v. $WCC(v,S)$, quantifying the extent by which v should be a member of S, is defined as:

$$WCC(v,S) = \begin{cases} \frac{t(v,S)}{t(v,V)} \cdot \frac{vt(v,V)}{|S \setminus v| + vt(v,V \setminus S)} & \text{if } t(v,V) > 0 \\ 0 & \text{otherwise,} \end{cases}$$

[1] NECTAR code and documentation may be downloaded from: https://github.com/amirubin87/NECTAR.

where V is the set of graph nodes. The cohesion level of a community S is defined as $WCC(S) = \frac{1}{|S|}\sum_{v\in S} WCC(v,S)$. Finally, the quality of a partition $\mathscr{C} = \{S_1,\ldots,S_k\}$ is defined as the following weighted average: $WCC(\mathscr{C}) = \frac{1}{|V|}\sum_{i=1}^{k}|S_i|\cdot WCC(S_i)$.

NECTAR uses Weighted Overlapping Community Clustering (WOCC) - our generalization of WCC that can be applied to overlapping community detection.

1.2 NECTAR: a Detailed Description

The high-level pseudo-code of NECTAR is given by Algorithm 1. The input to the `NECTAR` procedure (see line 4) is a graph $G =< V,E >$ and an algorithm parameter $\beta \geq 1$ that is used to determined the number of communities to which a node should belong in a dynamic manner (as we describe below).

`NECTAR` proceeds in iterations (lines 12–27), which we call *external iterations*. 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.

We implemented two overlapping community objective functions: the extended modularity function [20], denoted $Q^E(\mathscr{C})$, and WOCC - our generalization of the WCC function [18]. These implementations are described in our technical report [19,21]. 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. If the average number of closed triangles per node in G is above the *trRate* threshold, then WOCC is more likely to yield good performance and it is used, otherwise the extended modularity objective function is used instead (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).

Each internal iteration (comprising lines 13–22) proceeds as follows. First, `NECTAR` computes the set C_v of communities to which node v currently belongs (line 14). Then, v is removed from all these communities (line 15). Next, the set S_v of v's neighboring communities (that is, the communities of $\mathscr{C}$ that contain one or more neighbors of v) is computed in line 16. 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 $\mathscr{C}$) is computed in line 17. 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 18–19).[2] Thus, the number of communities to which a node belongs may change dynamically throughout the computation, as does the set of communities $\mathscr{C}$.

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 (initialized to 0 in line 13) is incremented (lines 20–21).

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

Figure 1: NECTAR algorithm.

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

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 done by the `merge` procedure (whose code is not shown), called in line 23. It receives as its single parameter a value α and merges any two communities whose relative overlap is α or more. If the number of communities was reduced by `merge`, the counter of stable nodes is reset to 0 (lines 24–25).

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 27), whichever occurs first. We have set the maximum number of iterations to 20 (line 1) in order to strike a good balance between detection quality and runtime. In practice, the algorithm converges within a fewer number of iterations in the vast majority of cases. For example, in our experiments on synthetic graphs with 5000 nodes, NECTAR converges after at most 20 iterations in 99.5% of the executions.

LM is a hierarchical clustering algorithm that has a second phase.We implemented a hierarchical version of NECTAR. However, since in all our experiments the best results were obtained in the first hierarchy level, we only describe the non-hierarchical version of NECTAR (Algorithm 1).

1.3 Experimental Evaluation

Xie et al. [22] conducted a comparative study of state-of-the-art overlapping community detection algorithms. We compare NECTAR with the following 5 of the key performers out of the 14 algorithms they evaluated: the *Greedy Clique Expansion* (GCE) algorithm [23], the *Cfinder* algorithm [12], the *Order Statistics Local Optimization Method* (OSLOM) [13], the *Community Overlap PRopagation Algorithm* (COPRA) [24], and the *Speaker-Listener Label Propagation Algorithm*

(SLPA) [10]. In addition, we also evaluate the following four algorithms: *Fuzzy-Infomap* [14], *Big-Clam* [25], *Link-Clustering* (LC) [26], and *DEMON* [27]. Details regarding these algorithms and the parameters we used when invoking them can be found in our technical report [19, 21].

We conducted competitive analysis using both synthetic networks and real-world networks with ground-truth. We evaluated results using the widely-used *Normalized Mutual Information* (NMI) [5], *Omega-index* [28], and *Average F1 score* [29] metrics (descriptions of these metrics can be found in [21]). Our evaluation shows that NECTAR outperformed all other algorithms in terms of average detection quality and provided best or second-best results for almost all networks, as we describe now.

Synthetic Networks: Lancichinetti et al. [30] introduced a set of benchmark graphs (henceforth the LFR benchmark), parameterized on: the number of nodes, n, the average node degree, k, the number of overlapping nodes, O_n, the number of communities an overlapping node belongs to, O_m, community sizes (varied in our experiments between $20-100$ for big communities and between $10-50$ for small communities), and more. We mostly use the LFR parameter values used by [22].[3] We generate 10 instances for each combination of parameters and take the average of the results for each algorithm and each metric over these 10 instances. For each algorithm, we present the results for the algorithm parameter value that maximizes this average.

Figure 1.1 presents the average performance of the four best algorithms in terms of NMI as a function of O_m (the number of communities to which each of the O_n overlapping nodes belongs), for $k \in \{10, 40\}$ and $O_n \in \{2500, 5000\}$. The Omega-index and average-F1 score results follow the same trends and are thus omitted for lack of space. They can be found in our technical report [19].

With only a few exceptions, it can be seen that the performance of the algorithms decreases as O_m increases. This can be attributed to the fact that the size of the solution space increases with O_m.

We focus first on the results on graphs with a higher number of overlapping nodes ($O_m = 2500$) and high average degrees ($k = 40$). The rate of triangles in these graphs is high (approx. 30 on average) and so NECTAR employs WOCC. NECTAR is the clear leader for big communities. It achieves the best results for almost all values of O_m and its relative performance improves as O_m increases, confirming that the combination of NECTAR's search strategy and the WOCC objective function is suitable for graphs with significant overlap. Cfinder improves its relative performance as O_m increases and is the second performer for $O_m \in \{4, 7, 8\}$. For small communities, Cfinder has the lead with NECTAR being second best and OSLOM third for most values of O_m, and NECTAR taking the lead for $O_m = 8$.

We now describe the results on graphs with lower numbers of overlapping nodes ($O_m = 500$) and low average degree ($k = 10$). The rate of triangles in these graphs is low (approx. 3.5 on average) and so NECTAR employs extended modularity. NEC-

[3] For more details on parameter values used for LFR, refer to [21].

TAR provides the best performance for both small and large communities for almost all values of O_m. The relative performance of Cfinder deteriorates as compared with its performance on high-overlap graphs. It is not optimized for sparser graphs, since its search for communities is based on locating cliques. OSLOM is second best on these graphs, having the upper hand for $O_m = 1$ and providing second-best performance for $O_m > 1$. These results highlight the advantage of NECTAR's capability of selecting the objective function it uses dynamically according to the properties of the graph at hand.

Summarizing the results of the tests we conducted on 96 different synthetic graph types, NECTAR is ranked first, with average rank of 1.58, leading in 33 out of 96 of the tests, followed by OSLOM, with average rank of 2.79.

Real-World Networks: We conducted our competitive analysis on two real-world networks - Amazon's product co-purchasing network and the DBLP scientific collaboration network. We downloaded both from Stanford's Large Network Dataset Collection [31]. The Amazon graph consists of $334,863$ nodes and $925,872$ edges. Nodes represent products and edges are between commonly co-purchased products. Products from the same category are viewed as a ground-truth community.

The DBLP graph consists of $317,080$ nodes and $1,049,866$ edges. Nodes correspond to authors and edges connect authors that have co-authored a paper. Publication venues (specifically, conferences) are used for defining ground-truth communities. Thus, the set of authors that have published in the same conference is viewed as a ground-truth community.

In [17], Yang and Leskovec rate the quality of ground-truth communities of Amazon and DBLP (as well as those of additional networks) using six scoring functions, such as modularity, conductance, and cut ratio. They rank ground-truth communities based on the average of their ranks over the six corresponding scores and maintain the $5,000$ top ground-truth communities per each network. These are the ground-truth communities provided as part of the datasets of [31].

The left part of figure 1.2 presents the results of the seven best algorithms on Amazon. The right part refers to results on DBLP. The rate of triangles in the Amazon graph is low, and so NECTAR employs extended modularity. NECTAR provides the best performance with an overall score of 2.062, approximately 3.5% more than InfoMap, which is second best. NECTAR has second-best average F1 score, lagging only slightly behind Cfinder. In terms of Omega-index, NECTAR is second-best as well, lagging behind InfoMap, and Cfinder is the last performer.

In the DBLP network, the rate of triangles is high, and so NECTAR employs WOCC. Cfinder has the highest overall score, enjoying a small margin of approximately 2.5% w.r.t. NECTAR, which is second-best. LC is the third performer, with a score lower than NECTAR's by approximately 8%. In terms of NMI, Cfinder is first with a score of 0.657 and NECTAR is third best, lagging behind by approximately 5.5%. NECTAR has the highest average F1 score, but Cfinder's score is only approximately 1% smaller. COPRA obtains the third score, nearly 17% less than NECTAR's. All algorithms fair poorly in terms of their Omega-index.

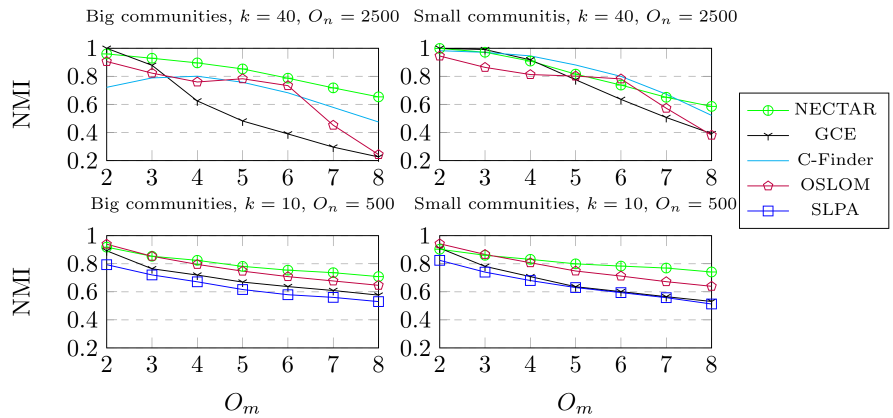

Fig. 1.1: Four best performers over synthetic networks in terms of NMI

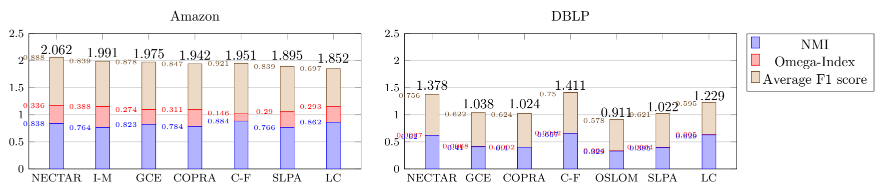

Fig. 1.2: Seven best performers over real-world networks

In order to assess the impact of dynamic objective function selection, we compared NECTAR with two variants that consistently used either Q^E or WOCC. In cases of disagreement, NECTAR's score was, on average, 30% higher than that of the WOCC version and 13% higher than the Q^E version.

We also measured time complexity on numerous networks, while varying the number of nodes and the average node degree. NECTAR's average running time was second best among all evaluated algorithms.

1.4 Conclusions

We introduced NECTAR, a novel overlapping community detection algorithm that generalizes Louvain's search heuristic and selects dynamically which objective function to optimize, depending on the structure of the graph at hand. Our evaluation shows that NECTAR outperforms all other algorithms in terms of average detection quality. Analysis of our empirical results shows that extended modularity yields better results on networks with low average node degrees and low community overlap, whereas WOCC yields better results on networks with higher degrees and overlap. The fact that NECTAR is able to provide excellent results on both types of networks highlights the importance of objective function dynamic selection, as well as the general applicability of Louvain's search heuristic.

References

1. S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3, pp. 75–174, 2010.
2. N. J. Krogan, G. Cagney, H. Yu, G. Zhong, X. Guo, A. Ignatchenko, J. Li, S. Pu, N. Datta, A. P. Tikuisis *et al.*, "Global landscape of protein complexes in the yeast saccharomyces cerevisiae," *Nature*, vol. 440, no. 7084, pp. 637–643, 2006.
3. G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee, "Self-organization and identification of web communities," *Computer*, vol. 35, no. 3, pp. 66–70, 2002.
4. E. Le Martelot and C. Hankin, "Fast multi-scale detection of relevant communities in large-scale networks," *The Computer Journal*, p. bxt002, 2013.
5. 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.
6. A. Prat-Pérez, D. Dominguez-Sal, and J.-L. Larriba-Pey, "High quality, scalable and parallel community detection for large real graphs," in *Proceedings of the 23rd international conference on World wide web*. ACM, 2014, pp. 225–236.
7. A. V. Esquivel and M. Rosvall, "Compression of flow can reveal overlapping-module organization in networks," *Physical Review X*, vol. 1, no. 2, p. 021025, 2011.
8. 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.
9. M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.
10. J. Xie and B. K. Szymanski, "Towards linear time overlapping community detection in social networks," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2012, pp. 25–36.
11. S. Gregory, "Finding overlapping communities in networks by label propagation," *New Journal of Physics*, vol. 12, no. 10, p. 103018, 2010.
12. B. Adamcsek, G. Palla, I. J. Farkas, I. Derényi, and T. Vicsek, "Cfinder: locating cliques and overlapping modules in biological networks," *Bioinformatics*, vol. 22, no. 8, pp. 1021–1023, 2006.
13. A. Lancichinetti, F. Radicchi, J. J. Ramasco, S. Fortunato *et al.*, "Finding statistically significant communities in networks," *PloS one*, vol. 6, no. 4, p. e18961, 2011.
14. S. Gregory, "Fuzzy overlapping communities in networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2011, no. 02, p. P02017, 2011.
15. U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner, "On finding graph clusterings with maximum modularity," in *Graph-Theoretic Concepts in Computer Science, 33rd International Workshop, WG 2007, Dornburg, Germany, June 21-23, 2007. Revised Papers*, 2007, pp. 121–132.
16. J. Síma and S. E. Schaeffer, "On the np-completeness of some graph cluster measures," in *SOFSEM 2006: Theory and Practice of Computer Science, 32nd Conference on Current Trends in Theory and Practice of Computer Science, Merín, Czech Republic, January 21-27, 2006, Proceedings*, 2006, pp. 530–537.
17. 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.
18. 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.
19. D. H. Yehonatan Cohen and A. Rubin, "NECTAR - Technical Report," https://github.com/amirubin87/NECTAR/blob/master/Technical-Report-NECTAR.pdf, 2016.
20. 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.

21. Y. Cohen, D. Hendler, and A. Rubin, “Node-centric detection of overlapping communities in social networks,” *arXiv preprint arXiv:1607.01683*, 2016.
22. J. Xie, S. Kelley, and B. K. Szymanski, “Overlapping community detection in networks: The state-of-the-art and comparative study,” *ACM Computing Surveys (csur)*, vol. 45, no. 4, p. 43, 2013.
23. C. Lee, F. Reid, A. McDaid, and N. Hurley, “Detecting highly overlapping community structure by greedy clique expansion,” *arXiv preprint arXiv:1002.1827*, 2010.
24. U. N. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” *Physical Review E*, vol. 76, no. 3, p. 036106, 2007.
25. J. Yang and J. Leskovec, “Overlapping community detection at scale: a nonnegative matrix factorization approach,” in *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 2013, pp. 587–596.
26. Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, “Link communities reveal multiscale complexity in networks,” *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.
27. M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi, “Demon: a local-first discovery method for overlapping communities,” in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 615–623.
28. 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.
29. 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.
30. A. Lancichinetti, S. Fortunato, and F. Radicchi, “Benchmark graphs for testing community detection algorithms,” *Physical review E*, vol. 78, no. 4, p. 046110, 2008.
31. J. Leskovec and A. Krevl, “SNAP Datasets: Stanford large network dataset collection,” http://snap.stanford.edu/data, Jun. 2014.