Scalable Link Community Detection: A Local Dispersion-aware Approach

Panagiotis Liakos
University of Athens
Athens 15703, Greece
Email: p.liakos@di.uoa.gr

Alexandros Ntoulas*
Samsung Research America
Mountain View, CA 94043
Email: ntoulas@gmail.com

Alex Delis
University of Athens
Athens 15703, Greece
Email: ad@di.uoa.gr

***Abstract*—Real-life systems involving interacting objects are typically modeled as graphs and can often grow very large in size. Revealing the community structure of such systems is crucial in helping us better understand their complex nature. However, the ever-increasing size of real-world graphs, and our evolving perception of what a community is, make the task of community detection very challenging. One such challenge, is the discovery of the possibly overlapping communities of a given node in a billion-node graph. This problem is very common in modern large social networks like Facebook and LinkedIn. In this paper, we propose a scalable local community detection approach to efficiently unfold the communities of individual target nodes in a given network. Our goal is to reveal the groupings formed around nodes (e.g., users) by leveraging the relations of the different contexts the nodes participate in. Our algorithm, termed Local Dispersion-aware Link Communities or LDLC, measures the similarity of pairs of links in the graph as well as the extent of their participation in multiple contexts. Then, it determines the ordering that we should group the links in order to form communities. Our approach is not affected by constraints existent in previous techniques (e.g., the need for several seed nodes or the need to collapse multiple overlapping communities to one). Our experimental evaluation using ground-truth communities for a wide range of large real-world networks show that LDLC significantly outperforms state-of-the-art methods on both accuracy and efficiency.**

I. Introduction

Networks are a powerful tool for modeling relations and interactions of entities in the real world. Real-world networks are continuously growing and are often massive; yet they exhibit a high level of order and organization, which allows the study of properties such as the power-law degree distribution and the small-world structure [6], [8]. Another important characteristic of networks is the presence of community structures [12]. At a high level, communities are groups of nodes that share a common functional property or context, e.g., two people that attended the same school, or two movies with the same actor. In several cases, communities in a network are distinct, e.g., *Bulls* vs. *Knicks* fans. However, it is often the case that communities overlap. Figure 1 illustrates the communities of an individual in a social network, i.e., her family, co-workers, basketball buddies and friends from college. It is obvious that the communities may overlap in different ways. For example, a co-worker may also be a basketball buddy and a friend from college. Such overlapping communities may have a complex structure of connections that are not easy to discern and are more challenging to identify compared to non-overlapping ones.

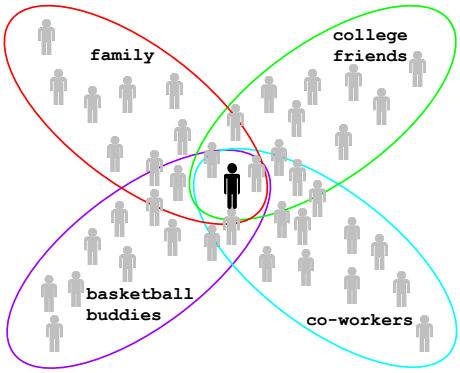


Fig. 1: Illustration of the social circles of an individual. Her family, co-workers, basketball buddies and friends from college are distinct yet overlapping communities.

Effectively extracting the community structure of a node in a network has many useful applications. For example, we can provide more informative and engaging social network feeds by better understanding the membership of an individual to various organizational groups. We can also suggest common friends of an individual to connect because they share mutual interests. We can create match-making algorithms for online players based on the similarity of their game play. Finally, we can identify groups of customers with similar behavior and enhance the efficiency of recommender systems.

Early community detection approaches focused either on grouping the nodes of a network or on searching for links that should be removed to separate the clusters [9]. However, these approaches did not consider the fact that communities may overlap, and ultimately could not provide an accurate representation of a network's community structure. Algorithms that followed [1], [7], [14], [27], [28], [30] allow for nodes to belong to several overlapping communities by employing techniques such as link clustering, matrix factorization, and personalized PageRank vectors. Still, these approaches are not applicable to the massive graphs of the Big Data era, as they focus on the *entire* graph structure and do not scale with regards to both execution time and memory consumption. Recent efforts [16], [19], [20] locally expand an exemplary

*This work was done before author joined Samsung Research America.

seed set in the community of interest and manage to scale to large networks. Such approaches employ random walks to estimate the likelihood of a node to participate in the target community. Studies of real-world networks show that two nodes are more likely to be connected if they share multiple communities in common [32]. For example, people belonging to both the co-workers and basketball buddies communities of Figure 1, are expected to know each other with high probability. Hence, as the overlapping area is in fact denser than the actual communities, seed set expansion methods are driven towards nodes that reside in the overlap. In addition to this, all scalable methods require multiple seeds to avoid detecting multiple overlapping communities as a single one. This is a challenge, as it is usually the case that we are interested in all communities of a single node, instead of seeking one community involving multiple predefined nodes. Finally, seed set expansion approaches are shown to perform well when detecting relatively large communities, whereas high quality communities are in fact small [32].

In this paper, we focus on the neighbors of a single node in the network, i.e., its egonet, and aim at extracting the –possibly overlapping– communities in which this node belongs to. We build upon the ideas of *link clustering* [1], [7] and employ *similarity* measures that allow us to effectively handle densely connected overlaps between communities. Our intuition is that when grouping pairs of links we should capture the *extent* to which a link belongs to multiple overlapping communities. To this end, we utilize a dispersion-based tie strength measure that helps us quantify the participation of a link's adjacent nodes to more than one communities. Our approach is both *efficient* and *scalable* as we focus on local parts of graphs featuring a target node and its neighbors. As we show in our experimental evaluation, we produce a more accurate and intuitive representation of the community structure around a node for a number of real-world networks.

In summary, we make the following contributions:

- We propose a local community detection algorithm that effectively reveals the overlapping nature of real-world network communities of individual target nodes.
- We operate with less input from the user (one single seed vs multiple) and generate communities of equal or better quality.
- We experimentally evaluate our algorithm against state-of-the-art approaches using publicly available networks. Our results show that our approach significantly outperforms current methods using popular evaluation metrics.
- We reduce the execution time notably, by focusing on the neighborhood of a node and thus, manage to handle billion-edge scale graphs.

Our paper is organized as follows: We first introduce some definitions and metrics that will be useful in describing our approach in Section II. In Section III, we describe our hierarchical overlapping community detection algorithm named *Local Dispersion-aware Link Communities* (`LDLC`). In Section IV, we extensively evaluate our approach both qualitatively and quantitatively. Section V reviews related work and finally, Section VI concludes our paper.

II. Background

In this section we review some basic principles and definitions for our work. First, we provide the definition of the *egonet* and then we discuss measures that are used to estimate the strength of *ties* in networks. Finally, we give the definition of *partition density* and detail the dataset used in our study.

A. Egonet

Large-scale graph mining methods are often based on local neighborhoods of nodes [13]. The set of nodes that are one hop away from a given node allows for a variety of scalable analyses that build intuition about that node and its role. In the context of social networks, this one hop neighborhood is frequently called the *egonet*. Figure 1 depicts such an egonet of an individual and the overlapping communities she is part of. We aim at extracting the community structure formed by the nodes connected to a single target node. Thus, we focus on the egonets of target nodes and so, we are able to scale to graphs of extreme volume.

B. Tie Strength Measures

The impact of the *closeness* between nodes in a network's dynamics has been studied extensively [15], [21]. Understanding the complex nature of interacting objects calls for quantifying the *strength* of ties to distinguish the connections of particular importance. We outline here the tie strength measures that we employ in the context of this work:

1) Embeddedness: Intuitively, a large number of shared neighbors between nodes indicates a *strong* tie, whereas a few mutual neighbors indicate a *weak* tie. Therefore, a frequently used measure to estimate the tie strength between two nodes is *embeddedness*, formally defined as:

$$emb(i,j) = |N_+(i) \cap N_+(j)| \tag{1}$$

where $N_+(i)$ is the set of nodes adjacent to i.

2) Jaccard similarity coefficient: The Jaccard similarity coefficient is a frequently used measure of similarity of two sets. In the case of two nodes in a network, the Jaccard similarity coefficient can be applied on the corresponding sets of neighbors:

$$J(i,j) = \frac{|N_+(i) \cap N_+(j)|}{|N_+(i) \cup N_+(j)|} \tag{2}$$

3) Absolute and Recursive Dispersion: Backstrom and Kleinberg [2] propose the use of *dispersion*-based measures for identifying spouses or romantic partners in a network. They analyze real data from Facebook and conclude that high dispersion is indeed present, not only to spouses or romantic partners, but to people who share multiple relevant aspects of their social environment in general.

Formally, we consider the egonet G_u of u in G and define *absolute dispersion* as:

$$disp(u,v) = \sum_{\substack{s,t \in C_{uv} \\ s<t}} d_v(s,t) \tag{3}$$

TABLE I: Graphs of our dataset reaching up to 1.8 billion edges.

Graphs	Type	Nodes	Edges	Av. Degree	Av. Community Size
DBLP	Co-authorship	317,080	1,049,866	3.31	22.45
Amazon	Co-purchasing	334,863	925,872	2.76	13.49
Youtube	Social	1,134,890	2,987,624	2.63	14.59
LiveJournal	Social	3,997,962	34,681,189	8.67	27.80
Orkut	Social	3,072,441	117,185,083	38.14	215.72
Friendster	Social	65,608,366	1,806,067,135	27.53	46.81

where C_{uv} is the set of common neighbors of u and v in G_u, and $d_v(s,t)$ is a distance function equal to 1 when s and t are not directly linked themselves and have no common neighbors in G_u other than u and v, and 0 otherwise.

For a fixed value of $disp(u,v)$, increased embeddedness is a negative predictor of whether v is *close* to u. Thus, absolute dispersion is more effective when normalized using embeddedness. In addition to this, its performance is found to strengthen when applying it recursively as follows. First, we consider $x_v = 1$ for all neighbors v of u. Then, we iteratively update x_v using the formula:

$$x_v = \frac{\sum\limits_{w \in C_{ij}} x_w^2 + 2 \sum\limits_{\substack{s,t \in C_{ij} \\ s<t}} d_v(s,t) x_s x_t}{emb(u,v)} \tag{4}$$

The value produced after the *third iteration* of (4) is empirically found to perform the best [2]. We will refer to this value as *recursive dispersion* of v in G_u for the rest of this paper.

C. Partition Density

Agglomerative community detection algorithms provide us with a dendrogram describing the hierarchical organization pattern of communities. To obtain meaningful communities from the dendrogram it is necessary to determine the level at which to cut the tree at. Ahn et al. [1] introduced *partition density* D, and cut the dendrogram at the level that produces its optimal value. Partition density is formally defined as follows:

$$D = \frac{2}{|E|} \sum_{c \in C} e_c \frac{e_c - (n_c - 1)}{(n_c - 2)(n_c - 1)} \tag{5}$$

where C is the set of communities discovered, e_c is the number of links in a community $c \in C$, and n_c is the number of nodes all the links in e_c touch. Partition density does not suffer a resolution limit like *modularity* [10], as every term in Equation (5) is local in each community c.

D. Networks of our Dataset

In this work, we employ *all six* of the real-world networks with available ground-truth communities that are provided by the Stanford Network Analysis Project (SNAP).[1] In particular, our evaluation is based on the top-5,000 highest quality communities of each of these networks [29]. Table I provides the details of our dataset. *DBLP* is a co-authorship network and ground-truth is formed from authors who published in the same journal or conference. *Amazon* is a product co-purchasing network and the annotated communities associated with it are based on the categories of the products. Finally, *Youtube*, *LiveJournal*, *Orkut*, and *Friendster* are all social networks and user-defined groups are considered as ground-truth communities. We observe in Table I, that our dataset features a graph that exceeds 1.8 billion edges, namely *Friendster*. We also see that, the average community size of most networks is relatively small, with the exception of *Orkut* with an average size of 215.72.

[1] https://snap.stanford.edu/data/#communities

III. Local Dispersion-aware Link Communities

In this section we describe in detail our approach for local community detection. We commence by examining the coverage ratio of egonets on the ground-truth communities of the networks in our dataset. We then discuss the difficulties that existing methods based on seed set expansion and link clustering face, due to the nature of real-world overlapping communities. We show that the use of dispersion-based measures of tie strength can alleviate such issues. Finally, we present our algorithm along with a brief analysis.

A. Egonet Coverage Ratio

Community detection methods that focus on the *global structure* of graphs fail to scale to the massive volume that real-world networks reach. We aim at detecting communities for large-scale graphs efficiently. To this end, we begin discussing our approach by investigating the fraction of nodes of ground-truth communities that are part of *egonets* of nodes that belong to the corresponding communities.

Figure 2 illustrates the coverage ratio of egonets regarding the ground-truth communities of the networks of our dataset. For every ground-truth community of all six networks of our dataset, we examined the coverage of the egonets of each of the nodes belonging to the community. The average coverage ratio depicted in Figure 2, results from the egonets of the nodes with the largest coverage for each ground-truth community. We observe that the coverage ratio is very high for all networks, ranging from 87% to 97%, with the exception of *Orkut* at slightly under 67%. The lower coverage ratio of *Orkut* is attributed to the larger average community size of this network, and it remains low even when using the 2-step geodesic neighborhood of nodes, as reported in [16]. Our findings are aligned with the empirical observations of Yang and Leskovec [32], who report that high quality communities usually consist of no more than 100 nodes and community

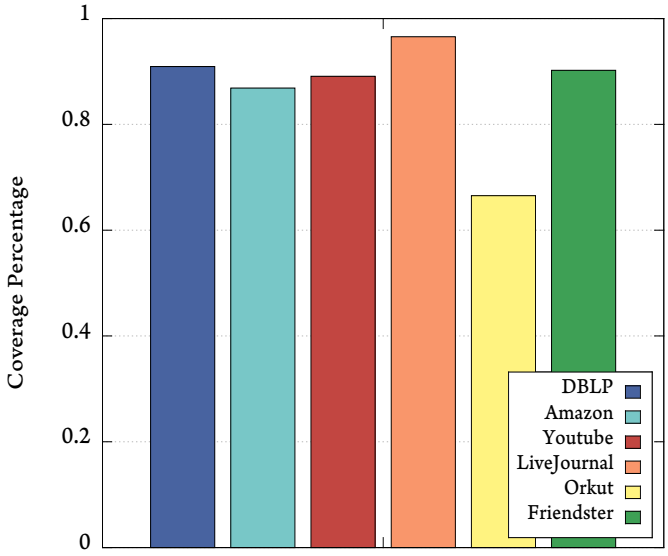


Fig. 2: Egonet coverage ratio for the ground-truth communities of the different graphs provided by SNAP. We show that the coverage ratio for all graphs, with the exception of *Orkut*, is above 87%. The ratio is lower for *Orkut* due to its large average ground-truth community size.

members tend to organize themselves around hub node(s) that are linked with most of the nodes in the community.

The large coverage ratio of egonets on ground-truth communities verifies our intuition that high quality communities can be detected when focusing on egonets of nodes. This allows us to significantly reduce complexity by focusing on a small part of a possibly extremely large network. Even in the case of nodes with extremely high-degree, dealing with the corresponding egonet instead of the global structure of the graph is beyond comparison with regard to efficiency. Space complexity is also reduced greatly, as the memory footprint of the egonet is insignificant when compared to the whole network.

B. Effective Detection of Overlaps Using Hierarchical Link Clustering and Dispersion-based measures

Studies on the structure of real-world networks have revealed that there is an increasing relationship between the number of shared communities and the probability of nodes being linked with an edge [32]. Hence, the nodes residing in overlapping parts of communities are more densely connected than the nodes residing in the non-overlapping parts. Moreover, *connector* nodes, i.e., nodes linked with most of the members of a community, belong to the overlap [32].

Recent local community detection methods [16], [19], [20] expand seed sets by utilizing the dynamics of random walks initiating from the seeds. The participation of a node in the target local community is determined by the corresponding probability score that results from these random walks. Naturally, nodes that reside in the dense overlapping area of multiple communities of a particular node have high

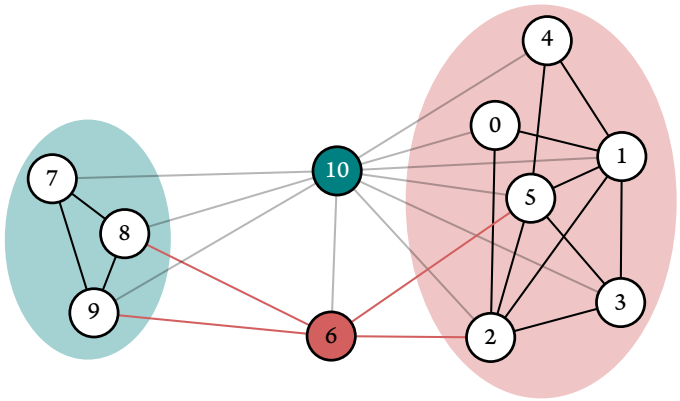


Fig. 3: Social communities in the egonet of an individual (**10**) in a social network. Using a force-directed layout we can easily identify two well-connected groups of acquaintances. A special tie between (**10**) and (**6**) is evident, as (**6**) is the only vertex having links (red colored) towards both communities.

probability scores for random walks starting off this node. In addition to this, nodes outside the overlapping area that are selected in the resulting community due to their probability scores, do not necessarily belong to the same community, as each random walk starting from a seed node is likely to reach a different community. Hierarchical link clustering approaches [1], [7] examine the similarity of pairs of links, and thus, avoid grouping nodes that actually belong to disparate communities. However, they are also unable to handle overlaps appropriately, as they consider that communities in whole are more densely connected themselves than their overlapping parts [31].

Figure 3 illustrates the egonet of an individual (**10**) in a social network. The use of a force-directed layout enables us to easily identify the organizational groups shaped around this node. In particular, we observe that the neighbors of node **10** form two well-connected groups. We also notice, that the only node in the egonet that maintains links (red-colored) with nodes of both groups other than **10**, is **6**. The relationship between nodes **10** and **6** is a particular case of a strong tie which is frequent in networks and has to be considered when identifying overlapping communities. Node **6** acts as a *connector* in the egonet of **10** and is linked with nodes that are not themselves well-connected, as they belong to different organizational groups.

We address the task of local community detection by merging pairs of links in the egonet of a target node. Links often depict a particular relation, e.g., a friendship between two nodes, whereas nodes are usually part of multiple groups. Thus, grouping links instead of nodes allows for the participation of nodes into multiple overlapping communities. To quantify the relevance of two edges e_{ik} and e_{jk}, we can properly adopt the Jaccard similarity coefficient in the context of links [1]. The use of the common node k of the two links would introduce bias without providing useful information. Therefore, using Equation (2), the similarity of the pair (e_{ik}, e_{jk}) is defined as:

$$J(e_{ik}, e_{jk}) = J(i, j) = \frac{|N_+(i) \cap N_+(j)|}{|N_+(i) \cup N_+(j)|} \quad (6)$$

where i and j are both adjacent to k.

The communities that result after performing link clustering in the egonet of Figure 3 using Equation (6), and cutting at the level of optimal partition density, are: **(0,1,2,3,4,5,10)**, **(6,7,8,9,10)** and **(2,5,6,7,8,9,10)**. We see that the third community groups numerous nodes that are not linked together (**2**,**5** with **7**,**8**,**9**). This behavior is exhibited due to the use of the Jaccard similarity coefficient, as this measure is unable to capture how well the neighbors of two nodes are interconnected.

Here, we propose the use of the recursive dispersion measure *along* with the Jaccard similarity coefficient in order to estimate the relevance of pairs of links. Dispersion-based measures fit perfectly in the task of overlapping community detection. Through their use, we are able to single out connector nodes that lie in overlapping parts of communities. For example, using Equation (3) we obtain that node **6** exhibits the highest absolute dispersion in the egonet of **10** with a value of 4. Hence, we can favor groupings of pairs of links with adjacent nodes that share a lot of common neighbors (high Jaccard similarity coefficient) only if these neighbors are also well interconnected (low recursive dispersion). In this way, connector nodes are involved in groupings at a higher level of the resulting dendrogram, which then depicts more accurately the hierarchical structure of the communities in the egonet. Formally, we define the similarity S of two pairs of links (e_{ik}, e_{jk}) to be:

$$S(e_{ik}, e_{jk}) = \frac{J(e_{ik}, e_{jk})}{rec(i) + rec(j) + rec(k)} \quad (7)$$

where $rec(i)$ is the recursive dispersion of i in the egonet of the target node.

Returning on the exemplary egonet of Figure 3, using Equation (7) instead of (6) we detect through hierarchical link clustering the communities: **(0,1,2,3,4,5,10)**, **(7,8,9,10)**, **(2,5,6,10)**, and **(6,8,9,10)**. We observe that the nodes of all communities are much more well-connected. Moreover, node **6** is featured in two communities, in which every two distinct vertices are adjacent, i.e., they form *cliques*. Thus, through Equation (7), we are able to penalize the high dispersion that node **6** exhibits in this egonet, and avoid forming communities featuring nodes of different organizational groups.

C. Our Proposed LDLC Algorithm

We present here the LDLC algorithm for finding local communities in large-scale graphs. LDLC is a hierarchical agglomerative clustering algorithm that detects the communities of a target node in a network by progressively merging pairs of links in the corresponding egonet of this node. The order in which we select these pairs of links is determined by their ranking according to Equation (7). When all pairs are merged, a dendrogram is produced and depicts the hierarchical organization of the communities the target node belongs to. We may cut this dendrogram at the level that produces the optimal partition density, or alternatively, we can cut it at the level that produces the desired number of communities.

Algorithm 1: LDLC(G, u)

input : An undirected network $G = (V, E)$ and a node $u \in V$.
output : A dendrogram depicting the hierarchical (possibly overlapping) communities of G_u.

```
1  begin
2      G_u(V_u, E_u) ← egonet(G, u);
3      rec ← dict();
4      foreach v ∈ G_u, v ≠ u do
5          rec[v] ← 1;
6      for iteration ← 1 to 3 do
7          foreach v ∈ V_u, v ≠ u do
8              rec[v] ← (Σ_{w∈C_uv} x_w^2 + 2 Σ_{s,t∈C_uv, s<t} d(s,t) x_s x_t) / emb(u,v);
9      similarities ← heap();
10     for k ∈ V_u do
11         for (e_ik, e_jk) ← combinations(N_+(k), 2) do
12             J(e_ik, e_jk) ← |N_+(i) ∩ N_+(j)| / |N_+(i) ∪ N_+(j)|;
13             S(e_ik, e_jk) ← J(e_ik, e_jk) / (rec[i] + rec[j] + rec[k]);
14             similarities ← (S(e_ik, e_jk), (e_ik, e_jk));
15     foreach (similarity, (e_ij, e_jk)) ∈ similarities do
16         join_clusters(e_ik, e_jk);
17         if len(clusters) == 1 then
18             break;
```

We outline our suggested LDLC in Algorithm 1. The algorithm accepts a graph $G(V, E)$ and a node $u \in V$ as its input and produces a dendrogram depicting the rich hierarchical structure of u's (possibly overlapping) communities. We start by loading in memory the egonet of u, i.e., node u, its adjacent nodes, and the edges among them (Line 2). Every edge $e \in E_u$ is initially considered to be a community of its own, with the two adjacent nodes as its members.

Lines 3–8 compute the recursive dispersion of all neighbors of u, $v \in V_u$ using Equation (4). Afterwards, for every node in the egonet, we examine the similarity of all possible pairs of its links. The use of a *heap* allows us to maintain the similarities of pairs of links sorted (Line 9). We first calculate the distance of two links using the Jaccard similarity coefficient (Line 12), and then we balance this distance using the previously calculated recursive dispersion measure, as specified in Equation (7). In particular, we divide the value of Jaccard similarity coefficient with the sum of the recursive dispersion of the nodes involved in the links (Line 13), and insert the result in the heap holding the similarities of all pairs (Line 14).

Finally, we iterate through the sorted similarities and group the respective links (Lines 15–16). At every grouping stage, we keep track of the action that takes place to allow for the construction of the dendrogram. Moreover, we calculate the partition density using Equation (5), to determine at the end the best level at which to cut the tree at. When the tree is built, i.e., when we are left with a single cluster, LDLC terminates (Lines 17–18). The dendrogram we produce reveals the overlapping nature of the network's communities through a rich and intuitive hierarchical structure.

IV. Experimental Evaluation

We compare LDLC against three local community detection algorithms based on seed set expansion, namely LEMON [20], LOSP [16], and HeatKernel [19]. We first discuss the specifications of our experimental setting. Then, we proceed with the evaluation of our LDLC by answering the following questions:

- How well does LDLC overcome the need of constraints other methods have, such as requiring multiple seeds to avoid mixing-up multiple overlapping communities, and detecting mostly large communities?
- How well does LDLC perform in detecting communities compared to other methods?
- How efficient is LDLC when compared to other local community detection approaches?

A. Experimental Setting

Our dataset comprises six social, co-authorship, and co-purchasing networks of different sizes, which are outlined in Section II-D. We implemented LDLC using Python 2.7 and the Snap.py interface[2] for the SNAP system. Our algorithm is publicly available.[3] We conducted our timing experiments on a Dell PowerEdge R630 server with an Intel® Xeon® E5-2630 v3, 2.40 GHz with 8 cores, and a total of 256GB of RAM. Our approach could be easily run in parallel as each node can act unilaterally, but we restricted to using only one core to avoid treating the rest of the approaches unfairly.

B. Qualitative Evaluation

We begin our discussion on experimental results by illustrating the behavior of our LDLC against LEMON, when discovering the communities of a target node in the *DBLP* co-authorship network.

Figure 4 depicts the egonet of the target node which we use as a seed to both algorithms (white colored node), as well as the communities detected by the two algorithms. The force-directed layout we use to enhance the visualization, reveals that the nodes form two well-connected groups. The nodes of the grouping in the right actually belong to one of *DBLP*'s high quality ground-truth communities to which none of the nodes of the left grouping belongs to. Moreover, we observe, that the pink colored node is part of the left group but features a link with a node that is part of the right group and is not connected with any of the pink node's neighbors other than the white node. This results to a high value of absolute dispersion for the pink node in the egonet of the white node.

Figure 4a illustrates part of the community that is detected using LEMON. In particular, providing the white colored node as a seed to LEMON, results in a detected community of 81 nodes in total, featuring all the neighbors of the seed node, as well as nodes that are only connected to the target's neighbors. The numbers on the nodes in Figure 4a indicate their ranking according to their likelihood to belong to the target community. The community detected by LEMON exhibits unexpected or undesired attributes. First, high quality ground-truth communities are reported to be much smaller than the community detected by LEMON. In particular, the high quality communities of *DBLP* have an average community size of 22.45 nodes, as shown in Table I. Second, using the target node as the single seed results in the participation in the detected community of nodes that belong to different social circles. In particular, LEMON performs random walks starting from the target node to calculate the likelihood of a node belonging to the detected community. Naturally, nodes of different social circles are likely to exhibit high likelihood and LEMON is unable to distinguish between the different and possibly overlapping communities of the target node. This behavior is evident in Figure 4a. We observe that nodes ranked from 2 to 7 according to their likelihood, reside in the middle part of the left well-connected group of the seed's neighbors. The node that LEMON adds to the community immediately after, ranked 8th, does not share a single link with these nodes, and clearly belongs to another community. Similarly, LEMON continues to add nodes in the detected community from diverse areas around the seed node, until it meets a stopping criterion. Therefore, we see that LEMON favors nodes that reside in dense areas regardless of their relevance to one another. Overcoming this issue would require *multiple cherry-picked* seeds that would increase the likelihood of nodes that are actually part of the same community. This is equally true for other methods that employ random walks for seed set expansion, including LOSP.[4] Last, the pink colored node continues to exhibit high dispersion in the community detected by LEMON.

Figure 4b illustrates the communities discovered in the egonet of the white colored node using LDLC. We cut the tree produced by LDLC at the level that produces the optimal partition density and observe that our algorithm detects two communities, depicted with pink and teal color, respectively. The pink community has a size of 12 nodes, and the teal community a size of 33 nodes. The average size of the two communities of LDLC (22.5) is very close to the average size of the ground-truth communities of this network. Both detected communities are well-connected. In addition, the pink-colored community is a very accurate detection of an actual ground-truth community. Finally, the pink-colored node is featured in both detected communities and does not exhibit high dispersion in either community.

We saw that previous approaches may not detect communities well in situations like the one that we described in this qualitative evaluation. Of course, there are other examples where previous approaches can accurately identify communities. Our goal was to show the strengths of our method through a concrete example. To measure performance more objectively, we now turn to comparing the accuracy of previous local community detection techniques and LDLC by using our ground truth datasets.

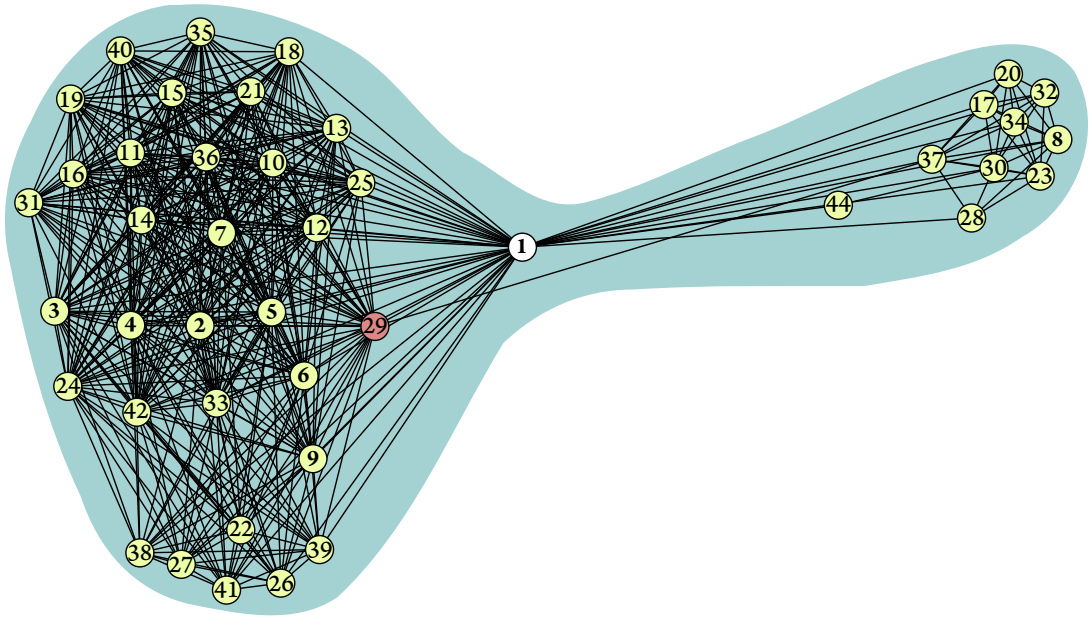
C. Evaluation via Ground-Truth

Evaluating and comparing communities detected by different algorithms is not a trivial task. Large networks exhibit extremely complex organization and cannot be visualized in

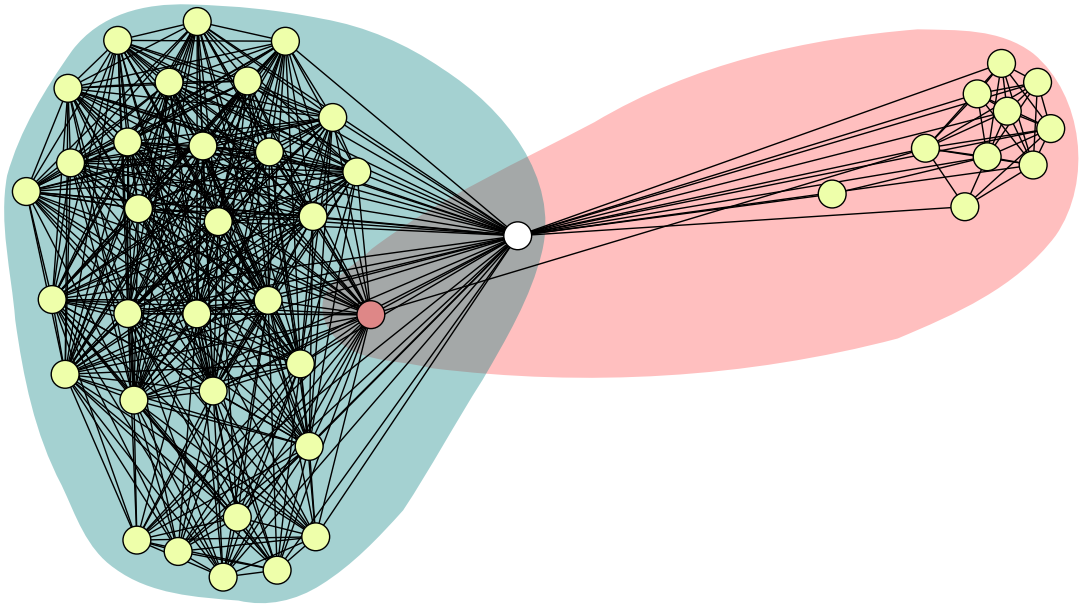
[2]https://snap.stanford.edu/snappy/index.html

[3]https://bitbucket.org/panagiotisl/ldlc

[4]As the authors show in [16] (Figure 2) the presence of three seeds is essential to enable LOSP to distinguish between two overlapping cliques.



(a) **LEMON**



(b) **LDLC**

Fig. 4: The egonet of a node in the *DBLP* graph. `LEMON`'s detected community (4a) features, among others, all the nodes in the egonet. Numbers indicate the `LEMON`'s ranking of the nodes according to their likelihood to belong to the detected community. `LDLC` uses hierarchical link clustering in the egonet of the target node and penalizes the links with nodes exhibiting high dispersion to come up with two communities, colored teal and pink (4b).

meaningful ways. However, we can measure the accuracy of a community detection algorithm given the presence of ground-truth communities [32]. In particular, we can quantify the similarity of a detected community C and a ground-truth community T using *F1 score*, which is defined as:

$$F_1(C,T) = \frac{2 * Precision(C,T) * Recall(C,T)}{Precision(C,T) + Recall(C,T)} \quad (8)$$

where precision is the fraction of detected nodes that are relevant and recall is the fraction of relevant nodes that are retrieved:

$$Precision(C,T) = \frac{|C \cap T|}{|C|} \quad (9)$$

$$Recall(C,T) = \frac{|C \cap T|}{|T|} \quad (10)$$

As there is no standard way of selecting a seed, we

TABLE II: F1 Score comparison.

Algorithm	DBLP	Amazon	Youtube	LiveJournal	Orkut	Friendster
LDLC	**0.843**	0.894	**0.560**	**0.876**	**0.438**	**0.688**
LEMON [20]	0.525	**0.910**	0.190	-	0.170	-
LOSP [16]	0.691	0.845	0.413	0.674	0.216	-
HeatKernel [19]	0.257	0.325	0.177	0.131	0.055	0.078

followed the procedure performed in [19]. In particular, we execute LDLC for all ground-truth communities of each network of our dataset, using every single node as an individual seed. For each ground-truth community, we kept the seed that produced the community with the highest F1 score. Table II shows the average F1 score of LDLC for all ground-truth communities of each network. In addition, we present results of 3 state-of-the-art local community detection algorithms on the same datasets. In particular, we used the publicly available implementation of LEMON[5] to perform experiments through the same exhaustive procedure. We also executed LEMON using 3 random seeds as suggested in [20]. The results we obtained are worse than the ones reported in [20] for *both* cases, as the optimal initialization setting of LEMON differs for the various networks of our dataset. Therefore, we opt to present in Table II the results reported in [20] instead. The implementation of LOSP is not publicly available. Therefore, we include the results of LOSP on the same dataset reported in [16]. Finally, we also report results of HeatKernel from [19]. We note that the results of LOSP and HeatKernel are obtained using a subset of only 500, and 100 ground-truth communities for each network, respectively.

We see in Table II that our LDLC manages to outperform all three state-of-the-art algorithms for all the networks of our dataset, with the exception of the *Amazon* co-purchasing graph for which LEMON is slightly better. The average F1 score of LDLC is significantly larger for all other networks, and the improvement is more evident on the social networks of our dataset, i.e., *Youtube*, *LiveJournal*, *Orkut*, and *Friendster*. For *DBLP*, *Youtube*, and *LiveJournal* the results of LDLC are impressive and much more accurate than all three other methods. Regarding *Orkut*, accurate detection is a particularly hard task, as the size of the ground-truth communities is unusually large in this network. Nonetheless, LDLC is much more effective than the other methods. For the friendship graph of the *Friendster* social network, we employed SNAP in order to load the graph in our infrastructure. *Friendster* almost reaches 2 billion edges and both LEMON and LOSP have failed to report results for this graph due to memory consumption. HeatKernel employs pylibbvg[6] and also manages to load graphs of this scale. We see in Table II that LDLC is able to achieve an F1 score of 0.688, which clearly outperforms HeatKernel. The results regarding the *Amazon* network differentiate due to the particular nature of its communities, which allows all 4 algorithms to achieve high accuracy. More specifically, *Amazon* is a co-purchasing network and, thus, does not feature any connector nodes [32]. In addition, the overlapping ground-truth communities of *Amazon* are usually nested communities, that are subsets of other communities [32].

[5]https://github.com/YixuanLi/LEMON

[6]https://pypi.python.org/pypi/pylibbvg

D. Execution Time comparison

We further evaluate LDLC as far as the execution time is concerned. In particular, for every graph of our dataset we executed LDLC for $5,000$ random nodes of the graph, and report here the average execution time needed. We perform the same experiment using LEMON.

Table III shows the results we obtained for the two algorithms and restate the results as reported in [16] for LOSP, as its implementation is not publicly available. We observe that LDLC is significantly faster than both other methods. In particular, we are able to respond in real-time for the communities of all the graphs of our dataset, including *Friendster* that comprises $1,806,067,135$ edges. We see in Table III that LDLC significantly outperforms both LEMON and LOSP with regard to execution time. This is expected, as LDLC operates only on the egonet of a target node. To produce the egonet we simply need to apply intersection on the sets of neighbors of all neighbors of the target node. Instead, LEMON and LOSP perform multiple random walks to generate a local neighborhood around the target node, a procedure that is much more costly timewise. In addition, the local neighborhood of LEMON or LOSP is usually significantly larger than the egonet of the target node. Therefore, LDLC is applied on a much smaller portion of the original graph, compared to LEMON and LOSP. We note, that the average execution time of LDLC for the *Friendster* graph is smaller than that for *LiveJournal* and *Orkut*, as the egonets of the first are sparser. Thus, LDLC has to iterate over fewer pairs of links in the grouping phase for the graph of *Friendster* and terminates faster.

V. Related Work

The problem of identifying communities emanates from research on graph partitioning, which has been active since the 1970s [18]. Girvan and Newman, with their seminal paper on community detection [12], build on Freeman's *betweeness centrality* measure [11] and define *edge betweeness* as the number of shortest paths between pairs of vertices that run along an edge. Using this measure, they iteratively remove the edges with high betweeness, as they have a tendency to connect different clusters, and thus, reveal the underlying community structure of a network. The algorithm is computationally expensive, but this work sparked significant research in the field of community detection [9].

Many clustering methods aim at maximizing *modularity*, a measure introduced by Newman and Girvan [23]. Modularity captures the quality of a specific proposed division of

TABLE III: Execution time comparison.

Algorithm	DBLP	Amazon	Youtube	LiveJournal	Orkut	Friendster
LDLC	**0.0063** sec	**0.0007** sec	**0.0048** sec	**0.1471** sec	**0.3742** sec	**0.0642** sec
LEMON	9.2781 sec	9.9206 sec	12.2579 sec	-	13.1432 sec	-
LOSP	0.38 sec	0.04 sec	3.85 sec	1.47 sec	4.74 sec	-

a network into communities, by examining how higher the internal cluster density is than the external cluster density. One such method is that of Clauset et al. [4]. There, the proposed algorithm discovers a hierarchical community structure and identifies the best level to cut the tree at as the one that produces the division that maximizes modularity. Blondel et al. [3] propose `Louvain`, another greedy modularity maximization algorithm. Nodes are iteratively aggregated into communities as long as such a move locally improves modularity. Methods of this class are know to suffer from a resolution limit [10].

Another popular direction in the field of community detection, is the use of *random walks*. Pons and Latapy [25] use random walks to measure the similarity between vertices. In another line of work, `Infomap` [26] finds the shortest multilevel description of a *random walker* to get a hierarchical clustering of the network.

The previous methods, hierarchically nested or else, do not take into account the fact that communities in networks may overlap [24]. Palla et al. [24], propose the `Clique Percolation Method`, a local approach based on *k-cliques*. Overlaps between communities are allowed as a given node can be part of several *k-clique* percolation clusters at the same time. A revolutionary idea in overlapping community detection was introduced in two approaches that were developed almost simultaneously [1], [7]. The core of these approaches is that instead of focusing on grouping nodes, communities should be formed by considering groups of links. This allows for a natural incorporation of overlaps between communities while also retaining a hierarchical community structure. Ahn et al. [1] additionally report a comparison of their proposed algorithm with previous approaches, proving that it outperforms all of them.

Later research efforts focused on providing more scalable approaches. Coscia et al. [5] use *egonet* analysis methods and propose `DEMON` that allows nodes to vote for the communities they see locally in an effort to improve the quality of overlapping partitions. Yang and Leskovec [28] report that, contrary to previous belief, community overlaps are more densely connected than the non-overlapping parts. This relaxes the assumption that governed all previous efforts on overlapping community detection. Building on their empirical observations, they also propose `BigClam` [30], a community detection method that uses matrix factorization to detect communities. `BigClam` requires as an input the number of communities to look for, or else should be guided with the minimum and maximum number of communities as well as the number of tries it should make. Gleich and Seshadhri [14] formalized the problem of community detection as finding vertex sets with small *conductance*, where conductance of a cluster is a measure of the probability that a one-step random walk starting in that cluster, leaves the cluster. They proposed the use of personalized PageRank vectors to identify communities with good conductance scores. A similar approach is investigated in [27], where a number of alternative seeding phases before the use of personalized PageRank vectors is examined. However, minimizing conductance leads to the identification of dense areas of a network as single communities, when they are in fact overlapping parts of multiple communities [31]. These approaches are more efficient than previous overlapping methods but fail to handle massive scale graphs.

Recent approaches depart from the direction of detecting communities on the *global* graph structure. Instead, they detect *local* communities in time functional to the size of the community, and provide support for large scale graphs. Kloster and Gleich [19] propose a deterministic local algorithm to compute heat kernel diffusion and study the communities it produces. The authors compare with PageRank diffusion on real-world datasets and report that their approach is able to detect smaller, more accurate communities, with slightly worse conductance. Li et al. [20] propose `LEMON` that uses seeds to perform short random walks and form an approximate invariant subspace termed *local spectra*. Then, `LEMON` looks for the minimum 1-norm vector in the span of this *local spectra* such that the seeds are in its support. Building on the findings of `LEMON`, He et al. propose `LOSP` [16] that is additionally able to detect small communities and performs better when initiated with a single seed. In another line of work, Metwally et al. [22] employ general purpose clustering algorithms to detect click rings that launch advertising traffic fraud attacks. However, their techniques are not applicable on *single* graphs, as they use *multi-faceted* graphs, where each *facet* is a set of edges that represents the relationships between the nodes in a specific context. Our approach focuses on local communities but employs hierarchical clustering of pairs of links in the egonet of a target node, using *tie strength* measures that effectively handle networks with dense overlapping parts of communities. Thus, we efficiently reveal a more accurate hierarchical community structure in large scale networks.

VI. Conclusion

In this paper we propose and develop `LDLC`, a novel local community detection algorithm for large scale graphs. `LDLC` focuses on the egonet of a target node and performs hierarchical agglomerative clustering on the egonet's pairs of links. We investigate measures that evaluate the strength of ties in networks, building on the notion that mutual neighbors of nodes may be or may be not well interconnected. The nodes involved in ties that belong in the second category, act as connector nodes between overlapping communities. Therefore, in a hierarchical approach they should be considered for grouping when the higher levels of the respective dendrogram are forming. We achieve that, by using the recursive dispersion

measure to balance the similarity of two links and prioritize the grouping of pairs of links with mutual neighbors that function in a single context. Thus, our approach is able to handle overlapping communities appropriately and provides increased accuracy, while also revealing the rich hierarchical structure of the communities of a node in the network. We compare LDLC with three state-of-the-art local community detection methods to highlight the effectiveness of our approach when handling overlapping areas of multiple communities. Moreover, we examine the accuracy of all algorithms against ground-truth communities and find that LDLC significantly outperforms all of them for a wide range of publicly available networks. Finally, we conduct timing experiments to showcase the improved efficiency LDLC offers for large scale graphs.

In the near future, we plan to investigate the performance of LDLC by exploiting node attributes to assign weights to links of networks, and adopting the Tanimoto coefficient [1] to capture the importance of weighted links. For example, we can assume that members of a social network group of a high school's alumni should be linked strongly in case they are born in the same year. We believe that a comparison of LDLC's performance on the respective unweighted and weighted networks will be extremely interesting. Furthermore, a drift from the currently available ground-truth communities depicting metadata groups [17] to communities that better portray the functional roles of a network's nodes, will allow for a more accurate comparison of community detection techniques. To this end, we will collect data from social network groups where membership signifies affinity.

ACKNOWLEDGMENTS

This work has been partially supported by the EU H2020 *"GALENA"* grant with agreement n. 641515. Panagiotis would like to express his gratitude to Katia Papakonstantinopoulou, Nikos Leonardos, and Katerina El Raheb for invaluable discussions and feedback.

REFERENCES

[1] 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.

[2] L. Backstrom and J. Kleinberg, "Romantic partnerships and the dispersion of social ties: A network analysis of relationship status on facebook," in *Proc. of the 17th ACM Conf. on Computer Supported Cooperative Work & Social Computing*, 2014, pp. 831–841.

[3] 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.

[4] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.

[5] M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi, "DEMON: a local-first discovery method for overlapping communities," in *Proc. of the 18th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2012, pp. 615–623.

[6] I. de Sola Pool and M. Kochen, "Contacts and influence," *Social networks*, vol. 1, no. 1, pp. 5–51, 1978.

[7] T. Evans and R. Lambiotte, "Line graphs, link partitions, and overlapping communities," *Physical Review E*, vol. 80, p. 016105, 2009.

[8] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," in *ACM SIGCOMM computer communication review*, vol. 29, no. 4. ACM, 1999, pp. 251–262.

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

[10] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, 2007.

[11] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[12] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proc. of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[13] D. F. Gleich and M. W. Mahoney, "Mining large graphs," in *Handbook of Big Data*, ser. Handbooks of modern statistical methods, P. Bühlmann, P. Drineas, M. Kane, and M. van de Laan, Eds. CRC Press, 2016, pp. 191–220.

[14] D. F. Gleich and C. Seshadhri, "Vertex neighborhoods, low conductance cuts, and good seeds for local community methods," in *Proc. of the 18th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2012, pp. 597–605.

[15] M. S. Granovetter, "The strength of weak ties," *American journal of sociology*, pp. 1360–1380, 1973.

[16] K. He, Y. Sun, D. Bindel, J. E. Hopcroft, and Y. Li, "Detecting overlapping communities from local spectral subspaces," in *IEEE International Conference on Data Mining, Atlantic City, NJ, USA*, 2015, pp. 769–774.

[17] D. Hric, R. K. Darst, and S. Fortunato, "Community detection in networks: Structural communities versus ground truth," *Physical Review E*, vol. 90, no. 6, p. 062805, 2014.

[18] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell system technical journal*, vol. 49, no. 2, pp. 291–307, 1970.

[19] K. Kloster and D. F. Gleich, "Heat kernel based community detection," in *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA*, 2014, pp. 1386–1395.

[20] Y. Li, K. He, D. Bindel, and J. E. Hopcroft, "Uncovering the small community structure in large networks: A local spectral approach," in *Proc. of the 24th Int. Conf. on World Wide Web*, 2015, pp. 658–668.

[21] P. V. Marsden and K. E. Campbell, "Measuring tie strength," *Social forces*, vol. 63, no. 2, pp. 482–501, 1984.

[22] A. Metwally, J. Pan, M. Doan, and C. Faloutsos, "Scalable community discovery from multi-faceted graphs," in *IEEE Int. Conf. on Big Data, Santa Clara, CA, USA*, 2015, pp. 1053–1062.

[23] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, p. 026113, Feb. 2004.

[24] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[25] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *Computer and Information Sciences-ISCIS 2005*, 2005, pp. 284–293.

[26] M. Rosvall and C. T. Bergstrom, "Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems," *PloS one*, vol. 6, no. 4, p. e18209, 2011.

[27] J. J. Whang, D. F. Gleich, and I. S. Dhillon, "Overlapping community detection using seed set expansion," in *Proc. of the 22nd ACM Int. Conf. on Information & Knowledge Management*, 2013, pp. 2099–2108.

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

[29] ——, "Defining and evaluating network communities based on ground-truth," in *Proc. of the 12th IEEE International Conference on Data Mining*, 2012, pp. 745–754.

[30] ——, "Overlapping community detection at scale: a nonnegative matrix factorization approach," in *Proc. of the 6th ACM int. Conf. on Web Search and Data Mining*, 2013, pp. 587–596.

[31] ——, "Overlapping communities explain core–periphery organization of networks," *Proc. of tihe IEEE*, vol. 102, no. 12, 2014.

[32] ——, "Structure and overlaps of ground-truth communities in networks," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 2, p. 26, 2014.