Effective Detection of Overlapping Link Communities

[redacted]

***Abstract*—Real-life systems entailing interacting objects have been routinely modeled as graphs or networks. Revealing the community structure of such systems is crucial in helping us better understand their complex nature. Networks and the relationships they portray are exploited by proposed community detection techniques that seek to facilitate the discovery of separate, overlapping, nested or fully hierarchical communities. Nevertheless, our perception of what a community is in a network of interacting objects, has evolved over the years. In this respect, the decomposition of networks into possibly overlapping organizational groups and our enhanced understanding of their intricate interactions remain open challenges. We address these issues through an agglomerative approach that groups pairs of links and provides a richer hierarchical structure than previous efforts. We attain this objective by exploiting the dispersion of established relationships among objects in the network. Our algorithm measures the *similarity* of such links as well as the extent of their participation in multiple contexts, to determine the order in which pairs of links should be grouped. Moreover, our technique termed `Dispersion-aware Link Communities` or `DLC` can handle both unweighted and weighted networks. Our experimental results with a popular network strongly demonstrate that our approach overcomes issues earlier techniques stumbled upon. Furthermore, we investigate the performance of our algorithm against *ground-truth* communities for a wide range of networks and show that `DLC` outperforms state-of-the-art methods.**

I. Introduction

Networks are a powerful tool for modeling relations and interactions of entities in the real world. Real-world networks are characterized by a high level of order and organization and allow the study of properties such as the power-law degree distributions and the small-world structure. Another important characteristic of networks is the presence of community structures [13]. At a high level, communities are groups of nodes that share a common functional property or role, e.g. two people that went to the same high school, or two movies with the same actor.

Although in several cases communities in a network are distinct (e.g. *Bulls* vs. *Knicks* fans) it is often the case that communities overlap. As an example, consider Figure 1 that illustrates the communities of an individual in a social network. The Figure displays the family, co-workers, basketball buddies and friends from college groups. It is obvious that the communities may overlap in various ways. For example, a co-worker may also be a basketball buddy and a friend from college. At a high level, identifying overlapping communities is more challenging compared to non-overlapping ones due to the fact that overlapping communities may have a more complex structure of connections that are not easy to discern.

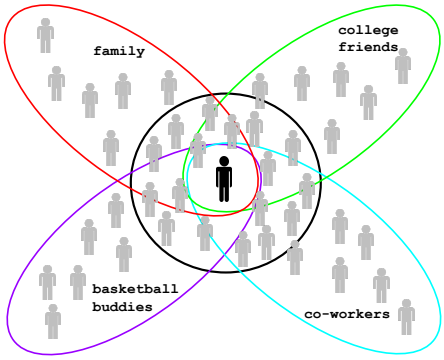


Fig. 1: Illustration of the social circles of an individual. Her family, co-workers, basketball buddies and friends from college are distinct yet overlapping communities.

Effectively extracting the communities from a network has many useful applications. To mention but a few: we can discover people with mutual interests in a social network and suggest them to connect. We can determine players that are have similar styles in online games and create match-making algorithms based on their gameplay. We can identify groups of customers with similar behavior and enhance the efficiency of recommender systems. Finally, we could improve news dissemination online by utilizing community membership to promote updates.

Early community detection approaches focused on grouping the nodes of a network or search for edges that should be removed to separate the clusters [10]. However, these approaches do not take into account that communities may overlap, and ultimately cannot provide an accurate representation of a network's community structure. Ahn et al. [1] proposed an approach that instead of focusing on nodes it uses links, which typically have a unique identity compared to nodes which may serve several roles within the same network. Their approach performs hierarchical clustering of the edges of a network, based on the *similarity* of the nodes directly attached to the edges. In this way, dendrograms which capture hierarchical community structures of the network are produced. The approach allows for nodes belonging to several overlapping communities.

The use of measures that rely solely on the *similarity* of pairs of links, results in identifying as communities the non-overlapping parts of communities that happen to be more densely connected compared to the overlapping parts. This is due to an implicit assumption on the structure of communities, i.e. that communities are expected to be more densely connected components. However, contrary to this implicit assumption, Yang and Leskovec, studied communities in real-world social, collaboration and information networks [27], and point out that two nodes are more likely to be connected if they share multiple communities in common [26], [28]. 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. As a result, since the density in the area of community overlap is high for such networks, the algorithm in [1] is misled by the high *similarity* of the respective edges and identifies the overlapping area as a single community. Algorithms [26], [28], [14], [25] that followed the observation in [27], unfortunately do not provide the hierarchy of community structure but instead allow only nesting of communities.

In our work, we build upon the ideas of link clustering and the observation that overlapping parts of communities are denser than non-overlapping parts. Our intuition is that when grouping pairs of edges we should capture the *extent* to which a link belongs to multiple overlapping communities. We do so by proposing a new metric based on dispersion. We proceed in an agglomerative manner and first group the similar pairs of edges that belong to single communities. Additionally, we purposely delay the grouping of the pairs of edges that lie in the overlap. As we will show in our experimental evaluation, the result is a better and more intuitive representation of the community structure in a number of real-world networks. The lower levels of our resulting dendrograms feature pairs of links that connect nodes sharing a single functional role. On the other hand, higher levels of the hierarchy contain pairs of links that act as brokers between different, yet overlapping communities.

In summary, we make the following contributions in this paper:

- We propose an algorithm that effectively reveals the overlapping nature of real-world network communities.
- We study how to handle both unweighted and weighted networks and retrieve a rich and intuitive hierarchical community structure.
- We experimentally evaluate our approach against the state-of-the-art approach using publicly available networks.

Our paper is organized as follows: We first introduce some notation 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 `DLC` (from Dispersion-aware Link Community detection). 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. Firstly, we discuss *embeddedness* and *absolute* and *recursive dispersion*, that are used to measure the strength of *ties*. Then, we detail the *similarity* measures used in the work of Ahn et al. [1]. Finally, we give the definition of some performance metrics, including *Average F1 score*, a measure that tests the accuracy of community detection algorithms against *ground-truth* communities.

A. Embeddedness

The *closeness* between nodes in a network and its impact on the network's dynamics has been studied in [15], [20]. 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*, that captures how much the neighbors of two nodes overlap. *Embeddedness* is formally defined as:

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

where $N_+(i)$ is the set of neighbors of node i.

In the case of social networks, individuals who share a particular focus are more likely to share joint activities with each other, as opposed to people that do not share that focus [8]. Therefore, we expect individuals to organize their relations around *relevant aspects* of their social environment, also termed as *foci*. As relationship partners share neighbors from several different *foci*, *embeddedness* has been used for the identification of couples [9].

B. Absolute and Recursive Dispersion

A more powerful measure of *tie* strength that targets individuals who span *foci* is that of Backstrom and Kleinberg [2], that takes into account the *dispersion*, i.e., the extent to which two individuals' mutual friends are not themselves well-connected.

Absolute dispersion is defined as:

$$disp(i,j) = \sum_{s,t \in C_{ij}} d(s,t) \tag{2}$$

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

For a fixed value of $disp(i,j)$, increased *embeddedness* is a negative predictor of whether j is the partner of i. Thus, *absolute dispersion* should be normalized using *embeddedness*. In addition to this, its performance strengthens by applying it recursively. We initially consider $x_{ij} = 1$ for all neighbors j of i. Then we iteratively update x_{ij} using the following formula:

$$x_{ij} = \frac{\sum\limits_{w \in C_{ij}} x_{iw}^2 + 2 \sum\limits_{s,t \in C_{ij}} d(s,t) x_{is} x_{it}}{emb(i,j)} \tag{3}$$

The value produced after the *third iteration* of (3) is empirically found to perform the best [2]. We will refer to this value as *recursive dispersion* of i and j for the rest of this paper.

C. Alternative Similarities

The algorithm of Ahn et al. [1], to which we will refer to as `LC` in the following, is an example of an agglomerative approach. However, in contrast to earlier similar approaches, `LC` focuses on links instead of nodes. The links of a network represent a node's functionality in many contexts, and thus, `LC` allows overlaps between communities. In particular, `LC` estimates the *similarities* between pairs of links using the `Jaccard` coefficient. *Similarity* S between edges e_{ik} and e_{jk}, is defined to be:

$$S(e_{ik}, e_{jk}) = \frac{|N_+(i) \cap N_+(j)|}{|N_+(i) \cup N_+(j)|} \tag{4}$$

where i and j are both neighbors of k and $N_+(i)$ denotes the set of neighbors of node i.

Applying the method in weighted networks is made possible through the adoption of the `Tanimoto` coefficient [1]. According to this, *similarity* S between edges e_{ik} and e_{jk} is:

$$S(e_{ik}, e_{jk}) = \frac{a_i a_j}{|a_i|^2 + |a_j|^2 - a_i a_j} \tag{5}$$

where i and j are both neighbors of k and a_i is a vector of the weights of links between node i and the nodes in the first-order neighborhoods of both nodes i and j.

D. Partition Density

Hierarchical community detection algorithms provide us with a dendrogram describing all the resulting communities, as well as their hierarchical structure. Ahn et al. [1] suggest the use of the *partition density*, D, to cut the dendrogram at the level that produces its optimal value. *Partition density*, is formally defined as follows:

$$D = \frac{2}{|E|} \sum_{c \in C} e_c \frac{e_c - (n_c - 1)}{(n_c - 2)(n_c - 1)} \tag{6}$$

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.

Alternatively, the tree can simply be cut at the level that produces the desired number of resulting communities.

E. Performance Metrics

Evaluating and comparing communities detected by different algorithms is not a trivial task. Large networks exhibit extremely complex organization and cannot be visualized in meaningful ways. However, we can measure the accuracy of a community detection algorithm given the presence of *ground-truth* communities, with the use of *Average F1 score*, as it is defined in [28]. We repeat here the definition for completeness:

Given a network $G(V, E)$, we consider a set of detected communities $\hat{C}$ and a set of *ground-truth* communities C^*. As there is no standard way of assigning a detected community $\hat{C}_i$ to a *ground-truth* community C^*_i, we use the *F1-score* of the best matching *ground-truth* community to each detected community, and the *F1-score* of the best matching detected community to each *ground-truth* community. In particular, we define *Average F1 score* as:

$$\begin{aligned} AF1(\hat{C}, C^*) = \frac{1}{2}\Big(&\frac{1}{|C^*|} \sum_{C_i \in C^*} F1(C_i, \hat{C}_{g}(i)) \\ &+ \frac{1}{|\hat{C}|} \sum_{\hat{C}_i \in \hat{C}} F1(C_{g'}(i), \hat{C}_i)\Big) \end{aligned} \tag{7}$$

where the best-matching g and g' are defined as follows:

$$g(i) = \arg\max_j F1(C_i, \hat{C}_j) \tag{8}$$

$$g'(i) = \arg\max_j F1(C_j, \hat{C}_i) \tag{9}$$

In addition to *Average F1 score*, the *coverage percentage* and *mean overlap* are other metrics used in the literature [1]. The first, measures the percentage of nodes that participate in the resulting communities, and the second, expresses the mean number of communities each node participates in.

III. Our Dispersion-aware Link Communities Approach

We present here our approach for revealing the hierarchical community structure of a network. We first discuss our observation regarding the reasons previous hierarchical approaches extract overlapping parts of different communities as a single separate community. Then, we propose a way to alleviate this issue through the use of *tie* strength measures that capture the notion of nodes functioning in a single or multiple contexts. After that, we discuss the measure used in our approach. Next, we detail our algorithm, termed `Dispersion-aware Link Communities` (`DLC`). Finally, we provide the modifications that are necessary for it to function with weighted networks.

A. Observation

Community detection algorithms have evolved from grouping related nodes into separate functional subunits, to discovering communities that allow overlaps. Yang and Leskovec recently empirically observed that overlapping detection methods discover communities based on a false assumption [27]. In particular, they consider non-overlapping parts of communities to be more densely connected than the overlapping parts. We provide here an example that highlights why using the `Jaccard` coefficient as a *similarity* measure of two links leads to the identification of overlapping communities that follow this false assumption.

Figure 2 depicts the well-known character co-occurrence network from Victor Hugo's novel, entitled "Les Misérables".

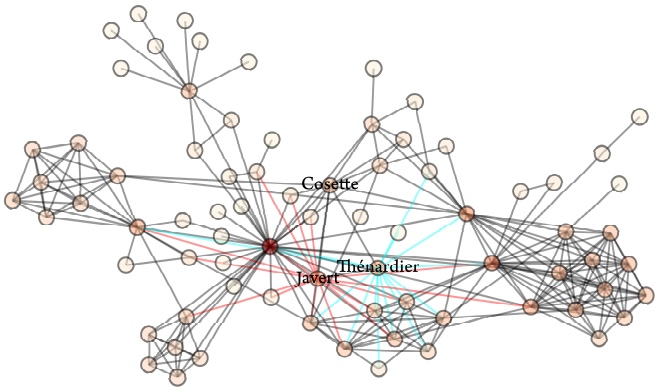


Fig. 2: Character co-occurrence network from Victor Hugo's novel, "Les Misérables". *Javert*, and *Thénardier* are acquainted with characters clustered in multiple groups of the network and their relationship with *Cosette* appears to be *similar*.

We observe that three of the main characters, namely *Cosette*, *Javert*, and *Thénardier*, appear in a dense area. We also notice that the links of *Javert*, illustrated using red lines, are quite *similar* to those of *Thénardier*, illustrated with cyan lines.

Using (4) we ascertain that the `Jaccard` similarity corresponding to the pair of edges connecting *Cosette* with *Javert* and *Cosette* with *Thénardier* is indeed relatively high. In particular, *Javert* and *Thénardier* have cumulatively a total of 24 neighbors and share 11 of them. Therefore, through (4) we get that the *similarity* of this pair of edges is 0.46. Due to this score, `LC` is eventually led to identify a community that includes all three characters.

However, in Hugo's novel, *Cosette* leaves the *Thénardier* family at a very young age, whereas her acquaintance with *Javert* occurs many years later. The reason for this effect is that *Javert* and *Thénardier* share a number of acquaintances that are not acquainted to each other. Therefore, choosing to include an edge pair that merges the communities they belong to, leads to the unification of two distinct, yet overlapping, communities into a single one. Worse still, this unification occurs at an early stage of the algorithm and the hierarchical structure of the network is also detected incorrectly.

`LC`'s inability to handle this type of overlaps is also reported in [29], following observations concerning the presence of *connector* nodes and densely connected overlaps of communities in a broad range of networks.

B. Employing dispersion-based measures for overlapping community detection

The relationship between *Javert* and *Thénardier* is a particular case of a strong *tie* which is frequent in networks and has to be considered when identifying overlapping communities. The two characters act as *connector* nodes for several overlapping communities. To identify connections of this type we utilize the new *dispersion*-based *tie* strength measures detailed in [2].

Backstrom and Kleinberg propose the use of *absolute dispersion*, as defined in (2), as a much more effective measure than *embeddedness* 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 span *foci* in general. The latter includes the relationship of *Javert* and *Thénardier*. Ranking links between nodes using the *recursive dispersion*, as defined in (3), is found to perform the best when identifying such relationships.

We propose the use of *dispersion*-based measures to single out the links that belong in overlapping parts of communities. This allows for prioritizing *similar* links with adjacent nodes that do not span *foci*. Thereby, we manage to favor groupings of pairs of edges that lie in the non-overlapping parts of communities and delay those pairs that lie in the overlapping parts. In this way, overlapping communities are joined at a higher level of the resulting tree which then depicts more accurately the hierarchical structure of the network and its communities.

C. Normalized recursive dispersion

Recursive dispersion exhibits high values of standard deviation for all the networks that were examined in this work. Additionally, the distributions of the *recursive dispersion* values for all networks are heavy-tailed.

Backstrom and Kleinberg [2] employ this measure to rank pairs of nodes. Therefore, a high value of standard deviation and a heavy-tailed distribution of *recursive dispersion* values have no impact on their work. However, we employ this measure to balance the *similarity* of two edges and disfavor the pairs that span *foci*. Thus, we need to alleviate the problem of *recursive dispersion* governing the *similarity* of a pair.

To do so, we perform a preprocessing step that standardizes the range of *recursive dispersion* values. In particular, we opt to normalize the values of *recursive dispersion* by applying feature scaling, to bring them into the range $[1, 1000]$, as well as log transformation. After both these actions have been applied, we increase the value by 1, resulting in values: $1 \leq nrd(i,j) \leq 4 \ \forall e_{ij} \in E$. As we will show in Section IV, this range of values performs well.

We formally define the *normalized recursive dispersion* of nodes i, j with $e_{ij} \in E$ to be:

$$nrd(i,j) = log(1 + 999\frac{(rd(i,j) - min\_rd)}{(max\_rd - min\_rd)}) + 1 \quad (10)$$

where $rd(i,j)$ is the *recursive dispersion* of nodes i and j, and max_rd, min_rd are the maximum and minimum values of *recursive dispersion* for the graph G.

D. `DLC` Algorithm

We present here the main contribution of this work, the `DLC` algorithm.

`DLC` performs hierarchical agglomerative clustering on the links of a network. It builds the hierarchy of communities by progressively merging pairs of links. The order on which the merging occurs, is determined by the *similarity* of pairs of links. To capture this *similarity* we employ both (4) and (10).

Algorithm 1: DLC(G)

input : An undirected graph $G = (V, E)$.
output : A dendrogram depicting the hierarchical structure of communities of G.

1 **begin**
2 $rd \leftarrow$ dict();
3 **for** $iter \leftarrow 1$ **to** 3 **do**
4 **foreach** $(i, j) \in E$ **do**
5 $rd[(i,j)] \leftarrow \frac{\sum_{w \in C_{ij}} x_w^2 + 2 \sum_{s,t \in C_{ij}} d(s,t) x_s x_t}{emb(i,j)}$;
6 $nrd \leftarrow$ normalize(rd);
7 $similarities \leftarrow$ heap();
8 **for** $k \in V$ **do**
9 **for** $(e_{ik}, e_{jk}) \leftarrow combinations(n_+(k), 2)$ **do**
10 **if** $(i, j) \in nrd$ **then**
11 $dispersion \leftarrow nrd[(i,j)]$;
12 **else**
13 $dispersion \leftarrow \frac{nrd[(i,k)] + nrd[(j,k)]}{2}$;
14 $S(e_{ik}, e_{jk}) \leftarrow \frac{|n_+(i) \cap n_+(j)|}{|n_+(i) \cup n_+(j)|}$;
15 $similarity \leftarrow \frac{S(e_{ik}, e_{jk})}{dispersion}$;
16 $similarities \leftarrow (similarity, (e_{ik}, e_{jk}))$;
17 **foreach** $(similarity, (e_{ij}, e_{jk})) \in similarities$ **do**
18 join_clusters(e_{ik}, e_{jk});
19 **if** *len(clusters)* == 1 **then**
20 **break**;

In particular, DLC uses the *normalized recursive dispersion* measure to balance the *similarity* S of pairs of links. When all clusters are merged, the resulting dendrogram is cut at the level that produces the optimal *partition density*. Alternatively, it can be cut at a level that produces the desired number of communities.

Algorithm 1 is a simplified version of DLC, appropriate for unweighted networks. We detail the modifications needed for a weighted network in Section III-F. The algorithm accepts a graph $G(V, E)$ as its input and produces a dendrogram depicting the rich hierarchical structure of its (possibly overlapping) communities.

Initially, we consider every link of graph G to be a community of its own. Lines 2–5 compute the *recursive dispersion* of all edges $e \in E$. Then, using this measure in Line 6, we compute the *normalized recursive dispersion* value of each edge, according to (10). Afterwards, for every node of the graph, we examine the *similarity* of all possible pairs of its edges. We first calculate the distance of two edges using (4) (Line 14), and then we balance this distance using the previously calculated *normalized recursive dispersion* measure. In particular, we divide the value of (4) with the *normalized recursive dispersion* of the adjacent nodes of the two edges. In case the adjacent nodes are not linked we use the average of the *normalized recursive dispersion* of the nodes of each edge (Lines 10–13, 15–16). Finally, we iterate through the sorted *similarities* and group the respective edges (Lines 17–18). At every grouping stage, we keep the action that takes place and allows the construction of the dendrogram, and we calculate the partition density 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, DLC terminates (Lines 19–20).

E. Complexity of DLC

Computing the *recursive dispersion* for every edge, requires finding the set of common neighbors of its two nodes, and then examining for every possible pair of neighbors if they are linked through another path in the network. This requires total time O($|E|d^3$), where $|E|$ is the number of edges in the network and d the network's average degree. Then, DLC calculates the *similarity* for every pair of edges. In particular, for every node we examine $\binom{d}{2}$ pairs and find their common neighbors. The computed *similarity* at every step is inserted in a heap. This step takes time O($|V|d^3 log(|V|)$), where $|V|$ is the number of nodes in the network. Finally, DLC iterates through the computed *similarities* and groups the respective clusters, which takes time O($|V|d^2 log(|V|)$). Therefore, the overall complexity is O($|V|d^3 log(|V|)$). We note here, that the average degree of real-world networks, such as the ones we examine in our experiments, is relatively small.

LC also features a phase in which for every node we examine $\binom{d}{2}$ pairs to find their common neighbors and insert their *similarity* in a heap. Therefore, its complexity is equivalent to that of DLC.

F. Weighted networks

Weights of links are valuable additional information that should be considered in network analysis, as they offer a more accurate representation of the strength of a *tie* between nodes [10]. For instance, in a collaboration network links can be weighted according to the number of papers co-authored by pairs of scientists, to differentiate the authors that work frequently together from those that co-authored only a few papers.

LC is appropriate for both weighted and unweighted networks, as (5) can be used instead of (4) when the edges are weighted; DLC uses *dispersion*-based measures that can be modified similarly to consider weights as well.

Absolute dispersion depends on the number of mutual neighbors of two nodes that are not connected to each other, i.e., their weight is 0 in a weighted network. *Normalized recursive dispersion* additionally depends on the *embeddedness* of two nodes. Therefore, to use this measure in weighted networks, we simply have to alter (3) and take the weights of links into account, as follows:

$$x_{ij} = \frac{\sum\limits_{w \in C_{ij}} x_{iw}^2 + 2 \sum\limits_{s,t \in C_{ij}} d(s,t) x_{is} x_{it}}{a_i a_j} \qquad (11)$$

where a_i is a vector of the weights of links between node i and the nodes in the first-order neighborhoods of both nodes i and j.

IV. Experimental Evaluation

We implemented and tested our approach on a number of publicly available and well-studied networks, and compared our results with the state-of-the-art approach of Ahn et al. [1]. In this section, we first describe the dataset used for our experiments and list the details of our implementation. Then, we investigate the *recursive dispersion* scores of a small

network	nodes	edges	mean degree	communities
Les Miserablés	77	254	6.6	-
congress	526	14,198	53.98	903
philosophers	1,231	7,303	9.71	1,162
dblp	317,080	1,049,866	6.62	13,477
amazon	334,863	925,872	5.53	151,037

TABLE I: The undirected networks examined in our experiments.

network, as well as their impact when used in hierarchical link clustering. After that, we present results of our algorithm against the LC approach and examine the differences between the discovered communities. Additionally, we provide a comparison between the two algorithms using the *Average F1 score* metric. Finally, we compare the behavior of the two algorithms using metrics presented in [1].

A. Experimental Setting

Our dataset comprises five networks of various sizes, four of which are accompanied with *ground-truth* communities. The properties of these networks and their respective *ground-truth* communities are listed in Table I.

Below, we furnish the fundamental characteristics of the five networks we have experimented with:

- *Les Miserablés*[1]: The co-appearance network of characters in the novel "Les Miserablés", extracted by Knuth [18]. The nodes of this network are the characters of the novel. Pairs of nodes are linked with an edge if the corresponding characters encounter each other in the novel. No *ground-truth* community is available, however, the small size of the network and the popularity of the novel allows for a qualitative evaluation.
- *congress*[2]: The network of legislative collaborations between US representatives of the House and the Senate during the 111^{st} US congress (2009-2011) [6]. Nodes are politicians who are linked with an edge if they have at least 75 co-sponsorships to bills. Edges that were created by bills with more than 10 co-sponsors were removed. Each bill is related to several subjects and *ground-truth* communities are formed using subjects politicians frequently worked on.
- *philosophers*[3]: A network of famous philosophers extracted from pages of the english-language Wikipedia[4] [1]. Wikipedia maintains a list of philosophers and hyperlinks in each philosopher's article are used to create edges between them. Wikipedia also maintains a taxonomy of philosophers according to their field, theory or area, which is used to produce *ground-truth* communities.
- *dblp*[5]: A co-authorship network with authors as nodes and an edge linking them in case they have published at least one paper together [27]. *Ground-truth* communities are formed using authors who published work to a certain venue, i.e. journal or conference.
- *amazon*[5]: A network formed after crawling the Amazon website. Nodes are products and an edge linking them exists in case they are frequently co-purchased [27]. *Ground-truth* communities are formed using the product categories that the Amazon website provides.

[1]Available here: http://www-personal.umich.edu/~mejn/netdata/

[2]Available here: http://www.michelecoscia.com/?page_id=42

[3]We thank Sune Lehmann for generously sharing the *philosophers* dataset.

[4]Wikipedia: http://en.wikipedia.org

[5]Available here: https://snap.stanford.edu/data/

rank	node u	node v	normalized *recursive dispersion*
1	*Gueulemer*	*Madame Thénardier*	4.0
2	*Gavroche*	*Gueulemer*	3.986
14	*Madame Thénardier*	*Thénardier*	3.061
21	*Fantine*	*Madame Thénardier*	2.953
22	*Cosette*	*Madame Thénardier*	2.953
29	*Bamatabois*	*Fantine*	2.773
37	*Cosette*	*Thénardier*	2.684
51	*Marius*	*Thénardier*	2.614
80	*Javert*	*Jean Valjean*	2.524
88	*Javert*	*Thénardier*	2.483

TABLE II: A selection of links from *Les Misérables* co-appearance network with high *normalized recursive dispersion* values involving popular characters of the novel.

For experiments against LC we used the author's Python implementation of the algorithm[6]. Our algorithm, implemented using Python as well, is also available online[7].

B. Impact of normalized recursive dispersion

Table II shows some exemplary values of *normalized recursive dispersion* for edges of *Les Misérables* network, ordered descendingly. We opted to list pairs of characters that are popular along with their ranking, instead of simply presenting those that ranked the highest, to make more evident that this measure indeed captures those edges that act as bridges to two or more overlapping communities.

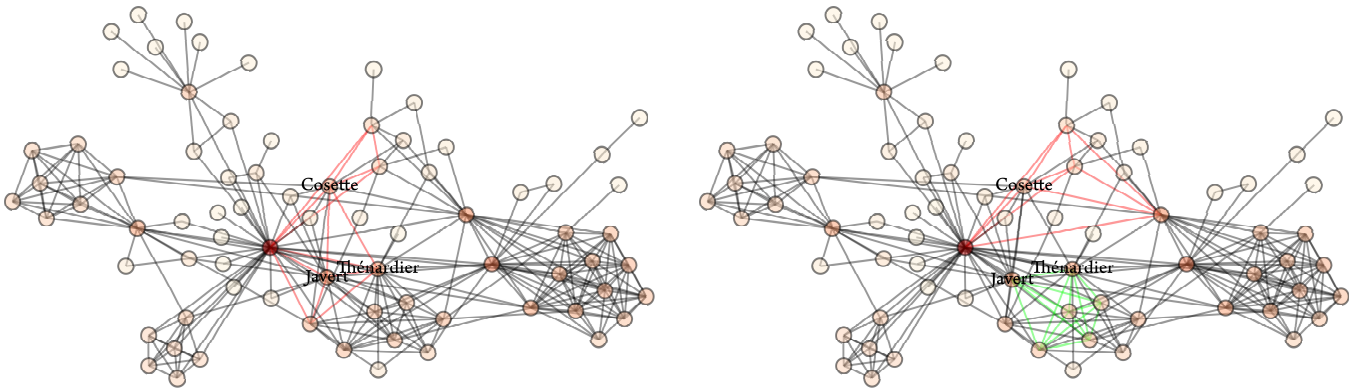
DLC uses *normalized recursive dispersion* to balance the values of *similarities* between edges of a network and prioritize those pairs that do not have adjacent nodes exhibiting high *dispersion*. The impact of the use of *normalized recursive dispersion* is evident in Table III. We present pairs of edges that are balanced using the *normalized recursive dispersion* values listed in Table II, along with their *similarity* and ranking using LC and DLC. We observe that all these pairs of edges ranked lower with DLC than with LC, and in certain cases the difference in ranking is quite large. The examples are again picked in order to involve popular characters to enhance the understanding of the reader. The pairs of edges in Table II either belong in multiple communities and should be moved lower to nest them, or do not mutually belong in a community and should be moved low enough to let the clusters form without them being considered.

[6]Available here: https://github.com/bagrow/linkcomm

[7]Available here: http://tinyurl.com/nfrese5.

edge e_{ik}	edge e_{jk}	*similarity*	*normalized recursive dispersion*	LC rank	DLC rank
(*Gueulemer*, *Babet*)	(*Madame Thénardier*, *Babet*)	0.5333	4.0	866	1,446
(*Gavroche*, *Babet*)	(*Gueulemer*, *Babet*)	0.3077	3.9857	1,286	1,875
(*Madame Thénardier*, *Jean Valjean*)	(*Thénardier*, *Jean Valjean*)	0.6111	3.061	736	1,082
(*Fantine*, *Javert*)	(*Madame Thénardier*, *Javert*)	0.2174	2.9527	1,487	1,927
(*Babet*, *Madame Thénardier*)	(*Cosette*, *Madame Thénardier*)	0.2105	2.9522	1,495	1,949
(*Bamatabois*, *Jean Valjean*)	(*Fantine*, *Jean Valjean*)	0.1905	2.7733	1,583	1,984
(*Cosette*, *Javert*)	(*Thénardier*, *Javert*)	0.2609	2.684	1,364	1,676
(*Marius*, *Cosette*)	(*Thénardier*, *Cosette*)	0.2333	2.6141	1,437	1,742
(*Javert*, *Cosette*)	(*Jean Valjean*, *Cosette*)	0.4865	2.5235	985	1,094
(*Javert*, *Cosette*)	(*Thénardier*, *Cosette*)	0.4583	2.4829	1,045	1,149

TABLE III: Link pairs for *Les Misérables* co-appearance network that are disfavored with DLC in comparison to LC because they span *foci* and ought to be considered for grouping at a higher level of the dendrogram.



(a) One of the communities LC mistakenly comes up with, due to the special *tie* between *Javert* and *Thénardier*, denoted with red links.

(b) Two communities detected using DLC, denoted with red and green links respectively, that involve the characters of the community LC came up with.

Fig. 3: Representation of partial results of algorithms LC and DLC for the character co-appearance network of *Les Misérables*, using a force-directed layout that strengthens the visual aspect of the difference in results.

C. Interpretation of Resulting Communities

The smallest network of our dataset, i.e., *Les Misérables*, does not have a set of *ground-truth* communities associated with it. However, the popularity of the novel and the small size of its co-appearance network allows for the investigation of specific examples that showcase our contribution.

We continue building on the example concerning two of the characters from *Les Misérables* that appear to span *foci*, namely, *Javert* and *Thénardier*. As the edges whose adjacent nodes follow similar behavior tend to have high Jaccard similarity, we expect LC to mistakenly identify the overlapping part of multiple communities as a single one. Indeed, one of the communities that LC comes up with features: *Javert*, *Thénardier*, *Madame Thénardier*, *Jean Valjean*, *Cosette*, *Gillenormand* and *Mademoiselle Gillenormand*. However, the relationships between members of the *Thénardier* family and *Cosette*, *Javert*, or members of the *Gillenormand* family are highly irrelevant. To make this even more evident, we provide an illustration of this community with Figure 3a. We use a force-directed layout that makes nodes that are not connected to be drawn apart. It is clearly visible that the community found by LC, denoted using red colored links, groups nodes that are not actually part of a single community.

Using DLC, we manage to overcome this issue and come up with much more meaningful communities involving the aforementioned characters. In particular, we find a community featuring *Cosette*, *Marius*, *Jean Valjean*, *Gillenormand*, and *Mademoiselle Gillenormand*, i.e., the young couple of the novel and their respective 'guardians'. Moreover, we find a community that more accurately illustrates the relationship between *Javert* and *Thénardier*, with its additional members being the *Patron-Minette* gang, i.e., *Gueulemer*, *Claquesous*, *Babet*, and *Montparnasse*. We remind the reader that in the novel, *Thénardier* employs the *Patron-Minette* gang to rob *Valjean* only to see their plans fail due to the intervention of *Javert*, who manages to rescue *Valjean*.

We would like to note here that DLC, as well as LC, create a dendrogram representing the hierarchical structure of the network. The higher we cut the tree, the more tangled the resulting communities will appear. However, DLC provides an improved representation of the nesting that occurs in real networks as it favors links that do not span *foci*. In particular, links with large *dispersion*, that are essentially acting as *brokers* between disconnected nodes [2], are used in groupings later than with LC. As a consequence, the links that span *foci* appear in a higher position in the dendrogram, and DLC captures more accurately the concept of *brokerage* between communities [4].

Table IV depicts all resulting communities of the two algorithms when we cut the tree at the respective level of optimal *partition density*. Nodes are ordered using *modularity* to enhance the visual aspect of the table, by bringing closer

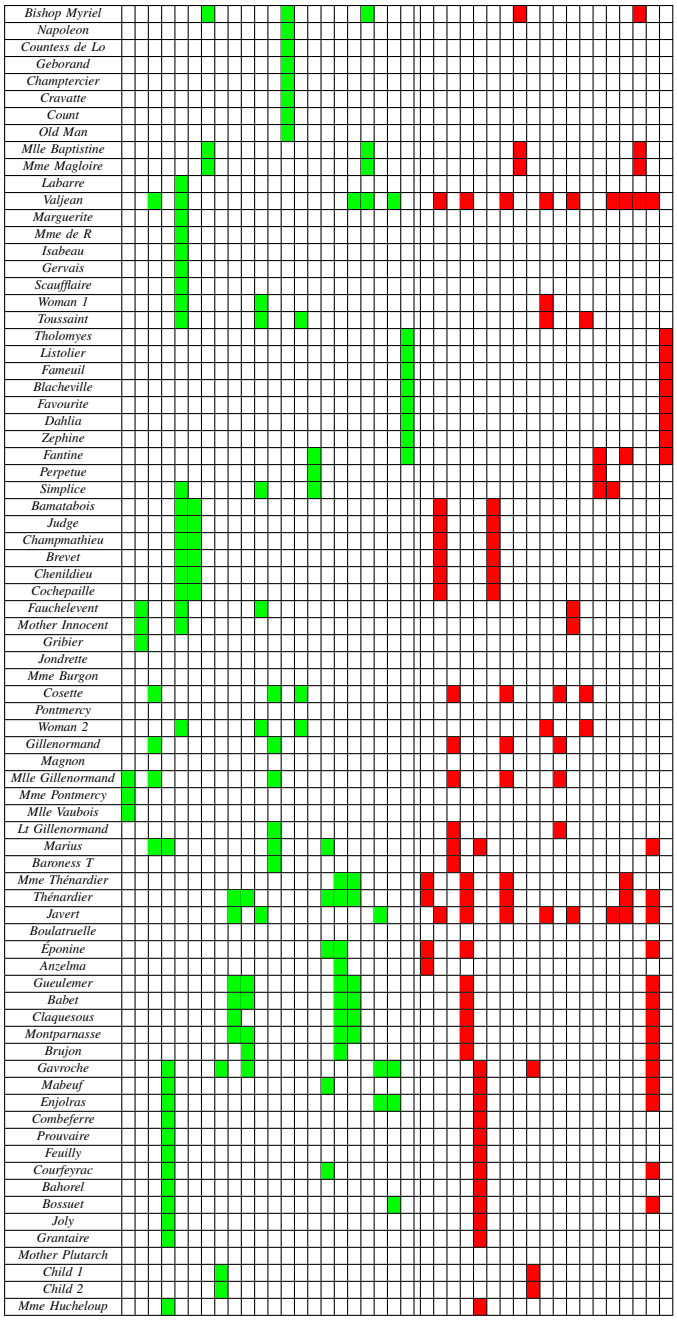


TABLE IV: Communities detected using DLC (green) and LC (red) for the character co-appearence network of *Les Misérables*. We ordered the characters based on *modularity* to bring close nodes that possibly share communities.

nodes that possibly share communities. We notice that *Valjean* and *Javert* appear in much fewer communities with DLC than with LC. In addition to this, the presence of both of them in some of the communities of LC is clearly problematic. A good illustration of this is DLC's community of *Fauchelevent*, *Mother Innocent*, and *Gribier*, where the grouping of a small and well-defined cluster with the link of *Valjean* and *Javert* is delayed, as opposed to LC's community of *Fauchelevent*, *Mother Innocent*, *Valjean*, and *Javert*. In addition to this, we

network	average f1 score	
	LC	DLC
congress	0.2191	**0.2532**
philosophers	0.3353	**0.3406**
dblp	0.3328	**0.3647**
amazon	0.3564	**0.3565**

TABLE V: Average F1 score for the networks of our dataset that posses *ground-truth* communities. DLC outperforms LC for all the networks of our dataset.

notice that DLC manages to form meaningful communities out of minor characters of the novel, as opposed to LC that ignores a lot of them. For example, DLC forms a community that groups all acquaintances of *Bishop Myriel* that have no other links in the network, i.e., *Napoleon*, *Countess de Lo*, *Geborand*, *Champtercier*, *Cravatte*, the *Count*, and the *Old Man*.

D. Experimental Evaluation of DLC

As the size of a network grows, sophisticated visualization techniques offer representations with no evident organization from which we cannot easily extract any information. Therefore, a common practice in evaluating community detection algorithms is comparing their results with *ground-truth* communities of networks.

Most of the networks of our dataset posses such *ground-truth* communities and therefore allow for a quantitative evaluation of the performance of our overlapping community detection algorithm. We compare here our DLC algorithm against LC and allow both methods to decide the number of resulting communities based on the optimal *partition density*. For both algorithms we do not consider communities with fewer than two nodes, as *nontrivial* communities should have three or more nodes [1].

We see in Table V that DLC outperforms LC for all networks of our dataset. This is expected as DLC captures more accurately the overlapping structure that has been observed in such networks. The improvement is less evident in the *amazon* network. This is also expected as communities of this network tend to have high values of embeddedness, which results into well separated groups [16] and almost eliminates DLC's advantage over LC.

We note here, that achieving high scores using this metric is difficult as the *ground-truth* communities that are frequently used, differ from *structural* communities and are actually metadata groups [16]. As such, they are not recoverable from network topology alone. Additionally, the number of resulting communities plays a crucial role in the accuracy obtained. The number of communities is an input parameter for many community detection algorithms, and thus, ranking of algorithms based on comparisons against *ground-truth* communities may differ depending on the experimental setting.

We further compare DLC with LC using performance benchmarks from Ahn et al. [1]. Table VI shows the results of both DLC and LC algorithms as far as the *coverage quality* and *overlap quality* are concerned. We renamed this measures to *coverage percentage* and *mean overlap* respectively, to highlight their true meaning, as good performance in these

network	coverage percentage		mean overlap		# communities	
	LC	DLC	LC	DLC	LC	DLC
Les Miserablés	0.7143	0.9221	1.4805	1.7273	19	22
congress	0.9544	0.9867	8.5589	1.0684	342	14
philosophers	0.8099	0.7945	2.6353	2.6572	595	690
dblp	0.8977	0.8972	1.9061	1.9165	130,755	132,294
amazon	0.7846	0.7859	1.5125	1.5132	107,745	107,678

TABLE VI: *Coverage percentage* and *mean overlap* of LC and DLC algorithms for the networks of our dataset.

benchmarks can be misleading, especially in the case of *overlap quality*.

We observe that the *coverage percentage* of DLC is better than that of LC for small networks and almost identical for large networks. This is expected as DLC delays the grouping of edges that span *foci* —and have a high value of similarity— in favor of pairs of edges that would be ignored with LC. In addition to this, we observe similar behavior between the two algorithms as far as *mean overlap* is concerned, with the exception of the smaller networks. This is due to the difference in the number of detected communities, which plays a crucial role for this measure.

V. Related Work

The problem of identifying communities emanates from research on graph partitioning, which has been active since the 1970s [17]. Girvan and Newman, with their seminal paper on community detection [13], build on Freeman's *betweeness centrality* measure [12] 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 [10].

Many clustering methods aim at maximizing *modularity*, a measure introduced by Newman and Girvan [21]. *Modularity* captures the quality of a specific proposed division of 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. [5]. There, the proposed algorithm discovers a hierarchical community structure and identifies the best level to cut the tree 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 [11].

Another popular direction in the field of community detection, is the use of *random walks*. Pons and Latapy [23] use *random walks* to measure the similarity between vertices. In another line of work, Infomap [24] 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 [22]. Palla et al. [22], 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 reports 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. [6] 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 [26] 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 [28], 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 score. A similar approach is investigated in [25], 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 [26]. Li et al. [19] search for small community structure in large networks by considering only a subset of the network such that the seeding nodes are in its support. However, they report results concerning the identification of one seeded community against algorithms that uncover the whole community structure of networks. These approaches manage to detect communities in large networks but do not reveal their hierarchical structure. Instead, they only permit overlaps and nestings. Additionally, they do not consider weighted networks.

Our approach belongs to the class of hierarchical community detection algorithms and uses links instead of nodes, as in [1] and [7]. We however, examine the use of *similarity* measures that handle networks with overlapping parts of communities that are denser than non-overlapping parts. Thus, we reveal a more accurate hierarchical community structure for both unweighted and weighted networks.

VI. Conclusion

In this paper we propose and develop a novel overlapping community detection algorithm, termed DLC. Our algorithm builds link communities through hierarchical agglomerative

clustering according to the *similarity* of the networks' pairs of links. We take into account a recent observation stating that overlapping parts of a network's communities are denser than non-overlapping parts. We investigate measures that evaluate the strength of *ties* in networks, building on the notion that mutual neighbors of nodes may span multiple *foci* or be clustered in a single context. The nodes involved in *ties* that belong in the first category, act as *connector* nodes between overlapping communities. Therefore, they should be considered for grouping in a hierarchical approach when the higher levels of the respective dendrogram are forming. We achieve that, by using *normalized recursive dispersion* to balance the *similarity* of two edges and prioritize the grouping of pairs of edges with mutual neighbors that function in a single context. Our approach reveals the rich hierarchical structure of network communities and handles both unweighted and weighted networks. We compare `DLC` with `LC` [1] and detail how the differences in their functionality alter the forming of communities in a popular network. In addition to this, we examine the accuracy of both algorithms against *ground-truth* communities and find that `DLC` outperforms `LC` for a wide range of publicly available networks.

We will further investigate the performance of `DLC` by exploiting node attributes to assign weights to links of networks. For example, we can assume that members of 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 `DLC`'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 to communities that better portray the functional roles of a network's nodes, will allow for a more accurate comparison of community detection techniques. We plan to collect data from only those social network groups where membership signifies affinity, and thus, create *ground-truth* communities of improved quality.

Acknowledgments

[redacted] would like to thank [redacted] and [redacted] for invaluable discussions and feedback. This work has been partially supported by [redacted].

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] R. S. Burt, "The social structure of competition," *Explorations in economic sociology*, vol. 65, p. 103, 1993.

[5] 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.

[6] 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.

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

[8] S. L. Feld, "The focused organization of social ties," *American journal of sociology*, pp. 1015–1035, 1981.

[9] D. H. Felmlee, "No couple is an island: A social network perspective on dyadic stability," *Social Forces*, vol. 79, no. 4, pp. 1259–1287, 2001.

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

[11] 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.

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

[13] 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.

[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] 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.

[17] 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.

[18] D. E. Knuth, *The Stanford GraphBase: A Platform for Combinatorial Computing*, 1993.

[19] 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.

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

[21] 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.

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

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

[28] ——, "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.

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