Broader Perception For Local Community Identification

Master Thesis Information Science
May 2010

Institute for Computing and Information Sciences
Radboud University
Nijmegen, The Netherlands

Author:	Frank Koopmans
Email:	ftwkoopmans@gmail.com
Thesis Number:	111 IK
Supervisor:	Theo van der Weide
Second Corrector:	Elena Marchiori

Abstract. The local identification of communities in a network can be much more effective than partitioning the entire network when only a small portion of a large network is of interest. A local approach is also favored when it is difficult to obtain information about the entire network, the world wide web is a prime example. Such local identification algorithms typically evaluate nodes outside the local community that are directly connected to the local community as potential community members without the need for knowledge about the entire network. But while being a recognized and relevant technique in community identification, the quality of local community identification lags behind their global counterparts that use the entire network. In order to improve the evaluation of community candidates in local algorithms we propose the use of contextual information beyond direct neighbors. This will decrease the gap between relevant network knowledge of global and local methods while remaining a local approach. Benchmarks on synthetic networks show our approach increases the quality of locally identified communities in general and a decrease of the dependency on specific source nodes.

I. INTRODUCTION

The science of complex systems is a popular interdisciplinary field of research. Using a network to represent the elements and interactions in a complex network is a commonly used tool to study complex system phenomena [1, 2]. Such analysis is done (among many others) on protein networks, social networks and (parts of) the internet [3, 4].

Complex systems, and thus their representation as a network, often exhibit a priori unknown structure of building blocks which have a (distinct) function. These are expressed in the network as sets of nodes that are among each other densely connected and have relatively few connections to the rest of the network. Depending on the type of the complex system and the scientific discipline, such building blocks are referred to as *modules*, *communities* or *clusters*. Identifying and observing such communities may lead to a greater understanding of the structure and functioning of a complex system.

The effort of identifying communities in a network is known to be NP-complete [5, 6]. And some networks may be very large, the internet is a prime example with over 20 billion potential nodes [7]. Many different techniques for community identification have been developed, with varying computational costs and accuracy [8, 9]. A widely used measure for evaluating community structure is *modularity*. Basically, it measures the fraction of edges within communities for a given partition of a network [10]. Optimizing the modularity measure for a network results in a partition of sparsely connected communities. This has proven to be a fast and quite effective method for finding communities (and thus revealing the modular structure of a network) [11]. However, there are some downsides to maximizing modularity in practise [12] just like every approach to community identification has its flaws.

Modularity considers a partition of a network and separated communities, whereas many real networks are made of highly *overlapping* cohesive nodes [13]. So a node can be a member of more than one community in some network. Communities become more difficult to distinguish as overlap increases. If an algorithm tries to partition the network in Figure 1b then the outcome seems quite random since there will be no structural difference when placing the overlap node in either community. Placing the node in both communities seems to be the only sensible option. The approach of regular modularity maximization as mentioned earlier is unable to detect heterogeneities in node memberships [14], but there are extensions to modularity that will take overlap into account [15].

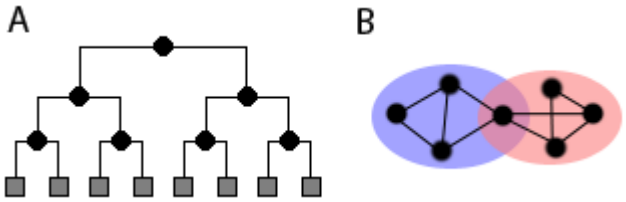

FIG. 1. Left: A dendrogram that reveals hierarchial community structure. Right: Overlap between two communities.

In addition to community structure a network can also exhibit a *hierarchical structure* in which communities further divide into subcommunities, possibly over multiple scales [16, 17]. In each hierarchy level the number of (sub)communities and their structure may vary. Figure 1a illustrates the commonly used dendrogram representation of subcommunity structure.

In some cases one would rather identify a specific community instead of all communities in the entire network in order to reduce the amount of information that needs to be processed or acquired. For example, finding the community around a given website on the internet or finding all friends of a specific person in a social network. In such cases there should be no need to process the entire network. A *local community* is a community in the network identified by starting from a *source node* while using only information from the context of the local community, thus no knowledge of the entire network is required.

Not having knowledge of the entire network is a strong

advantage for the efficiency of a local identification algorithm, but it can be a handicap for its quality. A lot of work has been done on local identification algorithms [18–20] and even though their quality seems lacking compared to global methods we think the concept has great potential, especially in large networks.

A third class of community identification methods is a mixture of global and local approaches [21, 22]. It detects communities at a global level(throughout the entire network) and uses local optimization of some fitness function. The performance of such an approach can be highly effective at detecting both overlap and hierarchical community structure [23, 24], any improvement made to local community identification techniques will benefit such approaches too.

In section two we review the process of local community identification, compare the concept of local and global identification, discuss the ideal community and review how it is usually approximated. We propose an improvement that can be applied to local identification algorithms in general in section three. In section four we present the results of several benchmarks used to test the improvements coined in section three followed by discussion thereof.

II. LOCAL COMMUNITY IDENTIFICATION

In this section we will first discuss some key properties of local communities and their identification algorithms. Then we compare local and global approaches to community identification, discuss the definition of the ideal local community and finally review existing local algorithms.

We define the *universe* of a community C, denoted as $U(C)$, as all nodes that are outside of C and adjacent to any node within C.

$$U(C) = \left\{ v \in C - N \mid \exists_{u \in C} \left[u \to v \right] \right\}$$

The *boundary* of C, denoted as $B(C)$, are all nodes within C that have at least one neighbor outside of C (thus within $U(C)$).

$$B(C) = \left\{ u \in C \mid \exists_{v \in U(C)} \left[u \to v \right] \right\}$$

Figure 2 illustrates these different areas in a network from a local perspective.

In a local approach we start with a specific node, commonly referred to as the source node, as the only member of the local community. Knowledge of the network is then expanded by crawling all nodes in the universe. The local community is built by evaluating these yet unknown nodes adjacent to the community and adding the best candidate to the local community and updating the universe at each step. This is repeated until no suitable candidates are left.

The *community definition* determines the quality of a given community within a network, we will denote it as μ. For example, a basic measure could be the fraction of internal edges, $internaledges(C)/degree(C)$. We will discuss several community definitions from current work later in this section.

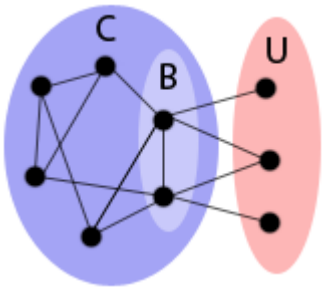

FIG. 2. An example network illustrating the Community, Boundary and Universe.

The *selection criteria* determine what node(s) are added to the community at each algorithm step. This is dependent on the community definition and determines the strategy for building the community. For example: at each algorithm step, add candidate member $v \in U(C)$ that complies with $max(\mu(C \cup v))$ to C.

The *stopping criteria* tell the algorithm when to stop building the community. A straight forward stopping criterium could be; stop adding nodes to the community when even the best possible addition does not further improve the quality of the community. In this example, let $v \in U(C)$ be the best addition to C as determined by the selection criteria. Then the selection criterium $\mu(C) > \mu(C \cup v)$ tells the algorithm when the job is done.

A local community is constructed around a given source node. We emphasize this fundamental property of a local community because it actually defines the local community; the set of nodes that are considered related by the source node.

In a social network one could find all friends of person u. This will yield a list of nodes that are directly, or indirectly, connected to person u and are considered a friend by u. If we would set out to find the friends of any friend of u, for instance person v, this might yield the exact same community but it does not have to be. Just because u thinks of v as his friend does not imply v thinks u is his friend.

In conclusion, we consider a local community around a source node as the relation between all nodes in the set, starting from the source node, from the perspective of the source node. The quality of such a local community is quantified by community definition μ.

A. Global versus Local

A major difference in philosophy between global and local methods of identifying communities is in their goal. The goal of a global approach is to partition the network

such that the community definition is optimized over all communities. A local approach aims to find the ideal community around a given source node without paying any attention to (consequences for) the rest of the network.

In a traditional global partition no overlap is allowed so we think there is no use for comparison with a local approach at all. In such a case, the local approach is supposed to approximate a community that was constructed based on considerations of the impact on other communities by the global algorithm. A local community cannot do that because it has no knowledge of other communities, only the local community currently under construction is known.

If one is interested in identifying all communities in a network a local approach could be applied to every node in the network, though less efficient than a global approach it will yield all communities.

As we have seen earlier in this section, a local approach acknowledges overlap in community structure. If we scale a local approach to a global approach it will still acknowledge overlap. Therefor, we can only compare a global and a local approach properly if we require the global approach to recognize overlapping communities too.

When identifying all communities in a network and any two communities are overlapping, there are more nodes within the set of communities than the total amount of nodes in the network. Another consequence of allowing overlap in a global approach is the potential to create a community for each individual node in the network. Thus, we can theoretically construct a local community C_v around every node $v \in N$ and a set of communities Z by a global approach that allows overlap such that every $C_v \in Z$.

In the previous section we discussed community hierarchy and its presence in many networks, while other networks may have a flat community structure. When a local algorithm agglomerates nodes to construct a community it may, or may not, detect the different levels of hierarchy in community structure as the community grows. This may be hard to observe since all the local algorithm knows is the set of nodes agglomerated so far and some contextual information around that local community. As opposed to a global approach that may access all network information at all times. A global approach can start from the largest community and work its way down internally, assuring that whatever it finds is a part of the community it started with.

So locally, we can hardly tell if we are crossing a hierarchy level while agglomerating nor can we tell at which hierarchy level we are (as seen from the top level). On the other hand, once the algorithm is done one could assume the top hierarchial level community has been identified, equal to that of a global method. Then from there on work back down and find the subcommunities like a global approach would.

B. What is the Ideal Local Community?

Previously we formulated a local community as a set of related nodes from the perspective of the source node. So each node in the local community has a relation with the source node. Furthermore the quality of a community as a whole is quantified by community definition μ.

Let us denote an edge from node u to node v, $(u, v) \in E$, as $u \to v$. We introduce node relation φ to indicate if two nodes are related, where related implies there is a path. Because we want to express the relation between nodes within the community we require the entire path to be within the community.

$$\varphi(C, u, v) \equiv u \to v \vee \exists_{x \in C} \left[u \to x \wedge \varphi(C, x, v) \right]$$

A set of nodes is a proper community if all nodes are related from the perspective of the source node. We introduce Φ as a valid community measure for a given community C and its source node v.

$$\Phi(C, v) = v \in C \wedge \forall_{x \in C - v} \left[\varphi(C, v, x) \right]$$

Note that Φ does not tell us how good the quality of the community is. In order to find the ideal local community we consider all possible communities and the ones with maximum μ scores are the ideal communities. We speak of multitudes here because it is possible that there is more than one community we can construct that yield the same μ score.

Let ζ be the set of all possible communities around a source node. We obtain ζ by filtering all communities from the powerset of the network that are found valid by Φ.

$$\zeta(v) = \left\{ C \subseteq N \;\middle|\; \Phi(C, v) \right\}$$

Now we can find the set of ideal communities around source node v using $max(\mu(\zeta(v)))$.

In practise, we are unable to compute the ideal communities because of the time complexity of this measure. The amount of elements in the powerset of a set with n elements equals 2^n. So we cannot use this measure as a reference for community identification algorithms in practise, but we can use it as a goal to approximate in any local identification algorithm implementation.

C. Current Work

Local greedy agglomeration algorithms that iterate universe nodes are widely used for local community identification. What varies most in different approaches to local community identification is the community definition and the stopping criteria. Such an algorithm starts of with the source node as the only member of the community and then for each iteration all universe nodes are evaluated and the best node is added to the community, until the stopping criteria are met. Some algorithms add

the twist of randomly removing nodes during agglomeration to allow algorithms to reevaluate past additions and prevent ever growing communities.

Using this concept there is only limited knowledge of the network. After the algorithm halts only the nodes in $C \cup U(C)$ have been evaluated. The time complexity is significantly lower than the ideal community computation suggested earlier because for each iteration we only need to evaluate nodes in the universe that were affected by mutations from the previous algorithm step.

Now that we have seen the basic approach of a local algorithm we will discuss a variety of local community definitions. The intuitive notion of a community, many internal and few external connections, can be found in most definitions. As illustrated by the following definitions, the intuitive notion leaves room for many (subtle) differences in approach.

1. Various Definitions

Clauset coined local modularity as a community size independent quality measure for local communities based on modularity [18]. It considers the fraction of edges from the boundary that are internal to the community:

$$R(C) = \frac{\sum_{ij} A_{ij} \left[j \in B(C), i \in C \right]}{\sum_{ij} A_{ij} \left[j \in B(C) \right]}$$

where A_{ij} is the adjacency matrix indicating edges from node j to node i in the network. Agglomerating candidate nodes is done by efficiently computing and comparing $\triangle R$ for all members of the universe. In each algorithm step the node with the highest $\triangle R$, say node v, is added to the community and then the boundary and universe are updated only where v was of influence. R lies on the interval $0 < R \leq 1$ ($R = 0$ when C is totally disconnected or the entire network), where its value is directly proportional to the sharpness of the boundary given by $B(C)$.

Schaeffer introduced a community quality measure as the product of local and relative density [19]. The desired properties of many internal connections is measured by *local density* and few external connections by *relative density*. Let $d_{int}(C) = |\left\{ u \to v \;\middle|\; u, v \in C \right\}|$ be the internal degree of C and let $d_{ext}(C) = |\left\{ u \to v \;\middle|\; u \in C, v \in U(C) \right\}|$ be the external degree of C. Then local density [25] compares the internal degree of C with a clique of the same size:

$$\delta_l(C) = \frac{2 d_{int}(C)}{|C|(|C| - 1)}$$

And relative density is defined as the fraction of edges that are completely internal to C:

$$\delta_r(C) = \frac{d_{int}(C)}{d_{int}(C) + d_{ext}(C)}$$

The community quality measure coined by Schaeffer amounts to $f(C) = \delta_l(C).\delta_r(C)$. Note that only $\delta_r(C)$ is widely used as a simple and intuitive quality measure.

Evaluating a node v for addition to C is done by efficiently calculating the influence of v on the quality of C, $C' = C \cup v$, as follows. Let $k = |\left\{ v \to u \;\middle|\; u \in C \right\}|$ be be the amount of edges from v to C and $l = d(v) - k$ the other edges from v. Then the internal and external degree of C' can be incrementally computed by $d_{int}(C') = d_{int}(C) + k$ and $d_{ext}(C') = d_{ext}(C) - k + l$.

Chen, Zaane and Goebel coined local community quality ratio L using the average internal degree of the community and the average external degree of the boundary [20]. Internal connections are evaluated by L_{in} as the average internal degree of the community:

$$L_{int} = \frac{d_{int}(C)}{|C|}$$

And the external connections are evaluated by L_{ex} as the average external degree of the boundary:

$$L_{ext} = \frac{d_{ext}(C)}{|B(C)|}$$

Combining these leads to L:

$$L = \frac{L_{int}}{L_{ext}}$$

Efficiently evaluating the addition of a node v to C can be done analogous to the method proposed by Schaeffer. The algorithm proposed for using L [20] adds post processing, the examination phase, to the standard greedy agglomeration algorithm where past additions to the community are evaluated again.

2. Observations

During our benchmarks of various community definitions we found that local modularity seems to optimize the boundary of the community instead of the local community as defined by the intuitive notion. In some cases this measure is reluctant to add a node that causes a change in the boundary while it should do so to further improve, according to the intuitive notion of a community.

Let $d_{int}(C, v) = |\left\{ v \to u \;\middle|\; u \in C \right\}|$ be the amount of edges from node v to community C and let $d_{ext}(C, v) = |\left\{ v \to u \;\middle|\; u \notin C \right\}|$ be the amount of edges from node v to nodes outside of community C. Let $z(C, n) = d_{int}(C, n)/d_{ext}(C, n)$ be the fraction of edges towards C for node n.

Local modularity will only accept a node v that moves boundary node b to $C - B(C)$ in two cases:
1) $d_{ext}(C, v) > 0 \wedge z(C, v) > z(C, b)$, which makes sense.
2) v only has edges toward C, $d_{ext}(C, v) = 0$, and $z(C, b) < R(C)$.

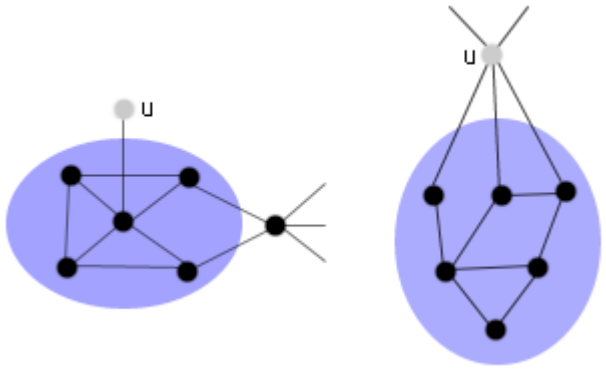

FIG. 3. The grey node, u, is currently not a member of C but intuitively it should be in both cases. However, not all community definitions seem to agree.

In the latter case v should always be added, the z ratio of b should not matter at all because adding a node that increases internal and decreases external edges is always a good thing. An example is illustrated in Figure 3a where node u is not added because it will remove a boundary node with a z ratio larger than $R(C)$, $R(C) = \frac{8}{11}$ vs $R(C \cup u) = \frac{4}{6}$.

Theoretically, L suffers similar problems to local modularity when nodes are added to C that move boundary nodes to $C - B(C)$. Because L_{ext} evaluates the average degree of the boundary it may optimize the boundary instead of the external connections of the community as a whole, the example in 3a applies to L as well. Currently $L(C) = \frac{7}{5}$ and adding the node u causes the quality to go down because L_{ext} still yields 1 and the average internal degree decreases, $L(C \cup u) = \frac{8}{6}$.

Another example where L behaves counter intuitive is illustrated in Figure 3b. Intuitively, node u belongs to the community because it is well connected and adding it causes a decrease in $d_{ext}(C)$. $L_{int}(C) = \frac{7}{6}$ and $L_{ext}(C) = \frac{3}{3}$ thus $L(C) = \frac{7}{6}$. Adding the node u causes the average outdegree of the boundary, L_{ext}, to double while we intuitively feel the boundary is improving. $L_{int}(C \cup u) = \frac{10}{7}$ and $L_{ext}(C \cup u) = \frac{2}{1}$ thus $L(C \cup u) = \frac{5}{7}$ and $L(C \cup u) < L(C)$.

The $f(C)$ quality measure is troubled by the example in 3a as well because the loss of local density outweighs the gain in relative density. We observe the shift in local and relative density before adding u, $\delta_l(C) = \frac{14}{20}$ and $\delta_r(C) = \frac{7}{10}$, and after adding u, $\delta_l(C \cup u) = \frac{16}{30}$ and $\delta_r(C \cup u) = \frac{8}{10}$. The source of this problem is clearly distinct from the local modularity and L issues where we think the quality function is not weighing the right properties (too much focus on the boundary instead of the community as a whole). For the $f(C)$ measure the balance of weighing internal and external community quality seems a bit off which is a common problem for any quality measure that needs to take two factors into account.

Besides boundary mutations there is another common problem with community definitions we would like to mention. In sparse networks there may be paths that consist of nodes with only one other neighbor each, a chain of nodes. If such a chain starts in the universe of a community some definitions are tempted to agglomerate the entire chain because it increases internal edges and does not worsen the external connections of the community. This example stresses the importance of balance between internal and external community strength.

We conclude that a community definition should be based on the intuitive concept of a community. And since the intuitive notion is twofold it feels natural that a community definition takes both the internal and the external aspects of community quality into account. As we have seen with $f(C)$, balance between the weight of multiple quality measures may have large influence on the outcome of the evaluation of a community.

We have elaborated various community definitions in theory and shown subtle differences in interpretation of the intuitive community definition. As they all have their niche and perform well on specific networks or desired community characteristics, choosing to use either depends on the problem at hand.

III. BROADENING THE LOCAL SCOPE

A strong advantage of global community identification algorithms over their local counterparts is knowledge of the entire network. The limited knowledge of the network is a handicap for local algorithms because that makes it hard to judge the consequences of adding a node to the community. For a local algorithm, there is no way of telling what lies beyond the universe of a community. For example, even though a node u in the universe may seem like a bad addition to the local community it may be a good addition after adding nodes beyond u as well. But how could the local algorithm know this when all it can evaluate are universe nodes?

A typical problem for local algorithms is their dependance on the right source node. If the algorithm starts agglomerating from a position such that it will only have connections to high outdegree nodes right at the start, it will usually terminate. Figure 4 illustrates such a situation. The cause is straight forward, local algorithms select the best candidate from the universe until the community can no longer improve. And if the algorithm sees no more improvements it will terminate, thus in this example the algorithm is likely to stop with $|C| = 2$ as a result.

A commonly applied band-aid solution is to make the algorithm ignore its stopping criteria until hitting a predetermined size or quality. However, this will only solve the problem with starting nodes up to a given predetermined threshold. Suppose we find a local community C_n with n nodes and some μ score that cannot improve by directly adding any universe node. Perhaps adding a universe node and the node after will lead to a community

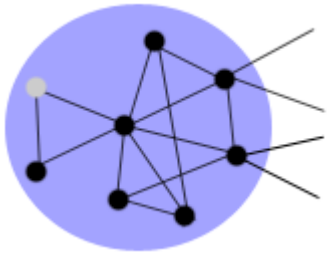

FIG. 4. Suppose a local algorithm starts at the grey node, how can the community evolve beyond two nodes?

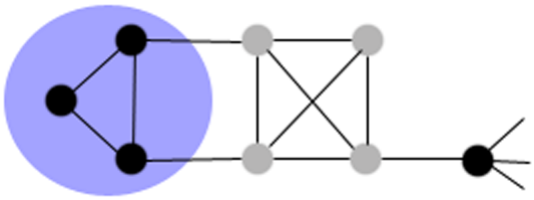

FIG. 5. A local identification algorithm will halt at the currently identified community because it cannot see the benefit of adding nodes beyond the universe.

C_{n+m} such that $\mu(C_{n+m}) > \mu(C_n)$. A local approach could not have seen this improvement beyond the universe while a global approach can because it has more contextual information. Such shortsightedness causes local algorithms to halt at every minor barrier they face.

We suggest the improvement of local identification algorithms in general by adding more contextual information to the selection criteria. Instead of evaluating each node in the universe by its edges one could check its neighbors as well. Perhaps adding a node from the universe and some of its neighbors combined yields a better result than only adding the universe node. By making a local algorithm look ahead further than one step we decrease the shortsightedness and allow for a more informed and balanced judgement on community membership, at the cost of computational and situational crawling complexity.

And if there are multiple nodes outside of a community that could be a member of that community it may be worthwhile to check the combination of these external nodes. The reason for combining universe nodes and nodes beyond them is the expectance that nearby nodes may have connections among each other. After all, two universe nodes u and v are both just one edge away from a tightly connected community C so there is a chance that u and v are related as well. If u and v are connected, adding both of them to C may result in more internal and less external edges than only adding either of them.

We propose the following example application of adding more contextual information to the selection criteria. For each universe node $u \in U(C)$ we could check all possible combinations of adding that node and its neighbors to verify the potential improvement on short term by u. We then find the best candidates as follows:

$$\begin{aligned} nb(u) &= \{v \in N \mid u \to v\} \\ cand(C) &= \{powerset(nb(u)) \cup u \mid u \in U(C)\} \end{aligned}$$

where the best candidate is the node set from $cand(C)$ that yields the highest $\mu(C)$ increase upon agglomeration. This extension can be applied to most commonly used greedy local agglomeration algorithms.

Figure 5 illustrates an example graph where a local algorithm identified a community of four nodes so far. By only looking at the universe the algorithm will decide the community is fine as it is, there is no room for improvement. However, if we look at the network we observe that the grey nodes are a welcome addition because they are well connected to the current community and that will result in less external connections. By making the algorithm look ahead as suggested above it will come to the same conclusion.

If algorithm speed, thus time complexity, is less of an issue we could further extend the usage of contextual information. Following the above algorithm one could consider all possible combinations of candidates from the universe and beyond to find the optimal additions to a community at a given agglomeration step. However, the time complexity will increase rapidly so we might limit the scope by considering only the best two neighbors for each universe node combined with all universe nodes. So this example approach would add each universe node and its best two sets from $cand(C)$ to a stack and then compute the powerset of this stack to generate all community candidate sets. The set with the best μ influence is added.

We only consider extra node information one step beyond the universe because of the time complexity involved. Especially with dense networks and high node degrees the time complexity of the greedy agglomeration algorithm extensions suggested here can increase rapidly. Also, the further away a node is from the community, the less likely it is (at that time) to be a member of that community so it does not seem useful to look very far ahead. Further research should consider different distances for the lookahead approach and determine an optimum (if any).

We gave two examples of adding more contextual information to the selection criteria of local community identification algorithms here to illustrate the concept, many more (effective and efficient) varieties are possible.

In conclusion, we suggest the usage of more contextual information than the universe when evaluating candidates for a local community. We proposed two example additions to local identification algorithms following this paradigm. Using additional contextual information is a broad concept with its roots in the comparison between local and global algorithm advantages that can be applied to many existing and future local community identifica-

tion algorithms. We aim at eliminating shortsightedness and decreasing the gap between relevant network knowledge of global and local methods. Thus we can make a more informed decision about community candidates.

IV. VALIDATION

Many networks have a varying density and we think the addition to local identification algorithms suggested in section three can be a major improvement in situations where nodes are sparsely connected. If there are only few universe nodes or the neighbors of the universe nodes are sparsely connected it can be hard to judge whether the local community has been correctly identified or if further improvement lies beyond. Another common problem for local community identification is the dependance on a well connected source node, local algorithms tend to struggle when starting with a low degree node. The aim of our experiment is to verify an increase in community quality for both problems when the improved algorithm is applied.

A. Experiment setup

We implement a basic local identification algorithm that may optionally use our suggested lookahead strategy proposed in section three and we test the commonly used local modularity and relative density community definitions. Because our aim is to consider if we can identify a stronger community with the lookahead strategy than without we generate synthetic networks with a flat community structure.

There are several methods for generating synthetic networks that can be used to benchmark community identification algorithms [26–29]. For our test we will generate a network according to the Barabsi–Albert model [2] and rewire it to create a flat community structure as proposed by Bagrow [26]. We simplify the network generated by the Barabsi–Albert model by removing multiple edges between nodes. The rewiring is done by creating k sets of nodes (representing the communities) in the network and then rewiring inter-community edges to intra-community edges while preserving the degree distribution. The resulting benchmark networks contain 128 nodes equally divided among 4 communities.

Since we aim to verify the improvement of local identification algorithms in the area where these often struggle we will generate graphs with an average degree of 4, 5 and 8. This will result in networks that contain a lot of potentially troublesome source nodes. We run four tests: a standard local algorithm and the lookahead algorithm with both LM and RD as quality measures. These tests are applied to every node in the network.

B. Quantifying Community Similarity

The quality of a community identification algorithm is commonly evaluated by measuring the similarity between the algorithm output and some reference set of communities. These reference communities are the supposed "real" communities and a higher similarity with the reference set indicates better algorithm quality.

A widely used measure for such similarity is *mutual information* which originates from information theory. The symmetric mutual information measure for two discrete random variables X and Y, denoted as $I(X,Y)$, indicates how much information they share. In other words, mutual information measures how much we know about X when Y is known and vice versa [30].

$$I(X,Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y).log\left(\frac{p(x,y)}{p_m(x).p_m(y)}\right)$$

where p_m is the marginal probability distribution function ($Pr(X=x)$). There is no upper bound and therefor this measure is commonly normalized to the interval [0..1] for practical comparison as follows.

Let the Shannon entropy be defined as:

$$H(X) = \sum_{x\in X} p(x).log\frac{1}{p(x)}$$

From the observation that $I(X,Y) \leq min(H(X),H(Y))$ and $H(X) = I(X,X)$ follows that the mutual information measure may be normalized by the arithmetic or geometric mean of $H(X)$ and $H(Y)$ [31].

When used to compare two sets of communities we observe that every element of X and Y is a community which consist of unique nodes. Let N be the amount of nodes in the entire network. Then due to the homogenous probability distribution within these communities the mutual information is defined as [31]:

$$I(X,Y) = \sum_{x\in X}\sum_{y\in Y} \frac{|x\cap y|}{N}.log\left(\frac{\frac{|x\cap y|}{N}}{|x|.|y|}\right)$$

and then the *normalized mutual information* using the geometric mean of the Shannon entropy is defined as:

$$NMI(X,Y) = \frac{\sum_{x\in X}\sum_{y\in Y}\frac{|x\cap y|}{N}.log\left(\frac{|x\cap y|.N}{|x|.|y|}\right)}{\sqrt{\left(\sum_{x\in X}\frac{|x|}{N}.log\frac{N}{|x|}\right)\left(\sum_{y\in Y}\frac{|y|}{N}.log\frac{N}{|y|}\right)}}$$

However, we do not intend to compare two sets of communities as is customary for evaluating global community identification algorithms. Instead, we are interested in the similarity between a single community, the one identified by a local algorithm, and some reference community. When the similarity between two single local communities is desired, we do not consider the rest of

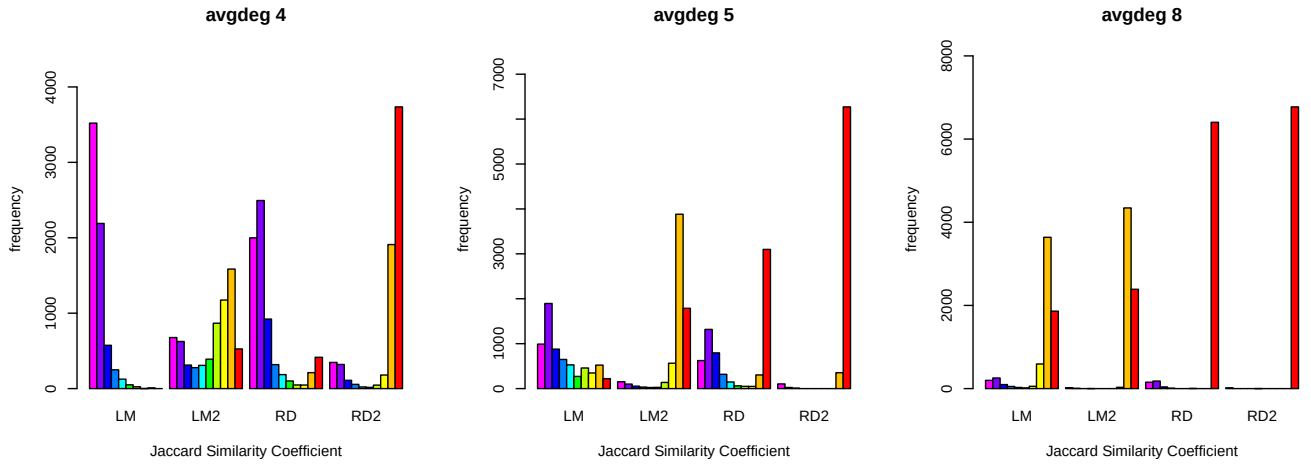

FIG. 6. A barplot shows the frequency of JSC scores for tests with Local Modularity and Relative Density with both the standard algorithm and the lookahead enabled algorithm. The bars indicate how often any score in the range of [0..1] occurs (divided among 10 bars).

the network relevant. In some cases there may even be a lack of knowledge about the entire domain, for instance when considering the similarity between a manually constructed community in the web graph and the result of a local algorithm. We observe that using (normalized) mutual information for comparing two single local communities can bring an undesired side effect and is therefor not a suitable measure for our experiment.

Let X and Y only contain a single community. In case we find a single community which is equal to the reference community and the entire graph then $X = Y$ and $|X| = |Y| = N$ thus $I(X,Y) = 0$ and $NMI(X,Y) = 0$. However, when the same X and Y would have been situated in a network where $N > |X|$ (and $X = Y$) then $NMI(X,Y) > 0$. Another example where using mutual information to compare two single communities yields undesired results: Let $|X| = N$ and $X \subset Y$. Then $|X \cap Y|.N = |X|.|Y|$ thus $I(X,Y) = 0$ and $NMI(X,Y) = 0$.

So for the purpose of quantifying the similarity between two communities we will adopt the Jaccard Similarity Coefficient (JSC) instead of the mutual information measure that we deem more suited for evaluating global approaches to community identification. The Jaccard Similarity Coefficient is defined as the generality of both sets divided by their commonality:

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

C. Results

Running these tests on 50 generated networks yields the plot of the JSC score frequency shown in Figure 6. In the networks with an average degree of 4 and 5 we observe a significant increase in high similarity for both the local modularity and the relative density community quality measure when the lookahead algorithm is applied. Networks with a larger average degree have a relatively low gain of the lookahead algorithm as illustrated by the similarity plot on a network with an average degree of 8. The plot confirms our theory of the lookahead algorithm improving community quality and decreasing the dependance on a specific source node. This is indicated in the plot by a higher frequency in high JSC scores and a lower amount of outliers.

We do observe a few outliers still remaining when applying the lookahead algorithm, it is not the final solution to local algorithm effectiveness. There are a couple of reasons why even the lookahead algorithm is struggling for some source nodes.

First of all, when the boundaries of two communities are not very sharp and the source node is a boundary node (as defined by the synthetic graph structure) the algorithm may start of in the wrong direction and identify the wrong community. Suppose we start with node v that is a member of community C according to the synthetic structure and the algorithm identifies community $C' \cup v$ where C' is another community defined by the synthetic structure. Then the result of the algorithm may be a quite strong community but according to the similarity measure it is really bad because there is very little overlap between the reference community and the found community.

The local algorithm may also find a strong community that is a subset of the community it is supposed to find. The gap between the found community and agglomerating until the algorithm identifies the larger and stronger community may be too large for the lookahead algorithm to recognize. There lies a tradeoff between the time complexity of the lookahead algorithm and the effectiveness of identifying improvement beyond the universe candidates.

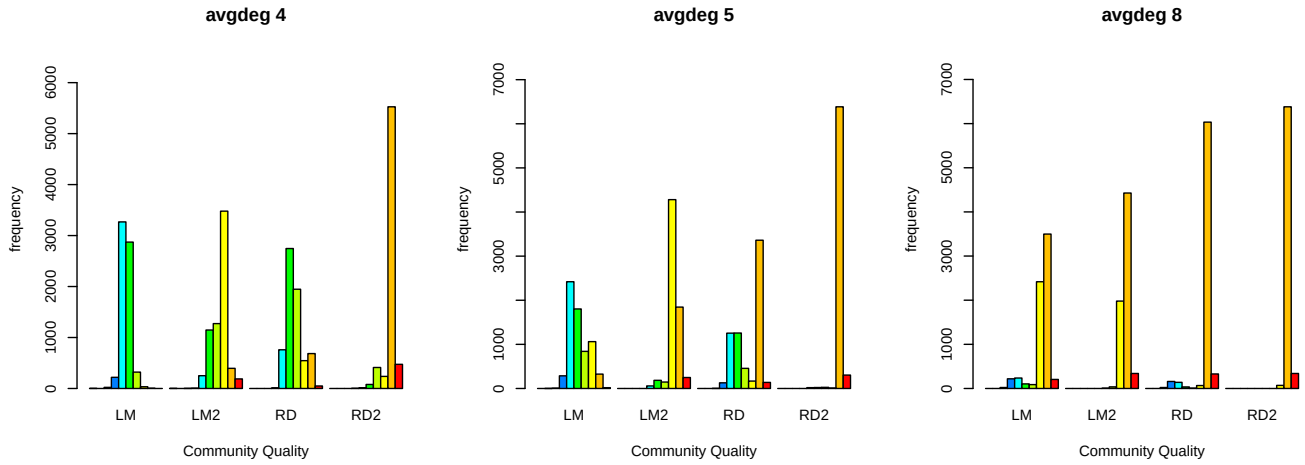

FIG. 7. A barplot shows the frequency of μ scores for tests with Local Modularity and Relative Density with both the standard algorithm and the lookahead enabled algorithm. The bars indicate how often any score in the range of [0..1] occurs (divided among 10 bars).

Since the ideal community is hard to compute due to time complexity constraints the quality of a community (as identified by an algorithm) is usually measured by comparison to the community structure as created in the synthetic benchmark network. However, when the goal of a local algorithm is to identify the strongest community around a source node (as is the case in this paper) that synthetic reference community may not be the strongest community. For instance, there may be some community definition μ that considers a reference community plus some nodes from a neighboring reference community as a better community. The cause is the nature of algorithms that construct synthetic graph structure. They take a set of nodes and rewire up to a certain ratio of internal and external connections without any regard of nodes outside of that synthetic community, nor is there any regard for the presence of a stronger subcommunity. We note that the measured quality of any algorithm as compared to some reference community is dependant on the quality of the reference community as well.

While running our tests we observed this phenomenon as the algorithm identified local communities that are stronger than the synthetic community we used as a reference in a few cases. While this is intuitively interpreted as a desirable outcome because goal is to identify the strongest community, this is shown as a bad JSC score in the plot since it only considers the similarity between two given sets. In our experiment we assume the reference community is the best community out there so an improved μ score is not taken into account.

The plot in Figure 7 shows the μ scores for all tests. When comparing these results with the similarity plot of Figure 6 we observe that the amount of outliers in the JSC score are larger than the amount of outliers in the μ score. This supports the suggestion that some similarity measure outliers can be explained by the algorithm identifying a strong community that is quite different than the synthetic reference community. So the algorithm may behave as desired in that case and the problem could lie with the reference community being far from the ideal community. If this is the case, the results presented here would be even more positive. Further research on the structure and quality of the reference community as generated by synthetic graph construction algorithms could provide more insight.

V. CONCLUSIONS

In this paper we have reviewed problems of widely used local community identification methods and propose an improvement that can be applied to local community identification algorithms in general.

We suggest adding contextual information beyond the universe of a local community when evaluating local community candidates in order to eliminate the shortsightedness of a local algorithm and thereby allowing more informed evaluation of community candidates. This improvement to local algorithms will increase the quality of locally identified communities in general and decrease the dependency on specific source nodes, which is a common problem for local algorithms.

We provided an example approach for this concept and ran benchmark tests on local algorithms with and without the suggested improvement. The results of superior quality of local communities identified by the improved local algorithm look promising, especially in situations where the algorithm is exploring low degree nodes. The results also show a decreased amount of outliers in the quality of the local communities yielded by the improved local algorithm, this indicates the decreased dependency on specific source nodes.

We intend to perform further research on the application, effectiveness and computational cost of (different

variations of) adding more contextual information to selection criteria for local algorithms. Other areas of interest for improving local identification algorithms include synthetic benchmark networks with guaranteed properties (such as optimal clusters), dynamic stopping criteria and removal strategies for local community members in order to improve a local community.

[1] S.H. Strogatz, "Exploring complex networks," Nature, **410**, 268–276 (2001).

[2] Reka Albert and Albert-Laszlo Barabasi, "Statistical mechanics of complex networks," Reviews of Modern Physics, **74**, 47–97 (2002), arXiv:cond-mat/0106096v1.

[3] M.E.J. Newman, "The structure and function of complex networks," SIAM Review, **45**, 167–256 (2003), arXiv:cond-mat/0303516v1.

[4] Gergely Palla, Albert-Laszlo Barabasi, and Tamas Vicsek, "Quantifying social group evolution," Nature, **446**, 664–667 (2007), arXiv:0704.0744v1.

[5] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Grke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner, "On finding graph clusterings with maximum modularity," Lecture Notes in Computer Science, **4769**, 121–132 (2007).

[6] J. Šíma and S.E. Schaeffer, "On the np-completeness of some graph cluster measures," Lecture Notes in Computer Science, **3831**, 350357 (2006), arXiv:cs/0506100v1.

[7] "The size of the world wide web," `http://www.worldwidewebsize.com`.

[8] Leon Danon, Jordi Duch, Albert Diaz-Guilera, and Alex Arenas, "Comparing community structure identification," J. Stat. Mech., P09008 (2005), arXiv:cond-mat/0505245v2.

[9] Andrea Lancichinetti and Santo Fortunato, "Community detection algorithms: a comparative analysis," Phys. Rev. E, **80**, 056117 (2009), arXiv:0908.1062v1.

[10] M.E.J. Newman and M. Girvan, "Finding and evaluating community structure in networks," Phys. Rev. E, **69**, 026113 (2004), arXiv:cond-mat/0308217v1.

[11] M.E.J. Newman, "Fast algorithm for detecting community structure in networks," Phys. Rev. E, **69**, 066133 (2004), arXiv:cond-mat/0309508v1.

[12] Benjamin H. Good, Yves-Alexandre de Montjoye, and Aaron Clauset, "The performance of modularity maximization in practical contexts," (2009), arXiv:0910.0165v1.

[13] Gergely Palla, Imre Derenyi, Illes Farkas, and Tamas Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," Nature, **435**, 814 (2005), arXiv:physics/0506133v1.

[14] Erin N. Sawardecker, Marta Sales-Pardo, and Lus A. Nunes Amaral, "Detection of node group membership in networks with group overlap," Eur. Phys. J B, **67**, 277–284 (2009), arXiv:0812.1243v1.

[15] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri, "Extending the definition of modularity to directed graphs with overlapping communities," J. Stat. Mech., P03024 (2009), arXiv:0801.1647v4.

[16] M. Sales-Pardo, R. Guimera, A. Moreira, and L. Amaral, "Extracting the hierarchical organization of complex systems," Proc. Natl. Acad. Sci. U.S.A., **104**, 15224–9 (2007), arXiv:0705.1679v1.

[17] Aaron Clauset, Cristopher Moore, and M.E.J. Newman, "Hierarchical structure and the prediction of missing links in networks," nature, **453**, 98–101 (2008), arXiv:0811.0484v1.

[18] Aaron Clauset, "Finding local community structure in networks," Phys. Rev. E, **72**, 026132 (2005), arXiv:physics/0503036v1.

[19] S.E. Schaeffer, "Stochastic local clustering for massive graphs," Lecture Notes in Artificial Intelligence, **3518**, 354360 (2005).

[20] Jiyang Chen, Osmar Zaane, and Randy Goebel, "Local community identification in social networks," ASONAM, 237–242 (2009).

[21] J.P. Bagrow and Erik Bollt, "A local method for detecting communities," Phys. Rev. E, **72**, 046108 (2005), arXiv:cond-mat/0412482v2.

[22] Jeffrey Baumes, Mark Goldberg, and Malik Magdon-Ismail, "Efficient identification of overlapping communities," Lect. Notes Comput. Sc., **3495**, 27 (2005).

[23] Andrea Lancichinetti, Santo Fortunato, and Janos Kertesz, "Detecting the overlapping and hierarchical community structure of complex networks," New Journal of Physics, **11**, 033015 (2009), arXiv:0802.1218v2.

[24] Conrad Lee, Fergal Reid, Aaron McDaid, and Neil Hurley, "Detecting highly overlapping community structure by greedy clique expansion," (2010), arXiv:1002.1827v1.

[25] Ulrik Brandes, Marco Gaertler, and Dorothea Wagner, "Experiments on graph clustering algorithms," in *In 11th Europ. Symp. Algorithms* (2003) pp. 568–579.

[26] J.P. Bagrow, "Evaluating local community methods in networks," J. Stat. Mech., P05001 (2008), arXiv:0706.3880v2.

[27] A. Arenas, A. Daz-Guilera, and C.J. Perz-Vicente, "Synchronization processes in complex networks," Physica D, **224**, 27–34 (2006), arXiv:nlin/0610057v1.

[28] A. Arenas, A. Daz-Guilera, and C.J. Perz-Vicente, "Synchronization reveals topological scales in complex networks," Physical Review Letters, **96**, 114102 (2006), arXiv:cond-mat/0511730v2.

[29] Andrea Lancichinetti and Santo Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," Physical Review E, **80**, 016118 (2009), arXiv:0904.3940v2.

[30] Thomas M. Cover and Joy A. Thomas, *Elements of Information Theory* (Wiley, 1991).

[31] Alexander Strehl and Joydeep Ghosh, "Cluster ensembles - a knowledge reuse framework for combining multiple partitions," Journal on Machine Learning Research (JMLR), **3**, 583–617 (2002).