Context-Sensitive Detection of Local Community Structure

L. Karl Branting
The MITRE Corporation
7525 Colshire Drive
McLean, Virginia
USA
Email: lbranting@mitre.org

***Abstract*—Local methods for detecting community structure are necessary when a graph's size or node-expansion cost make global community-detection methods infeasible. Various algorithms for local community detection have been proposed, but there has been little analysis of the circumstances under which one approach is preferable to another. This paper describes an evaluation comparing the accuracy of five alternative local community-detection algorithms in detecting two distinct types of community structures—vertex partitions that maximize modularity, and link clusters that maximize partition density—in a variety of graphs. In this evaluation, the algorithm that most accurately identified modularity-maximizing community structure in a given graph depended on how closely the graph's degree distribution approximated a power-law distribution. When the target community structure was partition-density maximization, however, an algorithm based on spreading activation generally performed best, regardless of degree distribution.**

I. Introduction

Many complex systems—such as power grids, nervous systems, sports leagues, collaborating researchers and musicians, and the World Wide Web—are amenable to representation as a graph consisting of vertices (representing entities) and edges (representing relationships or events). Communities within such graphs, consisting of subgraphs whose vertices are more highly connected to each other than to vertices outside the subgraph, often correspond to meaningful components of the systems represented by the graphs. Detection of such communities can therefore be a powerful tool for understanding complex systems.

Numerous algorithms of varying complexity have been developed to identify communities in graphs. One popular approach is to search for a partition of the vertices of a graph that optimizes a global utility function, such as *modularity* [New04]. A related approach searches for an edge partition that maximizes global *partition density* [YYA10]. The partition-density maximizing edge partition typically induces overlapping vertex communities.

In many cases it is not feasible to determine the *globally* optimal community structure, either because the entire graph is too large to fit in memory or because the cost in time or other resources of accessing the entire graph is prohibitive. Under these circumstances, the search for community structure must be limited to to the neighborhood of the graph *local* to a given query vertex.

The process of local community search typically consists of incrementally adding individual vertices to a community initialized with a query vertex, sometimes followed by, or interleaved with, a winnowing step that removes vertices that detract from the community structure [Cla05], [LWP08], [Bag08], [CZR09], [Bra10a]. Any implementation of this process requires policies for (1) selection (how to choose the next vertex to add to the community), (2) termination (when to stop adding vertices), and (3) filtering (which vertices, if any, to remove from the community).

The focus of this work is on improving vertex selection, independent of choice of termination or filtering policies. There are two justifications for this focus. First, it is typically easier to optimize individual design elements separately than to try to optimize all simultaneously. Second, termination and filtering policies are necessarily dependent on the characteristics of the selection policy. The more accurate the selection policy, the fewer the vertices that must be selected to obtain all vertices in a given community and the fewer the vertices that must be filtered to remove all nodes not in that community.

The selection policies of alternative local community detection algorithms differ in their policies regarding links from a candidate for selection to vertices outside of the current community. Some algorithms are *xenophobic* in that they penalize candidates in proportion to the number of their edges to non-community vertices. *Non-xenophobic* algorithms ignore or reward such edges.

Section II sets forth a schema common to many local algorithms and shows that these algorithms can be distinguished in terms of this schema based on whether their candidate selection criteria are xenophobic. A new evaluation criterion for local community detection algorithms that takes account of the relative centrality of vertices within the target community is proposed in Section III. Section IV describes a comparative evaluation on a set of standard natural and artificial graphs. This evaluation shows that the relative performance of xenophobic and non-xenophobic algorithms depends on (1) the edge distribution of the graph to which they are applied, (2) the target community structure, and (3) the centrality criterion for vertices within the target community.

II. Algorithms for Local Community Detection

Many local community detection algorithms can be viewed as implementing a common schema that assigns each vertex in the graph to one of three sets at each processing step:

- C, the Community under construction, which is typically initialized with the query vertex.
- N, Neighboring vertices not in C but sharing an edge with at least one element of C.
- U, Unexplored vertices, *i.e.*, those not adjacent to any element of C.

Optionally, C can be further partitioned into a boundary, $C_{boundary}$, consisting of every node in C that has at least one edge to a node in N, and C_{core}, which consists of the vertices in C that have no edges to N, *i.e.*, $C_{core} = C - C_{boundary}$. The local community detection algorithm schema is as follows:

Algorithm 1: Local-community structure algorithm schema

```
$C \leftarrow \{queryVertex\}$
$N \leftarrow neighbors(queryVertex)$
while (!terminationCriterion) do
    select the ‘best’ vertex $n \in N$
    $C \leftarrow C \cup \{n\}$
    $N \leftarrow (N - n) \cup neighbors(n) - C$
end
return filter(C)
```

Local community detection algorithms differ in their criterion for selecting the ‘best’ vertex $n \in N$. Note that in this schema, all neighbors of each vertex $n \in N$ are known, whereas neighbors of vertices in U are in general not known. Edges are assumed to be undirected.

A. Xenophobic Vertex Selection

The vertices in a community typically have more edges to vertices in the same community (internal edges) than to vertices outside the community (external edges). Conversely, vertices outside the community typically have more external than internal edges. Most local community detection algorithms use heuristics to try to estimate the relative number of internal and external edges for the actual community based on the current partial community under construction by the algorithm. Unfortunately, such estimates are necessarily approximate if the partial community is incomplete.

Clauset [Cla05] proposed a vertex selection criterion under which the vertex is selected that makes the largest increase (or smallest decrease) in *local modularity*, $R = \frac{I}{T}$, where T represents the number of edges incident to $C_{boundary}$ (*i.e.*, including both edges between pairs of nodes in C and those connecting a node in C to a node in N), and I represents the number of edges incident to $C_{boundary}$ that are internal to C (*i.e.*, that connect pairs of nodes in C). The intuition behind maximizing R is that R “is directly proportional to sharpness of the boundary given by $C_{boundary}$.” The procedure “avoids crossing a community boundary until absolutely necessary” [Cla05].

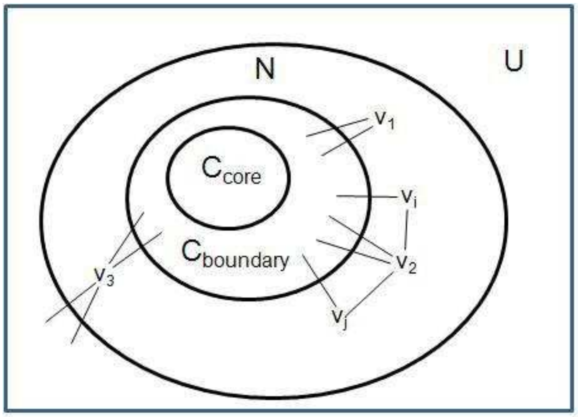

Fig. 1. Vertices v_1, v_2, and v_3 are candidates for addition to C.

A second selection criterion, termed outwardness, was proposed in [Bag08]. The outwardness of a vertex v, Ω_v, is:

$$\Omega_v = \frac{(k_v^{out} - k_v^{in})}{k_v} \tag{1}$$

where k_v is the degree of vertex v, k_v^{out} is the number of edges from v to vertices outside of the community C, (*i.e.*, to N or U), and k_v^{in} is the number of edges from v to vertices in C. At each stage, the vertex $v \in N$ with the lowest outwardness is selected to be moved to C, breaking ties at random.

A third selection criterion, based on [LWP08] is to choose the vertex that maximizes $M = \frac{ind(C)}{outd(C)}$, the ratio of $ind(C)$, the number of edges connecting pairs of nodes in C, to $outd(C)$, the number of edges connecting nodes in C to nodes outside of C.[1]

These three selection policies—(l) maximizing local modularity (L), (2) minimizing outwardness (Ω_v), and (3) maximizing M—share a common implicit assumption that, for any node $n \in N$, both edges from n to nodes in U and edges from n to other nodes in N make n less likely to be in the target community. This *xenophobic* assumption can sometimes lead a node that is very likely to be of low centrality to be selected before a node that might be of higher centrality.

Consider, for example, vertices v_1, v_2, and v_3 shown in Figure 1. Vertex v_2 may have higher centrality in the actual community than v_1 or v_3 because there are multiple paths from v_2 into C through edges to v_i and v_j, whereas no such alternative paths to C are possible for v_1, and no equally short alternative paths exist for v_3. However, v_2’s outwardness ($\frac{2-2}{4} = 0$) is higher than the outwardness of v_1 ($\frac{0-2}{2} = -1$) and is the same as the outwardness of v_3 ($\frac{2-2}{4} = 0$). Moreover,

[1] The algorithm of [LWP08] considers each $n \in N$ in ascending order of degree, adding to the community each n whose addition to C would increase M. Each element of C whose removal would increase M without disconnecting C is then removed. These two steps are repeated until no new vertices are added. The procedure described here differs from the algorithm of [LWP08] in that it selects the node that maximizes M, rather than the lowest degree node for which $\Delta M > O$, and in that it is purely a node-selection policy, with no node filtering.

local modularity would be higher after adding v_1 ($\frac{I+2}{T+0}$) than after adding v_2 or v_3 ($\frac{I+2}{T+2}$). Finally, adding v_1 would make $M = \frac{ind(C)+2}{outd(C)-2}$, which is higher than M after adding v_2 or v_3, $\frac{ind(C)+2}{outd(C)+0}$. Thus, under all three selection policies, v_1 would be selected before v_2, and v_2 and v_3 would be treated identically even though v_2 is more strongly connected to C than is v_3.

B. Non-Xenophobic Vertex Selection

The observation that there are scenarios in which maximizing local modularity, minimizing outwardness, and maximizing M all can lead low-centrality vertices to be selected before potentially higher-centrality vertices suggests that better performance might sometimes be obtained by selection criteria that distinguish edges internal to N from those between vertices in N and vertices in U, rewarding the former and ignoring the latter. Two such approaches to such selection criteria are described below.

The first is *spreading activation*, in which excitation is propagated along links from the query vertex to each node that has been expanded. The node $n \in N$ having the highest activation is selected to be added to C. This procedure rests on an implicit assumption that activation represents the strength of the connections through the graph from the query vertex to n. A second approach is density-based selection, in which the node $n \in N$ that contributes to the most highly interconnected community is selected at each step, regardless of the number of links from n to U. These two approaches ignore links from a candidate node $n \in N$ to U, and both reward edges from n to other nodes in N.

Spreading Activation

Numerous approaches to spreading activation have been explored in the history of computer science, *e.g.*, [CL75], [Cre97]. *MaxActivation* is a particularly simple form of spreading activation appropriate for local community detection [Bra10a].

In MaxActivation, activation is propagated outward from the query vertex. Each node's activation is the sum of activations received along each edge from a node of equal or lesser distance to the query vertex. The activation received along an edge is the sender's activation multiplied by a global edge-attenuation factor. To avoid ordering effects, updates of all vertices at a given distance from the query vertex are performed concurrently.

In the MaxActivation algorithm for selecting the highest-activation vertex, set forth below in Algorithm 2, the symbol δ represents the attenuation factor, $0.0 < \delta \leq 1.0$. Activation of vertices can be calculated incrementally after each update to C, but for simplicity of presentation the algorithm is shown below as applied in batch mode to all the vertices in $C \cup N$.

If $\delta < \frac{1}{\arg\max_{v \in G}(deg(v))}$, then the activation of each vertex v is guaranteed to be a monotonically decreasing function of the path length from v to the query vertex. MaxActivation doesn't permit any activation to flow from vertices farther from the query vertex to vertices closer to the query vertex and permits activation between vertices at the same distance from the query vertex to propagate only one step. MaxActivation is thus *non-xenophobic*, since edges from v to vertices in U are ignored (having no effect on v's activation) and edges from v to vertices in N increase v's activation (since activation flows to v from each such vertex).[2]

Algorithm 2: MaxActivation Node Selection Algorithm

```
queryVertex.activation ← $1.0$
currentPly ← {queryVertex}
previousPly ← $\phi$
while (currentPly $\neq \phi$) do
    nextPly ← {v | v ∈ (C ∪ N) ∧ ∃ edge(v,w) ∧ w ∈ currentPly ∧ v ∉ currentPly ∧ v ∉ previousPly}
    foreach v ∈ nextPly do
        v.activation ← $0.0$
        v.tmp ← $0.0$
    end
    spread activation from current to next ply
    foreach {edge(w,v)| w ∈ currentPly ∧ v ∈ nextPly} do
        v.activation += w.activation $*\delta$
    end
    spread activation between members of nextPly
    foreach {edge(w,v)| w, v ∈ nextPly } do
        v.tmp += w.activation $*\delta$
        w.tmp += v.activation $*\delta$
    end
    sum activation from both sources
    foreach v ∈ nextPly do
        v.activation += v.tmp
    end
    update plies
    previousPly ← currentPly
    currentPly ← nextPly
end
return $\arg\max_{n \in N}$(n.activation)
```

Density-Based Selection

An alternative non-xenophobic selection criterion is to select the $n \in N$ that makes the community as interconnected as possible. MaxDensity [Bra10a], shown below in Algorithm 3, is an approach to density-based selection that uses a simple criterion for this selection: choosing the $n \in N$ that has the most edges to vertices in C. Ties are broken by choosing the $n \in N$ with the most edges to other vertices in N, and any

[2] An alternative approach to spreading activation based on the Katz index [Kat53] assigns activation to node $n \in N$ equal to $= \sum_{l=1}^{\infty} \delta^l \cdot |\{w_l(q,n)\}|$, where $\{w_l(q,n)\}$ is the set of all walks of length l from query vertex q to vertex n and δ is an attenuation factor. This approach exhibited behavior very similar to that of MaxActivation in the evaluation set forth below and for brevity is omitted.

remaining ties are broken by selecting the $n \in N$ with the shortest path to the query vertex.

Like MaxActivation, MaxDensity ignores edges from v to vertices in U, and rewards edges from v to vertices in N, since ties are broken by selecting the vertex with largest number of such edges.

Algorithm 3: MaxDensity Node Selection Algorithm

```
D ← {n | $\arg\max_{n \in C}$(|{edge(v,n), v ∈ C }|) }
if (|D| > 1) then
    D ← {n | $\arg\max_{n \in D}$(|{edge(v,n), v ∈ N }|)}
    if (|D| > 1) then
        D ← {n | $\arg\min_{n \in D} pathlength(n, query)$}
    end
end
return random member of D
```

III. Evaluation Criteria for Local Community Detection

Ideally, a local community-detection algorithm would be evaluated by comparing its *return set*, *i.e.*, the community that it finds, to the optimal community under some global criterion. In practice, this approach to evaluation is possible only on graphs whose size and accessibility make global optimization tractable. However, comparative evaluations on tractable graphs may generalize to graphs for which global optimization is intractable. Accordingly, the evaluation set forth below is based on graphs small enough to be amenable to global optimization.

Evaluation relative to a global criterion depends on the choice of both the community-structure criterion to be optimized (*e.g.*, modularity or partition density) and the utility function for vertices in the optimal community *e.g.*, weighing vertices by degree or betweenness centrality within the target community.

The evaluation described below compares alternative local community-structure algorithms relative to two distinct global criterion: the vertex partition that maximizes *modularity* [New04]; and the edge partition that maximizes *partition density* [YYA10]. Modularity is the best-known global community-structure criterion and is widely used despite its known limitations, such as a resolution limit and a bias toward equal sized communities [FB07]. The partition-density criterion for link clustering is not subject to a resolution limit and permits overlapping communities, but produces communities somewhat different from those produced by maximizing modularity.

For a given seed vertex, s, and global community criterion, a *target community* is a community containing s that would be optimal under the criterion. For example, if the criterion were maximal modularity, then the target community for s would be the community containing s in highest-modularity level of the dendrogram created by the algorithm of [New04].

Given a target community T, the quality of a local community-detection algorithm can be calculated by means of a utility function, $util_T$, defined over the vertices of T. For example, the quality of a k-element return set C can be measured as the sum of the utilities of C's vertices, $\sum_{v \in C} util_T(v)$. This sum can be normalized onto the [0.0 .. 1.0] interval by dividing it by the sum of the k highest utility vertices of the community. The resulting measure of solution quality is termed *Normalized Utility-Weighted Recall* (NUWR). The Normalized Utility-Weighted Recall of community C with respect to target community T, NUWR, is shown in equation 2:

$$NUWR_T = \frac{\sum_{v \in C} util_T(v)}{\arg\max_{S \subseteq T, |S| = min(|C|,|T|)} \sum_{v \in S} util_T(v)} \tag{2}$$

NUWR formalizes the intuition that if two return sets differ only in a single pair of vertices with different utilities, the return set with the higher utility vertex is preferable to the return set with the lower utility vertex. Similarly, if every vertex in the target community has identical utility, then all return sets consisting of k community vertices will have identical NUWR, consistent with the intuition that all such return sets are equally good. Local community extraction algorithms can be ranked by comparing the NUWRs of the return sets of each algorithm when search is terminated, *e.g.*, when k vertices have been expanded.

The evaluation below used three different vertex utility functions:

1) Degree centrality, the proportion of edges in T that are incident to a given vertex n.
2) Betweenness centrality, the proportion of geodesics between pairs of vertices in the target community that traverse n [WF94].
3) Membership, assigning the same value, 1.0, for every $n \in T$ and 0.0 for all other vertices.

For all three vertex utility functions, $util(n) = 0$ for $n \notin T$. Note that if $k = |T|$ and the vertex utility function is 'membership,' then NUWR$_T$ is equivalent to recall, since under these circumstances $NUWR_T = \frac{|truePositives|}{|T|} = \frac{|truePositives|}{|truePositives|+|falsePositives|}$.

IV. Experimental Data

The behaviors of the local community detection algorithms described in Section II were compared on natural (social, cultural, and biological) graphs described in previous community detection research and on artificial graphs. Each of the graphs was small enough to permit calculation of the globally optimal community structure.

A. Natural Graphs

A number of standard social, cultural, and biological graphs have been described in the community-detection literature. The following data sets were used in the experiments:

- The Western US Power Grid (**power**). [4941 vertices, 6594 edges] [WS98].
- Network Science (**netsci**). A co-authorship network of scientists working on network theory and experiments [1589 vertices, 2742 edges] [New06].

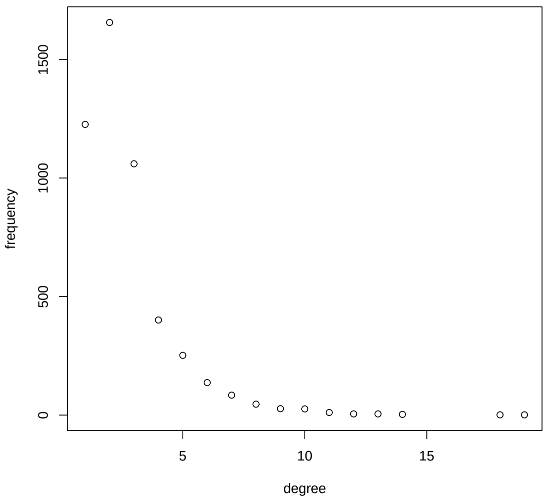

Fig. 2. Degree distribution for the Western US power grid network.

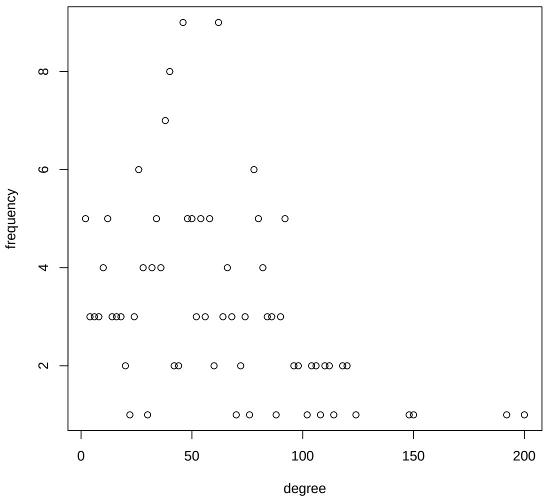

Fig. 3. Degree distribution for a network of jazz musicians.

- Word Adjacencies (**adjnoun**). Adjacency network of common adjectives and nouns in the Novel David Copperfield by Charles Dickens.[112 vertices, 425 edges] [New06].
- Les Miserables. Co-appearance network of characters in the Victor Hugo novel Les Miserables (**lesmis**).[77 vertices, 254 edges] [Knu93].
- The neural network of the nematode C. Elegans (**c.elegans**). [297 vertices, 2359 edges] [WS98].
- Zachary's karate club (**zachary**). [34 vertices, 78 edges] [Zac77].
- Dolphin social network (**dolphin**). A social network of frequent associations among 62 dolphins in a community living off Doubtful Sound, New Zealand [62 vertices, 159 edges] [LSB$^+$03].
- Jazz. A network of jazz musicians who have performed together (**jazz**). [198 vertices, 2742 edges] [GD03].
- American college football (**football**). A network of America football games between Division IA colleges during the regular Fall 2000 season [115 vertices, 616 edges] [GN02].

B. Artificial Graphs

A common data set for testing community-extraction algorithms consists of random networks of 128 vertices divided into 4 equal-sized communities with average vertex degree of 16 [NG04], [RB07], [Bag08]. In experiment 2, the average proportion of edges connected to other vertices in the same community (internal edge proportion) was 0.67 (weak community structure), 0.83 (moderate community structure), and 0.9 (strong community structure). All communities were of size 32; thus, k was equal to 32 in each trial. The three artificial graphs are referred to as $r.67$, $r.83$, and $r.90$, respectively, in the discussion below.

C. Network Degree Distributions

The degree distributions of the nine natural and three artificial graphs described above differ widely. For example, Figure 2 shows vertex frequency as a function of vertex degree for the Western US Power Grid network. This distribution has a heavy tail suggesting a power-law or exponential distribution. The degree distributions of the Network Science, Les Miserables, and Word Adjacencies networks display a similar heavy tail.

By contrast, the degree distribution of the random graphs is more symmetric, suggestive of the normal distribution to be expected of a random graph. The degree distributions of the remaining graphs, typified by the Jazz network shown in Figure 3, are harder to characterize, with little resemblance either to normal or heavy-tailed distributions.

One way to characterize the differences among these graphs is suggested by the convention of plotting degree distributions on log-log graphs. Graphs whose degree distributions are heavy-tailed, *i.e.*, that are well-approximated by power-law or exponential functions, typically appear to be linear when displayed in this fashion. If linear regression is performed on the log of the distribution values, a good fit will be obtained if the distribution is exponential or power-law, but the fit will be poor for other distributions, such as linear or normal. For example, the log-log plot of the degree distribution for the Western US Power Grid network, shown in Figure 4, is nearly linear, with $R^2 = 0.881$.

Figure 5 shows the least-squares linear fit of the log-log degree distributions of the 9 natural and 3 artificial graphs. R^2 is from 0.881 to 0.646 for the four heavy-tailed networks, but is less than 0.04 for two of the random graphs and is in

Graph	power	netsci	adjnoun	lesmis	c.elegans	dolphin	zachary	jazz	football	r.67	r.83	r.90
R^2	*0.881*	*0.821*	*0.669*	*0.646*	*0.5154*	*0.478*	*0.291*	*0.153*	*0.116*			
MaxM	0.636	0.846	0.445	0.706	**0.776**	**0.837**	**0.890**	**0.818**	0.738	0.789	0.892	0.936
MaxR	0.324	0.800	0.380	0.708	0.660	0.614	0.606	0.722	0.292	0.413	0.345	0.355
MinOmega	0.492	0.830	0.290	0.539	0.359	0.545	0.527	0.349	0.331	0.300	0.300	0.322
MaxDensity	0.647	0.856	0.419	0.635	0.576	0.768	0.766	0.807	**0.826**			
MaxActivation	**0.702**	**0.885**	**0.538**	**0.727**	0.669	0.824	0.826	0.803	0.733	0.769	0.912	0.942

TABLE I

MEAN NUWR IN 1000 TRIALS FOR 5 SELECTION POLICIES APPLIED TO 9 SOCIAL, CULTURAL, AND BIOLOGICAL GRAPHS NETWORKS.

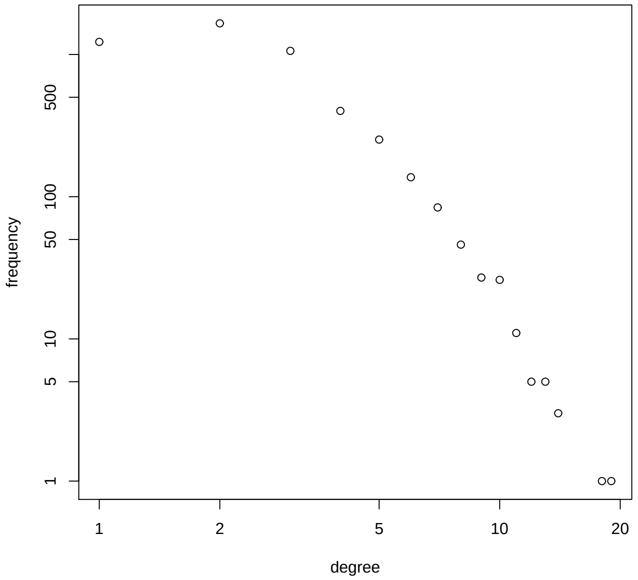

Fig. 4. Degree distribution of the Western US power grid plotted with log-log axes. The fit of this curve to a linear regression line has $R^2 = 0.881$.

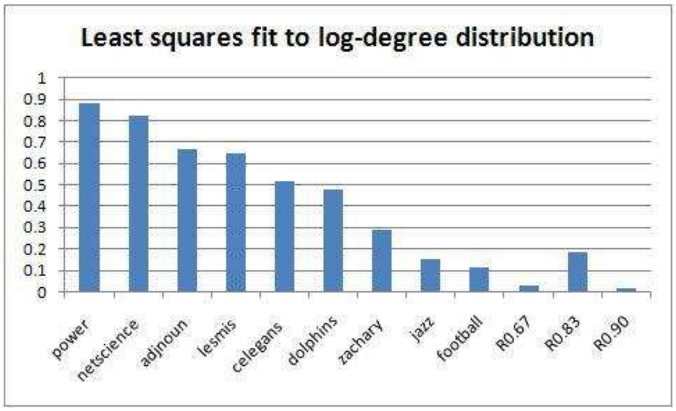

Fig. 5. R^2 statistic for linear regression of log-log degree distribution.

between for the remaining networks.[3]

V. EXPERIMENTAL PROCEDURE

To facilitate comparison of the behaviors of the local community detection algorithms under alternative criteria, two distinct global community structures were calculated for each graph: the modularity-maximizing structure, determined by the agglomerative clustering algorithm of [New04] (the *modularity structure*[4]); and the community structure induced by the partition-density maximizing edge partition [YYA10] (the *edge-partition structure*).

The modularity structure is a vertex partition, so a single vertex can belong to only a single community. The partition-density structure, in contrast, permits vertices to belong to multiple communities.

For each community, the betweenness and degree centrality of each vertex in that community was precomputed. Each vertex that belonged to two or more edge-partition communities was assigned its highest betweenness and degree centrality value in any of the containing communities.

The evaluation consisted of a series of trials, each of which started with the random selection of a query vertex, s, from the graph. For each global optimization criterion (modularity structure and edge-partition structure) in turn, the target community T under that criterion structure was retrieved, and each algorithm was then invoked on the graph with s as the query vertex and maximum return set size $|T| = k$ as a termination condition.[5] The NUWR was calculated for the k-element set of vertices returned by the algorithm using each of the three utility functions: betweenness centrality, degree centrality, and membership. For each of the three utility functions, an NUWR of 1.0 would mean that every community vertex, and no non-community vertex, was returned by the algorithm, whereas an NUWR of 0.0 would mean that no community vertices were found. The three utility functions differ in that betweenness centrality and degree centrality assign a higher weight to vertices that play a more central role in T, whereas membership treats all elements of T identically.

One thousand trials were performed for each algorithm on each graph. In MaxActivation, the attenuation factor, δ, was

[3] Clauset et al. [CSN09] describe a procedure for fitting degree distributions to a power-law function and provide code for this procedure at http://www.santafe.edu/~aaronc/powerlaws/. Under this procedure, none of the 12 graphs has a statistically significant fit to a power-law distribution.

[4] The highest modularity partition of a graph does not necessarily correspond to the actual community structure [FB07], and alternative metrics sometimes lead to better community structure ([RB07], [Bra10b], [KER08]. However, modularity is the best-known community-structure criterion, so for reproducibility of the results described here, the partition that globally optimizes modularity was chosen as the first target community structure.

[5] For edge-partition communities, k was the size of the largest community containing s.

set to 0.05.

MaxM, MaxR, and MinOmega are instantiations of the local community structure schema (shown in Algorithm 1, above) that maximize M, maximize R, and minimize Ω (outwardness), respectively, with no filtering. MaxR and MinOmega are equivalent to the algorithms of [Cla05] and [Bag08], respectively, whereas MaxM differs from the algorithm [LWP08] in that (1) MaxM selects the node that maximizes M, breaking ties in favor of the lowest degree node, rather than the lowest degree node for which $\Delta M > O$ and (2) MaxM performs no node filtering.

The first experiment evaluated the ability of each algorithm to find the same community as would be found through globally maximizing modularity. Tables I, II, and III show the NUWR of each algorithm on each graph, where utility within each target community was measured by betweenness centrality, degree centrality, and membership, respectively. In all three tables, MaxDensity had the highest NUWR for the artificial graphs (MaxM tied MaxDensity on R0.90 for betweenness centrality with an NUWR of 1.0), MaxActivation had the highest NUWR for the graphs whose degree distribution most closely matches a power law distribution, and MaxM had the highest NUWR for the remaining graphs. The choice of vertex utility functions affected the relative performance of MaxM and MaxActivation only on the adjnoun and lesmis graphs (MaxActivation had higher NUWR on both when the utility function was betweenness centrality, MaxM had higher NUWR on both when the utility function was membership, and when the utility function was degree centrality, MaxActivation was better for adjnoun and MaxM better for lesmis), but in all three cases every graph in which MaxActivation performed better than MaxM had a higher R^2 (*i.e.*, closer match to a power-law distribution) than any graph for which MaxM performed better.

The second experiment followed the same procedure as the first but used edge-partition structure as the target community structure for evaluation of the algorithms. Thus, the second experiment evaluated the extent to which each algorithm found the same community structure as would have been found by link clustering algorithm of [YYA10]. Tables IV, V, and VI show the NUMW of each algorithm on the same 12 graphs as above, where once again the utility within each target community is measured by betweenness centrality, degree centrality, and membership, respectively.

In the second experiment, MaxActivation had the highest NUWR, regardless of vertex utility function, for all but 2 graphs (MaxDensity was best on $r.67$ under betweenness centrality, and MaxM was best on zachary for membership).

A. Discussion

The relative accuracy of the alternative vertex selection criteria in identifying a globally optimal community, starting from a random member of that community, depended on the character of the graph and the nature of the target community. When the target community structure was globally maximal modularity, MaxActivation performed best in heavy-tailed graphs (which have high R^2), and MaxDensity was most accurate (with one tie from MaxM) in random graphs (which had very low R^2). In the remaining graphs, MaxM was the most accurate. The choice of vertex utility functions (between centrality, degree centrality, or membership) merely shifted the R^2 value where the relative ranking of MaxActivation and MaxM switch.

When the target community structure was produced by partition-density maximizing link-clustering, MaxActivation had higher NUWR values for all but two cases, regardless of vertex utility function. This suggests that local spreading activation is a proxy for the link distance metric of [YYA10].

It may seem counterintuitive that non-xenophobic algorithms, such as MaxActivation and MaxDensity, could ever have higher NUWR than xenophobic algorithms, such as MaxM, that use information (edges to vertices in U) that is ignored by the former. However, the empirical analysis suggests that in heavy-tailed networks the number of edges from a candidate vertex $n \in N$ to vertices in U is simply not an informative indicator of n's centrality in the target community. In these networks, n's centrality seems best modeled by the number and length of known paths from n to into the community, as expressed by activation, irrespective of links from n to U.

VI. Conclusion

This paper has shown that local community detection algorithms can be distinguished based on whether their vertex selection criterion is xenophobic. In an empirical evaluation on 12 natural and artificial graphs, the relative performance of xenophobic and non-xenophobic algorithms depended on three factors: (1) the degree distribution of the graph, (2) the target community structure, and (3) the centrality criterion for vertices within the target community.

To evaluate the relative accuracy of alternative vertex selection policies, a criterion was proposed, Normalized Utility-Weighted Recall (NUWR), that measures, relative to a target community structure and centrality measure, how closely a return set of k nodes matches the k most central nodes of the community.

These results suggest that there is no one-size-fits-all local community detection algorithm, but instead algorithms should be selected based on the characteristics of the graph and the nature of the community to be detected.

This work does not address the challenging problem of devising a termination policy that maximizes the likelihood of getting most or all of a community (*i.e.*, maximizing recall) while minimizing the proportion of non-community nodes (*i.e.*, maximizing precision). However, identifying better policies that optimize vertex-selection order will set the stage for development of such techniques. As better vertex-selection policies are devised, it may become easier to improve termination policies as well, leading to much more accurate local community detection techniques. The work described here is intended to be a step on this road.

Graph	power	netsci	adjnoun	lesmis	c.elegans	dolphin	zachary	jazz	football	r.67	r.83	r.90
R^2	*0.881*	*0.821*	*0.669*	*0.646*	*0.5154*	*0.478*	*0.291*	*0.153*	*0.116*	*0.030*	*0.184*	*0.018*
MaxM	0.612	0.858	0.456	**0.739**	**0.773**	**0.892**	**0.865**	**0.829**	**0.766**	0.818	0.908	0.927
MaxR	0.338	0.809	0.366	0.659	0.635	0.471	0.714	0.749	0.327	0.388	0.358	0.398
MinOmega	0.478	0.822	0.328	0.434	0.381	0.525	0.480	0.378	0.308	0.295	0.297	0.323
MaxDensity	0.660	0.867	0.434	0.611	0.564	0.766	0.771	0.820	0.816	**0.929**	**0.989**	**1.000**
MaxActivation	**0.710**	**0.894**	**0.549**	0.716	0.659	0.811	0.837	0.800	0.731	0.771	0.914	0.942

TABLE II
Modularity, degree.

Graph	power	netsci	adjnoun	lesmis	c.elegans	dolphin	zachary	jazz	football	r.67	r.83	r.90
R^2	*0.881*	*0.821*	*0.669*	*0.646*	*0.5154*	*0.478*	*0.291*	*0.153*	*0.116*	*0.030*	*0.184*	*0.018*
MaxM	0.604	0.930	**0.426**	**0.724**	**0.717**	**0.865**	**0.819**	**0.798**	**0.755**	0.811	0.906	0.926
MaxR	0.332	0.876	0.340	0.663	0.588	0.476	0.680	0.719	0.329	0.378	0.349	0.392
MinOmega	0.432	0.881	0.257	0.371	0.351	0.459	0.442	0.352	0.308	0.288	0.293	0.324
MaxDensity	0.565	0.907	0.271	0.418	0.402	0.596	0.584	0.636	0.804	**0.902**	**0.989**	**1.000**
MaxActivation	**0.650**	**0.947**	0.391	0.522	0.506	0.700	0.715	0.616	0.716	0.717	0.886	0.922

TABLE III
modularity/recall.

Graph	power	netsci	adjnoun	lesmis	c.elegans	dolphin	zachary	jazz	football	r.67	r.83	r.90
R^2	*0.881*	*0.821*	*0.669*	*0.646*	*0.5154*	*0.478*	*0.291*	*0.153*	*0.116*	*0.030*	*0.184*	*0.018*
MaxM	0.138	0.099	0.705	0.237	0.561	0.470	0.501	0.698	0.580	0.781	0.871	0.871
MaxR	0.140	0.123	0.649	0.265	0.533	0.381	0.456	0.701	0.319	0.486	0.395	0.417
MinOmega	0.142	0.108	0.548	0.205	0.329	0.379	0.416	0.395	0.366	0.386	0.313	0.348
MaxDensity	0.175	0.169	0.760	0.397	0.694	0.581	0.777	0.806	0.724	**0.924**	0.925	0.950
MaxActivation	**0.171**	**0.170**	**0.743**	**0.403**	**0.740**	**0.610**	**0.799**	**0.835**	**0.677**	0.879	**0.931**	**0.954**

TABLE IV
Link-clustering, betweenness centrality.

Graph	power	netsci	adjnoun	lesmis	c.elegans	dolphin	zachary	jazz	football	r.67	r.83	r.90
R^2	*0.881*	*0.821*	*0.669*	*0.646*	*0.5154*	*0.478*	*0.291*	*0.153*	*0.116*	*0.030*	*0.184*	*0.018*
MaxM	0.805	0.469	0.854	0.862	0.627	0.790	0.873	0.782	0.667	0.693	0.823	0.864
MaxR	0.763	0.463	0.813	0.834	0.592	0.683	0.708	0.778	0.408	0.504	0.472	0.443
MinOmega	0.726	0.381	0.646	0.622	0.342	0.569	0.512	0.400	0.363	0.395	0.355	0.373
MaxDensity	0.903	0.474	0.797	0.873	0.563	0.756	0.796	0.790	0.786	0.740	0.867	0.916
MaxActivation	**0.976**	**0.483**	**0.957**	**0.947**	**0.894**	**0.962**	**0.894**	**0.899**	**0.994**	**0.912**	**0.967**	**0.982**

TABLE V
Link clustering, degree centrality.

Graph	power	netsci	adjnoun	lesmis	c.elegans	dolphin	zachary	jazz	football	r.67	r.83	r.90
R^2	*0.881*	*0.821*	*0.669*	*0.646*	*0.5154*	*0.478*	*0.291*	*0.153*	*0.116*	*0.030*	*0.184*	*0.018*
MaxM	0.814	0.484	0.742	0.786	0.600	0.762	**0.855**	0.683	0.649	0.711	0.838	0.854
MaxR	0.742	0.476	0.695	0.772	0.523	0.650	0.683	0.682	0.367	0.454	0.429	0.434
MinOmega	0.706	0.389	0.606	0.588	0.280	0.543	0.490	0.363	0.339	0.360	0.330	0.367
MaxDensity	0.893	0.489	0.728	0.849	0.453	0.761	0.692	0.785	0.775	0.780	0.880	0.910
MaxActivation	**0.964**	**0.493**	**0.870**	**0.910**	**0.753**	**0.937**	0.821	**0.844**	**0.981**	**0.846**	**0.934**	**0.956**

TABLE VI
link clustering, membership.

Acknowledgment

This work was funded under contract number CECOM W15P7T-09-C-F600. The MITRE Corporation is a not-for-profit Federally Funded Research and Development Center chartered in the public interest.

References

[Bag08] J. Bagrow, "Evaluating local community methods in networks," *J. Stat. Mech.*, vol. 2008, no. 05, p. P05001, May 2008.

[Bra10a] L. K. Branting, "Incremental detection of local community structure," in *International Conference on Advances in Social Network Analysis and Mining*. Los Alamitos, CA, USA: IEEE Computer Society, 2010, pp. 80–87.

[Bra10b] ——, "Information theoretic criteria for community detection," *Advances in Social Network Mining and Analysis, Lecture Notes in Computer Science*, vol. 5498, pp. 114–130, 2010.

[CL75] A. Collins and F. Loftus, "A spreading-activation theory of semantic processing." *Psychological Review*, vol. 82, no. 6, pp. 407–428, November 1975.

[Cla05] A. Clauset, "Finding local community structure in networks," *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, vol. 72, no. 2, p. 026132, 2005. [Online]. Available: http://link.aps.org/abstract/PRE/v72/e026132

[Cre97] F. Crestani, "Application of spreading activation techniques in information retrieval," *Artif. Intell. Rev.*, vol. 11, no. 6, December 1997.

[CSN09] A. Clauset, C. Shalizi, and M. Newman, "Power-law distributions in empirical data," *SIAM Review*, vol. 51, no. 4, pp. 661–703, 2009.

[CZR09] J. Chen, O. Zaiane, and G. R, "Local community identification in social networks," in *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Athens, Greece, July 20–22 2009.

[FB07] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, January 2007.

[GD03] P. M. Gleiser and L. Danon, "Community structure in jazz," *Advances in Complex Systems (ACS)*, vol. 06, no. 04, pp. 565–573, 2003.

[GN02] M. Girvan and M. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.

[Kat53] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, March 1953.

[KER08] P. Koutsourelakis and T. Eliassi-Rad, "Finding mixed-memberships in social networks," in *Proceedings of the 2008 AAAI spring symposium on social information processing*. Stanford, CA: AAAI, 2008.

[Knu93] D. Knuth, *The Stanford GraphBase: a platform for combinatorial computing*. New York, NY, USA: ACM, 1993.

[LSB$^+$03] D. Lusseau, K. Schneider, O. Boisseau, P. Haase, E. Slooten, and S. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.

[LWP08] F. Luo, J. Wang, and E. Promislow, "Exploring local community structures in large networks," *Web Intelli. and Agent Sys.*, vol. 6, no. 4, pp. 387–400, 2008.

[New04] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, p. 066133, 2004. [Online]. Available: doi:10.1103/PhysRevE.69.066133

[New06] ——, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, vol. 74, no. 3, pp. 036 104+, 2006. [Online]. Available: http://dx.doi.org/10.1103/PhysRevE.74.036104

[NG04] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks." *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 69, no. 2 Pt 2, February 2004. [Online]. Available: http://view.ncbi.nlm.nih.gov/pubmed/14995526

[RB07] M. Rosvall and C. Bergstrom, "An information-theoretic framework for resolving community structure in complex networks," *PNAS*, vol. 104, no. 7327, 2007.

[WF94] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge: Cambridge University Press, 1994.

[WS98] D. Watts and S. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, June 4 1998.

[YYA10] S. L. Yong-Yeol Ahn, James P. Bagrow, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 20 June 2010.

[Zac77] W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, 1977.