Incremental Detection of Local Community Structure

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

***Abstract*—Incremental methods for detecting community structure are necessary when a graph's size or node-expansion cost makes global community-detection methods infeasible. Previous approaches to local community detection, which conflate edges between vertices in the immediate neighborhood of a partially-known community with edges to more distant vertices, often select vertices in an order that is suboptimal with respect to the actual community structure. This paper describes two new algorithms—*MaxActivation* and *MaxDensity*—whose vertex-selection policies focus on edges among the vertices in the partially-known community and its immediate neighborhood, ignoring edges to more distant vertices. In an empirical evaluation on a collection of natural and artificial graphs of varying degrees of community cohesion, the relative performance of alternative algorithms depended upon the degree distribution of each graph. These results demonstrate that the selection of an algorithm for incremental community detection should be guided by the characteristics of the graph to which it will be applied.**

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). Meaningful components of such systems often correspond to communities within the associated graph, that is, to subgraphs whose vertices are more highly connected to each other than to vertices outside the community. Detection of such communities can therefore be a powerful tool for understanding complex systems.

Numerous algorithms of varying complexity and accuracy have been developed to identify communities in graphs. One popular approach is to search for a partition of the graph that optimizes a global utility function, such as modularity [New04]. As a practical matter, however, many graphs are only partly accessible, either because the entire graph is too large to fit in memory or because the cost in time or other resources of expanding all vertices in the entire graph is prohibitive. In such cases, it is not feasible to determine the globally optimal community structure. Instead, the objective of the search must be limited to determining the local community structure in the neighborhood of a 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]. 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).

An ideal vertex selection policy is that it choose vertices in decreasing order of their centrality (for a given centrality measure) in the actual target community that contains the query vertex, starting with the most central. Selecting vertices in this order would optimize solution quality because a solution containing the k most central vertices to the actual community is preferable, *ceteris paribus*, to a solution consisting of k vertices that are less central to the community, regardless of k. Intuitively, one vertex-selection policy is preferable to another if, for k no greater than the size of the actual community, the k vertices selected by the first policy collectively have higher centrality in the actual community than those selected by the second. This paper formalizes this intuition, proposing a criterion for local community detection, *normalized utility-weighted recall* (NUWR), based on node-betweenness centrality and modularity.

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.

This paper proposes two new algorithms for local community detection that use selection policies different from those of previous local community detection algorithms in that they select each successive vertex based only on edges to the partial community and its immediate neighbors. For some classes of graphs, this approach leads, counterintuitively, to better performance than previous approaches that take into consideration edges to vertices more than one step from the current community.

Section II reviews previous approaches to local community detection algorithms and describes two introspective algorithms. A criterion for local community detection is proposed in Section III. Section IV sets forth a comparative evaluation on a set of standard natural and artificial graphs.

II. Algorithms for Local Community Detection

Many local community detection algorithms share a common schema that at each step of the algorithm assigns each vertex in the graph to one of three sets:

- 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 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 under 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. Previous Local Community Detection Algorithms

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] proposes 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].

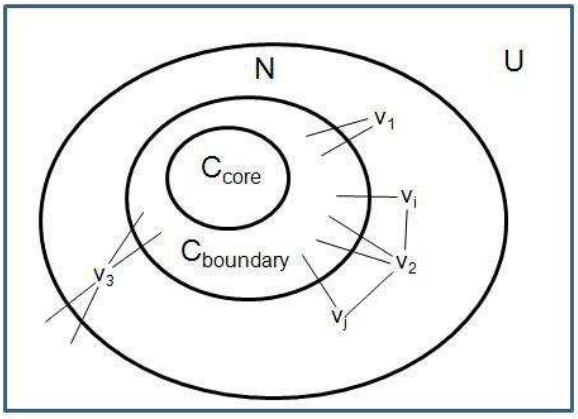

Fig. 1. Vertices v_1, v_2, and v_3 are candidates for addition to C.

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

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

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][1] 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.

These three selection policies—(l) maximizing local modularity (L), (2) minimizing outwardness (Ω_v), and (3) maximizing M—have in common that they fail to distinguish edges internal to N from edges connecting N to U. This can sometimes lead a node that is very likely to be of low centrality to be chosen 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

[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.

($\frac{2-2}{4} = 0$) 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, 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.

The observation that maximizing local modularity, minimizing outwardness, and maximizing M can all sometimes lead low-centrality vertices to be selected before potentially higher-centrality vertices suggests that better performance might be obtained by selection criteria that distinguish edges internal to N from those between vertices in N and vertices in U. 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 on the 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 are *introspective* in the sense that they focus on vertices close to C, ignoring links to U.

B. Introspective Community-Detection Algorithms

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 incremental community detection.

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.

Algorithm 2: MaxActivation Node Selection Algorithm

```
queryVertex.activation ← 1.0
currentPly ← {queryVertex}
previousPly ← φ
while (currentPly ≠ φ) 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 ∗δ
    end
    spread activation between members of
    nextPly
    foreach {edge(w,v)| w, v ∈ nextPly } do
        v.tmp += w.activation ∗δ
        w.tmp += v.activation ∗δ
    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∈N}(n.activation)
```

Density-Based Selection

An alternative introspective selection criterion is to select the $n \in N$ that makes the community as interconnected as possible. MaxDensity, 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 with the most edges to other vertices in N, and any remaining ties are broken by selecting the n with the shortest path to the query vertex.

III. Evaluation Criteria for Incremental Local Community Detection

In the absence of knowledge of the actual community structure, it is difficult to evaluate the output of a local community detection algorithm. However, in cases in which the actual community structure is known, local community detection algorithms can be evaluated by comparing their output to the actual structure. For example, a local community detection algorithm's selection policy can be evaluated by

Algorithm 3: MaxDensity Node Selection Algorithm

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

comparing the order in which vertices are added under the policy to the optimal order. Given an oracle that provides the actual community, C', and a utility function, $util$, defined over all community vertices (such as node betweenness), the quality of a return set (*i.e.,* proposed community), C, consisting of k vertices selected under a given selection policy, can be measured as the sum of the utilities of the vertices in the return set, $\sum_{v \in C} util(v)$. 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 actual community C', NUWR, is shown in equation 2:

$$NUWR = \frac{\sum_{v \in C} util(v)}{\arg\max_{S \subseteq C', |S| = min(|C|,|C'|)} \sum_{v \in S} util(v)} \quad (2)$$

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

Alternative evaluation metrics that have been applied to community detection are less informative when applied to vertex-selection policies. F-measure (the harmonic mean of recall and precision) and the Rand index [Ran71], [HA85] use unweighted counts, so they don't distinguish partial communities consisting of high centrality vertices from those consisting of low-centrality vertices. Incorporating precision into NUWR would not provide any additional information because, for fixed k and actual community size $|C'|$, precision and recall express the same information, *i.e.*, under these circumstances, precision = $\frac{|truePositives|}{|k|}$, and recall = $\frac{|truePositives|}{|C'|}$, so

$$precision = \frac{recall * |C'|}{k} \quad (3)$$

In the evaluation described below, the utility of each vertex in a community was calculated as the vertex's betweenness centrality [WF94] in the subgraph that consists only of community vertices and edges (*i.e.*, excluding non-community vertices and edges). Utility of zero was assigned to nodes outside of the community.

IV. Empirical Evaluation

The accuracies 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. In each trial, a query vertex s was randomly selected from the graph, and the canonical community C' for which $s \in C'$ was retrieved, together with the node-betweenness of each node in C', as calculated by applying the Jung[2] implementation of node betweenness to the subgraph consisting of the community's nodes and edges. Each algorithm was then invoked on the graph with s as the query vertex and a maximum community size of $|C'| = k$ as a termination condition. The NUWR was calculated for the k-element set of vertices returned by the algorithm. 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. One thousand trials were performed for each algorithm on each graph. In MaxActivation, the attenuation factor, δ, was set to 0.05.

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 first experiment:

- The Western US Power Grid [4941 vertices, 6594 edges] [WS98].
- Network Science. A co-authorship network of scientists working on network theory and experiments [1589 vertices, 2742 edges] [New06].
- Word Adjacencies. 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 [77 vertices, 254 edges] [Knu93].
- The neural network of the nematode C. Elegans [297 vertices, 2359 edges] [WS98].
- Zachary's karate club [34 vertices, 78 edges] [Zac77].
- Dolphin social network. 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 [198 vertices, 2742 edges] [GD03].
- American college football. A network of America football games between Division IA colleges during the regular Fall 2000 season [115 vertices, 616 edges] [GN02].

[2] http://jung.sourceforge.net/

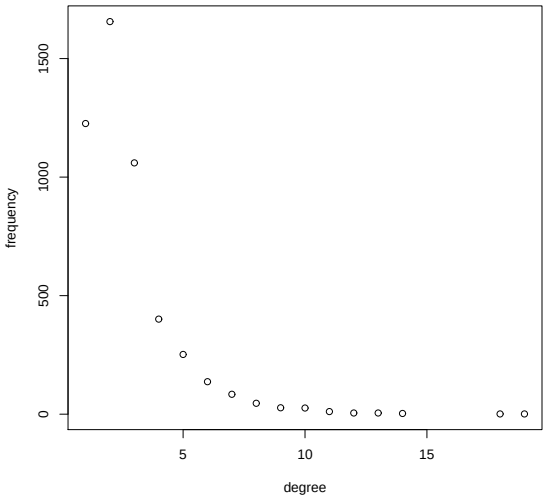

Fig. 2. Degree distribution for the Western US power grid network.

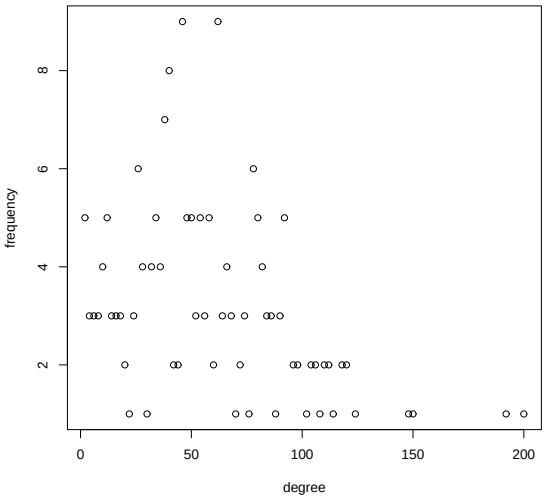

Fig. 3. Degree distribution for a network of jazz musicians.

The canonical community structure of each graph was determined using the [New04] algorithm, which finds the highest-modularity graph partition in the dendrogram generated by greedy agglomerative clustering, where at each iteration the pair of clusters is joined that results in the greatest increase, or lowest decrease, in modularity.[3]

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], [MC07], [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.

C. Network Degree Distributions

The degree distribution 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.

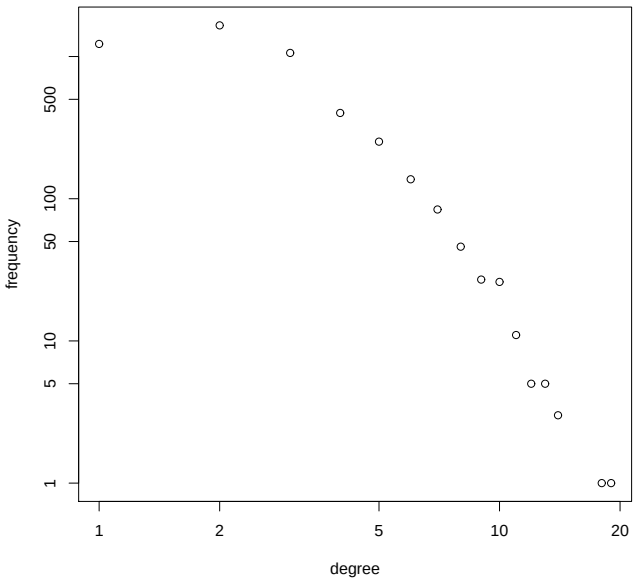

Fig. 4. Degree distribution of the Western US power grid plotted with log-log axes. The fit of this curve to a linear regression line has $R^2 = 0.881$.

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 graphed in this fashion. If linear regression is performed on

[3]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 ([MC07], [Bra08], [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 canonical community structure for the natural and artificial graphs.

Graph	power	netsci	adjnoun	lesmis	c.elegans	dolphin	zachary	jazz	football
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
MaxR	0.324	0.800	0.380	0.708	0.660	0.614	0.606	0.722	0.292
MinOmega	0.492	0.830	0.290	0.539	0.359	0.545	0.527	0.349	0.331
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

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

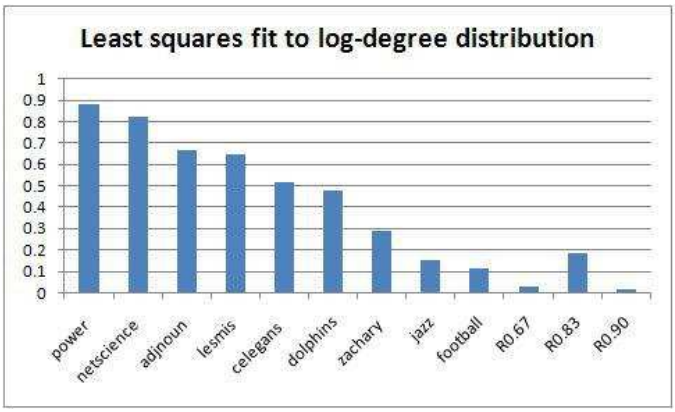

Fig. 5. R^2 statistic for linear regression of log-log degree distribution.

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 between for the remaining networks.[4]

D. Experiments

The first experiment evaluated the ability of each algorithm to find the same community as would be found through globally maximizing modularity. 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.

[4]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.

proportion internal nodes	0.67	0.83	0.90
R^2	0.030	0.184	0.018
MaxM	0.789	0.892	0.936
MaxR	0.413	0.345	0.355
MinOmega	0.300	0.300	0.322
MaxDensity	**0.927**	**0.985**	**1.000**
MaxActivation	0.769	0.912	0.942

TABLE II
MEAN NUWR IN 1000 TRIALS FOR 5 SELECTION POLICIES APPLIED TO 3 ARTIFICIAL NETWORKS.

As shown in Table I, MaxActivation had the highest NUWR for networks in which R^2 was 0.646 or higher—that is, those whose degree distribution resembles a power-law or exponential function—and MaxM had the highest NUWR for the remaining networks except for the Football network, for which MaxDensity had the highest NUWR.

The second experiment evaluated the algorithms on the three random graphs. As shown in Table II, MaxDensity had the highest NUWR for graphs whose proportion of internal edges was 0.67 (weak community structure), 0.83 (moderate community structure), and 0.90 (strong community structure).

E. Discussion

The relative accuracy of the alternative vertex selection criteria in identifying the globally optimal community starting from a random member of that community varied with the character of the graph. In heavy-tailed graphs, MaxActivation performed best; in random graphs and the Football network, which had very low R^2, MaxDensity was most accurate; in the remaining graphs, MaxM was most accurate. MinOmega was generally the lowest performing algorithm.

It may seem counterintuitive that the introspective algorithms, MaxActivation and MaxDensity, could ever have higher NUWR than non-introspective algorithms, such as MaxM, given that the latter uses information (edges to vertices in U) that is ignored by the former. 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 actual community. In these networks, the structure of individual communities seems best modeled by the number and length of paths into the community, as expressed by activation, irrespective of links into U. In the random graphs used in the evaluation, it appears that the simple heuristic of choosing the node that maximizes the number of internal edges is quite

effective. It is in graphs that are neither random nor heavy-tailed that the heuristic of preferring nodes that maximize the ratio of internal to external edges performs best.

V. Conclusion

This paper has shown that previous local community detection algorithms can sometimes select vertices in an order that is suboptimal with respect to the actual community structure because they fail to distinguish edges between members of the partially known community's immediate neighborhood from those to more distant nodes. To address this limitation, two new algorithms—MaxActivation and MaxDensity—were proposed that use introspective policies under which vertices are selected based only on the edges between vertices in the partially-known community and its immediate neighborhood.

To evaluate the relative accuracy of alternative vertex selection policies, a criterion was proposed, Normalized Utility-Weighted Recall (NUWR), that measures, relative to a given centrality measure and actual community structure, how closely a return set of k nodes matches the k most central nodes of the community. In an evaluation comparing five algorithms on nine natural and three artificial graphs, the highest NUWR depended on the degree distribution of the particular graph. The best solutions on graphs having heavy-tailed degree distributions were found by MaxActivation, the best solutions on random graphs were found by MaxDensity, and the best solutions on graphs in neither of these categories was found by MaxM. These results suggest that selection of algorithms for incremental community detection should be the guide to the characteristics of the graph to which they are applied.

This empirical evaluation is limited to the particular vertex utility function chosen for the evaluation (node-betweenness centrality) and, in the case of the natural graphs, to the particular global community structure on which the vertex utility function was based (globally maximal modularity). Different results could be expected if the algorithms were compared with respect to different community structures or different vertex utility functions. Indeed, a long-term objective of research in this field may be to demonstrate how to adapt community detection techniques to maximize any particular community structure or vertex-utility criteria specified by a user. For the present, however, modularity and node betweenness centrality are very commonly used criteria, so local community detection algorithms that perform well with respect to these criteria may be of broad utility.

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.

Acknowledgment

This work was funded under contract number CECOM W15P7T-09-C-F600. The MITRE Corporation is a non-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.

[Bra08] L. Branting, "Overcoming resolution limits in MDL community detection," in *Proceedings of the second KDD-SNA workshop on social network analysis*, Las Vegas, NV, USA, 2008.

[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.

[HA85] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, 1985.

[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.

[MC07] R. M. and B. C., "An information-theoretic framework for resolving community structure in complex networks," *PNAS*, vol. 104, no. 7327, 2007.

[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

[Ran71] W. M. Rand, “Objective Criteria for the Evaluation of Clustering Methods,” *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846–850, 1971.

[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.

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