ORIGINAL ARTICLE

Context-sensitive detection of local community structure

L. Karl Branting

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 vertex selection policies in detecting two distinct types of community structures—vertex partitions that maximize modularity, and link partitions that maximize partition density—in a variety of graphs. In this evaluation, the vertex selection policy that most accurately identified vertex-partition 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. These results indicate that local community detection should be context-sensitive in the sense of basing vertex selection on the graph's degree distribution and the target community structure.

L. K. Branting (✉)
The MITRE Corporation,
7525 Colshire Drive,
McLean, VA, USA
e-mail: lbranting@mitre.org

1 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* (Newman 2004) or *minimum description length* (Chakrabarti 2004; Rosvall and Bergstrom 2007). A related approach searches for an edge partition that maximizes global *partition density* (Ahn and Bagrow 2010). 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 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 (Clauset 2005; Luo et al. 2008; Bagrow 2008; Chen et al. 2009; Branting 2010a). 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 as a function of the number of their edges to non-community vertices. *Non-xenophobic* algorithms ignore or reward such edges.

Section 2 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 Sect. 3. Section 4 describes an experimental data set, comprising 12 standard natural and artificial graphs, and analyzes the vertex-degree distributions of each. A comparative evaluation set forth in Sect. 5 and discussed in Sect. 6, shows that the relative performance of xenophobic and non-xenophobic algorithms depends on (1) the degree 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.

2 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.

2.1 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 (2005) 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" (Clauset 2005).

A second selection criterion, termed outwardness, was proposed in Bagrow (2008). The outwardness of a vertex v, Ω_v, is:

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

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 Luo et al. (2008) is to choose the vertex that maximizes $M = \frac{\text{ind}(C)}{\text{outd}(C)}$, the ratio of $\text{ind}(C)$, the number of edges connecting pairs of nodes in C, to $\text{outd}(C)$, the number of edges connecting nodes in C to nodes outside of C.[1]

These three selection policies—(1) 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 Fig. 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
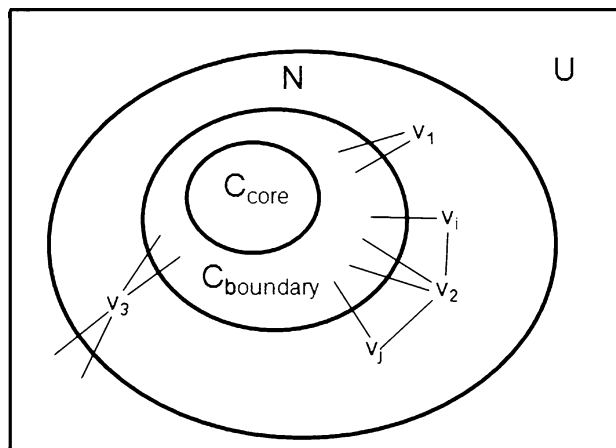

Fig. 1 Vertices v_1, v_2, and v_3 are candidates for addition to C

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, 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 $M = \frac{\text{ind}(C)+2}{\text{outd}(C)-2}$, which is higher than M after adding v_2 or v_3, $\frac{\text{ind}(C)+2}{\text{outd}(C)+0}$. 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.

2.2 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

---

[1] The algorithm of Luo et al. (2008) 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 Luo et al. (2008) 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.

candidate node $n \in N$ to U, and both reward edges from n to other nodes in N.

2.2.1 Spreading activation

Numerous approaches to spreading activation have been explored in the history of computer science (e.g., Collins and Loftus 1975; Crestani 1997). *MaxActivation* is a particularly simple form of spreading activation appropriate for local community detection (Branting 2010a).

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 does not 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]

2.2.2 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 (Branting 2010a), 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 remaining ties are broken by selecting the $n \in N$ with the shortest path to the query vertex.

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 $\arg\max_{n \in N}$*(n.activation)*

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 the largest number of such edges.

---

[2] An alternative approach to spreading activation based on the Katz (1953) index 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 but for brevity is omitted.

Algorithm 3: MaxDensity Node Selection Algorithm

$D \leftarrow \{n \mid \arg\max_{n \in C}(|\{edge(v,n), v \in C\}|)\}$

if *(|D| > 1)* **then**

    $D \leftarrow \{n \mid \arg\max_{n \in D}(|\{edge(v,n), v \in N\}|)\}$

    if *(|D| > 1)* **then**

        $D \leftarrow \{n \mid \arg\min_{n \in D} pathlength(n, query)\}$

    end

end

return *random member of D*

---

3 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 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* (Newman 2004), and the edge partition that maximizes *partition density* (Ahn and Bagrow 2010). 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 (Fortunato and Barthelemy 2007). 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 Newman (2004).

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 NUWR of community C with respect to target community T, NUWR, is shown in (2):

$$\text{NUWR}_T = \frac{\sum_{v \in C} util_T(v)}{\arg\max_{S \subseteq T, |S| = \min(|C|, |T|)} \sum_{v \in S} util_T(v)} \quad (2)$$

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 two different vertex utility functions:

1. Degree centrality, the proportion of edges in T that are incident to a given vertex n.
2. Membership, assigning the same value, 1.0, for every $n \in T$ and 0.0 for all other vertices.

For both 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 R-precision (Baeza-Yates and Ribeiro-Neto 1999), since under these circumstances

$$\text{NUWR}_T = \frac{|truePositives|}{|T|} = \frac{|truePositives|}{|truePositives| + |falsePositives|}.$$

4 Experimental data

The behaviors of the local community detection algorithms described in Sect. 2 were compared on standard benchmark graphs described in previous community detection research. Each of the graphs was small enough to permit calculation of the globally optimal community structure.

4.1 Natural graphs

A number of standard social, cultural, and biological graphs have been described in the community detection

literature. The experiments used following data sets, drawn from corpus of network data collected by Newman (2009):

- The Western US Power Grid (**power**) [4,941 vertices, 6,594 edges] (Watts and Strogatz 1998).
- Network Science (**netsci**). A co-authorship network of scientists working on network theory and experiments [1,589 vertices, 2,742 edges] (Newman 2006).
- Word Adjacencies (**adjnoun**). Adjacency network of common adjectives and nouns in the novel David Copperfield by Charles Dickens [112 vertices, 425 edges] (Newman 2006).
- Les Miserables. Co-appearance network of characters in the Victor Hugo novel Les Miserables (**lesmis**) [77 vertices, 254 edges] (Knuth 1993).
- The neural network of the nematode *C. elegans* (**c.elegans**) [297 vertices, 2,359 edges] (Watts and Strogatz 1998).
- Zachary's karate club (**zachary**) [34 vertices, 78 edges] (Zachary 1977).
- 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] (Lusseau et al. 2003).
- Jazz. A network of jazz musicians who have performed together (**jazz**) [198 vertices, 2,742 edges] (Gleiser and Danon 2003).
- American college football (**football**). A network of America football games between Division IA colleges during the regular Fall 2000 season [115 vertices, 616 edges] (Girvan and Newman 2002).

4.2 The Girvan–Newman benchmarks

The Girvan–Newman (GN) benchmarks, introduced in Girvan and Newman (2002), consist of random networks of 128 vertices divided into 4 equal-sized communities with an average vertex degree of 16 (Newman and Girvan 2004; Rosvall and Bergstrom 2007; Bagrow 2008).[3] 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 GN graphs are referred to as $r.67$, $r.83$, and $r.90$, respectively, in the discussion below.

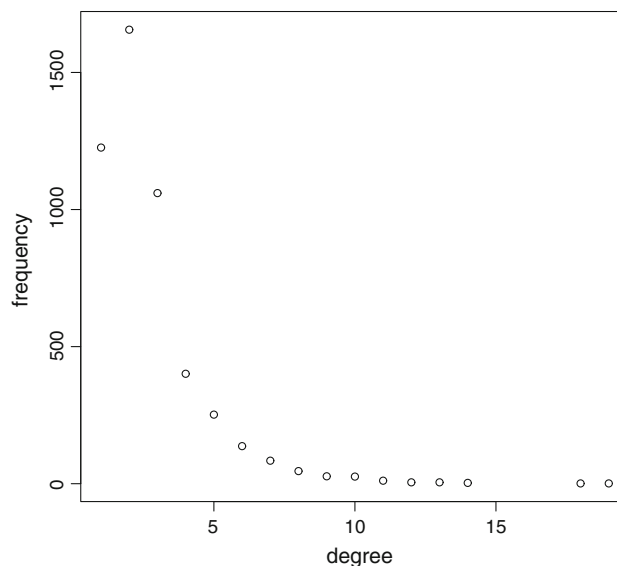[3] For a discussion of alternatives to the GN benchmarks, see Lancichinetti et al. (2008).

Fig. 2 Degree distribution for the Western US Power Grid network

4.3 Network degree distributions

The degree distributions of the nine natural and three GN graphs described above differ widely. For example, Fig. 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.

In 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 Fig. 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 Fig. 4, is nearly linear, with $R^2 = 0.881$.

Figure 5 and Table 1 show the least-squares linear fit of the log–log degree distributions of the nine natural and three GN graphs. R^2 ranges from 0.881 to 0.646 for the four
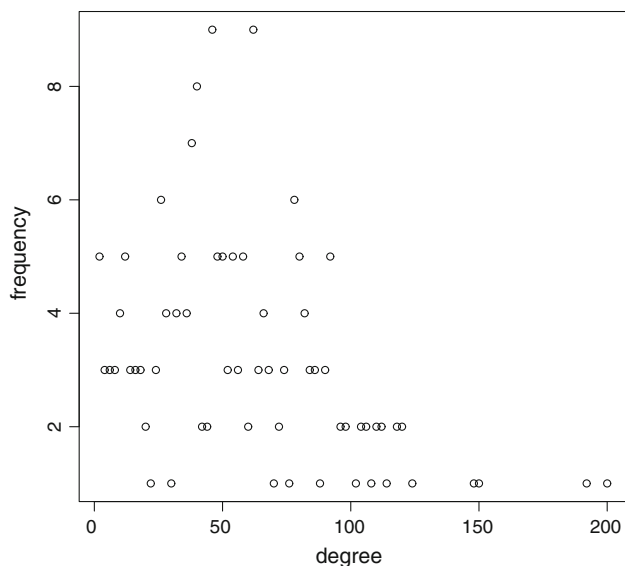
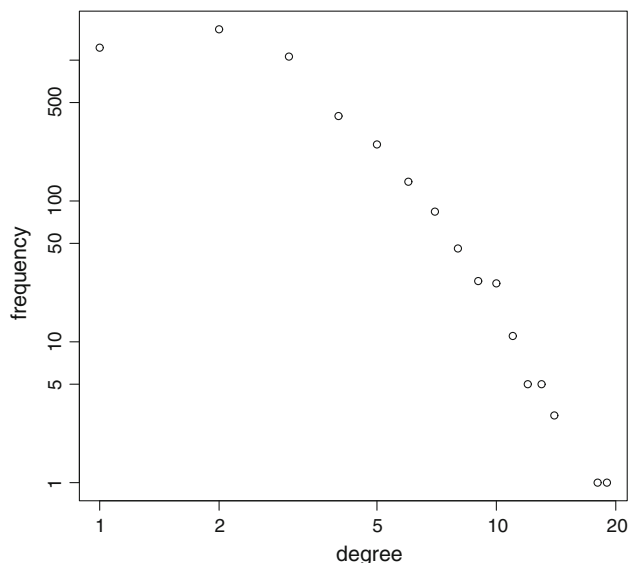**Fig. 3** Degree distribution for a network of jazz musicians

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 $R^2 = 0.881$

heavy-tailed networks, but is <0.04 for two of the random graphs, and is in between for the remaining networks.[4]

5 Experimental procedure

To facilitate comparison of the behaviors of the local community detection algorithms under alternative criteria,
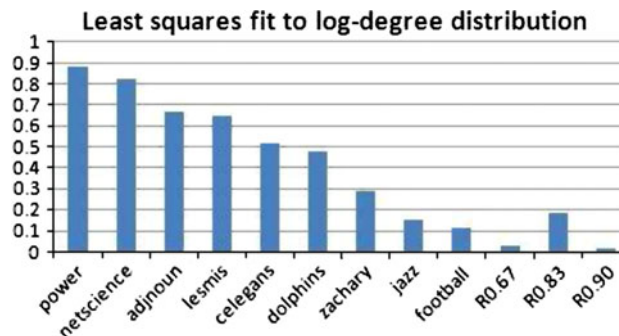

Fig. 5 R^2 statistic for linear regression of log–log degree distribution

two distinct global community structures were calculated for each of the graphs: the *vertex-partition structure*, determined for the natural graphs by the agglomerative clustering algorithm of Newman (2004)[5] and for the GN graphs consisting simply of the original vertex partition used to generate each graph; and the *edge-partition structure*, consisting of the community structure induced by the partition-density maximizing edge partition (Ahn and Bagrow 2010).

In vertex-partition structures, a single vertex can belong to only a single community, whereas in partition-density structures, by contrast, vertices may to belong to multiple communities.

For each community in the graphs, the degree centrality of each vertex in that community was precomputed. Each vertex that belonged to two or more partition-density communities was assigned its highest 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 target community structure (vertex partition or link partition) 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.[6] The NUWR was calculated for the k-element set of vertices returned by the algorithm using each of the two utility functions: degree centrality and membership. For each of the two utility functions, an NUWR of 1.0 would mean that every community vertex, and no non-community vertex, was returned by the

---

[4] Clauset et al. (2009) 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.

[5] The highest modularity partition of a graph does not necessarily correspond to the actual community structure (Fortunato and Barthelemy 2007), and alternative metrics sometimes lead to better community structure (Rosvall and Bergstrom 2007; Branting 2010b; Koutsourelakis and Eliassi-Rad 2008). 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.

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

Table 1 Fit of the degree distribution of each network to a linear regression line

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

Table 2 Mean NUWR in 1,000 trials where the target community structure was a vertex partition found by **maximizing modularity** (natural graphs) or the 'true' structure (GN graphs) and the vertex utility function was **degree centrality**

Graph	power	netsci	adjnoun	lesmis	c.elegans	dolphin	zachary	jazz	football	$r.67$	$r.83$	$r.90$
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 3 Mean NUWR in 1,000 trials where the target community structure was found a vertex partition found by **maximizing modularity** (natural graphs) or the 'true' structure (GN graphs) and the vertex utility function was **community membership**

Graph	power	netsci	adjnoun	lesmis	c.elegans	dolphin	zachary	jazz	football	$r.67$	$r.83$	$r.90$
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

algorithm, whereas an NUWR of 0.0 would mean that no community vertices were found. The two utility functions differ in that degree centrality assigns 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 on each graph, with all five algorithms tested on the same random seed, s, in each trial. In MaxActivation, the attenuation factor, δ, was 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 Clauset (2005) and Bagrow (2008), respectively, whereas MaxM differs from the algorithm (Luo et al. 2008) in that (1) MaxM selects the node that maximizes M, breaking ties in favor of the lowest degree node, rather than selecting 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 2 and 3 show the NUWR of each algorithm on each graph, where utility within each target community was measured by

degree centrality and membership, respectively (the highest value in each column in Tables 2–5 is shown in bold). In both tables, MaxDensity had the highest NUWR for the GN graphs, MaxActivation had the highest NUWR for the graphs whose degree distribution most closely matched 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 Word Adjacencies graph (MaxActivation had higher NUWR on the adjnoun graph when the utility function was degree centrality, and MaxM had higher NUWR on the Word Adjacencies graph when the utility function was membership), but in both 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 partition-density 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 the link-clustering algorithm of Ahn and Bagrow (2010) (i.e., the partition-density community). Tables 4 and 5 show the NUMW of each

Table 4 Mean NUWR in 1,000 trials where the target community structure was found by **maximizing partition density** and the vertex utility function was **degree centrality**

Graph	power	netsci	adjnoun	lesmis	c.elegans	dolphin	zachary	jazz	football	r.67	r.83	r.90
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 5 Mean NUWR in 1,000 trials where the target community structure was found by **maximizing partition density** and the vertex utility function was **community membership**

Graph	power	netsci	adjnoun	lesmis	c.elegans	dolphin	zachary	jazz	football	r.67	r.83	r.90
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 6 Vertex selection criterion leading to the highest mean NUWR under alternative target community structures and degrees of fit to a power-law degree distribution

Fit to power-law distribution	Community structure	
	Vertex partition	Link partition
High	MaxActivation	MaxActivation
Medium	MaxM	MaxActivation (MaxM for Zachary)
Low	MaxDensity	MaxActivation

algorithm on the same 12 graphs as above, where once again the utility within each target community is degree centrality and membership, respectively.

In the second experiment, MaxActivation had the highest NUWR, regardless of vertex utility function, for all but 1 graph (MaxM was best on Zachary's Karate Club when the utility function was membership). The results of both experiments are summarized in Table 6, which shows the vertex selection criterion leading to the highest mean NUWR for each of six combinations of community structure type and degree of fit to a power-law degree distribution.

6 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, the nature of the target community, and to a less extent the vertex utility function. When the target community structure was vertex

partition found by globally maximizing modularity (natural graphs) or from the structure used to generate the graphs (GN graphs), 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 one case, regardless of vertex utility function. This suggests that local spreading activation is a good proxy for the link distance metric of Ahn and Bagrow (2010).

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) ignored by non-xenophobic algorithms. However, the empirical evaluation suggests that the

number of edges from a candidate vertex $n \in N$ to vertices in U is simply not an informative indicator in heavy-tailed networks 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 the community, as expressed by activation, irrespective of links from n to U.

7 Related work

A wide variety of techniques have been developed for global community detection. Many of these techniques involve a utility function over graph partitions, such as modularity (Newman and Girvan 2004), minimum description length (Chakrabarti 2004; Rosvall and Bergstrom 2007; Branting 2010b), or log-likelihood (Zhang et al. 2007), together with a search strategy for finding partitions that optimize the utility function, such as greedy agglomerative (Newman 2004) or divisive (Newman and Girvan 2004) clustering, simulated annealing (Rosvall and Bergstrom 2007), genetic algorithms (Jin et al. 2010), or spectral clustering (Newman 2009). The division between the utility function and the search strategy is less clearly defined in other approaches, such as "barycentric" clustering (Cohen 2008), marker passing algorithms (Raghavan et al. 2007), and information flow estimation (Rosvall and Bergstrom 2008). A recent link-clustering algorithm takes a non-partitional approach, permitting communities to overlap (Ahn and Bagrow 2010). Comparative evaluations of many approaches to community detection are set forth in Danon et al. (2005) and Leskovec et al. (2010). All these algorithms require access to the entire graph.

Recent work has addressed the problem of detecting local community structure based on incremental search of the neighborhood surrounding one or more query vertices. These techniques are necessary when it is infeasible to search the entire graph. An early contribution was formulation of a criterion defined over local, as opposed to global, community structure (Muff and Rao 2005).

Viewed from the perspective of the algorithm schema set forth in Sect. 1, above, previous work in local community detection has typically attempted to identify a single criterion useful both for (1) selecting the 'best' vertex in N to add to C and (2) identifying the boundary of C community under construction. In the procedure described in Bagrow and Bollt (2005), community construction stops when the "change in total emerging degree" exceeds a threshold α. In Clauset (2005), the stopping criterion is a maximum community size, k, but the node-selection criterion, R, is intended to identify nodes on the boundary of the true community. Bagrow (2008) introduced a vertex selection criterion, "outwardness," that did not perform as well as other criteria in the evaluation

described above, but the primary focus was on recognizing community boundaries, i.e., termination criteria, an issue which is beyond the scope of this paper.

Luo et al. (2008) propose a vertex selection criterion, M, that performed very well in the evaluation described above. The procedure of Luo et al. (2008) differs from that described here, however, in that it interleaves adding to C vertices in N for which $\Delta M > O$ with filtering vertices in C whose removal would increase M. Chen et al. (2009) propose a criterion, L, representing the average internal degree of nodes in C over the average external degree of nodes in $C_{boundary}$, that is closely related to Clauset's R. In a manner parallel to Luo et al. (2008), Chen et al.'s algorithm includes a phase that filters vertices whose removal would increase the average internal degree of C.

The work described in this paper differs from previous work in that it focuses specifically on vertex selection, independent of filtering or termination. However, this work is complementary to research on termination and filtering techniques, since such techniques can be combined with whatever vertex selection technique is most appropriate for a given graph in light of the graph's vertex degree distribution and the target community structure.

8 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 GN graphs, the relative performance of vertex selection criteria 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, 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 that local community detection should be *context-sensitive* in the sense of selecting vertices based on the characteristics of the graph, the nature of the community to be detected, and the centrality criterion within the community.

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.

References

Ahn Y-Y, Bagrow JP (2010) Link communities reveal multiscale complexity in networks. Nature 466(7307):761–764

Bagrow J (2008) Evaluating local community methods in networks. J Stat Mech 2008(5):P05001

Bagrow J, Bollt E (2005) Local method for detecting communities. Phys Rev E 72(4):046108

Branting LK (2010a) Incremental detection of local community structure. In: International conference on advances, in social network analysis and mining. IEEE Computer Society, Los Alamitos, CA, USA, pp 80–87

Branting LK (2010b) Information theoretic criteria for community detection. Adv Soc Netw Min Anal Lect Notes Comput Sci 5498:114–130

Baeza-Yates R, Ribeiro-Neto B (1999) Modern information retrieval. Addison-Wesley, Reading

Chakrabarti D (2004) Autopart: parameter-free graph partitioning and outlier detection. In: Proceedings of the European conference on machine learning and practice of knowledge discovery in databases, pp 112–124

Chen J, Zaiane O, Goebel R (2009) Local community identification in social networks. In: Proceedings of the international conference on advances in social networks analysis and mining (ASONAM), Athens, Greece, 20–22 July

Clauset A (2005) Finding local community structure in networks. Phys Rev E (Stat Nonlinear Soft Matter Phys) 72(2):026132 (online). http://link.aps.org/abstract/PRE/v72/e026132

Clauset A, Shalizi C, Newman M (2009) Power-law distributions in empirical data. SIAM Rev 51(4):661–703

Cohen J (2008) Barycentric graph clustering. http://www2.computer.org/cms/Computer.org/dl/mags/cs/2009/04/extras/msp2009040029s2.pdf

Collins A, Loftus F (1975) A spreading-activation theory of semantic processing. Psychol Rev 82(6):407–428

Crestani F (1997) Application of spreading activation techniques in information retrieval. Artif Intell Rev 11(6):453–482

Danon L, Díaz-Guilera A, Duch J, Arenas A (2005) Comparing community structure identification. J Stat Mech Theory Exp 2005(9):P09008 (online). http://stacks.iop.org/1742-5468/2005/i=09/a=P09008

Fortunato S, Barthelemy M (2007) Resolution limit in community detection. Proc Natl Acad Sci 104(1):36–41

Girvan M, Newman M (2002) Community structure in social and biological networks. Proc Natl Acad Sci USA 99(12):7821–7826

Gleiser PM, Danon L (2003) Community structure in jazz. Adv Complex Syst (ACS) 6(4):565–573

Jin D, He D, Liu D, Baquero C (2010) Genetic algorithm with local search for community mining in complex networks. In: ICTAI (1)'10, pp 105–112

Katz L (1953) A new status index derived from sociometric analysis. Psychometrika 18(1):39–43

Koutsourelakis P, Eliassi-Rad T (2008) Finding mixed-memberships in social networks. In: Proceedings of the 2008 AAAI spring symposium on social information processing. AAAI, Stanford, CA

Knuth D (1993) The Stanford GraphBase: a platform for combinatorial computing. ACM, New York

Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. Phys Rev E 78:046110

Leskovec J, Lang K, Mahoney M (2010) Empirical comparison of algorithms for network community detection. In: Proceedings of the international conference on World Wide Web (WWW), 26–30 April. ACM, Raleigh, NC

Luo F, Wang J, Promislow E (2008) Exploring local community structures in large networks. Web Intell Agent Syst 6(4):387–400

Lusseau D, Schneider K, Boisseau O, Haase P, Slooten E, Dawson S (2003) The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behav Ecol Sociobiol 54(4):396–405

Muff ACS, Rao F (2005) Local modularity measure for network clusterizations. Phys Rev E 72:056107

Newman MEJ (2004) Fast algorithm for detecting community structure in networks. Phys Rev E 69:066133 (online). doi: 10.1103/PhysRevE.69.066133

Newman MEJ (2006) Finding community structure in networks using the eigenvectors of matrices. Phys Rev E (Stat Nonlinear Soft Matter Phys) 74(3):036104+ (online). http://dx.doi.org/10.1103/PhysRevE.74.036104

Newman M (2009) Network data. http://www-personal.umich.edu/~mejn/netdata/

Newman ME, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E Stat Nonlinear Soft Matter Phys 69(2 Pt 2):026113 (online). http://view.ncbi.nlm.nih.gov/pubmed/14995526

Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76:036106

Rosvall M, Bergstrom C (2007) An information-theoretic framework for resolving community structure in complex networks. Proc Natl Acad Sci USA 104:7327–7331

Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci USA 105(4):1118–1123

Watts D, Strogatz S (1998) Collective dynamics of 'small-world' networks. Nature 393(6684):440–442

Zachary W (1977) An information flow model for conflict and fission in small groups. J Anthropol Res 33:452–473

Zhang H, Giles CL, Foley HC, Yen J (2007) Probabilistic community discovery using hierarchical latent gaussian mixture model. In: AAAI'07: proceedings of the 22nd national conference on artificial intelligence. AAAI Press, Menlo Park, pp 663–668