Finding local community structure in networks

Aaron Clauset*

Department of Computer Science, University of New Mexico, Albuquerque, New Mexico 87131, USA
(Received 22 February 2005; published 29 August 2005)

Although the inference of global community structure in networks has recently become a topic of great interest in the physics community, all such algorithms require that the graph be completely known. Here, we define both a measure of local community structure and an algorithm that infers the hierarchy of communities that enclose a given vertex by exploring the graph one vertex at a time. This algorithm runs in time $O(k^2d)$ for general graphs when d is the mean degree and k is the number of vertices to be explored. For graphs where exploring a new vertex is time consuming, the running time is linear, $O(k)$. We show that on computer-generated graphs the average behavior of this technique approximates that of algorithms that require global knowledge. As an application, we use this algorithm to extract meaningful local clustering information in the large recommender network of an online retailer.

PACS number(s): 89.75.Hc, 05.10.−a, 87.23.Ge, 89.20.Hh

I. INTRODUCTION

Recently, physicists have become increasingly interested in representing the patterns of interactions in complex systems as networks [1–4]. Canonical examples include the Internet [5], the World Wide Web [6], social networks [7], citation networks [8,9], and biological networks [10]. In each case, the system is modeled as a graph with n vertices and m edges, e.g., physical connections between computers, friendships between people, and citations among academic papers.

Within these networks, the global organization of vertices into *communities* has garnered broad interest both inside and beyond the physics community. Conventionally, a community is taken to be a group of vertices in which there are more edges between vertices within the group than to vertices outside of it. Although the partitioning of a network into such groups is a well-studied problem, older algorithms tend to only work well in special cases [11–15]. Several algorithms have recently been proposed within the physics community, and have been shown to reliably extract the known community structure in real world networks [16–21]. Similarly, the computer science community has proposed algorithms based on the concept of flow [22].

However, each of these algorithms require knowledge of the entire structure of the graph. This constraint is problematic for networks like the World Wide Web, which for all practical purposes is too large and too dynamic to ever be known fully, or networks that are larger than can be accommodated by the fastest algorithms [21]. In spite of these limitations, we would still like to make quantitative statements about community structure, albeit confined to some accessible and known region of the graph in question. For instance, we might like to quantify the local communities of either a person given their social network or a particular website given its local topology in the World Wide Web.

Here, we propose a general measure of local community structure, which we call *local modularity*, for graphs in which we lack global knowledge and that must be explored one vertex at a time. We then define a fast agglomerative algorithm that maximizes the local modularity in a greedy fashion, and test the algorithm's performance on a series of computer-generated networks with known community structure. Finally, we use this algorithm to analyze the local community structure of the online retailer Amazon.com's recommender network, which is composed of more than 400 000 vertices and 2 million edges. Through this analysis, we demonstrate the existence of a mesoscopic network structure that is distinct from both the microstructure of vertex statistics and the global community structure previously given in [21]. Interestingly, we find a wide variety of local community structures, and that generally, the local modularity of the network surrounding a vertex is negatively correlated with its degree.

II. LOCAL MODULARITY

The inference of community structure can generally be reduced to identifying a partitioning of the graph that maximizes some quantitative notion of community structure. However, when we lack global knowledge of the graph's topology, a measure of community structure must necessarily be independent of those global properties. For instance, this requirement precludes the use of the modularity metric Q, due to Newman and Girvan [17], as it depends on m, the total number of edges.

Suppose that in the graph $\mathcal{G}$, we have perfect knowledge of the connectivity of some set of vertices, i.e., the known portion of the graph, which we denote $\mathcal{C}$. This necessarily implies the existence of a set of vertices $\mathcal{U}$ about which we know only their adjacencies to $\mathcal{C}$. Further, let us assume that the only way we may gain additional knowledge about $\mathcal{G}$ is by visiting some neighboring vertex $v_i \in \mathcal{U}$, which yields a list of its adjacencies. As a result, v_i becomes a member of $\mathcal{C}$, and additional unknown vertices may be added to $\mathcal{U}$. This vertex-at-a-time discovery process is directly analogous to the manner in which "spider" or "crawler" programs harvest the hyperlink structure of the World Wide Web.
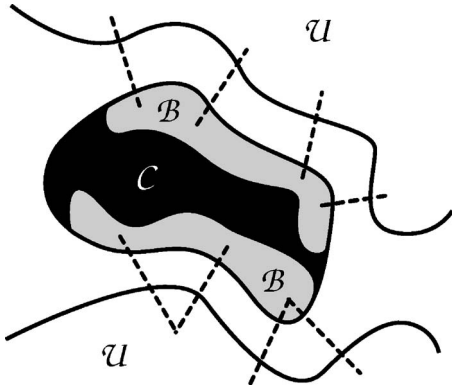
*Electronic address: aaron@cs.unm.edu

FIG. 1. An illustration of the division of an abstract graph into the local community $\mathcal{C}$, its boundary $\mathcal{B}$, and the edges that connect $\mathcal{B}$ to the largely unknown neighbors $\mathcal{U}$.

The adjacency matrix of such a partially explored graph is given by

$$
A_{ij} = \begin{cases} 1, & \text{if vertices } i \text{ and } j \text{ are connected,} \\ & \text{and either vertex is in } \mathcal{C}; \\ 0, & \text{otherwise.} \end{cases} \tag{1}
$$

If we consider $\mathcal{C}$ to constitute a local community, the most simple measure of the quality of such a partitioning of $\mathcal{G}$ is simply the fraction of known adjacencies that are completely internal to $\mathcal{C}$. This quantity is given by

$$
\frac{\sum\limits_{ij} A_{ij}\xi(i,j)}{\sum\limits_{ij} A_{ij}} = \frac{1}{2m^*}\sum_{ij} A_{ij}\xi(i,j), \tag{2}
$$

where $m^* = \frac{1}{2}\Sigma_{ij} A_{ij}$, the number of edges in the partial adjacency matrix, and $\xi(i,j)$ is 1 if both v_i and v_j are in $\mathcal{C}$, and 0 otherwise. This quantity will be large when $\mathcal{C}$ has many internal connections and few connections to the unknown portion of the graph. This measure also has the property that when $|\mathcal{C}| \gg |\mathcal{U}|$, the partition will almost always appear to be good.

If we restrict our consideration to those vertices in the subset of $\mathcal{C}$ that have at least one neighbor in $\mathcal{U}$, i.e., the vertices that make up the *boundary* of $\mathcal{C}$ (Fig. 1), we obtain a direct measure of the sharpness of that boundary. Additionally, this measure is independent of the size of the enclosed community. Intuitively, we expect that a community with a sharp boundary will have few connections from its boundary to the unknown portion of the graph, while having a greater proportion of connections from the boundary back into the local community. In the interest of keeping the notation concise, let us denote those vertices that comprise the boundary as $\mathcal{B}$, and the boundary-adjacency matrix as

$$
B_{ij} = \begin{cases} 1, & \text{if vertices } i \text{ and } j \text{ are connected,} \\ & \text{and either vertex is in } \mathcal{B}; \\ 0, & \text{otherwise.} \end{cases} \tag{3}
$$

Thus, we define the local modularity R to be

$$
R = \frac{\sum\limits_{ij} B_{ij}\delta(i,j)}{\sum\limits_{ij} B_{ij}} = \frac{I}{T}, \tag{4}
$$

where $\delta(i,j)$ is 1 when either $v_i \in \mathcal{B}$ and $v_j \in \mathcal{C}$, or *vice versa*, and is 0 otherwise. Here, T is the number of edges with one or more end points in $\mathcal{B}$, while I is the number of those edges with neither end point in $\mathcal{U}$. This measure assumes an unweighted graph, although the weighted generalization is straightforward [23].

A few comments regarding this formulation are worthwhile before proceeding. By considering the fraction of boundary edges that are internal to $\mathcal{C}$, we ensure that our measure of local modularity lies on the interval $0 < R < 1$, where its value is directly proportional to sharpness of the boundary given by $\mathcal{B}$. This is true except when the entire component has been discovered, at which point R is undefined. If we like, we may set $R = 1$ in that case in order to match the intuitive notion that an entire component constitutes the strongest kind of community. Finally, there are certainly alternative measures that can be defined on $\mathcal{B}$, however, in this paper we consider only the one given.

III. THE ALGORITHM

For graphs like the World Wide Web, in which one must literally crawl the network in order to discover the adjacency matrix, any analysis of local community structure must necessarily begin at some source vertex v_0. In general, if the explored portion of the graph has k vertices, the number of ways to partition it into two sets, those vertices considered a part of the same local community as the source vertex and those considered outside of it, is given by $2^{k-2} - 1$, which is exponential in the size of the explored portion of the network. In this section, we describe an algorithm that only takes time polynomial in k, and that infers local community structure by using the vertex-at-a-time discovery process subject to maximizing our measure of local modularity.

Initially, we place the source vertex in the community, $v_0 = \mathcal{C}$, and place its neighbors in $\mathcal{U}$. At each step, the algorithm adds to $\mathcal{C}$ (and to $\mathcal{B}$, if necessary) the neighboring vertex that results in the largest increase (or smallest decrease) in R, breaking ties randomly. That is, for each vertex $v_j \in \mathcal{U}$, we calculate the ΔR_j that corresponds to the change in local modularity as a result of joining v_j to $\mathcal{C}$. The vertex that results in the largest positive change is then joined. Finally, we add to $\mathcal{U}$ any newly discovered vertices, and update our estimate of R. This process continues until it has agglomerated either a given number of vertices k, or it has discovered the entire enclosing component, whichever happens first. Generally, these operations can be done quite quickly by an appropriate use of arrays and linked lists. The pseudocode for this process is given in Table I. As we will see in the two subsequent sections, this algorithm performs well on both computer-generated graphs with some known community structure and on real world graphs in which we can manually verify the sensibility of the inferred communities.

TABLE I. The general algorithm for the greedy maximization of local modularity, as given in the text.

---

add v_0 to $\mathcal{C}$

add all neighbors of v_0 to $\mathcal{U}$

set $\mathcal{B} = v_0$

while $|\mathcal{C}| < k$ **do**

for each $v_j \in \mathcal{U}$ **do**

compute ΔR_j from Eq. (5)

end for

find v_j such that its ΔR_j is maximum

add that v_j to $\mathcal{C}$

add all new neighbors of that v_j to $\mathcal{U}$

update R and $\mathcal{B}$

end while

The computation of the ΔR_j associated with each $v_j \in \mathcal{U}$ can be done quickly using an expression derived from Eq. (4):

$$\Delta R_j = \frac{x - Ry - z(1 - R)}{T - z + y}, \qquad (5)$$

where x is the number of edges in T that terminated at v_j, y is the number of edges that will be added to T by the agglomeration of v_j (i.e., the degree of v_j is $x+y$), and z is the number of edges that will be removed from T by the agglomeration. Because ΔR_j depends on the current value of R, and on the y and z that correspond to v_j, each step of the algorithm takes time proportional to the number of vertices in $\mathcal{U}$. This is roughly kd, where d is the mean degree of the graph; we note that this will be a significant overestimate for graphs with nontrivial clustering coefficients, significant community structure, or when $\mathcal{C}$ is a large portion of the graph. Thus, in general, the running time for the algorithm is $O(k^2 d)$, or simply $O(k^2)$ for a sparse graph, i.e., when $m \sim n$. As it agglomerates vertices, the algorithm outputs a function $R(t)$, the local modularity of the community centered on v_0 after t steps, and a list of vertices paired with the time of their agglomeration.

The above calculation of the running time is somewhat misleading, as it assumes that the algorithm is dominated by the time required to calculate the ΔR_j for each vertex in $\mathcal{U}$; however, for graphs like the World Wide Web, where adding a new vertex to $\mathcal{U}$ requires the algorithm to fetch a web page from a remote server, the running time will instead be dominated by the time-consuming retrieval. When this is true, the running time is linear in the size of the explored subgraph, $O(k)$.

A few comments regarding this algorithm are due. Because we greedily maximize the local modularity, a neighboring high degree vertex will only be agglomerated if a sufficient number of its neighbors have been explored. It is this behavior that prevents the algorithm from crossing a community boundary until absolutely necessary. Additionally, the algorithm is somewhat sensitive to the degree distribution of the source vertex's neighbors: when the source
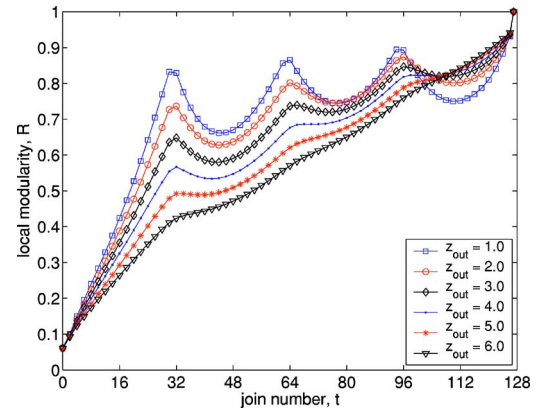

FIG. 2. (Color online) Average local modularity R as a function of the number of agglomerations t (details described in the text; error bars are omitted for clarity). By varying the expected number of intercommunity edges per node z_{out}, the strength of the community boundaries are varied.

degree is high, the algorithm will first explore its low degree neighbors. This implicitly assumes that high degree vertices are likely to sit at the boundary of several local communities. While certainly not the case in general, this may be true for some real world networks. We shall return to this idea in a later section.

Finally, although one could certainly stop the algorithm once the first enclosing community has been found, in principle, there is no reason that it cannot continue until some arbitrary number of vertices have been agglomerated. Doing so yields the hierarchy of communities that enclose the source vertex. In a sense, this process is akin to the following: given the dendrogram of the global community hierarchy, walk upward toward the root from some leaf v_0 and observe the successive hierarchical relationships, as represented by junctions in the dendrogram. Thus, the enclosing communities inferred by our algorithm for some source vertex is the community hierarchy *from the perspective of that vertex*.

IV. COMPUTER-GENERATED GRAPHS

As has become standard with testing community inference techniques, we apply our algorithm to a set of computer-generated random graphs that have known community structure [17]. In these graphs, $n=128$ vertices are divided into four equal-sized communities of 32 vertices. Each vertex has a total expected degree z that is divided between intra- and intercommunity edges such that $z = z_{in} + z_{out}$. These edges are placed independently and at random so that, in expectation, the values of z_{in} and z_{out} are respected. By holding the expected degree constant $z=16$, we may tune the sharpness of the community boundaries by varying z_{out}. Note that for these graphs, when $z_{out}=12$, edges between vertices in the same group are just as likely as edges between vertices that are not.

Figure 2 shows the average local modularity R as a function of the number of steps t, over 500 realizations of the graphs described above, with a randomly selected source
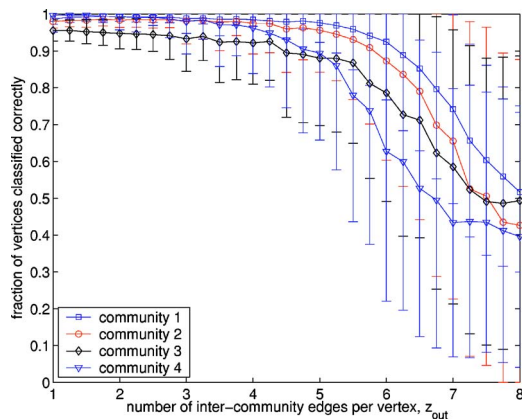
FIG. 3. (Color online) Fraction of correctly classified nodes, by community, as a function of the number of inter-community edges z_{out}. Although there the variance increases as the community boundaries become less sharp, the average behavior degrades gracefully.

FIG. 4. (Color online) Local modularity R for three items in the Amazon.com recommender network, shown on log-linear axes. For comparison, the time series for a random graph with the same degree distribution is shown. The large open symbols indicate the locations of the five strongest enclosing communities.

vertex. For the sake of clarity, only data series for $z_{out} \leqslant 6.0$ are shown and error bars are omitted. Sharp community boundaries correspond to peaks in the curve. As z_{out} grows, the sharpness of the boundaries and the height of the peaks decrease proportionally. When the first derivative is positive everywhere, e.g., for $z_{out} > 5$, the inferred locations of the community boundaries may be extracted by finding local minima in the second derivative, possibly after some smoothing.

From this information we may grade the performance of the algorithm on the computer-generated graphs in the following manner: we label each vertex with its true community membership, run the local inference algorithm with a randomly selected source vertex, detect the community boundaries as described above, and compute the fraction of correctly classified vertices for each community. Averaging over 500 instantiations provides an estimate, as a function of z_{out}, of the mean fraction of correctly classified vertices for each community (Fig. 3); error bars depict one standard deviation. As a method for inferring the first enclosing community, our algorithm correctly classifies, on average, more than 50% of the vertices even when the boundaries are weak, i.e., when $z_{out} = 8$. We note that for such computer-generated graphs, our algorithm's average performance approximates that of more global methods [17–19], although the variance in the quality of classification is substantially larger for weak boundaries. Indeed, such an increase is to be expected as the algorithm we describe here may be misled by large fluctuations in local structure, while global algorithms may not.

Recently, another approach to inferring community structure using only local information appeared [24]. This alternative technique relies upon growing a breadth-first tree outward from the source vertex v_0, until the rate of expansion falls below an arbitrary threshold. The uniform exploration has the property that some level in the tree will correspond to a good partitioning only when v_0 is equidistant from all parts of its enclosing community's boundary. On the other hand, by exploring the surrounding graph one vertex at a time, our algorithm will avoid crossing boundaries until it has explored the remainder of the enclosing community.
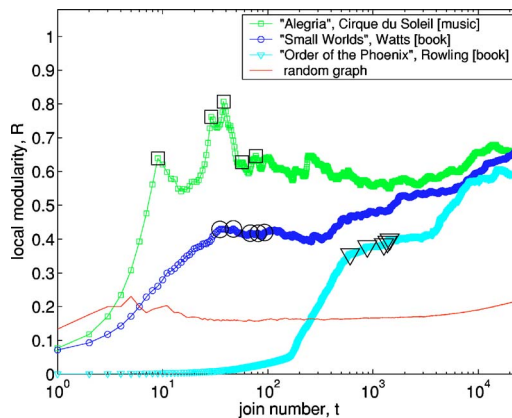
V. LOCAL COPURCHASING HABITS

In this section, we apply our local inference algorithm to the recommender network of Amazon.com, collected in August 2003, which has $n = 409\,687$ vertices, $m = 2\,464\,630$ edges, and thus a mean degree of 12.03. We note that the degree distribution is fairly right-skewed, having a standard deviation of 14.64. Here, vertices are items such as books and digital media sold on Amazon's website, while edges connect pairs of items that are frequently purchased together by customers. It is this copurchasing data that yields recommendations for customers as they browse the online store. Although, in general, the algorithm we have described is intended for graphs like the World Wide Web, the Amazon recommender network has the advantage that, by virtue of being both very large and fully known, we may explore global regularities in local community structure without concern for sampling bias in the choice of source vertices. Additionally, we may check the inferred the community structures against our, admittedly heuristic, notions of correctness.

As illustrative examples, we choose three qualitatively different items as source vertices: the compact disk *Alegria* by Cirque du Soleil, the book *Small Worlds* and the book *Harry Potter and the Order of the Phoenix* by J. K Rowling. These items have degree 15, 19, and 3117, respectively. At the time the network data was collected, the Harry Potter book was the highest degree vertex in the network, its release date having been June 2003. For each of these items, we explore $k = 25\,000$ associated vertices. Figure 4 illustrates the local modularity as a function of the number of steps t for each item; an analogous data series for a random graph with the same degree distribution [25] has been plotted for comparison. We mark the locations of the five principle enclosing communities with large open symbols, where their location was determined by the same second derivative test described above.

These time series have several distinguishing features. First, *Alegria* has the smallest enclosing communities, composed of $t = \{10, 30, 39, 58, 78\}$ vertices, and these communi-
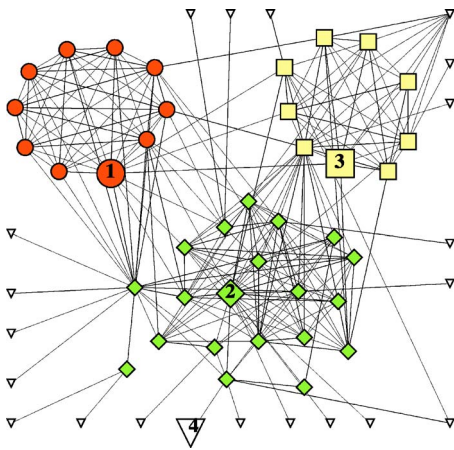
FIG. 5. The first three enclosing communities for Cirque du Soleil's *Alegria* in Amazon.com's recommender network; communities are distinguished by shape (circles, diamonds, squares, respectively). Connections to triangles represent connections to items in the remaining unknown portion of the graph. *Alegria* and *Order of the Phoenix* are denoted by 1 and 4, respectively.

FIG. 6. (Color online) The average local modularity over the first 250 steps for source vertices with degree at least d. The "knee" in the upper data series is located at $d=13$; the mean degree for the network is 12.03. The logarithmic falloff illustrates the negative correlation between the source vertex degree and the strength of the surrounding local community.

ties are associated with high values of local modularity. The first five enclosing communities all have $R>0.62$, while the third community corresponds to $R=0.81$, indicating that only about 20% of the boundary edges reach out to the rest of the network. In contrast, the communities of *Small Worlds* contain $t=\{36,48,69,82,94\}$ vertices, while the Harry Potter book's communities are extremely large, containing $t=\{607,883,1270,1374,1438\}$ vertices. Both sets have only moderate values of local modularity, $R \leq 0.43$. It is notable that the local modularity functions for all three items follow relatively distinct trajectories until the algorithm has agglomerated roughly 10 000 items. Beyond that point, the curves begin to converge, indicating that, from the perspectives of the source vertices, the local community structure has become relatively similar.

To illustrate the inferred local structure, we show the partial subgraph that corresponds to the first three enclosing local communities for the compact disc *Alegria* in Fig. 5. Here, communities are distinguished by shape according to the order of discovery (circle, diamond, and square, respectively), and vertices beyond these communities are denoted by triangles. Items in the first enclosing community are uniformly compact disks produced by Cirque du Soleil. Items in the second are slightly more diverse, including movies and books about the troupe, the Cirque du Soleil compact disk entitled *Varekai*, and one compact disk by a band called Era; the third group contains both new and old Cirque du Soleil movies. *Varekai* appears to have been placed outside the first community because it has fewer connections to those items than to items in the subsequent enclosing communities. Briefly, we find that the enclosing local communities of *Small Worlds* are populated by texts in sociology and social network analysis, while the Harry Potter book's communities have little topical similarity.

In Fig. 5, the labels 1 and 4 denote the items *Alegria* and *Order of the Phoenix*, respectively. It is notable that these items are only three steps away in the graph, yet have ex-
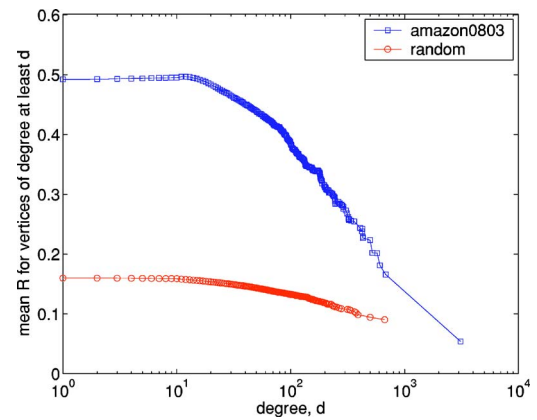
tremely different local community structures (Fig. 4). If an item's popularity is reflected by its degree, then it is reasonable to believe that the strength of the source vertex's local community structure may be inversely related to its degree. That is, popular items like *Order of the Phoenix* may connect to many well-defined communities by virtue of being purchased by a large number of customers with diverse interests, while niche items like Cirque du Soleil's *Alegria* exhibit stronger local community structure from more specific copurchasing habits. Such a structure appears to be distinct from both the macroscopic structure discovered using global community inference methods [21] and the microscopic structure of simple vertex statistics such as the clustering coefficient or assortative mixing, i.e., topological correlations based on degree or vertex-type similarity [26]. We call this intermediate level of topological structure "mesoscopic."

With the exception of social networks, the degree of adjacent vertices in most complex networks appears to be negatively correlated in most networks. This property is often called "disassortative" mixing [26], and can be caused by a high clustering coefficient, global community structure, or a specific social mechanism [27]. However, for the Amazon recommender network, we find that the assortativity coefficient is statistically no different from zero, $r=-3.01\times10^{-19}\pm1.49\times10^{-4}$, yet the network exhibits a nontrivial clustering coefficient, $c=0.17$ and strong global community structure with a peak modularity of $Q=0.745$ [21]. Returning to the suggested inverse relationship between the degree of the source vertex and the strength of its surrounding community structure, we sample for 100 000 random vertices the average local modularity over the first $k=250$ steps. We find the average local modularity to be relatively high, $\langle R_{amzn}\rangle=0.49\pm0.08$, while a random graph with the same degree distribution yields $\langle R_{rand}\rangle=0.16\pm0.01$. The variance for the Amazon graph is due to the contributions of high degree vertices. In Fig. 6, we plot from our random sample the average local modularity for all source vertices with degree of at least d. Notably, the average is

relatively constant until $d=13$, after which it falls off logarithmically. This supports the hypothesis that, in the recommender network, there is a weak inverse relationship between the degree of the source vertex and the strength of its surrounding local community.

Naturally, there are many ways to use the concept of local community structure to understand the properties of real world networks. Further characterizations of the Amazon graph are beyond the scope of this paper, but we propose a rigorous exploration of the relationship between the source vertex degree and its surrounding local community structure as a topic for future work.

VI. CONCLUSIONS

Although many recent algorithms have appeared in the physics literature for the inference of community structure when the entire graph structure is known, there has been little consideration of graphs that are either too large for even the fastest known techniques, or that are, like the World Wide Web, too large or too dynamic to ever be fully known. Here, we define a measure of community structure that depends only on the topology of some known portion of a graph. We then give a simple fast, agglomerative algorithm that greedily maximizes our measure as it explores the graph one vertex at a time. When the time it takes to retrieve the adjacencies of a vertex is small, this algorithm runs in time $O(k^2d)$ for general graphs when it explores k vertices, and the graph has mean degree d. For sparse graphs, i.e., when $m\sim n$, this is simply $O(k^2)$. On the other hand, when visiting a new vertex to retrieve its adjacencies dominates the running time, e.g., downloading a web page on the World Wide Web, the algorithm takes time linear in the size of the explored subgraph, $O(k)$. Generally, if we are interested in making quantitative statements about local structure, that is, when $k\ll n$, it is much more reasonable to use an algorithm which is linear or even quadratic in k, than an algorithm that is linear in the size of the graph n. Finally, we note that our algorithm's simplicity will make it especially easy to incoporate into web spider or crawler programs for the discovery of local community structures on the World Wide Web graph.

Using computer-generated graphs with known community structure, we show that our algorithm extracts this structure and that its average performance approximates that of more global inference algorithms [17–19] that rely on global information. We then apply our algorithm to the large recommender network of the online retailer Amazon.com, and extract the local hierarchy of communities for several qualitatively distinct items. Additionally, we show that a vertex's degree is inversely related to the strength of its surrounding local structure. This discovery points to the existence of topological structure in real world networks that is above the single-vertex structure such as the clustering coefficient, but below that of the global community structure. Finally, this algorithm should allow researchers to characterize the structure of a wide variety of other graphs, such as the World Wide Web, and we look forward to seeing such applications.

[1] S. H. Strogatz, Nature (London) **410**, 268 (2001).

[2] R. Albert and A.-L. Barabási, Rev. Mod. Phys. **74**, 47 (2002).

[3] S. N. Dorogovtsev and J. F. F. Mendes, Adv. Phys. **51**, 1079 (2002).

[4] M. E. J. Newman, SIAM Rev. **45**, 167 (2003).

[5] M. Faloutsos, P. Faloutsos, and C. Faloutsos, Comput. Commun. Rev. **29**, 251 (1999).

[6] J. M. Kleinberg, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, "The Web as a graph: Measurements, models and methods," in *Proceedings of the International Conference on Combinatorics and Computing*, No. 1627 in Lecture Notes in Computer Science (Springer-Verlag, Berlin, 1999), pp. 1–18.

[7] S. Wasserman and K. Faust, *Social Network Analysis* (Cambridge University Press, Cambridge, 1994).

[8] D. J. de S. Price, Science **149**, 510 (1965).

[9] S. Redner, e-print physics/0407137.

[10] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, Proc. Natl. Acad. Sci. U.S.A. **98**, 4569 (2001).

[11] B. W. Kernighan and S. Lin, Bell Syst. Tech. J. **49**, 291 (1970).

[12] M. Fiedler, Czech. Math. J. **23**, 298 (1973).

[13] A. Pothen, H. Simon, and K.-P. Liou, SIAM J. Matrix Anal. Appl. **11**, 430 (1990).

[14] J. Scott, *Social Network Analysis: A Handbook*, 2nd ed., Sage, London (2000).

[15] M. E. J. Newman, Eur. Phys. J. B **38**, 321 (2004).

[16] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **99**, 7821 (2002).

[17] M. E. J. Newman, Phys. Rev. E **69**, 026113 (2004).

[18] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, Proc. Natl. Acad. Sci. U.S.A. **101**, 2658 (2004).

[19] M. E. J. Newman, Phys. Rev. E **69**, 066133 (2004).

[20] F. Wu and B. A. Huberman, Eur. Phys. J. B **38**, 331 (2004).

[21] A. Clauset, M. E. J. Newman, and C. Moore, Phys. Rev. E **70**, 066111 (2004).

[22] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee, IEEE Computer **35** (3), 66 (2002).

[23] M. E. J. Newman, Phys. Rev. E **70**, 056131 (2004).

[24] J. P. Bagrow and E. M. Bollt, e-print cond-mat/0412482.

[25] B. Bollobás, *Random Graphs* (Academic, New York, 1985).

[26] M. E. J. Newman, Phys. Rev. E **67**, 026126 (2003).

[27] M. E. J. Newman and J. Park, Phys. Rev. E **68**, 036122 (2003).