Local method for detecting communities

James P. Bagrow[1] and Erik M. Bollt[2,1]
[1]*Department of Physics, Clarkson University, Potsdam, New York 13699-5820, USA*
[2]*Department of Math and Computer Science, Clarkson University, Potsdam, New York 13699-5815, USA*
(Received 24 December 2004; published 10 October 2005)

We propose a method of community detection that is computationally inexpensive and possesses physical significance to a member of a social network. This method is unlike many divisive and agglomerative techniques and is local in the sense that a community can be detected within a network without requiring knowledge of the entire network. A global application of this method is also introduced. Several artificial and real-world networks, including the famous Zachary karate club, are analyzed.

DOI: 10.1103/PhysRevE.72.046108 PACS number(s): 89.75.Hc, 05.10.−a, 87.23.Ge, 89.20.Hh

I. INTRODUCTION

It has been shown in the past that many interesting systems can be represented as networks composed of vertices and edges [1–4]. Such systems include the Internet [5], social and friendship networks [6], food webs [7], and citation networks [8,9]. For example, a social network may represent people as vertices and edges linking vertices when those people are on a first-name basis.

A topic of current interest in the area of networks has been the idea of communities and their detection. A community could be loosely described as a collection of vertices within a graph that are densely connected amongst themselves while being loosely connected to the rest of the graph [10–12]. Many networks exhibit such a community structure and this motivates our work. This description, however, is somewhat vague and open to interpretation. This leads to the possibility that different techniques for detecting these communities may lead to slightly different yet equally valid results. We emphasize this variation in Sec. II D.

Several techniques have been proposed to detect community structure inside of a network. The recent and highly successful betweenness centrality algorithm due to Newman and Girvan [13–15] performs well within a variety of networks but it is costly to compute [$O(n^2m)$ on a graph with n vertices and m edges] [15]. More importantly, while betweenness centrality has been shown to be a useful quantity for detecting community structure, it is knowledge not usually attainable to a vertex *within the graph*.

In this paper we ask, if a person were to move to a new town, what actions would he or she take to see what community or communities they belong to? Most community detection methods using hierarchical clustering fall within two categories: divisive and agglomerative [6,15]. Both forms, including those using betweenness and other methods, are global algorithms and do not represent feasible actions that a member of a network could undertake to identify the network's community structure. The method proposed here may better represent actions that members of a network would undertake to identify their own communities.

II. ALGORITHM

The proposed algorithm consists of an l shell spreading outward from a starting vertex. As the starting vertex's nearest neighbors and next-nearest neighbors, etc., are visited by the l shell, two quantities are computed: the emerging degree and total emerging degree. The emerging degree of a vertex is defined as the number of edges that connect that vertex to vertices the l shell has not already visited as it expanded from the previous $l-1, l-2, \ldots$ shells. Note that edges between vertices within the same l shell do not contribute to the emerging degree. Let us define the following notation for the emerging degree and total emerging degree:

$$k_i^e(j) = \text{emerging degree of vertex } i \text{ from a shell started at vertex } j,$$

$$K_j^l = \text{total emerging degree of a shell of depth } l \text{ starting from vertex } j. \tag{1}$$

The total emerging degree of an l shell is then the sum of the emerging degrees of all vertices on the leading edge of the l shell. This can also be thought of as the total number of *emerging edges* from that l shell [16]. We see that the total emerging degree at depth l is not necessarily the number of vertices at depth $l+1$. At depth 0, the total emerging degree is just the degree of the starting vertex. At depth l, it is the total number of edges from vertices at depth l connected to vertices at depth $>l$.

It follows from Eq. (1) that

$$K_j^0 = k_j,$$

$$K_j^l = \sum_{i \in S_j^l} k_i^e(j), \tag{2}$$

where S_j^l is the leading edge of the l shell—that is, the set of all vertices exactly l steps away from vertex j.

In addition, let us define the *change in total emerging degree*

$$\Delta K_j^l = \frac{K_j^l}{K_j^{l-1}}, \tag{3}$$

for a shell at depth l starting from vertex j.

The algorithm works by expanding an l shell outward from some starting vertex j and comparing the change in total emerging degree to some threshold α. When

TABLE I. Algorithm 1, the local algorithm to determine a starting vertex's community.[a]

$s \in V$; // *s is the starting vertex*

$K_s^{d-1} \longleftarrow 1$;

$Q \longleftarrow$ empty queue; // *search queue*

$C \longleftarrow$ empty queue; // *community queue*

enqueue $s \longrightarrow Q$;

$K_s^d \longleftarrow$ *emerging* $(Q, C, G(V, E))$;

$\Delta K_s^d \longleftarrow \frac{K_s^d}{K_s^{d-1}}$;

while $\Delta K_s^d > \alpha$ **do**

  $K_s^{d-1} \longleftarrow K_s^d$;

  foreach $q \in Q$ **do**

    dequeue $q \longleftarrow Q$;

    enqueue $q \longrightarrow C$;

    enqueue $neighbors(q) \longrightarrow Q$;

  end

  $K_s^d \longleftarrow$ *emerging* $(Q, C, G(V, E))$;

  $\Delta K_s^d \longleftarrow \frac{K_s^d}{K_s^{d-1}}$;

end

[a]Note that emerging is a function of the l shell and the graph that returns the total emerging degree.

$$\Delta K_j^l < \alpha, \tag{4}$$

the l shell ceases to grow and all vertices covered by shells of a depth $\leq l$ are listed as members of vertex j's community.

We describe our algorithm roughly as follows. For a starting vertex j, do the following.

1. Start an l shell, $l=0$, at vertex j (add j to the list of community members) and compute K_j^0.

2. Spread the l shell, $l=1$, add the neighbors of j to the list, and compute K_j^1.

3. Compute ΔK_j^1. If $\Delta K_j^1 < \alpha$, then a community has been found. Stop the algorithm.

4. Else repeat from step (ii) for the next l shell, until α is crossed or the entire connected component is added to the community list.

See the algorithm 1 in Table I for a more exact pseudocode.

Since there tend to be many interconnections within a community, so defined, as an l shell grows outward from some starting vertex within a community, the total emerging degree will tend to increase. See Sec. III for more discussion of an idealized graph model with community structure. When the l shell reaches the "border" of the community, the number of emerging edges will decrease sharply. This is because, at this point, the only emerging edges are those connecting the community to the rest of the graph which are, by definition, less in number than those within the community.

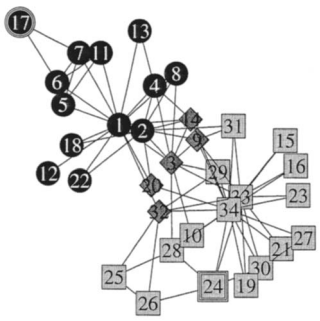

FIG. 1. Two local results on the Zachary karate club with $\alpha = 1.9$. Boxes and diamonds represent the output of algorithm 1 when starting from vertex 24, while circles and diamonds represent the output when starting from vertex 17.

By introducing a single parameter α and monitoring ΔK_j^l, the l shell's growth can be stopped when it has covered the community. It is this fact that allows for the starting vertex to detect its community locally: at the last depth before α is crossed, it does not matter where the emerging edges lead. See Sec. II A for results using our purely local method.

Our method is not perfect, however, and it is possible for the l shell to "spill over" the community it is detecting. This is dependent on how the starting vertex is situated within the graph: if it is closer (or equally close) to some noncommunity vertex or vertices than to some community vertices, the l shell may spread along two or more communities at the same time. To alleviate this effect, one can run the algorithm N times, using each vertex as a starting vertex, and then achieve a *group consensus* as to which vertices belong to which communities. This idea is discussed in Sec. II B.

The idea of having an expanding l shell encompass a community is not in itself new here. The hub-based algorithm in [16] expands multiple l shells simultaneously from

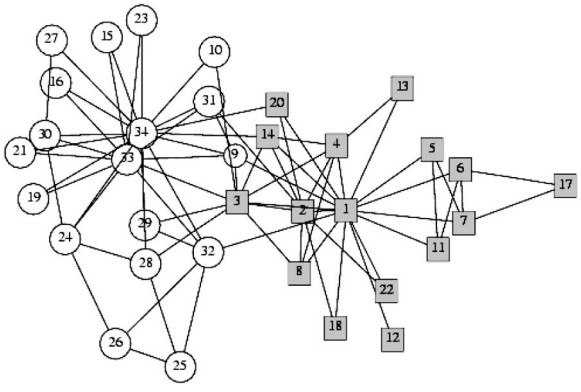

FIG. 2. Actual breakdown of the karate club [18].

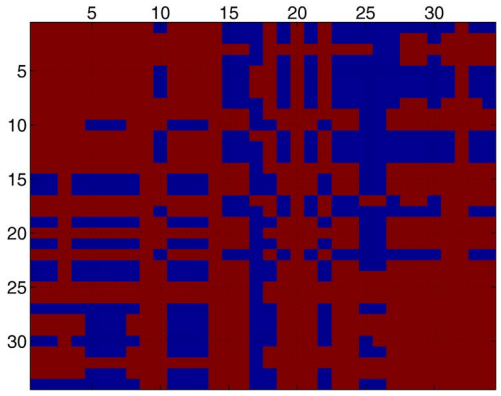

FIG. 3. (Color online) Membership matrix M for the Zachary karate club before sorting, with $\alpha=1.2$. Gray boxes indicate a value of 1, white 0.

the n vertices of the highest degree (the hubs) until all vertices are within an l shell. While computationally inexpensive, this method has the following drawback: the number of communities detected is arbitrarily preassigned and the algorithm neglects the possibility of having two hubs within the same community. In addition, it requires knowledge of the entire graph; it is a global algorithm, not local.

A. Select local results

We have applied algorithm 1 to the Zachary karate club as shown in Fig. 1, Figure 2 shows the actual split the club underwent. We complete two runs, one starting from vertex 17 and another from vertex 24. Five vertices (3, 9, 14, 20, and 32) were claimed by both runs as members of that starting vertex's community. Note that this graph and all subsequent graphs and dendrograms were drawn using [17].

We interpret our results as follows. The five vertices that are listed as members of both starting vertex's communities tend to fall on the "border" between the two groups. This makes sense to us since each vertex is linked roughly equally to both communities. One can imagine these five members had the most difficult choice to make when the club split. Far from being an unwanted result, this overlap could be used to predict vertices that may be more likely to switch communities in the future (in an evolving network) or which vertices are least isolated within a single community.

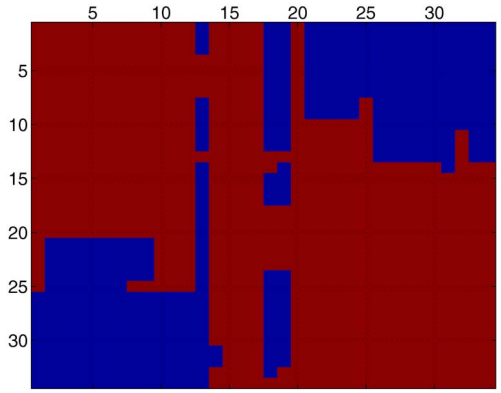

FIG. 4. (Color online) Sorted membership matrix $\tilde{M}$ for the karate club, with $\alpha=1.2$.

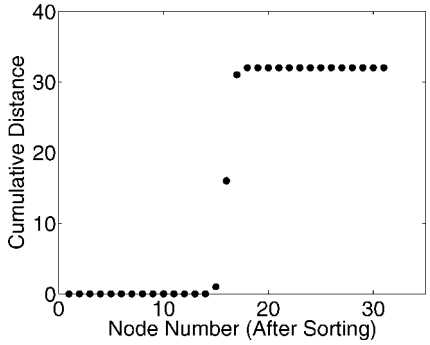

FIG. 5. Cumulative row distances for ideal graph 1. Generated using the membership matrix shown in Fig. 6.

B. Obtaining global information

Algorithm 1 is a method for a single vertex to determine something about its community membership. It seems reasonable that, by surveying all the locally determined membership listings, one should be able to generate an idea of the global structure of the network. Here we propose a simple method using a *membership matrix* to obtain such a picture and to overcome membership overlap (discussed in Sec. II A) when determining a "consensus" partitioning of the network.

For any given starting vertex j, algorithm 1 can return a vector v_j of size N, where the ith component is 1 if vertex i is a member of the starting vertex's community and 0 otherwise. These vectors can be assembled to form an $N\times N$ membership matrix

$$M = (\mathrm{v}_1|\mathrm{v}_2|\cdots|\mathrm{v}_N)^t, \tag{5}$$

where the jth row contains the results from using vertex j as the algorithm's starting point. This allows for a good way to visualize the resultant data when starting the algorithm from multiple vertices.

We define a *distance* between rows i and j of the mem-

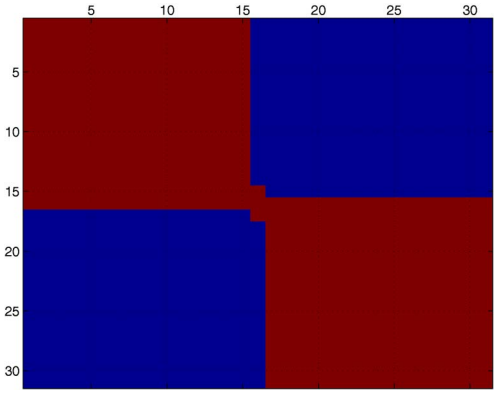

FIG. 6. (Color online) Membership matrix for ideal graph 1. Note that no sorting was required. $\alpha=1$, but for these idealized model graphs, the results are identical for a wide range of α values.

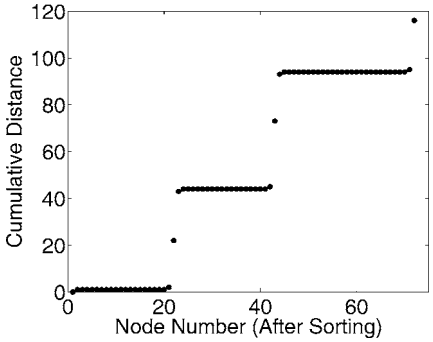

FIG. 7. Row distances for ideal graph 2. Generated using the membership matrix shown in Fig. 8.

bership matrix as the total number of differences between their components:

$$\text{distance}(i,j) = n - \sum_{k=1}^{n} \delta(M_{ik}, M_{jk}), \tag{6}$$

where $\delta(M_{ik}, M_{jk}) = 1$ if $M_{ik} = M_{jk}$ and 0 otherwise.

Now we perform a simple sorting algorithm on M. For the ith row do the following.

1. Find distance (i,j) for all rows $j > i$.
2. Pick the row that is the smallest distance to row i (call it row k) and interchange it with row $i+1$. This requires swapping rows $i+1$ and k and swapping columns $i+1$ and k. Columns are swapped because a row interchange is equivalent to a renumbering of the involved vertices, so that new numbering must be kept consistent throughout M.
3. Repeat for row $i+1$.

Unfortunately, the sorting step can be computationally expensive. Finding distance (i,j) costs $O(N)$. When the sorting algorithm begins at the first row, there are $N-1$ distances to find, so the cost of the first sort is $O(N(N-1)) \sim O(N^2)$. This is then repeated for the next row, costing $O(N(N-2)) \sim O(N^2)$, and so on for each additional row. Since there are N rows, the total cost is

$$\sum_{i=1}^{N} N(N-i) = N\left(N^2 - \frac{1}{2}N(N+1)\right) = O(N^3). \tag{7}$$

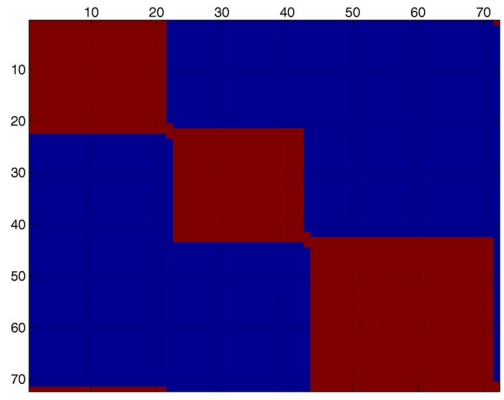

FIG. 8. (Color online) Membership matrix for ideal graph 2.

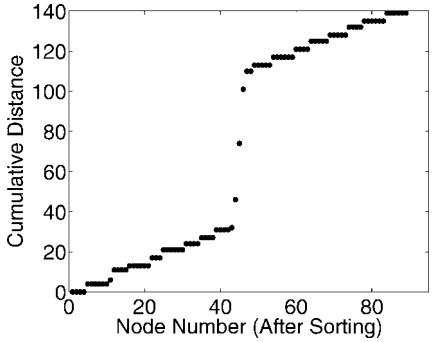

FIG. 9. Row distances for ideal graph 3, $\alpha = 0.87$. Generated using the membership matrix shown in Fig. 18.

The result of this sorting and renumbering is a membership matrix that is much more indicative of structure. Specifically, we have a sorted membership matrix

$$\tilde{M} = P^t M P, \tag{8}$$

where P is a permutation matrix effectively resulting from the above. Well-separated communities appear as blocks along the diagonal, and imperfections within the blocks can indicate substructure (see Figs. 3 and 4).

C. Finding a hierarchy of subcommunities

Sorting the membership matrix already provides a useful means of visualizing the results of all the different runs of the local algorithm, but this is not enough to determine how any present subcommunities relate to larger communities. Therefore, here we introduce a further operation to apply to $\tilde{M}$ to generate a dendrogram of the community structure. For row i, we compute a cumulative row distance (CD_i):

$$\text{CD}_1 = 0,$$

$$\text{CD}_i = \text{distance}(i, i-1) + \text{CD}_{i-1} = \sum_{j=2}^{i} \text{distance}(j-1, j). \tag{9}$$

Plotting the row number i versus the cumulative distance CD_i will yield a collection of points of increasing value fall-

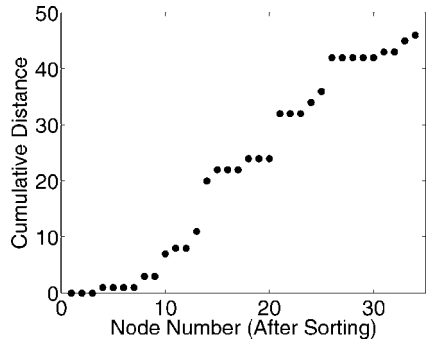

FIG. 10. Cumulative row distances for the karate club, computed using the membership matrix shown in Fig. 4 ($\alpha = 1.2$).

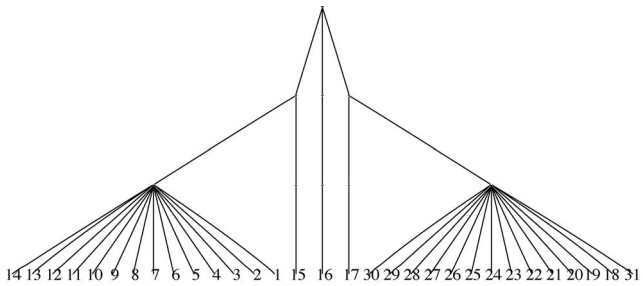

FIG. 11. Dendrogram for ideal graph 1. Notice that central member vertex 16 is idealized here as a community unto itself, which, in light of the form of the graph in Fig. 20, simply means that 16 is central between two communities. Notice also the special placement of members 15 and 17.

ing into discrete bands that indicate the members of each community. Note that the row number i is the new sorted number i for that vertex: one needs to keep track of all the individual sorting operations to maintain the original number of that vertex—that is, through the permutations P. This step of finding these cumulative distances costs $\sim O(N^2)$ operations. See Figs. 5–10, for plots of examples of these cumulative row distances for various networks. These plots are useful for visualization but are not strictly necessary to get the subcommunity hierarchy.

Finally, to yield a dendrogram of the community structure, the following operation is performed

1. $d \leftarrow 1$.
2. Compute distance $(i-1,i)$ for all $i=2,\ldots,n$.
3. Choose the smallest distance (often zero for identical rows) and call it D_{min}.
4. $C_d \leftarrow$ empty queue // *clustering queue*.
5. enqueue first vertex $\rightarrow C_d$.
6. *For* $i=2,\ldots,n$:
6.1. *If* distance $(i-1,i) > D_{min}$:
6.1.1. $d \leftarrow d+1$.
6.1.2. $C_d \leftarrow$ empty queue.
6.2. enqueue ith vertex $\rightarrow C_d$.
7. Repeat from step 3 for next smallest distance until all distances have been used.

Essentially, we are moving down the rows of $\tilde{M}$ and grouping together all the vertices whose corresponding rows are closer together than D_{min} until we arrive at a row that is farther away than D_{min}. Then we start a new group and begin grouping the subsequent vertices together until we *again* find a row that is farther away than D_{min} and so forth. This is then repeated using the next smallest distance as D_{min}. This has a course-graining effect: as we use larger distances for D_{min}, farther vertices will start grouping together.

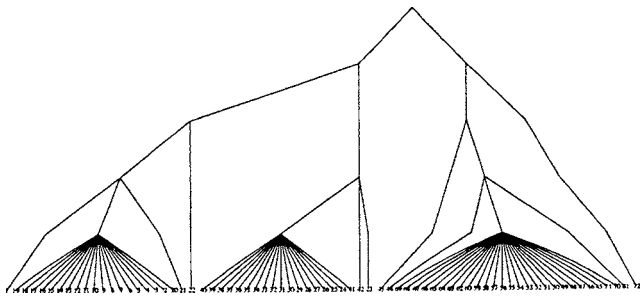

FIG. 12. Dendrogram for ideal graph 2. Again, as in Fig. 11, notice the results for central members 22, 43, and 72 and also the boundary members 1, 21, 23, 42, 44, and 61.

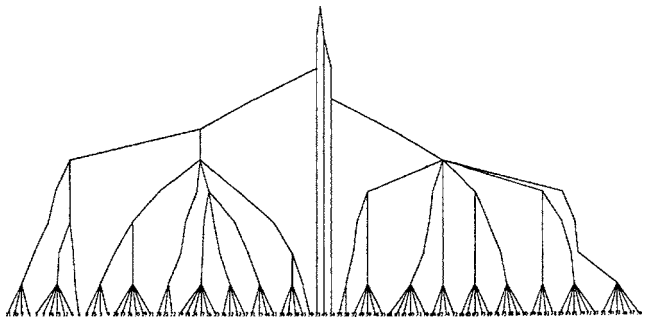

FIG. 13. Dendrogram for ideal graph 2, $\alpha=0.87$. Now with central members, for different scale communities, the dendrogram becomes more complicated in contrast to those in Figs. 11 and 12.

Grouping the rows of $\tilde{M}$ in this way is equivalent to grouping the vertices of the graph together into a subcommunity hierarchy. This is also similar in form to many agglomerative techniques, with the row distances of $\tilde{M}$ used as a similarity measure. These groupings can then be used to generate a dendrogram of the subcommunity structure if we assume that each vertex is a singleton before we started grouping and that after the largest distance is used, all vertices are grouped together. See Figs. 11–16 for such dendrograms.

D. Impact of α

The algorithm is based on a single parameter α which controls when to stop the spread of the l shell. When $\alpha=0$, the l shell will never stop until the entire connected component has been visited. As α increases in size, l shells will tend to stop growing sooner, until eventually they do not spread beyond the starting vertex and the final result will be N singleton communities. This is guaranteed to happen when $\alpha > k_{max}$, where k_{max} is the largest degree in the network.

The impact of varying α is readily apparent in Figs. 17–19. In Fig. 17, the smaller α allowed the l shells to spread farther: many l shells starting from vertices close to the main partition (the two edges of highest betweenness) have spread to the entire network. In Fig. 19, the larger value of α truncated the l shells before they had a chance to spread beyond the subcommunities of the starting vertices.

In contrasting Figs. 17–19, we emphasize that how one defines a community through an algorithm bears on the specific results. It is our contention that there is not a single true answer for a community partition. We hold that the flexibility of a parameter like α to allow for various levels of community courseness is in fact quite natural: the result is in the eye of the beholder—of the specific algorithm. In any case, as can be seen by our examples in Sec. IV, intermediate values of α lead to community partitions which agree well with many found in the literature, and we think that they make

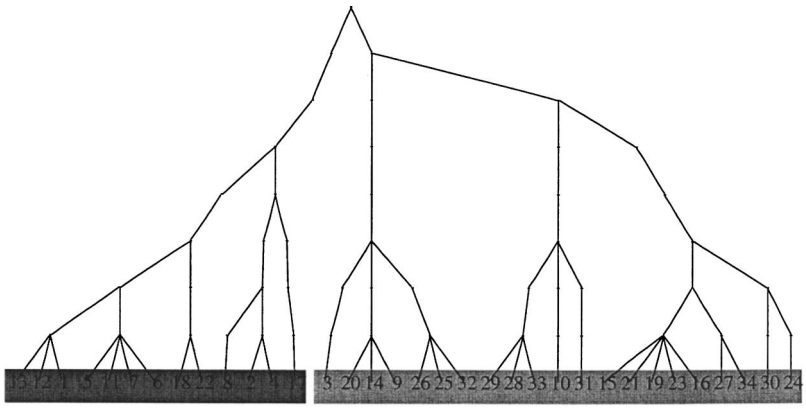

FIG. 14. Dendrogram for the karate club, using the membership matrix shown in Fig. 4 ($\alpha=1.2$).

good sense in light of our model interpretation as described in the next section.

One can think of α as a measure of the "friendliness" of the starting vertex, to use a social network analogy. When α is small ($\alpha \ll 1$), the l shells will spread to much of the network. This can indicate vertices that are more likely to include other vertices in their respective communities or, in a social network, people who are more welcoming of their neighbors. Similarly, when α is large ($\alpha \gg 1$), the l shells will stop growing immediately. This can be indicative of vertices that are unlikely to include other vertices in their community or hermitlike people who are unwilling to accept even their immediate neighbors into their communities, instead preferring to remain a singleton. In this sense, α can be thought of as an inverse measure of friendliness or social acceptance.

III. MOTIVATING EXAMPLES

We propose the following idealized models for a network with simple community structure, and we test our algorithm first in these cases. Our models demonstrate the high-density intraconnections and low-density interconnections. We define an idealized community of size N as a complete subgraph ($k_j=\langle k \rangle = N-1$), with one or more additional edges linking it to one or more additional ideal communities.

These networks represent the extreme fulfillment of the idea of a community. Each community has the maximum number of internal links possible while having close to the minimum number of external links. This results in the number of emerging edges dropping off very sharply at the border of each subgraph, leading to nearly identical results when α and the starting vertices are varied.

Several configurations of these ideal networks are analyzed (Figs. 20–22). These networks also provide a means of visualizing, understanding, and interpreting how the membership matrix may look for a given community structure. In addition, these networks contain single vertices situated between the subgraphs. Since these vertices are equally connected to multiple communities, the results from algorithm 1

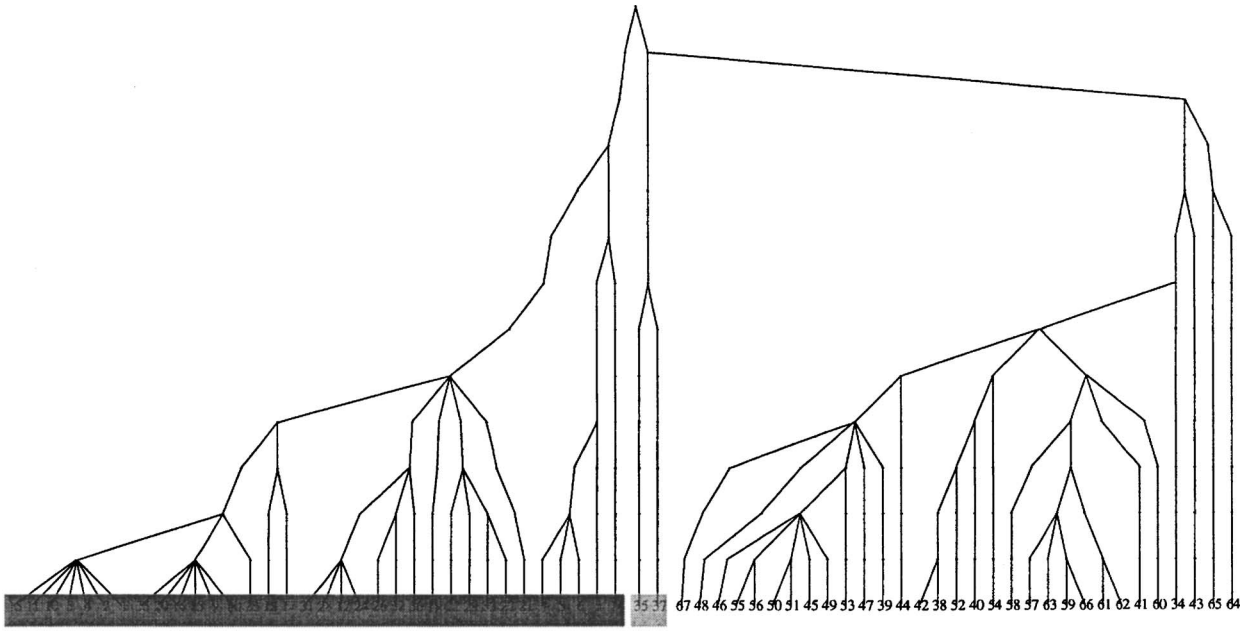

FIG. 15. Dendrogram for the books on politics network. Generated using the membership matrix shown in Fig. 25. Note that vertices 35 and 37 are rows 34 and 35, respectively, of the membership matrix.

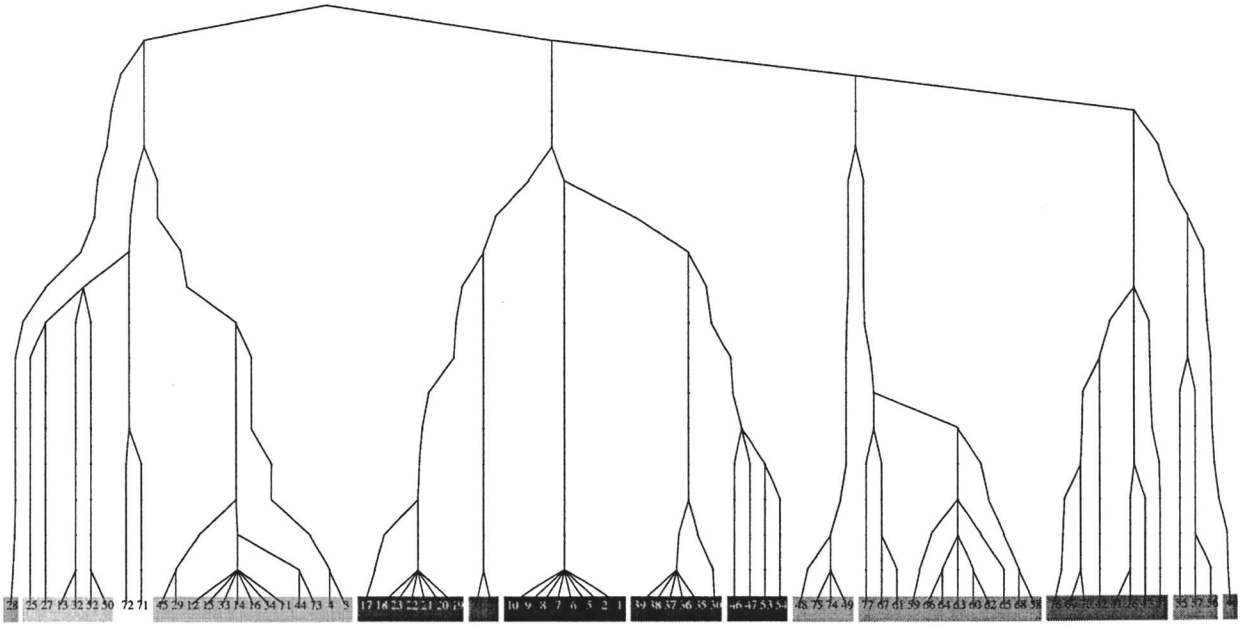

FIG. 16. Dendrogram for the Les Mis social network. Generated using the membership matrix shown in Fig. 27.

starting from these vertices will contain all the subgraphs that the vertices are linked to. This is evident in the rows of the membership matrix that overlap two or more blocks.

Through these models, we can better understand how to interpret the performance of our algorithm; we believe these models in fact make suitable benchmarks for other community partitioning algorithms found in the literature. One can easily assemble a graph with a given community structure, apply a community partitioning algorithm to it, and compare the results of that algorithm with the structure created when the graph was assembled.

In addition, it is useful to note that these networks require little or no sorting of their membership matrices. This is because the vertices are already numbered consecutively: vertices 1 through i are community 1, vertices $i+1$ through j are community 2, etc. This is, of course, a contrived result and cannot be expected in general.

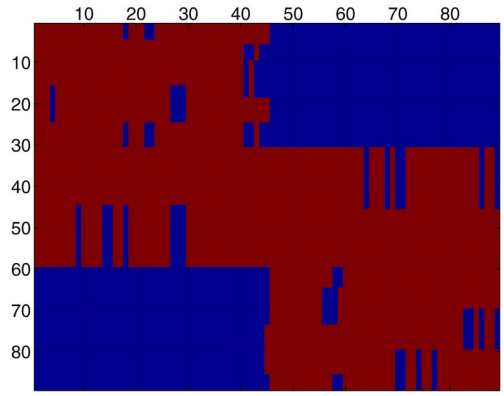

FIG. 17. (Color online) Membership matrix for ideal graph 3 after sorting, with $\alpha=0.25$. Note the increased number of "spilled" vertices in the the middle rows, as compared with Fig. 18.

IV. REAL-WORLD NETWORKS

The proposed algorithm performs extremely well on idealized networks (see Sec. III), but how does it perform on real-world networks? Here we analyze the Zachary karate club, the network of coappearances present in the novel *Les Misérables* by Victor Hugo, and the partisan network of copurchased books on American politics.

A. Zachary karate club

The Zachary karate club is perhaps the most famous network in terms of community structure [19]. The club suffered from infighting and eventually split in half, providing actual evidence of the community structure, at least at the topmost level. Thus, it provides an excellent means to compare the accuracy of any proposed detection methods.

For $\alpha=1.2$, we achieve a reasonable result: three vertices (3, 14, and 20) are labeled incorrectly as compared with the

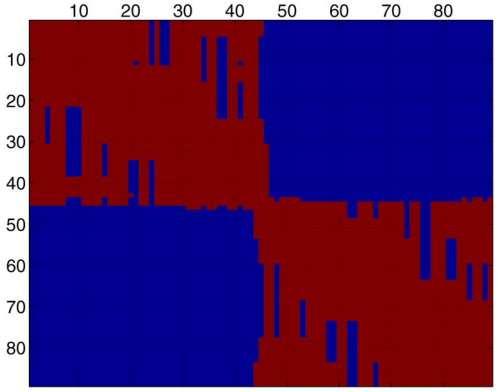

FIG. 18. (Color online) Membership matrix for ideal graph 3, $\alpha=0.87$.

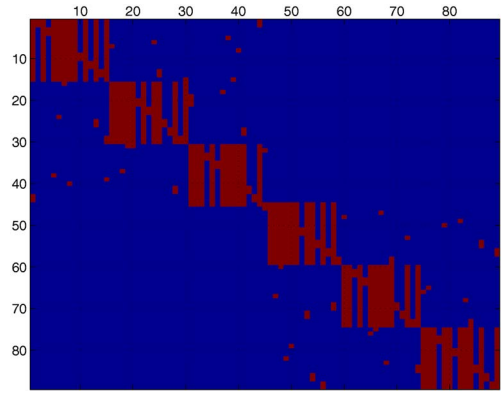

FIG. 19. (Color online) Membership matrix for ideal graph 3 after sorting, with $\alpha=3$. The larger value of α stops the l shells sooner, allowing for the smaller subcommunities to become evident instead of the main partitioning.

betweenness partitioning [15] and the actual split the club underwent [19]. Looking at the network itself (Fig. 23), all of the disputed vertices are situated on the "border" of one community or the other. One would expect these vertices to be the most likely to be labeled incorrectly because, unlike more idealized networks, these vertices are almost equally connected to both communities.

Figures 4, 10, and 14 contain the membership matrix, cumulative row distances plot, and dendrogram, respectively.

B. Books on politics

A network possessing a fairly simple two-community structure is the network of copurchased books on American politics shown in Fig. 24 [20]. As can be seen from Figs. 25 and 15, the results are extremely reasonable.

C. Les Misérables

Another network with an interesting community structure is the network of character coappearances from the novel *Les Misérables* by Victor Hugo [15]. This network, shown in Fig. 26, differs from the karate club and the political books networks in that there are several communities of smaller size

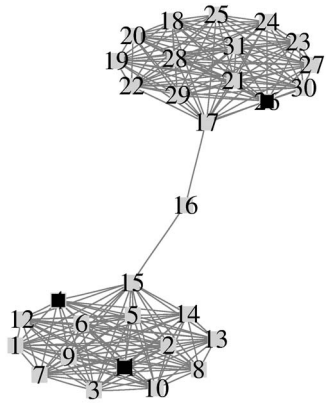

FIG. 20. Ideal graph 1: two complete subgraphs of size 15 bridged by a common vertex.

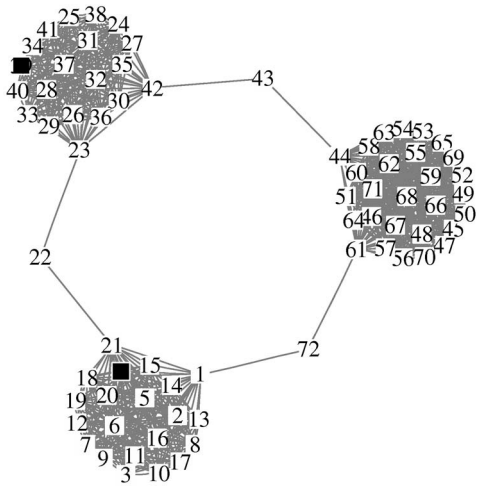

FIG. 21. Ideal graph 2: Three complete subgraphs, one larger than the others.

rather than two large communities. As can be seen from the membership matrix in Fig. 27, some of the communities separated quite well, while others were detected rather poorly, at least compared to the results in [15].

V. CONCLUSIONS

In this paper, we have introduced algorithm 1, a method for detecting community structure. This method is local and may be applied in situations where other methods are too inefficient—for example, when one is concerned with a single community and not the complete community structure of a graph. A single parameter α is used, making it very easy to tune the output of the algorithm, as was shown in Sec. II D.

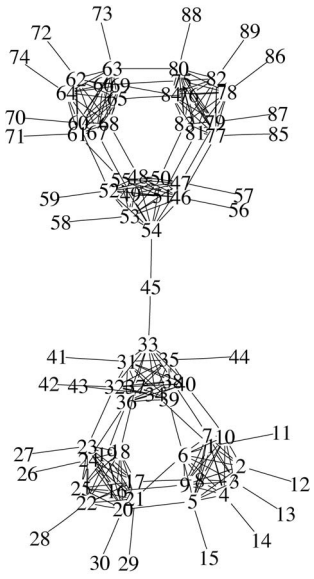

FIG. 22. Ideal graph 3: three complete subgraphs joined together by multiple edges with singletons attached bridged by a single vertex to another group of similar (but not identical) structure.

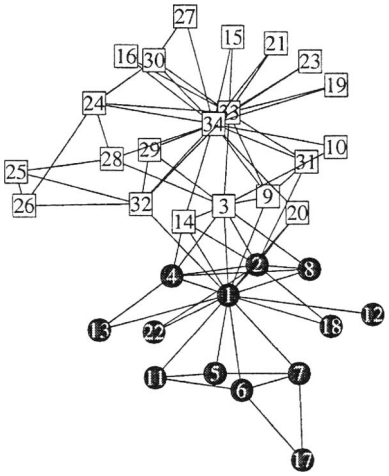

FIG. 23. The Zachary karate club. The shading indicates the membership of the two clusters of the topmost branch of the dendrogram (Fig. 14).

We have also proposed one possible method of applying algorithm 1 globally (see Secs. II B and II C). Sorting the membership matrix is expensive and limits this global application to smaller networks. In addition, the membership matrix is, unlike an adjacency matrix, not necessarily sparse for a sparse graph: it will only be sparse when the sizes of the individual communities are all much smaller than the size of the network as a whole. This limits the possibility of replacing the membership matrix with a more efficient data structure.

We feel that algorithm 1 is a useful result, due to both its simplicity and flexibility. For example, our global application could be easily altered to become more efficient: one imagines the cost of the sorting algorithm can be offset a great deal by only starting algorithm 1 from a fraction of the vertices, f, instead of all vertices in the graph. This reduces the cost of the sorting algorithm by a factor of f^3, a real savings

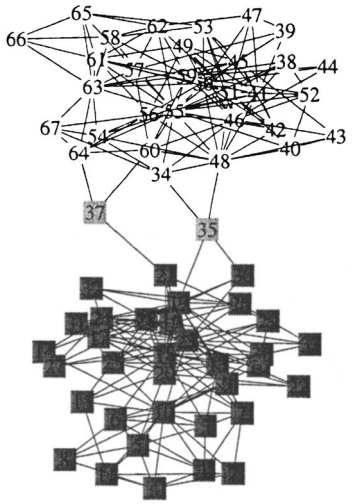

FIG. 24. The network of copurchased books on American politics [20]. Here a link is drawn between two vertices if those books were purchased together from a major online retailer.

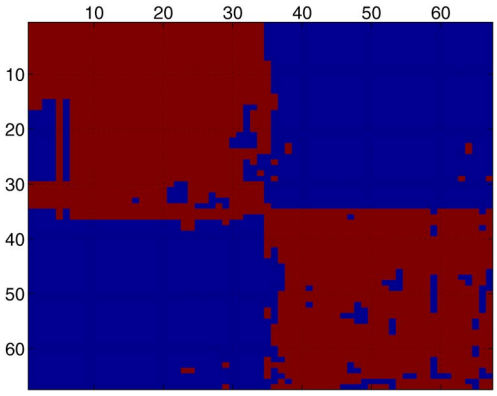

FIG. 25. (Color online) Membership matrix for the books on politics network. $\alpha=1.2$.

for small f, with the presumed trade-off being a reduction in accuracy as f decreases. Other applications of algorithm 1 besides the expensive use of the membership matrix may also be discovered.

The Zachary karate club has become an almost canonical representative of a community structure. The possibility remains, however, that outside factors may not be represented in the dataset. If this is true, then the club's fissure should not be used as the sole means of justification for a community detection method. For example, some of the border nodes, represented as diamonds in Fig. 1, could well have joined either community, based solely on the network at hand. This can lead to ambiguity in the final partition. The point is that the algorithm defines the community; the community should not define the algorithm.

Another concept, often neglected in determining communities, is the idea of a relative result. Who is to say that someone in a town agrees with what community the rest of the town feels he or she belongs to? It seems feasible that a vertex that is equally linked to two communities in a graph is just as likely to correspond to a person who thinks he or she is a member of both communities as it is to correspond to a person who feels they are independent of both larger communities. If one considers the output of algorithm 1 to be

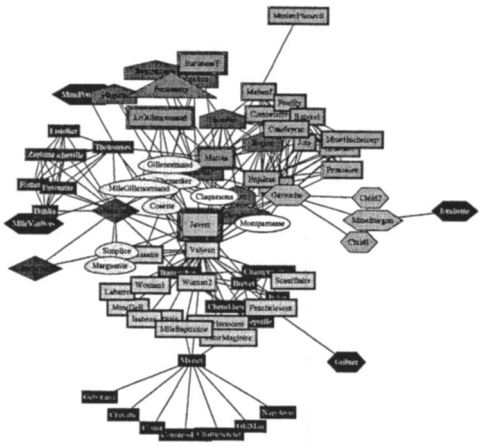

FIG. 26. The network of character coappearances from the novel *Les Misérables* by Victor Hugo. See Fig. 16 for partitioning.

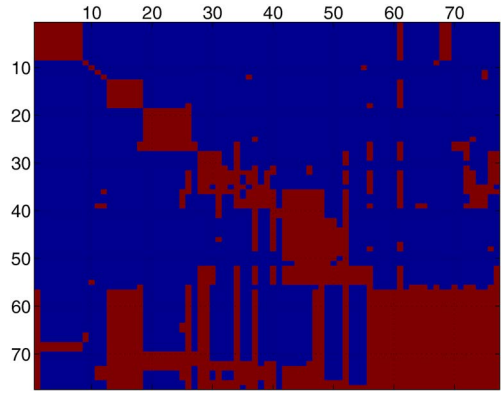

FIG. 27. (Color online) Membership matrix (after sorting) for the Les Mis network. $\alpha=6.9$.

what the starting vertex "believes" to be his or her community, then this method may prove to be a useful tool when analyzing relative results. This is not useful for many applications of community detection BOD where a final structure is necessary, such as minimizing cross talk between parallel processors in a computer, but it may prove very useful in areas such as social networks.

ACKNOWLEDGMENTS

The authors are grateful to Hernan Rozenfeld and Daniel ben-Avraham for useful discussions and to Mark Newman for useful data. This work was supported by the NSF under Grant No. DMS0404778.

[1] S. H. Strogatz, Nature (London) **410**, 268 (2001).
[2] R. Albert and A.-L. Barabási, Rev. Mod. Phys. **74**, 47 (2002).
[3] M. E. J. Newman, SIAM Rev. **45**, 167 (2003).
[4] S. N. Dorogovtsev and J. F. F. Mendes, *Evolution of Networks: From Biological Nets to the Internet and WWW* (Oxford University Press, Oxford, 2003).
[5] M. Faloutsos, P. Faloutsos, and C. Faloutsos, Comput. Commun. Rev. **29**, 251 (1999).
[6] J. Scott, *Social Network Analysis: A Handbook*, 2nd ed. (Sage Publications, London, 2000).
[7] J. A. Dunne, R. J. Williams, and N. D. Martinez, Proc. Natl. Acad. Sci. U.S.A. **99**, 12917 (2002).
[8] S. Redner, Eur. Phys. J. B **4**, 131 (1998).
[9] M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **98**, 404 (2001).
[10] S. Wasserman and K. Faust, *Social Network Analysis* (Cambridge University Press, Cambridge, England, 1994).
[11] G. W. Flake, S. R. Lawrence, C. L Giles, and F. M. Coetzee, IEEE Comput. **35**, 66 (2002).
[12] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, Proc. Natl. Acad. Sci. U.S.A. **101**, 2658 (2004)
[13] M. E. J. Newman, Phys. Rev. E **67**, 026126 (2003).
[14] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **99**, 7821 (2002).
[15] M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).
[16] L. da F. Costa, e-print cond-mat/0405022.
[17] GRAPHIZ, Graph Visualization Software homepage: http://www.graphviz.org/
[18] Image courtesy Mark Newman's site at http://www-personal.umich.edu/~mejn/networks/
[19] W. W. Zachary, J. Anthropol. Res. **33**, 452 (1977).
[20] V. Krebs, Political Patterns on the WWW—Divided We Stand homepage: http://www.orgnet.com/divided2.html