Community Structure in Directed Networks

E. A. Leicht[1] and M. E. J. Newman[1,2]

[1]Department of Physics, University of Michigan, Ann Arbor, Michigan 48109, USA
[2]Center for the Study of Complex Systems, University of Michigan, Ann Arbor, Michigan 48109, USA

(Received 8 October 2007; published 21 March 2008)

We consider the problem of finding communities or modules in directed networks. In the past, the most common approach to this problem has been to ignore edge direction and apply methods developed for community discovery in undirected networks, but this approach discards potentially useful information contained in the edge directions. Here we show how the widely used community finding technique of modularity maximization can be generalized in a principled fashion to incorporate information contained in edge directions. We describe an explicit algorithm based on spectral optimization of the modularity and show that it gives demonstrably better results than previous methods on a variety of test networks, both real and computer generated.

DOI: 10.1103/PhysRevLett.100.118703 PACS numbers: 89.75.Hc, 02.10.Ox, 02.50.−r

At the most fundamental level a network consists of a set of nodes or vertices connected in pairs by lines or edges, but many variations are possible, including networks with directed edges, weighted edges, labels on nodes or edges, and others. This flexible structure lends itself to the modeling of a wide array of complex systems, and networks have, as a result, attracted considerable attention in the recent physics literature [1,2].

Many networks are found to display ''community structure,'' dividing naturally into communities or modules with dense connections within communities but sparser connections between them. Communities are of interest both in their own right as functional building blocks within networks and for the insights they offer into the dynamics or modes of formation of networks, and a large volume of research has been devoted to the development of algorithmic methods for discovering communities—see [3] for a review. Nearly all of these methods, however, have one thing in common: they are intended for the analysis of undirected network data. Many of the networks that we would like to study are directed, including the World Wide Web, food webs, many biological networks, and even some social networks. The commonest approach to detecting communities in directed networks has been simply to ignore the edge directions and apply algorithms designed for undirected networks. This works reasonably well in some cases, although in others it does not, as we will see in this Letter. Even in the cases where it works, however, it is clear that in discarding the directions of edges we are throwing away a good deal of information about our network's structure, information that, at least in principle, could allow us to make a more accurate determination of the communities.

Several previous studies, including our own, have touched on this problem [4–7], but they have typically not tackled the community structure question directly. In this Letter we propose a method for finding communities in directed networks that makes explicit use of the information contained in edge directions. The method is an extension of the well established modularity optimization approach for undirected networks [8], an approach that has been shown to be both computationally efficient and highly effective in practical applications [3].

The premise of the modularity optimization method is that a good division of a network into communities will give high values of the benefit function Q, called the modularity, defined by [9]

$$Q = (\text{fraction of edges within communities}) - (\text{expected fraction of such edges}). \quad (1)$$

Large positive values of the modularity indicate when a statistically surprising fraction of the edges in a network fall within the chosen communities; it tells us when there are more edges within communities than we would expect on the basis of chance.

The expected fraction of edges is typically evaluated within the so-called configuration model, a random graph conditioned on the degree sequence of the original network, in which the probability of an edge between two vertices i and j is $k_i k_j/2m$, where k_i is the degree of vertex i and m is the total number of edges in the network. The modularity can then be written

$$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta_{c_i,c_j}, \quad (2)$$

where A_{ij} is an element of the adjacency matrix, δ_{ij} is the Kronecker delta symbol, and c_i is the label of the community to which vertex i is assigned. Then one maximizes Q over possible divisions of the network into communities, the maximum being taken as the best estimate of the true communities in the network. Neither the size nor the number of communities need be fixed; both can be varied freely in our attempt to find the maximum.

In practice, exact optimization of the modularity is computationally hard, so practical methods based on modularity optimization make use of approximate optimization

schemes such as greedy algorithms, simulated annealing, spectral methods, and others [8,10–13].

In recent work on complexity reduction in networks, Arenas *et al.* [5] have proposed a generalization of the modularity to directed networks, which can be understood in the following way. The expected positions of edges in a directed network depend on their direction. Consider two vertices, A and B. Vertex A has high out-degree but low in-degree while vertex B has the reverse situation. This means that a given edge is more likely to run from A to B than vice versa, simply because there are more ways it can fall in the first direction than in the second. Hence if we *observe* in our real network that there is an edge from B to A, it should be considered a bigger surprise than an edge from A to B and hence should make a bigger contribution to the modularity, since modularity should be high for statistically surprising configurations.

We put these insights to work as follows. Given the in- or out-degree sequence of our directed network, we can create a directed equivalent of the configuration model, which will have an edge from vertex j to vertex i with probability $k_i^{\text{in}}k_j^{\text{out}}/m$, where k_i^{in} and k_j^{out} are the in- and out-degrees of the vertices. (Note that there is no factor of 2 in the denominator now.) Then the equivalent of Eq. (2) for a directed network is

$$Q = \frac{1}{m}\sum_{ij}\left[A_{ij} - \frac{k_i^{\text{in}}k_j^{\text{out}}}{m}\right]\delta_{c_i,c_j}, \tag{3}$$

which is a special case of the formula given in [5]. Here A_{ij} is defined in the conventional manner to be 1 if there is an edge from j to i and zero otherwise. Note that edges $j \to i$ do indeed make larger contributions to this expression if k_i^{in} and/or k_j^{out} is small.

As in the undirected case we can make use of the modularity to find network communities by searching for the division of the network that maximizes Q. One can in principle make use of any of the methods previously applied to modularity maximization, such as simulated annealing or greedy algorithms. Here we derive the appropriate generalization of the spectral optimization method of Newman [13], which is both computationally efficient and appears to give excellent results in practice.

We consider first the simplified problem of dividing a directed network into just two communities. We define s_i to be $+1$ if vertex i is assigned to community 1 and -1 if it is assigned to community 2. Note that this implies that $\sum_i s_i^2 = n$. Then $\delta_{c_i,c_j} = \frac{1}{2}(s_i s_j + 1)$ and

$$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{k_i^{\text{in}}k_j^{\text{out}}}{m}\right](s_i s_j + 1) = \frac{1}{2m}\mathbf{s}^T\mathbf{B}\mathbf{s}, \tag{4}$$

where $\mathbf{s}$ is the vector whose elements are the s_i, $\mathbf{B}$ is the so-called modularity matrix with elements

$$B_{ij} = A_{ij} - \frac{k_i^{\text{in}}k_j^{\text{out}}}{m}, \tag{5}$$

and we have made use of $\sum_{ij}A_{ij} = \sum_i k_i^{\text{in}} = \sum_j k_j^{\text{out}} = m$. Our goal is now to find the $\mathbf{s}$ that maximizes Q for a given $\mathbf{B}$.

In the undirected case the modularity matrix is symmetric but in the present case it is, in general, not, and the lack of symmetry will cause technical problems if we blindly attempt to duplicate the eigenvector-based machinery presented for undirected networks in [13]. We can, however, restore symmetry to the problem by the following trick. Noting that Q is a scalar and therefore equal to its own transpose, we take the transpose of Eq. (4) to give $Q = (2m)^{-1}\mathbf{s}^T\mathbf{B}^T\mathbf{s}$ and then take the average of this expression and Eq. (4) to give

$$Q = \frac{1}{4m}\mathbf{s}^T(\mathbf{B} + \mathbf{B}^T)\mathbf{s}. \tag{6}$$

The matrix $\mathbf{B} + \mathbf{B}^T$ is manifestly symmetric, and it is on this matrix that we focus forthwith. Notice that $\mathbf{B} + \mathbf{B}^T$ is not the same as the modularity matrix for a symmetrized version of the network in which direction is ignored, and hence we expect methods based on the true directed modularity to give different results, in general, to methods based on the undirected version.

The leading constant $1/4m$ in Eq. (6) is conventional, but makes no difference to the position of the maximum of Q, so for the sake of clarity we neglect it in defining our optimization procedure.

The standard approach to optimization problems of this type is first to solve the "relaxed" problem in which the s_i are allowed to take any real value, not just ± 1. Enforcing the normalization constraint $\sum_i s_i^2 = n$ with a Lagrange multiplier and differentiating, we then find that the maximum of Q is achieved when $\mathbf{s}$ is parallel to the vector $\mathbf{v}$ satisfying $(\mathbf{B} + \mathbf{B}^T)\mathbf{v} = \beta\mathbf{v}$, where β is the largest (most positive) eigenvalue of $\mathbf{B} + \mathbf{B}^T$. In other words $\mathbf{s}$ should be chosen parallel to the leading eigenvector of $\mathbf{B} + \mathbf{B}^T$. Unfortunately, our problem carries the additional constraint that $s_i = \pm 1$, which normally prevents us from reaching this simple eigenvector optimum, but we do the best we can and make $\mathbf{s}$ as close as possible to parallel with $\mathbf{v}$, meaning we choose the value of $\mathbf{s}$ that maximizes $\mathbf{v}^T \cdot \mathbf{s}$. It is straightforward to show that this gives $s_i = +1$ if $v_i > 0$ and $s_i = -1$ if $v_i < 0$, where v_i is the ith element of $\mathbf{v}$. (If $v_i = 0$, then $s_i = \pm 1$ are equally good solutions to the maximization problem.)

Thus we arrive at a simple algorithm for splitting a network: we calculate the eigenvector corresponding to the largest positive eigenvalue of the symmetric matrix $\mathbf{B} + \mathbf{B}^T$ and then assign communities based on the signs of the elements of this eigenvector.

As in the undirected case, the spectral method typically provides an excellent guide to the broad outlines of the optimal partition, but may err in the case of individual vertices, a situation that can be remedied by adding a "fine-tuning" stage to the algorithm in which vertices are moved back and forth between communities in an effort to increase the modularity, until no further improvements can

be made [13]. We have incorporated such a fine-tuning in all the calculations presented here.

So far we have discussed the division of a network into two communities. There are a number of ways of generalizing the approach to more than two communities, but the simplest, which we adopt here, is repeated bisection. That is, we first divide the network into two groups, then subdivide those groups and so on, the process stopping when we reach a point at which further division does not increase the total modularity of the network.

The subdivision of a community contained within a larger network requires a slight generalization of the method above. Consider the change in modularity ΔQ of an entire network when a community g within it is subdivided. Defining s_i as before for vertices in g, we find

$$\begin{aligned}\Delta Q &= \frac{1}{2m}\Bigg[\sum_{i,j\in g}(B_{ij}+B_{ji})\frac{s_i s_j+1}{2} - \sum_{i,j\in g}(B_{ij}+B_{ji})\Bigg] \\ &= \frac{1}{4m}\sum_{i,j\in g}\Bigg[(B_{ij}+B_{ji}) - \delta_{ij}\sum_{k\in g}(B_{ik}+B_{ki})\Bigg]s_i s_j \\ &= \frac{1}{4m}\mathbf{s}^T(\mathbf{B}^{(g)}+\mathbf{B}^{(g)T})\mathbf{s}, \end{aligned} \tag{7}$$

where we have made use of $s_i^2 = 1$ for all i and

$$B_{ij}^{(g)} = B_{ij} - \frac{1}{2}\delta_{ij}\sum_{k\in g}(B_{ik}+B_{ki}). \tag{8}$$

In other words, $\mathbf{B}^{(g)}$ is the submatrix of $\mathbf{B}$ for the subgraph g with the average of the appropriate row and column sums subtracted from each diagonal element. Although $\mathbf{B}^{(g)}$, like $\mathbf{B}$, is in general asymmetric, the sum $\mathbf{B}^{(g)}+\mathbf{B}^{(g)T}$ is symmetric; hence Eq. (7) has the same functional form as Eq. (6), and we can apply the same method to maximize ΔQ.

Our complete algorithm for discovering communities or groups in a directed network is thus as follows. We construct the modularity matrix, Eq. (5), for the network and find the most positive eigenvalue of the symmetric matrix $\mathbf{B}+\mathbf{B}^T$ and the corresponding eigenvector. Each vertex is assigned to one of two groups depending on the sign of the appropriate element of the eigenvector, and then we fine-tune the assignments as described above to maximize the modularity. We then further subdivide the communities using the same method, but with the generalized modularity matrix, Eq. (8), fine-tuning after each division. If the algorithm finds no division giving a positive value of ΔQ for a particular community, then we can increase the modularity no further by subdividing this community and we leave it alone. When all communities reach this state the algorithm ends.

We now give a number of examples of the application of the method. For illustrative purposes, we first consider an artificial computer-generated network, designed specifically to test the performance of the algorithm. In this network of 32 vertices, vertex pairs are connected by edges independently and uniformly at random with some probability p. The edges are initially undirected. The network is then divided into two groups of 16 vertices each and edges that fall within groups are assigned directions at random, but edges between groups are biased so that they are more likely to point from group 1 to group 2 than vice versa.

By construction, there is no community structure to be found in this network if we ignore edge directions—the positions of the edges are entirely random—and this is confirmed in Fig. 1(a), which shows the results of the application of the undirected modularity maximization algorithm. If we take the directions into account, however, using the algorithm presented in this Letter, the two communities are detected almost perfectly: just one vertex out of 32 is misclassified—see Fig. 1(b).

Even in networks where there is clear community structure contained in the positions of the edges, it is still possible for the directions to contribute additional useful information. As an example of this type of behavior, consider the network shown in Fig. 2, which has 32 vertices and three communities. For two of the communities, containing 14 vertices each, there is a high probability of connection between pairs of vertices that fall in the same community but a lower probability if the vertices are in different communities. Structure of this kind, in which edge direction does not play a role, can in principle be found by algorithms designed for undirected networks. The third community, however, is different. It has four vertices, each of which has a high probability of connection to every other vertex. The only feature that distinguishes this third community as separate is the direction of its edges—two of the four vertices have high probability of ingoing edges, the other two have high probability of outgoing edges, and there are also a small number of additional edges running from the former to the latter.

Applied to this network, the standard undirected community detection algorithm finds the two large communities with ease, but the remaining community is not found and its vertices are dispersed by the algorithm among the other communities [Fig. 2(a)]. Our directed algorithm, on

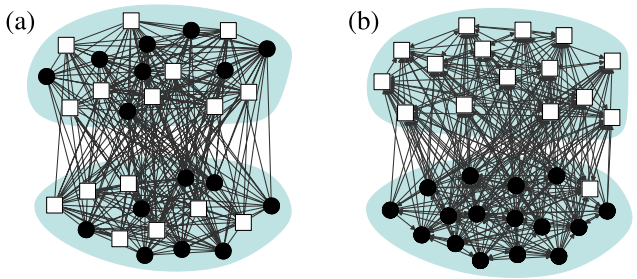

FIG. 1 (color online). Community assignments for the two-community random network described in the text using (a) a standard modularity maximization that ignores edge direction and (b) the algorithm of this Letter. The shaded regions represent the communities discovered by the algorithms; the true community assignments are denoted by vertex shape and color.

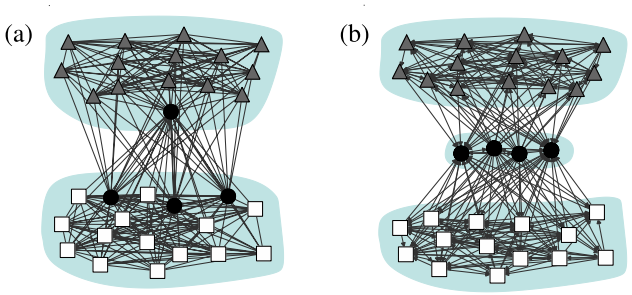

FIG. 2 (color online). Community assignments for the three-community random network described in the text as generated by (a) standard undirected modularity maximization and (b) the algorithm of this Letter.

the other hand, finds all three communities without difficulty [Fig. 2(b)]. Again the algorithm has made use of information contained in the edge directions to identify structures not accessible to other methods.

Turning now to real-world networks, consider Fig. 3, which shows a network representation of a sporting competition. Networks of this kind have received some attention in the recent literature for their clear but nontrivial community structure. The vertices in the network represent the teams in one of the regional competitions or "conferences" of U.S. universities in the game of American football. Edges join pairs of teams that played one another during the 2005 football season. Most previous studies have represented such networks as undirected, but useful information can be extracted from a directed version in which the edges point from the winner to the loser of each game [14].

Figure 3(a) shows the two communities found in this network when edge direction is taken into account. The teams are shaded according to whether they won or lost a majority of their (within-conference) games and, as the figure shows, the two communities correspond precisely to these two groups in this case—the algorithm has divided the more successful and less successful teams into different

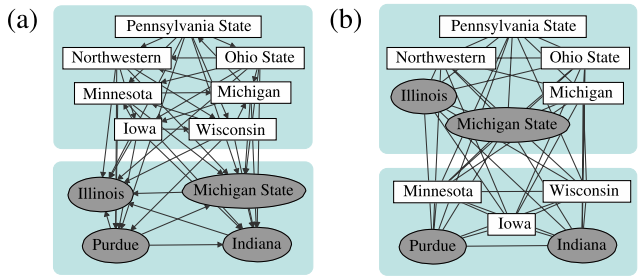

FIG. 3 (color online). Community assignments for the network of American football teams competing in the "Big Ten" conference in 2005 as generated by (a) the algorithm of this Letter and (b) a standard undirected modularity maximization. The shaded regions represent the communities discovered by the algorithms while vertex shapes and colors indicate whether teams won or lost a majority of their games during the season.

communities using the information contained in the edge directions of the network. (Similar results are seen in the networks for other years.) If, however, we ignore the edge directions of the network and apply the undirected modularity algorithm, the method entirely fails to identify the two groups, as shown in Fig. 3(b), indicating that in this case a crucial part of the community information is contained in the edge directions.

In summary, we have presented a method for detecting community structure in directed networks that makes explicit use of information contained in edge directions, information that most other algorithms discard. Our method is an extension of the established modularity maximization method widely used to determine community structure in undirected networks. We have applied the method to a variety of networks, both real and simulated, showing that it is able to recover known community structure and extract additional and revealing information not available to algorithms that ignore edge direction. The computational efficiency of the algorithm is essentially identical to that of the corresponding algorithm for undirected networks, and hence we see no reason to use the undirected algorithm on directed graphs; we recommend the use of the full directed algorithm in all cases where researchers wish to analyze both edge placement and edge direction.

The authors thank Luís Amaral, Sergio Gómez, Roger Guimerà, Marta Sales-Pardo, and Daniel Stouffer for useful conversations. This work was funded in part by the National Science Foundation under Grant No. DMS-0405348 and by the James S. McDonnell Foundation.

[1] R. Albert and A.-L. Barabási, Rev. Mod. Phys. **74**, 47 (2002).
[2] M. E. J. Newman, SIAM Rev. **45**, 167 (2003).
[3] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas, J. Stat. Mech. (2005) P09008.
[4] M. E. J. Newman and E. A. Leicht, Proc. Natl. Acad. Sci. U.S.A. **104**, 9564 (2007).
[5] A. Arenas, J. Duch, A. Fernández, and S. Gómez, New J. Phys. **9**, 176 (2007).
[6] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, Phys. Rev. E **76**, 036102 (2007).
[7] M. Rosvall and C. T. Bergstrom, Proc. Natl. Acad. Sci. U.S.A. **104**, 7327 (2007).
[8] M. E. J. Newman, Phys. Rev. E **69**, 066133 (2004).
[9] M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).
[10] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, Phys. Rev. E **70**, 025101 (2004).
[11] A. Medus, G. Acuña, and C. O. Dorso, Physica (Amsterdam) **358A**, 593 (2005).
[12] J. Reichardt and S. Bornholdt, Phys. Rev. E **74**, 016110 (2006).
[13] M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **103**, 8577 (2006).
[14] J. Park and M. E. J. Newman, J. Stat. Mech. (2005) P10014.