

Spectral clustering with epidemic diffusion

Laura M. Smith,¹ Kristina Lerman,² Cristina Garcia-Cardona,³ Allon G. Percus,³ and Rumi Ghosh⁴

¹California State University, Fullerton, California 92831, USA

²USC Information Sciences Institute, Marina del Rey, California 90292, USA

³Claremont Graduate University, Claremont, California 91711, USA

⁴HP Labs, Palo Alto, California 94303, USA

(Received 21 July 2013; published 21 October 2013)

Spectral clustering is widely used to partition graphs into distinct modules or communities. Existing methods for spectral clustering use the eigenvalues and eigenvectors of the graph Laplacian, an operator that is closely associated with random walks on graphs. We propose a spectral partitioning method that exploits the properties of epidemic diffusion. An epidemic is a dynamic process that, unlike the random walk, simultaneously transitions to all the neighbors of a given node. We show that the replicator, an operator describing epidemic diffusion, is equivalent to the symmetric normalized Laplacian of a reweighted graph with edges reweighted by the eigenvector centralities of their incident nodes. Thus, more weight is given to edges connecting more central nodes. We describe a method that partitions the nodes based on the componentwise ratio of the replicator's second eigenvector to the first and compare its performance to traditional spectral clustering techniques on synthetic graphs with known community structure. We demonstrate that the replicator gives preference to dense, clique-like structures, enabling it to more effectively discover communities that may be obscured by dense intercommunity linking.

DOI: [10.1103/PhysRevE.88.042813](https://doi.org/10.1103/PhysRevE.88.042813)

PACS number(s): 89.75.Hc, 89.75.Fb, 02.10.Ud

I. INTRODUCTION

Graph partitioning is used in many applications, including community detection [1], image segmentation [2], and data mining [3], where it is necessary to partition a graph into modules or clusters of similar, or similarly behaving, nodes. Spectral partitioning uses the eigenvectors associated with the k smallest eigenvalues of the graph Laplacian matrix (or its normalized version) to partition the graph into k clusters [4–7].

Existing methods for spectral partitioning are closely associated with random walks on graphs. A random walk is a stochastic dynamic process where transitions take place from a node to a random neighbor of that node, and it is described by the (normalized) graph Laplacian. The existence of a good partition implies that random walks take a long time to reach a stationary distribution on the graph [2,8], because they spend a long time within a module and seldom pass between modules [9]. This forms a basis for objective functions used to select which edges to cut so as to partition the graph, such as normalized cut and conductance, though these functions have trouble partitioning real-world graphs where many intermodule edges obscure the underlying structure [1].

Epidemic diffusion is another type of dynamic process on a graph. An epidemic undergoes transitions simultaneously to all the neighbors of a given node, rather than a single neighbor, and is often used to model the spread of a virus or an innovation through a social network [10,11]. Recently, Lerman and Ghosh introduced the replicator matrix [12], an analog of the graph Laplacian, to describe epidemic diffusion on graphs. They used the replicator to simulate dynamics of synchronization in a network of oscillators, showing that oscillators coupled via epidemic diffusion synchronize into different structures than oscillators coupled via random walk-like diffusion.

We propose a method for spectral graph partitioning based on epidemic diffusion. First, we show that the replicator is equivalent to the symmetric normalized Laplacian of a reweighted graph, where new edge weights are the product

of old edge weights and the eigenvector centralities of the two end points. The eigenvector centrality [13] of a graph is given by the eigenvector corresponding to the largest eigenvalue of the adjacency matrix. Therefore, edges linking central nodes are given a higher weight by the reweighting scheme.

The equivalence between the replicator and symmetric normalized Laplacian of a reweighted graph allows us to exploit well-known relationships between spectral clustering and graph partitioning. To use the replicator for spectral partitioning, we give a computationally efficient procedure that orders nodes based on the componentwise ratio of the second to first eigenvectors and selects a partition that minimizes a quality function computed on the reweighted graph. This tends to preserve dense structures, since edges linking more central nodes in such dense clusters are less likely to be cut.

We study the performance of the proposed spectral partitioning method using synthetic graphs with known community structure. We demonstrate that spectral clustering based on epidemics leads to a better recovery of ground truth communities than traditional methods based on the graph Laplacian, especially in graphs that are more challenging because of the presence of many edges between clusters. Our work suggests that epidemic diffusion can be a useful probe of graph structure, as it can illuminate properties of graphs that are distinct from those found by methods based on the random walk.

II. SPECTRAL CLUSTERING

An unweighted graph $G = (V, E)$, with vertices (or nodes) V and edges (or links) E , can be represented by a $|V| \times |V|$ adjacency matrix A , with $A_{ij} = 1$ if the edge $(i, j) \in E$, and $A_{ij} = 0$ otherwise. By convention $A_{ii} = 0$. We consider undirected graphs, where $A_{ij} = A_{ji}$. The degree of node i is defined as the number of edges incident on it, $d_i = \sum_j A_{ij}$. Other useful constructs are D , a diagonal degree matrix where $D_{ii} = d_i$, and the identity matrix I .

A. Graph Laplacian and spectral clustering

The graph Laplacian matrix is defined as $L = D - A$. The eigenvalues and eigenvectors of L capture many properties of the graph. In the simplest case, if the graph has k disjoint components, the k smallest eigenvalues of L are zero, and the associated eigenvectors are indicator functions assigning nodes to their respective cluster or community [7]. Even if the k smallest eigenvalues are not all zero, their corresponding eigenvectors can be used to partition nodes into k clusters by projecting these nodes onto a subspace of the first k eigenvectors and using standard clustering techniques such as k means [2,5]. The simplest spectral clustering method, spectral bisection, partitions nodes based on the values of the second eigenvector v of the adjacency matrix, or the graph Laplacian. A splitting value c is used to divide the nodes into different clusters based on whether $v_i < c$ or $v_i \geq c$ [6]. A range of splitting values have been used, including zero, the median value within the vector, the largest gap, and the value producing the best ratio cut, best conductance [14], or another measure.

In practice, normalized versions of the graph Laplacian produce better results in spectral clustering applications [3,5]. Two examples are the symmetric normalized Laplacian $L_s = I - D^{-1/2} A D^{-1/2}$ and the random walk Laplacian $L_{rw} = I - D^{-1} A$, so named because the matrix of transition probabilities for a random walk on a graph is given by $D^{-1} A$.

B. Graph cuts and their quality measures

Intuitively, a cluster is a set of nodes $S \subset V$ that are more tightly connected to each other than to nodes outside of the cluster. We use $\bar{S} = V \setminus S$ to denote the complement of S , which consists of nodes that are not in S . In order to bisect the graph into two disjoint clusters, one typically wants to minimize the number of cut edges between clusters,

$$E(S, \bar{S}) = \sum_{i \in S, j \in \bar{S}} A_{ij},$$

while maximizing cluster size, which may be measured by the number of nodes it contains, $|S|$, or the sum of the degrees of the nodes in the set, $v(S) = \sum_{i \in S} d_i$, also called volume of the set.

Several functions have been proposed for measuring the quality of a graph cut. The best known of these are ratio cut $\mathcal{R}(S)$ and normalized cut $\mathcal{N}(S)$:

$$\mathcal{R}(S) = \left(\frac{1}{|S|} + \frac{1}{|\bar{S}|} \right) E(S, \bar{S}) \quad (1)$$

$$\mathcal{N}(S) = \left(\frac{1}{v(S)} + \frac{1}{v(\bar{S})} \right) E(S, \bar{S}). \quad (2)$$

There is a relationship between graph cuts and spectral clustering. Deciding which edges to cut to optimize any of these quality functions is an NP-complete problem. Spectral clustering solves a relaxation of the problem, where the discrete indicator variables that assign nodes to clusters become continuous. Although in general there are no useful bounds for the approximation produced by this relaxation [7], in practice it often provides a simple and effective clustering method. Solutions to the relaxed optimization problem are

given by the second eigenvector of the graph Laplacian L or the normalized graph Laplacian L_s [6]. Relaxing ratio cut leads to spectral clustering using L , while relaxing normalized cut leads to spectral clustering using L_s [2,7]. Such relaxation methods have also been applied productively to the popular modularity maximization method for community detection [15,16]. By analogy with spectral bisection [6], the leading eigenvector approach assigns nodes to clusters based on the sign of the components of the leading eigenvector of the modularity matrix.

C. Spectral clustering and random walks

There exists a further relationship between spectral clustering, the partition quality function, and properties of random walks. A random walk on a graph is a stochastic process where transitions take place to a randomly chosen neighbor of a given node. Cluster properties of the graph can be expressed in terms of the transition matrix $D^{-1} A$ [17] of a random walk. Spectral clustering finds a partition such that a random walk stays within the same cluster for a long time and seldom jumps between clusters [2,9]. Therefore, the presence of a good partition (low normalized cut value) implies that it will take a random walk a long time to reach its equilibrium distribution.

III. EPIDEMIC DIFFUSION ON GRAPHS

An epidemic is a dynamic process that simultaneously undergoes transitions to every neighbor of the current node. Epidemics are used to model the spread of disease [18] and innovation [11] in social networks. Epidemics differ from random walks in important ways. First, rather than choosing a single neighbor to transition to or “infect” as the random walk does, an epidemic will attempt to “infect” every neighbor of a node. In a random walk, the probability of finding the walker in a given location is a conserved quantity that diffuses through the graph, and the random walk transition matrix is a stochastic matrix. Epidemics, on the other hand, replicate themselves with each successful transmission, without following a conservation law [12].

Lerman and Ghosh [12] introduced the replicator operator $R = \lambda_{\max} I - A$ to describe dynamics of synchronization in a network of nodes coupled via epidemic diffusion. Here λ_{\max} is the largest eigenvalue of A , also known as the epidemic threshold [19]. In this system, a dynamic variable u_i associated with node i can change its value based on the values of its neighbors according to

$$\frac{du}{dt} = -R u, \quad (3)$$

where R replaces the Laplacian used in the analogous heat equation that gives the (diffusive) evolution of a random walk on a graph [20]. By construction, the replicator has a steady state given by θ , the eigenvector of A associated with λ_{\max} : $A\theta = \lambda_{\max}\theta$. θ is also known as the *eigenvector centrality* [13] and was introduced by Bonacich to explain the importance of actors in a social network based on the importance of the actors to which they were connected.

Clusters of nodes with similar values of the dynamic variable u emerge as the system of coupled nodes evolves toward the steady state [12]. This motivates a community

detection method with nodes classified according to the rate of convergence to their steady-state values. For large time t , we approximate the solution to Eq. (3) using the two leading eigenvectors θ and ψ of \mathbf{R} ,

$$u_i(t) \approx c_1 \theta_i + c_2 e^{-\lambda_2 t} \psi_i = c_1 \theta_i \left[1 + \frac{c_2}{c_1} e^{-\lambda_2 t} \frac{\psi_i}{\theta_i} \right],$$

where c_1 and c_2 are constants, and λ_2 is the second smallest eigenvalue of \mathbf{R} associated with eigenvector ψ , guaranteed to be nonzero if the graph is connected. Therefore, convergence depends on ψ_i/θ_i , the componentwise ratio of the second to first eigenvectors. Note that eigenvectors of \mathbf{R} corresponding to \mathbf{R} 's two smallest eigenvalues are the same as the eigenvectors of \mathbf{A} corresponding to \mathbf{A} 's two largest eigenvalues.

A. Replicator as the symmetric normalized Laplacian of a reweighted graph

In a social network, one might expect nodes of high ‘‘importance’’ to attract other nodes, resulting in communities forming around nodes with large eigenvector centrality values θ_i . In this section we propose a modification of our graph, converting the unweighted network into a weighted one, where weights are given by the product of the eigenvector centralities of an edge's end points: $\tilde{A}_{ij} = A_{ij} \theta_i \theta_j$. Moreover, we show that the replicator on the unweighted graph given by \mathbf{A} is in fact exactly equivalent to the symmetric normalized Laplacian of the reweighted graph given by $\tilde{\mathbf{A}}$.

In the reweighted graph, the degree of node i is given by

$$\tilde{d}_i = \sum_j A_{ij} \theta_i \theta_j = \theta_i \sum_j A_{ij} \theta_j = \lambda_{\max} \theta_i^2.$$

For convenience, define Θ as the diagonal matrix whose elements are the components of eigenvector θ , i.e., Θ_{ii} . Then, from \tilde{A}_{ij} and \tilde{d}_i above,

$$\tilde{\mathbf{A}} = \Theta \mathbf{A} \Theta \quad \text{and} \quad \tilde{\mathbf{D}} = \lambda_{\max} \Theta^2. \quad (4)$$

We can now write the symmetric normalized Laplacian of the reweighted graph:

$$\begin{aligned} \tilde{\mathbf{L}}_s &= \mathbf{I} - \tilde{\mathbf{D}}^{-1/2} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-1/2} \\ &= \mathbf{I} - \left(\frac{1}{\sqrt{\lambda_{\max}}} \Theta^{-1} \right) \Theta \mathbf{A} \Theta \left(\frac{1}{\sqrt{\lambda_{\max}}} \Theta^{-1} \right) \\ &= \mathbf{I} - \frac{1}{\lambda_{\max}} \mathbf{A} = \frac{1}{\lambda_{\max}} \mathbf{R}. \end{aligned}$$

Hence, $\mathbf{R} = \lambda_{\max} \tilde{\mathbf{L}}_s$.

The equivalence between epidemics and the diffusive process of random walks is at first surprising. Diffusive processes conserve the total amount of the substance diffusing, whereas no such conservation law holds for epidemics [12]. The intuition for the equivalence of the two processes is the following. A node's eigenvector centrality gives the number of paths connecting it to all other nodes in the graph [21]; hence, the product of eigenvector centralities of a pair of nodes captures how much of the substance is newly created when the epidemic follows the edge linking the pair. By encoding the amount of nonconservation in edge reweighting, this scheme allows the epidemic to be reduced to diffusion.

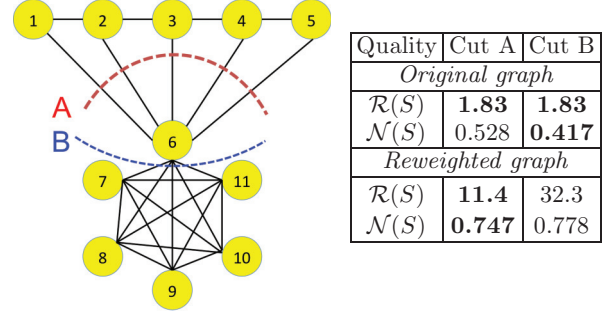


FIG. 1. (Color online) (Left) An example graph. The possible cuts are shown by the dotted curves A and B. (Right) Quality measures of cuts A and B on the original and reweighted graph.

B. Quality measure for the replicator

The equivalence proved above allows us to exploit the properties of the symmetric normalized Laplacian, along with its relationship to graph partitioning, for epidemic diffusion. Since the replicator is simply \mathbf{L}_s of the reweighted graph $\tilde{\mathbf{A}}$, spectral clustering using the replicator corresponds to a relaxation of normalized cut on this reweighted graph. The appropriate measure for assessing graph cut quality with the replicator is therefore normalized cut on the reweighted graph $\tilde{\mathcal{N}}(S)$.

C. An illustrative example

We use a simple example to highlight the differences between traditional graph partitioning and one based on epidemics. Consider the graph in Fig. 1, which shows a dense cluster connected through node 6 to a sparsely linked cluster. Such a configuration is common in social networks, where a high-degree hub linking different communities may obscure community boundaries. We expect a good partition to group node 6 with other nodes in its clique. However, the cut (B) that minimizes normalized cut $[\mathcal{N}(S)]$ groups node 6 with nodes 1–5 and assigns nodes 7–11 to the other cluster. Multiple cuts minimize ratio cut $[\mathcal{R}(S)]$, including one that groups together nodes 3–5.

Node 6 has the highest eigenvector centrality. Furthermore, nodes that belong to the clique have higher centrality values than other nodes. Consequently, in the *reweighted* graph, the edges linking node 6 to the rest of the clique are more ‘‘expensive’’ to cut, and nodes 6–11 are grouped together by the preferred cut (A) that minimizes both the ratio cut $\mathcal{R}(S)$ and the normalized cut $\tilde{\mathcal{N}}(S)$ on the reweighted graph. The quality measures of the cuts are shown in the table in Fig. 1. By giving edges linking central nodes a higher weight, epidemic-based graph partitioning thus preserves dense, clique-like structures. Accordingly, deleting these edges will have the greatest impact on reducing the spread of an epidemic [22].

D. Efficient spectral partitioning

We now describe an efficient method for spectral clustering using epidemic diffusion based on spectral bisection [6]. First, we create a vector \mathbf{v} that is the componentwise ratio of the second eigenvector ψ to the first eigenvector θ of the operator \mathbf{R} and sort its values. Next, we examine all $N - 1$ cuts in this ordering (where $N = |V|$) and pick one

corresponding to the partition that minimizes an appropriate quality measure. The quality measure we use with \mathbf{R} is normalized cut on the reweighted graph $[\tilde{\mathcal{N}}(S)]$. We compare the resulting partition with those produced by applying an analogous splitting procedure to \mathbf{L} , with quality measure $\mathcal{R}(S)$, and \mathbf{L}_s , with quality measure $\mathcal{N}(S)$ (on the original graph).

The proposed optimization procedure is exhaustive, since it tests all $N - 1$ possible cuts within the ordering produced by \mathbf{v} . It may seem that there would be some loss in accuracy from restricting our search to cuts in a one-dimensional projection, rather than searching over the entire subspace spanned by the first two eigenvectors θ and ψ . However, it has been observed [2,23] that the componentwise ratio of the second to first eigenvector of \mathbf{L}_s is precisely equal to the second eigenvector of the random walk Laplacian \mathbf{L}_{rw} , whose first eigenvector is a constant vector. Thus, our algorithm is effective because it is a computationally efficient procedure for finding the best normalized cut in the two-dimensional eigenspace of $\tilde{\mathbf{L}}_{rw}$, i.e., \mathbf{L}_{rw} on the reweighted graph. The advantages of using \mathbf{L}_{rw} in spectral clustering are discussed in Ref. [7].

IV. EVALUATION ON SYNTHETIC GRAPHS

We use synthetic graphs to gain better insight into the differences between operators \mathbf{L} , \mathbf{L}_s , and \mathbf{R} and the characteristics of graphs for which different operators find better solutions. Lancichinetti and Fortunato have proposed an algorithm to generate random graphs with known hierarchical community structure [24]. The N nodes are divided into macro communities, which are themselves composed of micro communities, and then edges between nodes are created using mixing parameters μ_1 and μ_2 . The parameter μ_1 designates the fraction of a node's edges that will connect to nodes in a different macro community, and μ_2 gives the fraction of edges that will connect to nodes in a different micro community within the same macro community. The remaining $(1 - \mu_1 - \mu_2)$ fraction of edges link to other nodes within the same micro and macro communities. These benchmark networks allow us to systematically explore the performance of different spectral clustering approaches.

Using software available on Ref. [25], we generated 100 graphs for each set of parameter values. We took $N = 100$ with two macro communities. We varied μ_1 and μ_2 between 0 and 0.5. The average clustering coefficient ranged between 0.23 and 0.6421, suggesting that the synthetic graphs have properties similar to those often found in real-world networks [26] (Fig. 2).

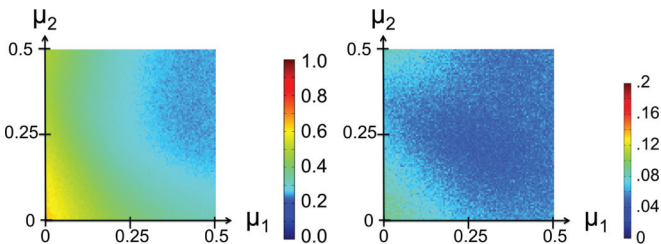


FIG. 2. (Color) Each pixel represents the mean average clustering coefficient (left) and the standard deviation (right) across 100 runs for fixed (μ_1, μ_2) .

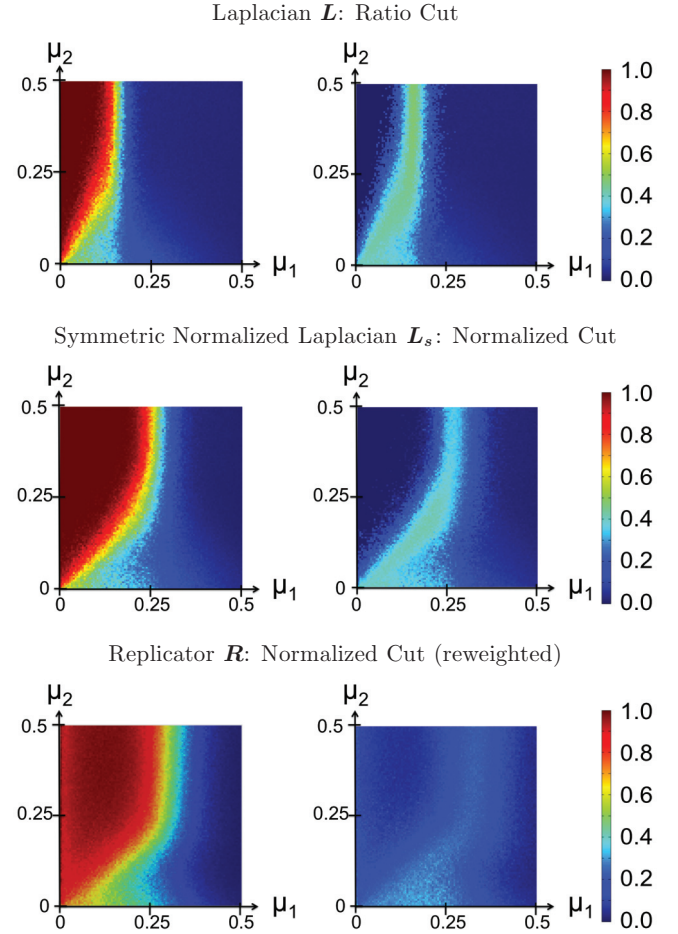


FIG. 3. (Color) NMI scores for minimizing the respective operators' quality measure. Each pixel represents the average (left) or standard deviation (right) NMI score across 100 runs for fixed (μ_1, μ_2) .

We partition each benchmark graph using \mathbf{L} , \mathbf{L}_s , and \mathbf{R} by minimizing their respective quality measures. To evaluate the resulting partitions, we use the normalized mutual information (NMI) measure [27], which compares the partition to the ground truth communities. When the value of this measure is 1.0, the partitioning method has successfully recovered the underlying community structure. We calculate the average and standard deviation of the NMI scores for a fixed set of parameters and display the results in Fig. 3.

As the proportion of a node's edges that connect to individuals in the opposite community, μ_1 , increases, it becomes more difficult to divide the network into the correct communities. We find that \mathbf{L} and \mathbf{L}_s give better results when μ_1 is small (very few links between the two communities). As μ_1 increases, \mathbf{R} dominates with a higher NMI score. Additionally, \mathbf{R} has the lowest standard deviation of the three operators, indicating a consistent performance in identifying the underlying communities.

V. CONCLUSION

Spectral partitioning traditionally uses the graph Laplacian. In this paper, we have introduced a method for spectral partitioning using the replicator, an operator describing

epidemic diffusion on graphs. We have shown that this operator is equivalent to the symmetric normalized Laplacian on a different graph, where edges are reweighted according to the eigenvector centrality measure. By reweighting the edges, a higher weight is placed on globally important nodes. Thus, this method tends to preserve cliques and other dense clusters.

We have introduced a spectral bisection approach based on the componentwise ratio of the second to the first eigenvector of \mathbf{R} , choosing the partition by splitting the sorted vector so as to minimize an appropriate quality measure. Comparing the performance of different methods on synthetic graphs with known community structure, we have shown that spectral partitioning using the replicator is better able to recover the underlying community structure, especially in cases where more edges between the two macro communities make it more difficult for the Laplacian and symmetric normalized Laplacian to identify communities. By reweighting the edges using

eigenvector centrality, the replicator assigns more importance to central nodes. Thus, the edges that pass between clusters are given less influence if they do not link nodes of high centrality. By limiting the cuts to influential edges, the method leads to a more accurate reconstruction of the community structure.

ACKNOWLEDGMENTS

The authors are grateful to Shanghua Teng, Arjuna Flenner, Yves van Gennip, and Blake Hunter for many instructive conversations and suggestions. This work has been funded by the Air Force Office of Scientific Research under Contracts No. FA9550-10-1-0569 and No. FA9550-10-1-0102, DARPA under Contract No. W911NF-12-1-0034, the National Science Foundation under Grant No. 1217605, and the Department of Energy Office of Science Advanced Computing Research (ASCR) program in Applied Mathematics.

-
- [1] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, in *Proceedings of the 17th International Conference on World Wide Web, WWW '08* (ACM, New York, 2008), pp. 695–704.
 - [2] J. Shi and J. Malik, *IEEE Trans. Pattern Analysis Mach. Intell.* **22**, 888 (2000).
 - [3] A. Bertozzi and A. Flenner, *Multiscale Model. Simulat.* **10**, 1090 (2012).
 - [4] F. R. K. Chung, *Spectral Graph Theory*, Vol. 92 of CBMS Regional Conference Series in Mathematics (American Mathematical Society, Providence, RI, 1996).
 - [5] A. Y. Ng, M. I. Jordan, and Y. Weiss, in *Advances in Neural Information Processing Systems* (MIT Press, Cambridge, MA, 2001), Vol. 14, pp. 849–856.
 - [6] D. A. Spielman and S.-H. Teng, *Linear Alg. Appl.* **421**, 284 (2007).
 - [7] U. von Luxburg, *Stat. Comput.* **17**, 395 (2007).
 - [8] M. Jerrum and A. Sinclair, in *Proceedings of the ACM Symposium on Theory of Computing, STOC '88* (ACM, New York, 1988), pp. 235–244.
 - [9] M. Rosvall and C. T. Bergstrom, *Proc. Natl. Acad. Sci. USA* **105**, 1118 (2008).
 - [10] R. M. Anderson and R. May, *Infectious Diseases of Humans: Dynamics and Control* (Oxford University Press, Oxford, 1991).
 - [11] E. M. Rogers, *Diffusion of Innovations*, 5th ed. (Free Press, New York, 2003).
 - [12] K. Lerman and R. Ghosh, *Phys. Rev. E* **86**, 026108 (2012).
 - [13] P. Bonacich and P. Lloyd, *Soc. Net.* **23**, 191 (2001).
 - [14] R. Kannan and S. Vempala, *Spectral Algorithms*, Vol. 4 of Foundations and Trends in Theoretical Computer Science (Now Publishers Inc., Hanover, MA, 2009).
 - [15] M. E. J. Newman, *Proc. Natl. Acad. Sci. USA* **103**, 8577 (2006).
 - [16] S. Fortunato, *Phys. Rep.* **486**, 75 (2010).
 - [17] L. Lovász, *Random Walks on Graphs: A Survey* (Cambridge University Press, Cambridge, 1993), pp. 353–397.
 - [18] H. W. Hethcote, *SIAM Rev.* **42**, 599 (2000).
 - [19] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos, in *Proceedings of the 22nd International Symposium on Reliable Distributed Systems* (IEEE, Piscataway, NJ, 2003), pp. 25–34.
 - [20] F. Chung, *Proc. Natl. Acad. Sci. USA* **104**, 19735 (2007).
 - [21] R. Ghosh and K. Lerman, *Phys. Rev. E* **83**, 066118 (2011).
 - [22] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos, in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12* (ACM, New York, 2012), pp. 245–254.
 - [23] Y. Weiss, in *Proceedings of the Seventh IEEE International Conference on Computer Vision* (IEEE, Piscataway, NJ, 1999), Vol. 2, pp. 975–982.
 - [24] A. Lancichinetti and S. Fortunato, *Phys. Rev. E* **80**, 016118 (2009).
 - [25] S. Fortunato, *Benchmark graphs to test community detection algorithms* (2011), <https://sites.google.com/site/santofortunato/inthepress2>.
 - [26] D. J. Watts, *Am. J. Sociol.* **105**, 493 (1999).
 - [27] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, *J. Stat. Mech.* (2005) P09008.