Community detection in graphs using singular value decomposition

Somwrita Sarkar* and Andy Dong†
Design Lab, University of Sydney, New South Wales 2006, Australia
(Received 28 October 2010; revised manuscript received 17 February 2011; published 21 April 2011)

A spectral algorithm for community detection is presented. The algorithm consists of three stages: (1) matrix factorization of two matrix forms, square signless Laplacian for unipartite graphs and rectangular adjacency matrix for bipartite graphs, using singular value decompostion (SVD); (2) dimensionality reduction using an optimal linear approximation; and (3) clustering vertices using dot products in reduced dimensional space. The algorithm reveals communities in graphs without placing any restriction on the input network type or the output community type. It is applicable on unipartite or bipartite unweighted or weighted networks. It places no requirement on strict community membership and automatically reveals the number of clusters, their respective sizes and overlaps, and hierarchical modular organization. By representing vertices as vectors in real space, expressed as linear combinations of the orthogonal bases described by SVD, *orthogonality* becomes the metric for classifying vertices into communities. Results on several test and real world networks are presented, including cases where a mix of disjointed, overlapping, or hierarchical communities are known to exist in the network.

DOI: 10.1103/PhysRevE.83.046114 PACS number(s): 89.75.Fb, 89.75.Kd

I. INTRODUCTION

A well studied but open problem in complex network research is to identify communities or modules in complex systems [1–4]. Many complex systems show the existence of structural modules that play significant and defined functional roles in the system [1,5,6]. In a graph, modules are identified as densely connected node groups with sparser between group connections. Identifying these modules is fundamentally important to reveal the deep structure of the entire network and functional patterns that may be causal in forming such a structure or result from this structure. In many other complex systems, for example the brain [7], the structure-function relationship is so far not well understood and the way in which modules are defined by community detection algorithms can have significant impact on advancing the understanding of such systems.

There are as yet a number of poorly addressed and open issues associated with the community detection problem [1]. These issues focus around the central observation that, in most community detection algorithms, the definition of a what comprises a module or a community remains method-guided, instead of guided by the underlying structure of the data set. For example, a method that defines a community in terms of strictly partitioned node sets will never discover overlapping structures if they exist naturally in the data. Specifically, a method is needed that has the capacity to reveal all of the following possibilities if and where any occurs in the underlying data: (a) there is no community structure, or the community structure is (b) strictly partitioned, (c) overlapping, or (d) partly both. Further, we would like this method to (1) *simultaneously* provide information on scale and hierarchical modular organization; (2) have representational flexibility to work equally well on weighted or unweighted unipartite or bipartite networks; (3) not be sensitive to specific domains or networks; and (4) not impose any *a priori* assumptions on properties of a community such as prespecifying the number or size of communities, strict community membership, or the number of hierarchical levels.

To address these aims, this paper presents an algortihm that relates the singular value decomposition (SVD) of a graph matrix followed by clustering of a reduced space representation to the community structure detection problem. We present two matrix representations, the signless Laplacian for unipartite graphs and the rectangular adjacency matrix for bipartite graphs. Matrix factorization using SVD allows vertices to be represented as vectors in terms of derived orthogonal bases and singular values; a vertex vector describes the degree of coupling of a vertex with all other vertices as a vector point in space. Then we compute an optimal reduced dimensional approximation of this vector representation in a linear least-squares sense. This causes the principal coupling patterns to be retained as "signal" and discards the rest as "noise." Similar to principal component analysis (PCA), this is a variance maximizing, covariance minimizing step. Finally, we identify communities by applying a clustering algorithm using dot products between vectors in this reduced dimensional space; this allows the simultaneous identification of partitions, overlaps, and hierarchy. A key role is played by the rate of decrease in the magnitude of the singular values. In particular, we show how the rate of decrease acts as a powerful heuristic to identify the optimal number of clusters and to test for the presence of hierarchy in the network.

The major contributions of the algorithm are: (1) it correctly identifies if a community structure does or does not exist in the network; (2) it operates on unipartite and bipartite graphs and unweighted and weighted graphs; (3) it detects overlapping and/or disjointed communities simultaneously including cases where a graph may contain a mix of these because vertices are not restricted to membership of only one community; (5) it simultaneously reveals hierarchical modular organization; (6) it does not impose the number or sizes of communities, overlaps, or the number of hierarchical levels as externally imposed parameters. Very cricually, the algorithm

*ssarkar@mail.usyd.edu.au
†andy.dong@sydney.edu.au

is computationally efficient as fast algorithms for SVD exist even for very large and sparse matrices with routines available in most commercially available software such as MATLAB; see [8,9] for fast SVD algorithms and running time estimates. The algorithm requires minimal code writing time.

We begin with a review of spectral algorithms for community detection to motivate our approach. We then present our algorithm and test it on several benchmark and real world networks.

II. BACKGROUND

The algorithm in this paper uses a modified spectral approach. We review two aspects of the best known spectral approaches, the spectral graph bipartitioning algorithm [10], and Newman's spectral modularity maximization algorithm [2]: (1) the matrix form that is used for spectral decomposition and (2) the index vector or matrix form in which a community is defined in terms of a strict partitioning of vertices into two or more communities. We then present an alternate spectral approach, where a reformulation of these two aspects allows us to propose an algorithm that has the properties outlined in the Introduction.

In general, all spectral graph partitioning approaches use the information contained in the eigenvectors and eigenvalues of a suitable matrix representation of a graph. We let **A** represent the *adjacency matrix* of a graph G with n nodes, where

$$A_{ij} = \begin{cases} 1 & \text{if an edge exists between nodes } i \text{ and } j, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

For an undirected graph, **A** is symmetric. Further, let **D** be the degree matrix, where

$$D_{ij} = \begin{cases} d_i & \text{degree of node } i \text{ when } i = j, \\ 0 & \text{when } i \neq j. \end{cases} \tag{2}$$

Then, the Laplacian matrix is

$$\mathbf{L} = \mathbf{D} - \mathbf{A}, \tag{3}$$

with

$$L_{ij} = \begin{cases} d_i & \text{when } i = j, \\ -1 & \text{when } i \neq j \text{ and } i \text{ is adjacent to } j, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Spectral approaches operate on **L** (or other variants) to partition the graph recursively, each time finding an optimal bisection [11]. The graph is first partitioned into two modules with respect to an optimization function, followed by a recursive reapplication of the bisection step to find more modules. The spectral bipartitioning algorithm [10] aims to minimize the *cut size R*, defined as the number of edges running between two groups of vertices into which a cut partitions a graph. Following [2], the minimization function that is solved is

$$\text{minimize } R = \tfrac{1}{4}\mathbf{s}^T\mathbf{L}\mathbf{s}, \tag{5}$$

where **L** is the Laplacian matrix, and an index vector $\mathbf{s} = \{s_1, s_2, \ldots, s_n\}$, with each $s_i = +1$ or -1 depending on which of the two modules vertex i is assigned to, with the normalization condition $\mathbf{s}^T\mathbf{s} = 1$. Following a parallel formulation, Newman's spectral approach [2] presents a *modularity function Q* that is maximized to find a partition that optimally divides the network into two modules. This function is

$$\text{maximize } Q = \frac{1}{4m}\mathbf{s}^T\mathbf{B}\mathbf{s}, \tag{6}$$

where m is the number of edges in the graph, and **s** is the partition vector as in Eq. (5). **B** is the modularity matrix that measures the difference between the actual number of edges existing between a pair of vertices and an expected number derived from an equivalent random graph with the same number of vertices and the same degree distribution but with no community structure [2].

For details of deriving these two forms [Eqs. (5) and (6)], we refer the reader to [2], and merely note here the equivalence between the forms of the two optimization functions in (5) and (6). Both these approaches represent **s** as a linear combination of the eigenvectors $\mathbf{v}_i$ of the Laplacian matrix **L** in Eq. (5) or the modularity matrix **B** in Eq. (6), as

$$\mathbf{s} = \sum_{i=1}^{n} a_i \mathbf{v}_i. \tag{7}$$

To solve the optimization problems and minimize Eq. (5) or maximize Eq. (6), **s** is chosen proportional to the eigenvector corresponding to the second smallest eigenvalue of **L** or the leading eigenvector of **B**. However, since **s** is, by definition, constrained to take on discrete $+1$ or -1 values, an approximate solution is resorted to in both cases, with $s_i = +1$ if the ith element of the corresponding eigenvector is positive and -1 if negative. This corresponds to the strict partition and bisection assumption: Vertices can belong to only one of two communities. For more than two communities, the vector **s** is replaced by an $n \times k$ matrix **S** with k communities, with $i = 1$ to n and $j = 1$ to k, such that

$$S_{ij} = \begin{cases} 1 & \text{when vertex } i \text{ belongs to community } j, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

The strict partitioning assumption still holds. Thus, *a priori* assumptions built into the definition of a community do not permit identification of overlapping communities, even when the eigenvectors and eigenvalues may implicitly contain this information. We note that forcing an approximate solution may throw away useful information contained in the eigenvectors and eigenvalues that can potentially be used to shed more light onto the community structure existing in the the graph.

Further, the spectral bipartitioning algorithm [10] asks the user to choose the number and relative sizes of communities beforehand and always provides a solution, whether or not a clear community structure exists in the graph. Newman's modified approach [2] addresses most of these limitations by using the modularity matrix instead of the Laplacian. However, his approach is valid for unipartite graphs only and maintains the strict partitioning assumption. Thus, overlapping modules are not revealed. Recent studies have also shown that that Newman's approach has a resolution limit problem [12]: It cannot detect smaller hidden substructures inside larger communities.

Extending Newman's approach, Barber [13] explores modularity in bipartite networks. Other approaches [14,15] focus on module identification in bipartite and directed networks. However, most of these methods maintain the strict partitioning assumption. Other recent approaches focus on finding overlapping communities [16,17], most noticeably the algorithm by Palla *et al.* [5]. A few of these look at hierarchical organization along with overlapping module detection [18,19], but these methods do not apply to bipartite graphs simultaneously. To the best of our knowledge, no single algorithm addresses all the requirements outlined in Sec. I. Further, from a methodological point of view, it seems natural and promising to extend and generalize the eigenvalue decomposition based spectral approach [2,10] by using the SVD. SVD robustly lends itself to any general rectangular or square matrix form and makes no specific demands on representation, making it quite suitable to handle many different kinds of graph representations.

A recent approach by Arenas *et al.* [20] shows the application of the SVD combined with dimensionality reduction for a related objective, but different to module detection. In [20], the aim was to characterize the interrelationships between modules and the contribution roles played by nodes in modules for a given defined partition. A *contribution matrix* is defined as a nodes versus modules map and describes a partition of interest. SVD and dimensionality reduction applied onto this contribution matrix shows the extent to which each node contributes to a module and thus the interrelationships between the modules.

In [20], a partition of the network into modules is prespecified [the partition matrix **S**, similar in form to Eq. (8)] and it is this partition that is studied. In drawing a parallel comparison to the latent semantic approach [9] from statistical natural language processing, the query "how much does a word belong to a cluster of documents?" is analogous to "how much does a node belong to a module?" Inherent in this idea is to measure the participation of a node in a module, once a partition into modules has been predefined.

The work presented in this paper differs significantly from the approach presented in [20]. We apply the truncated SVD onto a node versus node representation, with the main aim of detecting partitions of interest. No *a priori* assumptions on the modular organization are provided to the method. The modular organization is instead detected in a data-driven manner. Thus, in [20], the quality of one such partition will be revealed, but many partitions of interest may exist. The work presented in this paper detects multiple partitions of interest.

In [20], only the first two principal dimensions are studied in the dimensionality reduction step. The authors do not provide a formal reason, except that modular interrelationships are understood through projections on a plane. In this work, a dimensionality reduction step, through the study of decay of singular values, reveals that other projections in all lower-dimensional subspaces greater than 2, can also contain meaningful information about the modular organization of the network, multiple partitions of interest, and hierarchical modular organization.

Underlying these differences in aim and methodology, there are also similarities between [20] and the work presented in this paper, namely, the way in which the mathematical properties of the SVD are used for pattern recognition tasks. SVD and dimensionality reduction allow the extraction of an optimal lower-dimensional description of any data set by preserving the principal patterns of association in the data and discarding the rest as noise. While [20] uses this property to study a possible node-module schema, the work in this paper uses this property to formulate a module detection algorithm. This similarity offers proof that SVD is a robust matrix factorization technique to work with many different types of matrix formulations for similar pattern recognition objectives.

Because SVD promises to be a robust, efficient, and fast algorithm for this family of tasks, it will be interesting to combine the method presented in this paper with that presented in [20]: to first use truncated SVD to identify partitions of interest (as shown in this paper) and then use the same to study the modules and their interrelationships as suggested by different partitions (as shown in [20]).

III. COMMUNITY DETECTION AS A DIMENSION REDUCTION PROBLEM

By definition, a community should have the following properties: (1) two neighbors with all common neighbors are in the same community; (2) two vertices that are not neighbors but share all or many common neighbors are likely to be in the same community; (3) two vertices that are neither neighbors nor share common neighbors are likely to be in different communities; and, (4) a vertex with more than one neighbor but having none or few common neighbors with any of its neighbors is likely to fall in an overlap between communities.

In an $n \times n$ adjacency matrix, each vertex represents one dimension. If there is community structure in the graph, then the number of communities will always be much lower than n dimensions. Considering the above-stated properties, the adjacency matrix will have *redundancy*. If vertices have the same neighbors, then there are linearly dependent rows/columns; many common neighbors implies vertex vectors point in a similar direction in space (mutual dot products are high); no common neighbors implies independent rows and columns and different directions in space.

Thus, *what is the optimal number of dimensions that can best describe this independence-redundance relation by minimizing the redundancy*? In terms of pattern recognition theory, this is the "pattern" or "signal" we wish to detect.

IV. ALGORITHM

A. Data representation

We now consider the adjacency matrix **A** of a graph G with n nodes. For weighted edges, entries $A_{ij} = w_{ij} \geqslant 0$. The adjacency matrix is a purely local measure of a vertex's neighbors. This local measure is, however, insufficient to provide information on community structure as a "global" pattern of the graph. For example, if person A knows B and B knows C but not D, then an adjacency matrix does not incorporate the information that the induced or implicit coupling between A and C is higher than A with D, when there are no existing edges A-C or A-D. Due to the sparsity of most adjacency matrices, computing straight dot products

between the rows columns of **A** will, in general, not reveal the community structure.

We further consider an alternate matrix representation: the weighted or unweighted signless Laplacian matrix

$$|\mathbf{L}| = \mathbf{D} + \mathbf{A}, \tag{9}$$

with

$$L_{ij} = \begin{cases} s_i & \text{when } i = j, \\ 1 & \text{or } w_{ij} \text{ when } i \neq j \text{ and } i \text{ is adjacent to } j, \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where $s_i = \sum_j w_{ij}$. A chief reason for using the weighted or unweighted version of the signless Laplacian is the manner in which the diagonal and off-diagonal entries encode connectivity: It allows us to formulate a strong measure of the *relative strength of association* between nodes. As we will see later, modularity is expressed spatially in a vector space, where each node's position in space is representative of its connectivity with all other nodes. This spatial vector based measure not only depends upon the strength of individual node to node connections, but also on how many other nodes a particular node is connected to. The signless Laplacian allows us to capture this through the diagonal entries, while the standard adjacency matrix does not.

Further, this representation allows for self-loops: If a node is connected to itself, this will reflect as an increase in its corresponding degree by 2 in an undirected network for each loop. A representation that allows self-loops is important in many domains. For example, in the brain domain [7], a typical node in a connection matrix represents a group of neurons, which are considered a single unit. In such a case, each node should be considered connected to itself to account for the internal connections between a group of neurons.

Generalizing further, a bipartite graph with m type 1 (blue) nodes and n type 2 (yellow) nodes, can be represented as a rectangular m by n matrix. If G is bipartite, has vertex set V with $(m+n)$ nodes, then by definition there are two mutually exclusive node sets V_1 with m blue nodes and V_2 with n yellow nodes. All edges connect a blue node to a yellow node, with $V_1 \cap V_2 = \{\}$ and $V_1 \cup V_2 = V$. Then an m by n matrix **B** represents the m nodes in V_1 and the n nodes in V_2, where for $i = 1$ to m and $j = 1$ to n

$$B_{ij} = \begin{cases} 1 & \text{or } w_{ij} \text{ if an edge exists between } i \text{ and } j, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

One could represent **B** in square form similar to the $|\mathbf{L}|$ with the $(m+n) \times (m+n)$ matrix $\mathbf{A} = [\mathbf{0}\,\mathbf{B}; \mathbf{B}^T\,\mathbf{0}]$. The corresponding degree matrix $\mathbf{D} = [\mathbf{D}_1\,\mathbf{0}; \mathbf{0}\,\mathbf{D}_2]$, where $\mathbf{D}_1$ is the degree matrix for nodes in V_1 and $\mathbf{D}_2$ is the degree matrix for nodes in V_2. Then the signless Laplacian for the bipartite case is

$$|\mathbf{L}| = \begin{bmatrix} \mathbf{D}_1 & \mathbf{B} \\ \mathbf{B}^T & \mathbf{D}_2 \end{bmatrix}. \tag{12}$$

This representation causes a repetition of data and is therefore computationally inefficient. The method we present in this paper works directly with the matrix **B** for a bipartite graph. In a rectangular adjacency matrix for a bipartite graph, the diagonal element is a connectivity mapping between two different elements. Therefore, the form captures the relative strength of association between nodes directly even without using the degree or strength information.

As another example of a matrix representation, the authors have previously tried a slightly modified adjacency matrix formulation for performing design decomposition for large, complex engineering design systems such as aircraft engines [21,22]. Engineering systems are spatially organized in terms of material, energy, or information interactions. Thus, there exist design criteria such as "component X should be closest to component Y but must not be close to component Z." In this interpretation, any component is always "closest to itself." So the authors used a modified binary adjacency matrix with all 1's on the diagonals. The truncated SVD approach was able to successfully detect the subsystems in a large engineering system with this modified matrix formulation.

Further, it has been shown that the truncated SVD approach has been successfully applied to another matrix representation [20], the contribution matrix, a map of the nodes versus the modules, where the intention was to map the interrelations between modules and node contributions to individual modules.

In general, the truncated SVD approach, mainly because of the mathematical properties of the SVD and dimensionality reduction, is robust enough to handle multiple matrix formulations. In the Conclusions section, we discuss future possibilities of experimentation with other information-rich matrix forms.

For the rest of the paper, we work with $|\mathbf{L}|$ for unipartite and **B** for bipartite graphs.

B. Singular value decomposition

We present an analysis for the general rectangular matrix representation in **B**. The same analysis remains valid for the square form in $|\mathbf{L}|$, which is a special case of the more general rectangular case.

In **B**, the ith row vector in $\mathbb{R}^n$ shows the neighbors of blue vertex i, $i = 1$ to m. Similarly, the jth column vector in $\mathbb{R}^m$ shows the neighbors of yellow vertex j, $j = 1$ to n. A SVD of the matrix causes a linear transformation that diagonalizes the matrix into an orthogonal matrix times a diagonal matrix times an orthogonal matrix. These new left and right orthogonal bases provide us with a very convenient way to describe each vertex vector in terms of its coupling with all other vertices:

$$\mathbf{B} = \mathbf{U}\mathbf{S}\mathbf{V}^T. \tag{13}$$

U is an $m \times m$ orthonormal basis for $\mathbb{R}^m$; **V** is an $n \times n$ orthonormal basis for $\mathbb{R}^n$. The $m \times n$ diagonal matrix **S** contains singular values that contain scaling information on how a vector is stretched or shrunk when it goes from $\mathbb{R}^n$ to $\mathbb{R}^m$ and are arranged in a decreasing order of magnitude. The number of singular values is equal to the rank r of the matrix **A**. If we disregard the null space, then **U** is an $m \times r$ matrix, **S** is an $r \times r$ matrix, and $\mathbf{V}^T$ is an $r \times n$ matrix. Thus, **U** and **V** represent basis sets of eigenvectors for $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively, where the original correlated vertex coupling information is diagonalized and expressed in terms of uncorrelated independent vectors. The local vertex-vertex coupling information in **B** is decoupled: Vertex vectors can

now be expressed as linear combinations of the derived orthogonal bases and singular values.

In SVDâs basic action on a matrix [23] the row space of $\mathbf{B}$ is r-dimensional and inside $\mathbb{R}^n$, and the column space of $\mathbf{B}$ is r-dimensional and inside $\mathbb{R}^m$. Choosing special orthonormal bases $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r)$ for the row space and $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r)$ for the column space such that $\mathbf{B}\mathbf{v}_i$ is in the direction of $\mathbf{u}_i$, with $\mathbf{s}_i$ providing the scaling factor, we obtain

$$\mathbf{B}\mathbf{v}_i = \mathbf{s}_i\mathbf{u}_i \quad \text{or} \quad \mathbf{BV} = \mathbf{US}. \tag{14}$$

A row of $\mathbf{B}$ shows the neighbors of a blue vertex. The left-hand side of Eq. (14) says that a dot product between a row of $\mathbf{B}$ and the ith eigenvector in $\mathbf{V}$ is a measure of how much a vertex vector points in the same direction as the eigenvector dimension $\mathbf{v}_i$. Obviously, if two vertices share the same neighbors, or many common neighbors, they will all share a similar relationship with each eigenvector. Recall also that eigenvectors form an orthogonal basis; that is, they are perpendicular (uncorrelated) to each other. Thus, *orthogonality* or direction in space becomes a chief way in which a vertex vector can be classified. Under the orthogonality condition, all vectors with common neighbors will roughly point in the same direction in space. The right-hand side of Eq. (14) says that this is equal to a linear combination of a corresponding eigenvector $\mathbf{u}_i$ and the corresponding singular value s_i. The diagonalization of the data causes a $\mathbf{v}_i$ to be expressed in terms of only the ith eigenvector $\mathbf{u}_i$ and the ith singular value s_i. Therefore, SVD provides for a change of basis and allows us to represent the same matrix in the best possible diagonal form. Similarly,

$$\mathbf{B}^T\mathbf{U} = \mathbf{VS}. \tag{15}$$

Thus, the $\mathbf{US}$ or $\mathbf{VS}$ products provide a new abstract way of describing each of the m type 1 vertices and n type 2 vertices, respectively, as a linear combination of the corresponding orthogonal bases and the singular values. An ith type 1 vertex is

$$u_{i1}s_{11} + u_{i2}s_{22} + \cdots + u_{ir}s_{rr}, \quad i = 1 \text{ to } m. \tag{16}$$

A jth type 2 vertex is

$$s_{11}v_{1j} + s_{22}v_{2j} + \cdots + s_{rr}v_{rj}, \quad j = 1 \text{ to } n. \tag{17}$$

The representation in Eqs. (16) and (17) allow a continuous (and not discrete) representation of a vertex as a vector point in space, where its position is representative of its coupling with other vertices, in terms of its "membership" to an eigenvector. If we consider each eigenvector as a representative "community axis," then each individual term in Eqs. (16) and (17) tells us how much each node belongs to a community axis. Highly connected groups of vertices, that is, a community, will point to the same direction in space, owing to shared similar relationships with the eigenvectors. Each node can thus be expressed by representing them as linear combinations of orthogonal bases formed by the eigenvectors and the eigenvalues. A dot product between any two will be a measure of how closely they are coupled in the graph, which is, in turn, a measure of community belongingness. The more "orthogonal" two vectors are, the lower the possibility that they belong to the same community; the more "parallel" they are, the more likely it is that they belong to the same community. Note also that because the singular values are arranged in decreasing order of magnitude, each subsequent term in Eqs. (16) and (17) contributes less to the vertex vector representation compared to the previous one. Thus, each eigenvector dimension does not contribute equally to defining membership; those that correspond to larger singular values have a larger contribution to defining community membership.

This continuous measure allows us to address the overlapping community detection problem: A node with multiple neighbors but no or few common neighbors with any of them shows equally high or positive cosines with nodes in separate communities, or correspondingly with multiple eigenvector dimensions, and is therefore placed in an overlap between communities.

In the case of a unipartite undirected graph, the matrix $|\mathbf{L}|$ will be symmetric, and the $\mathbf{SU}$ and $\mathbf{SV}$ products will collapse into one. In the case of a directed one-mode graph, the matrix is not symmetric. We discuss the possible extension of this method for directed graphs as future work in the Conclusions section.

C. Dimension reduction

We can have, therefore, a *continuous measure of community membership* that considers orthogonality, or more generally direction in space, as the main index by which vertices can be classified, with a secondary index of how important an orthogonal dimension is for classification (by the magnitude of the singular values).

We now come back to the dimension question. Not all dimensions are important or needed for making the classification. By definition, the number of communities will always be lower than the number of vertices. We let each eigenvector dimension correspond to one community or module: the number of modules theoretically varies from 1 (all vertices in one module) and n (each vertex in its own module).

Now, consider the relation between the number of vertices, the number of communities, and the rank r of the matrix $\mathbf{B}$. If two vertices share the same set of neighbors, then there will be dependent rows and columns in $\mathbf{B}$. Therefore, in case of such data redundancy, a lower number of dimensions will be sufficient to capture that these two vertices should belong to the same community. Then the rank r is lower than n and there are at most r modules.

To appeal to physical intuition, refer to Fig. 1. Consider two extreme versions of a bipartite graph G with m type 1 vertices and n type 2 vertices. In the first extreme case, let the number of edges be exactly pairwise; that is, the number of pairs is equal to the lesser of m and n, with one blue vertex connected to exactly one yellow vertex and some remaining unpaired vertices. Each pair is a separate community; the number of communities will be equal to the number of pairs. The matrix representation is diagonal with the same numerical weight/degree along the diagonal and some rows/columns with 0 entries.

SVD will show that all the singular vectors and values are needed to detect the exact number of communities: All

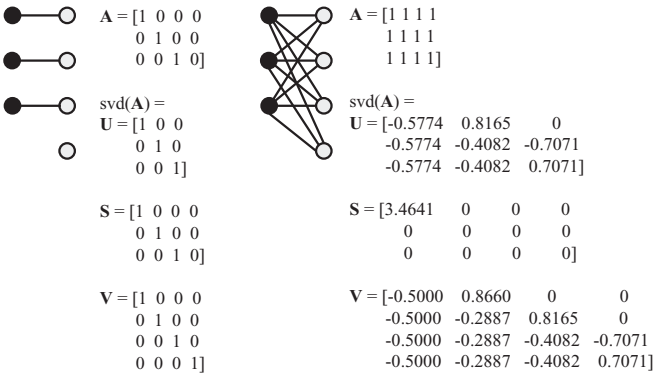


FIG. 1. Two extreme cases for the method.

columns in **B** are independent of each other; no dimensionality reduction is possible without loss of information. Each pair is represented by one dimension that is orthogonal to any other. All the m or n singular values are equally important. Thus, the solution says that there are r distinct communities, where r is the rank of the matrix, with one vertex pair belonging to each community.

As the other extreme case, consider a graph with all vertices connected to all the others. Obviously, the number of communities is exactly equal to 1. Correspondingly, SVD shows that only one singular vector and value is needed to detect the exact number of communities: All the columns in the original matrix are fully dependent (the same). Due to this data redundancy, only one dimension is enough to detect the exact solution. The rank of the matrix is 1. This is the fully "parallel" case, where all the vertices are represented only by one dimension. This case represents the other extreme for making a decision based on orthogonality.

Therefore, *the number of communities will always be lower than or equal to the rank of the matrix*.

Next we show the stronger condition that *the number of communities will always be much lower than the rank r* and depends on the k largest singular values and their rate of decay. The reduced dimensional approximations of **B** play a crucial role in identifying these. As discussed previously, if the graph has a community structure, the intrinsic dimensionality of the data is lower than the m or n features, because there are many vertices that are part of the same community by virtue of commonly shared neighbors. We are looking for an optimal lower dimensional description of the original data set that can reveal these communities by exercising this redundancy in the data set.

A well-known theorem in linear algebra [23] states that an optimal k-rank least squares approximation of the original matrix **B** is given by retaining the first k largest singular vectors and values:

$$\mathbf{B}_{\text{reduced}} = \mathbf{U}_k \mathbf{S}_k \mathbf{V}_k^T. \tag{18}$$

SVD can be viewed in terms of r rank 1 matrices [23]: The optimal rank k approximation to **B** is $\mathbf{u}_1\mathbf{s}_1\mathbf{v}_1^T + \mathbf{u}_2\mathbf{s}_2\mathbf{v}_2^T + \cdots + \mathbf{u}_k\mathbf{s}_k\mathbf{v}_k^T$. Instead of using all the r eigenvectors and singular values to describe the vertices of the original matrix, we now use only the first k. To represent the vertices in reduced dimensional space, we use only the first k terms of Eqs. (16) and (17).

A reduced rank approximation is a "best guess" on how strongly coupled two vertices are: It minimizes the square of the error in reconstructing the original data; that is, it gives the best optimal linear least squares approximation to the original data. In a principal components analysis (PCA) sense, it will treat the k largest singular vectors and values as "pattern" and remove the $r - k$ singular vectors and values as "noise."

D. Relationship to principal component analysis

Consider the matrices $\mathbf{B}\mathbf{B}^T$ and $\mathbf{B}^T\mathbf{B}$. They measure the dot products between the blue vertices and yellow vertices, respectively. Therefore, while the diagonal entries in these matrices are a measure of the variance of the data set. A dot product of a vertex vector with itself becomes a measure of vertex degree, or correlation with itself. The off-diagonal terms are a measure of the covariance, as a dot product of a vertex vector with another vertex vector is a measure of common neighbors, or correlation with each other. The off-diagonal entries therefore represent redundancy in the data set by way of correlation: A large and positive value of covariance indicates that two vertices share many common neighbors and are well correlated. A zero value indicates no shared neighbors.

To discover the community structure, the minimum number of reduced dimensions that optimally describes this redundancy is equivalent to choosing a basis to express the vertex vectors in which the first largest singular vector and value accounts for the dimension of largest variance, the second singular vector and value accounts for the dimension of second largest variance, orthogonal to the first, and so on. Recall that singular values are arranged in decreasing order of magnitude. An implicit assumption here is that the magnitude of each singular value captures the relative importance of each orthogonal dimension for classification and that the variance along a small number of principal components will be an optimal least squares characterization of the "pattern" in the data, that is, the community structure.

We can show that **U** and **V** contain the eigenvectors or principal components of the matrices $\mathbf{B}\mathbf{B}^T$ and $\mathbf{B}^T\mathbf{B}$, respectively, with the squares of the singular values as their common eigenvalues. Taking the SVD of **B** and $\mathbf{B}^T$ shows:

$$\mathbf{B}\mathbf{B}^T = \mathbf{U}\mathbf{S}\mathbf{S}^T\mathbf{U}^T, \quad \mathbf{B}^T\mathbf{B} = \mathbf{V}\mathbf{S}^T\mathbf{S}\mathbf{V}^T. \tag{19}$$

Thus, in a reduced rank approximation dot products between vertices sharing many common neighbors will increase and that between vertices that do not share common neighbors will be decreased, as the membership of each vertex with the $r - k$ least important eigenvector dimensions is discarded as "noise." Vertices that share many common neighbors will be oriented further in the same direction in space and their dot products will increase. This will remain true even if two vertices do not explicitly share an edge. Equivalently, vertices that do not share common neighbors will be oriented such that their dot products go lower or negative. Thus, a lower dimensionality approximation is crucial to reveal the community structure. The examples in the Results section demonstrate that if all

the r dimensions are considered, then this does not reveal the community structure.

E. Identifying the communities

The final step of actually detecting the communities is trivial. The k-reduced vector representations of the **US** and **VS** products are plotted as vector points in the reduced k-dimensional space. The higher the cosine between two vector representations of vertices, the higher is the suggestion that they belong to the same community. We run a K-means clustering algorithm [24] with the distance type set to "cosine" to locate the communities. The K-means algorithm produces disjointed clusters and requires the user to choose the number of clusters as a parameter.

Since we would like the method to automatically reveal the number of clusters and identify overlapping clusters, if any, we have developed an alternate approach. Computing cosine values between all the pairs of vertices generates an $m \times m$ (cosines between type 1 blue vertices), $n \times n$ (cosines between type 2 yellow vertices), or $m \times n$ (cosines between type 1 blue and type 2 yellow vertices) cosine matrix. We then reorder the rows and columns by means of a simple matrix reordering algorithm that orders cosines in decreasing order of magnitude simultaneously across the rows and the columns. This reveals the communities as block matrices with high cosine values, ordered along the main diagonal. Using both the cosine matrix reordering method and the K-means clustering method produces similar results, as expected.

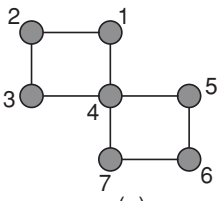
However, the cosine matrix reordering method does not place any restrictions on the disjointedness of the clusters and does not require the user to choose the number of clusters. In cases where communities overlap, the block matrices overlap, too, with the vertices that fall in the overlap belonging to all the respective clusters. We have developed a matrix reordering algorithm (Fig. 2) for visualizing the block matrices that reveal the clusters by ordering all similar cosine values together, but any reordering algorithm that orders similar values to cluster together in a matrix will produce the same results. Alternatively, using a fuzzy or soft K-means algorithm will show similar results too.

Figure 3(a) shows a very simple example: Node 4 is obviously part of both clusters. Figures 3(b), 3(c), and 3(d) show the original matrix, the $k = 2$ reduced matrix, and the cosine matrix at $k = 2$. Note that both A_{13} and A_{15} entries

```
function REORDER-MATRIX (Cosine-Matrix)
returns Reordered Cosine Matrix X

A = Cosine-Matrix
[r, c] = size(Cosine-Matrix)
   loop for i = 1 to c
      index = sortrows-descending(Cosine-Matrix, i)
      reorder rows and columns of Cosine-Matrix based on index
      INDEX = update (index)
      remove row and column i from Cosine-Matrix
   end
A = reorder rows and columns of A based on INDEX
return A
```

FIG. 2. Cosine matrix reordering algorithm.



(a)

A	1	2	3	4	5	6	7
1	2	1	0	1	0	0	0
2	1	2	1	0	0	0	0
3	0	1	2	1	0	0	0
4	1	0	1	4	1	0	1
5	0	0	0	1	2	1	0
6	0	0	0	0	1	2	1
7	0	0	0	1	0	1	2

(b)

(c)

A_{red}, k=2	1	2	3	4	5	6	7
1	0.8536	0.8536	0.8536	1.2071	0.0000	-0.3536	0.0000
2	0.8536	1.0000	0.8536	0.7071	-0.3536	-0.7071	-0.3536
3	0.8536	0.8536	0.8536	1.2071	0.0000	-0.3536	0.0000
4	1.2071	0.7071	1.2071	3.4142	1.2071	0.7071	1.2071
5	0.0000	-0.3536	0.0000	1.2071	0.8536	0.8536	0.8536
6	-0.3536	-0.7071	-0.3536	0.7071	0.8536	1.0000	0.8536
7	0.0000	-0.3536	0.0000	1.2071	0.8536	0.8536	0.8536

(d)

A_{cosine}	1	3	2	4	5	7	6
1	1.0000	1.0000	0.9135	0.7831	0.2265	0.2265	-0.1892
3	1.0000	1.0000	0.9135	0.7831	0.2265	0.2265	-0.1892
2	0.9135	0.9135	1.0000	0.4625	-0.1892	-0.1892	-0.5722
4	0.7831	0.7831	0.4625	1.0000	0.7831	0.7831	0.4625
5	0.2265	0.2265	-0.1892	0.7831	1.0000	1.0000	0.9135
7	0.2265	0.2265	-0.1892	0.7831	1.0000	1.0000	0.9135
6	-0.1892	-0.1892	-0.5722	0.4625	0.9135	0.9135	1.0000

FIG. 3. Dimension reduction demonstration. (a) An example graph, node 4 is part of both modules; (b) the signless Laplacian for the graph in (a); (c) $k = 2$ approximation to (b); (d) cosines between vertex vectors in $k = 2$ reduced space. The gray block matrices show the two modules with node 4 in the overlap.

are 0 in the original matrix, that is, no edge. In the reduced dimension matrix, however, while coupling between nodes 1 and 3, A_{13}, goes up to 0.8536, coupling between nodes 1 and 5, A_{15}, remains at 0. The cosine matrix corroborates this observation. In the original matrix, dot product between vertex vectors A_1 and A_3 is 0.3333, and that between A_1 and A_5 is 0.1667 (both low). In the reduced representation, these become 1.000 (high) and 0.2265 (low), respectively, showing that nodes 1 and 3 are classified as part of the same cluster, while node 5 is not part of this cluster. Note also that node 4 is correctly identified as part of both clusters, that is, as an overlapping node between the two communities.

F. Choosing the optimal number of modules

We now come to an important question: How does one choose the optimal number of modules? Choosing the value of the parameter k corresponds to choosing k clusters. We consider approximations $k = 1$ to r, where an r-dimensional approximation means considering the original matrix **B**. The singular values and their rate of decay plotted as a scree chart provides us with a heuristic to choose this value as the number of leading eigenvector dimensions and singular values that captures most of the "pattern" in the data set. These are preserved to compute a reduced dimensional representation.

The scree plot shows three definite features, which we discuss here and demonstrate in Sec. V. Figure 4(a) shows in a scree plot the decay of singular values for a typical 128-node Newman-Girvan type test network [2]. First, there is either a pronounced or gradual "elbow" in all scree charts, with the magnitude of singular values sharply falling for the first

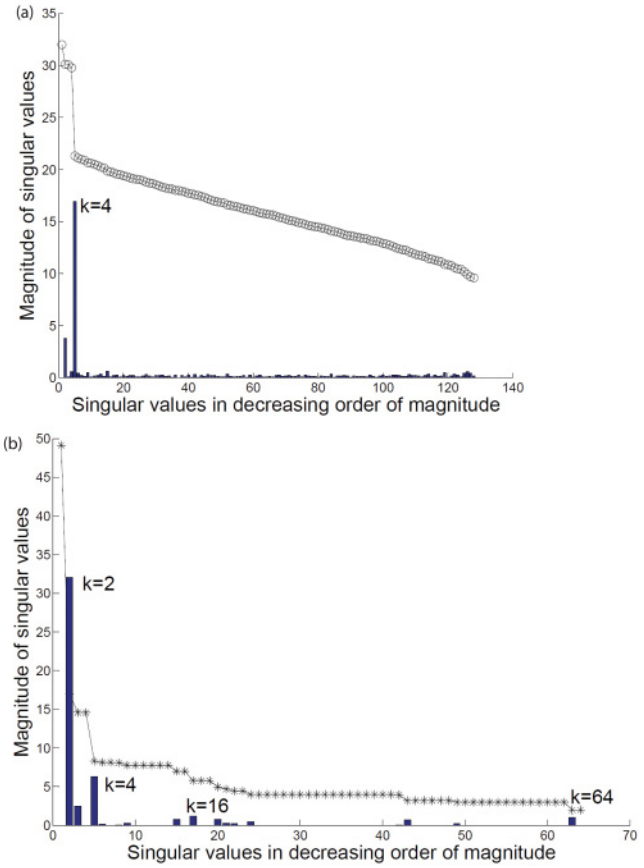


FIG. 4. (Color online) (a) Scree plot for decay of singular values shows the characteristic "elbow" corresponding to the largest difference in magnitude at $k = 4$ in a Newman-Girvan type network. (b) Scree plot for a typical scale-free hierarchical network from [4] shows levels, with highest differences at $k = 2$, 4, 16, and 64.

few values and becoming gradual thereafter; in all cases, the number of modules increases up to this characteristic k value and remains constant thereafter. This characteristic k value gives us the optimal number of modules.

Second, this is easily located by charting the differences between successive singular values. There is a peak difference, as shown in Fig. 4(a) that corresponds to the optimal number of modules.

Third, for networks with no hierarchy but only modular structure, such as a typical test Newman-Girvan type network, there is only one such peak [Fig. 4(a)]. For networks with hierarchy, such as a typical scale-free one from [4], the singular values are clustered in levels, and the scree plot takes on a stepped form with the decay occurring in steps: There are clusters of closely spaced singular values with small decreases then a large decrease, then another cluster, and so on. As a useful first order approximation, we believe that the number of such levels shows the number of hierarchical levels in the network. If the differences between singular values are arranged in descending order, then choosing the largest ones (corresponding to each stepped level) gives us the hierarchical organization at that level. Although this matches well with the idea of stable states in [25], the relationship between this method and [25] should be formally examined.

For example, for a typical 64-node scale-free hierarchical network of [4] we obtain the largest differences at $k = 2, 4, 16,$ and 64, which represents the true hierarchical organization in the network. We show the clustering results at these values in Sec. V. In contrast, no matter how much the k value is increased in a Newman-Girvan type "flat" network, no hierarchical structure is revealed, as increasing the k value to more than the number of modules does not show the larger modules breaking up into smaller ones. The singular values plot shows one large elbow and no stepped decay pattern. Increasing the k value is like viewing a network based on its principal patterns of node association, from a coarse-grained (lowest k) to a fine-grained view (highest k).

A class of multiresolution methods exist [1] that address the problem of hierarchy and scales of modular organization in networks. Most of these methods have a freely tunable parameter, which can be changed by a user to discover hierarchy and scales, if any exist in the network. The k parameter can be considered to be such a parameter, which the user can change in order to reveal multiple feasible partitions of the network.

The authors have found that for very tightly defined networks with a clear community structure, the k value has a one-to-one correspondence with the number of clusters. For example, the benchmark networks of Newman and Girvan [6] (see Sec. V) show a very symmetric structure: 4 modules corresponded to preserving 4 singular values. In a convergence test, if more singular values are retained, say 5 or 6 or $k = r$, the same 4 modules appear in the cosine matrix. In other words, the convergence test shows that the number of clusters increases with the k value to a certain threshold point. After this threshold point, if the number of singular values or k is increased, no increase in the number of clusters is visible. Thus, for example, for a 128-node Newman-Girvan type test network with 4 modules of 32 nodes each, retaining any number of k values from 4 to 128 will still show only 4 clusters.

This one-to-one relationship does not always hold when we come to messy real world networks with modules having different numbers of nodes, heterogenous node degrees, etc. For the real world cases, the convergence test, as described above, is a valuable tool: The k value gives a good hint of the number of modules; the number of modules roughly corresponds to the k value to a certain threshold point, after which the number of modules stabilizes. For example, as we see in the Sec. V B, in the Zachary example [26], at $k = 2$, the two main modules emerge that correspond to the actual split in the group. At $k = 4$, 4 submodules are seen. Beyond $k = 4$, however, the number of modules stabilizes, and we continue to see the 4 modules: The number of modules does not increase as the k value increases. However, the elbow in the scree plot appears at $k = 6$, and the results show that the 4 submodules are not clearly disjointed but have multiple overlaps.

V. RESULTS

We have applied the method to several benchmark and test networks and present the results in this section. To show the communities detected, we use the network itself, as well

as the cosine matrix form showing the block matrices that correspond to clusters as in Fig. 3. Throughout, we use MATLAB-generated grayscale images of the reordered cosine matrix to show the modules detected. Block matrices along the main diagonal show the modules. Off-diagonal colored patches show couplings that exist between the main modules.

We worked on a Dell Precision M6400 with 8GB RAM and programed the algorithm in MATLAB 2010. One thousand-node networks ran in 15–20 s. The smaller networks ran in negligible time.

A. Benchmark test networks

We generated regular, random, and networks with known community structure following [2], [27], and [4], respectively, to test the performance of the algorithm. Regular and random networks show no community structure. Figure 5(a) shows a regular network with 64 nodes, a regular ring lattice in which each node has a degree of 18. The application of the method shows that no modules or clusters are identified. Instead, a thick middle spine and two small triangular parts echo back the original ring lattice structure of the network. Figure 5(b) shows the application of the method on a 64-node random network with an average degree sequence of 18 at the same k value. The results show no modular structure along the block diagonal, but a random pattern.

Figure 5(c) shows the application of the method onto a 128-node Newman-Girvan type network with 4 communities of 32 vertices each. The mixing parameter $m = 0.9$; that is 90% of the edges fall between communities, with 10% edges falling between communities. The results show a perfect block diagonal structure with very high cosines (0.9–1.0) identifying the 4 clusters along the main diagonal, with sharp distinctions between cosines. That is, there are no shades of gray in the plot. This implies that within the modules identified, all the nodes share very high cosines with each other. Conversely, these nodes share very low cosines with the nodes of other communities. We increased the mixing parameter, gradually decreasing number of intracommunity edges and increasing the number of intercommunity edges. The method proved robust to this increasing noise: Figure 5(d) shows the results for a similar 128-node network but with $m = 0.6$. The method is able to identify the clusters, but shades of gray have started to appear. As m is increased further, distinctions between modules become more and more noisy from $m = 0.5$ onward, until no more modules are detected when roughly each vertex connects to as many vertices inside as outside its own community.

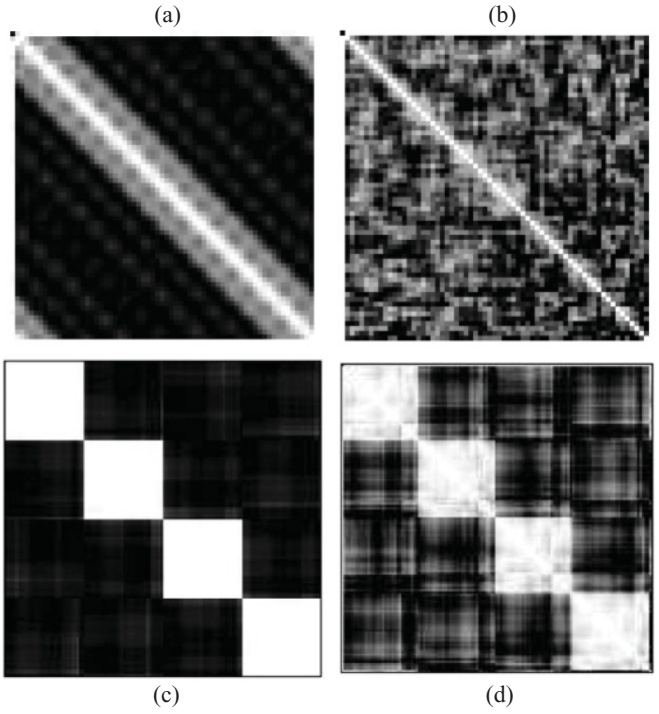


FIG. 5. Test network results. (a) Regular network with 64 nodes; (b) random network with 64 nodes and similar average degree; (c),(d) Newman-Girvan type network with 128 nodes. Mixing ratio: 0.9 and 0.6.

Further, we used the software provided by [27] to generate Newman-Girvan like networks, but with overlapping nodes. Figure 6(a) shows one example of a 128-node network with 5 communities, different community sizes, with 32 overlapping nodes being shared by at least 2 communities. The results match exactly with the known structure, and the algorithm was able to identify the overlapping nodes correctly. Note that the large clusters along the diagonal in the figure show the 5 main modules, the smaller clusters along the diagonal show the overlapping modules. The off-diagonal light gray areas show to which main modules each overlapping set of nodes belongs with.

Further, we have experimented with 1000-node Lancichinetti-Fortunato type networks with heterogenous node degrees and both nonoverlapping and overlapping nodes. In all cases the method proved robust and performed well in identifying the known clusters. Figure 6(b) shows an example of a 500-node network where all the 7 modules were exactly identified by the algorithm.

We also tested scale-free hierarchical test networks using the Ravasz *et al.* model [4]. These hierarchical scale-free networks have a known community structure with 4 densely connected nodes forming the basic module, with 1 internal and 3 external nodes. Four of these come together to form a 16-node two-level network, again one internal and three surrounding 4-node clusters. Sixteen of these then come together to form a three-level 64-node network, with one internal and three surrounding 16-node clusters. At each level the external nodes of the 4-node module are attached to the central node of the oldest 4-node module. Figure 7(a) shows an original 64-node network. Refer to the discussion in Sec. IV F that shows peak differences in singular values at $k = 2$, 4, and 16. Figures 7(b), 7(c), 7(d), and 7(e) show the results at these k values: 2, 4, and 16. At $k = 2$, we can see two clusters: The first central 16-node cluster and the surrounding three clusters. At $k = 4$, we can see the four 16-node clusters with the gray areas showing the first one's relationship to the other three. At $k = 16$, we can see all the 16 four-node clusters with the gray areas showing the secondary level relationships. Thus, we see that increasing the k value captures the most major patterns of association first, with the k values increasingly capturing more and more detail in the structure. Ravasz *et al.* mention

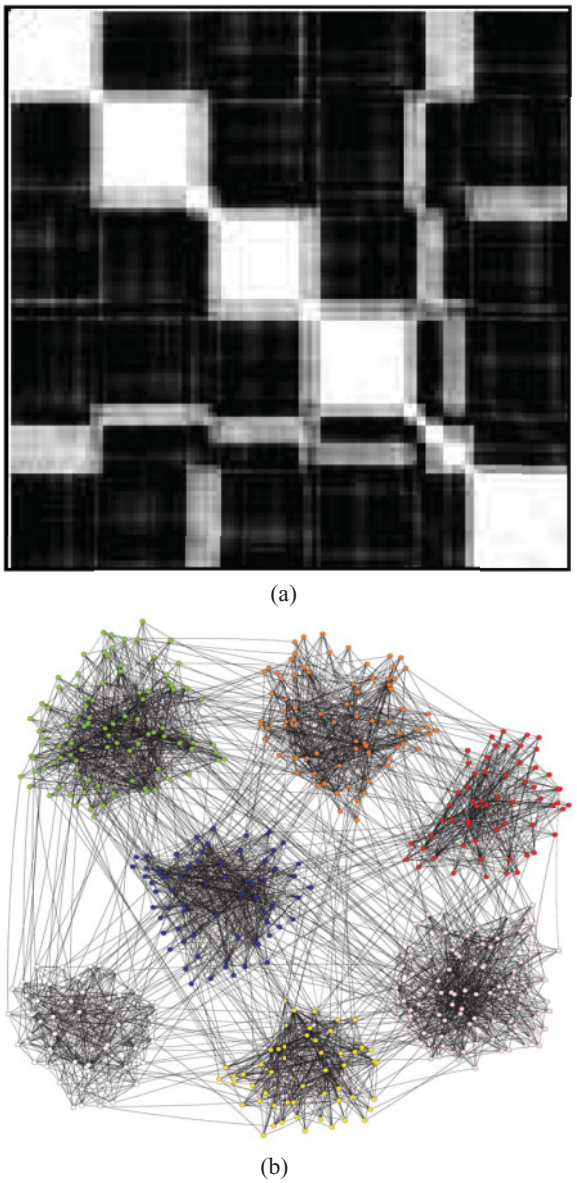


FIG. 6. (Color online) Test network results. (a) Newman-Girvan type network with 128 nodes, generated using software by [27]; the big white clusters are the 5 main modules, the smaller white clusters lie in between main clusters and show the overlapping nodes, the light gray areas show which two main modules each overlap cluster belongs with (b) Lancichinetti-Fortunato type test network with 500 nodes, heterogenous node degree sequences; results match perfectly with the known community clusters in all cases.

that most clustering algorithms fail to detect this hierarchical modular arrangement, while the application of our method easily reveals the entire structure.

B. Unipartite Zachary network

We also applied the method onto several real world networks that are frequently used to test the performance of community detection algorithms.

Our first example is the well-known study of the Zachary Karate Club network [26] where a group of people in a karate club split into two following a rift between the two main leaders of the group. Figure 8(a) shows the results at $k = 2$ using a cosine threshold of 0.7, matching exactly with the original rift in the group. Figure 8(b) shows a scree plot: a plot of the decay of the singular values. These appear grouped: the first two, followed by a large "eigen"-gap, then the next three and a large gap, followed by the tail, and hints at a hierarchical structure with two levels. As discussed in Sec. IV F, interesting modular structure can be observed when these gaps are considered. Figure 8(c) shows the original matrix. Figure 8(d) shows the cosine matrix at $k = 2$, showing the split of the group into two modules. The results, when visualized in the cosine matrix show not only the two major clusters, but also provide an idea on which vertices fall in the "overlap" region, that is, members who share roughly high coupling strength with both groups. Vertices 3, 9, 14, and 20, for example, fall in this region. From $k = 4$, the two modules divide into 4 submodules. Figure 8(e) shows the cosine matrix at $k = 6$, where roughly 4 main submodules can be seen within the two larger groups, with overlap regions. However, one could also interpret a higher number of smaller 6 modules. This shows that the division of the main modules into submodules is not clear and many nodes share overlaps with one or more submodules. The largest eigengaps appeared at $k = 2$, 6, 13, and 20. The 4 boxes in Fig. 8(e) show the results from the K-means clustering algorithm at $k = 20$ superimposed onto the $k = 6$ matrix that corresponds to the solution of a partition into 4 modules found by many researchers and reported in [1]. Beyond $k = 6$, roughly 4 modules continue to appear. Figure 8(e) shows the cosine

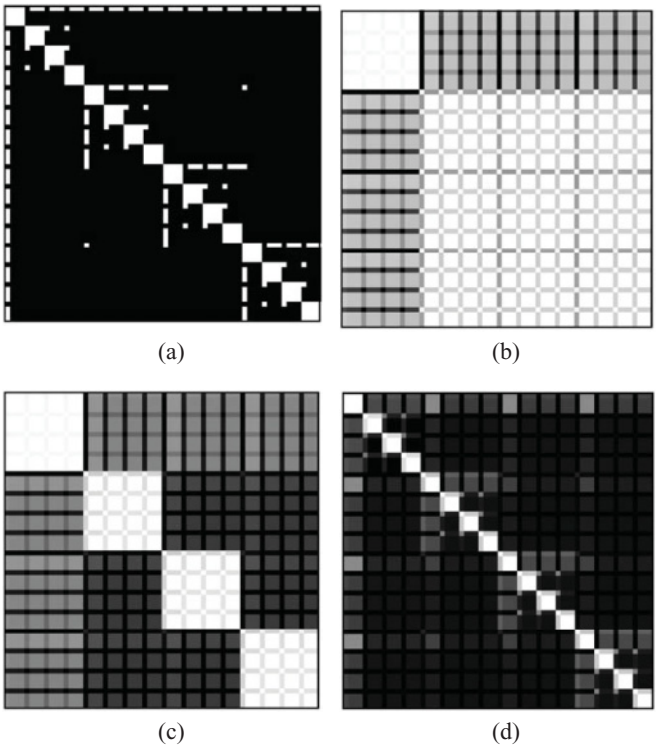


FIG. 7. Scale-free Hierarchical network results. (a) Original 64-node network, and results at (b) $k = 2$, (c) $k = 4$, and (d) $k = 16$. The white blocks show the modules detected.

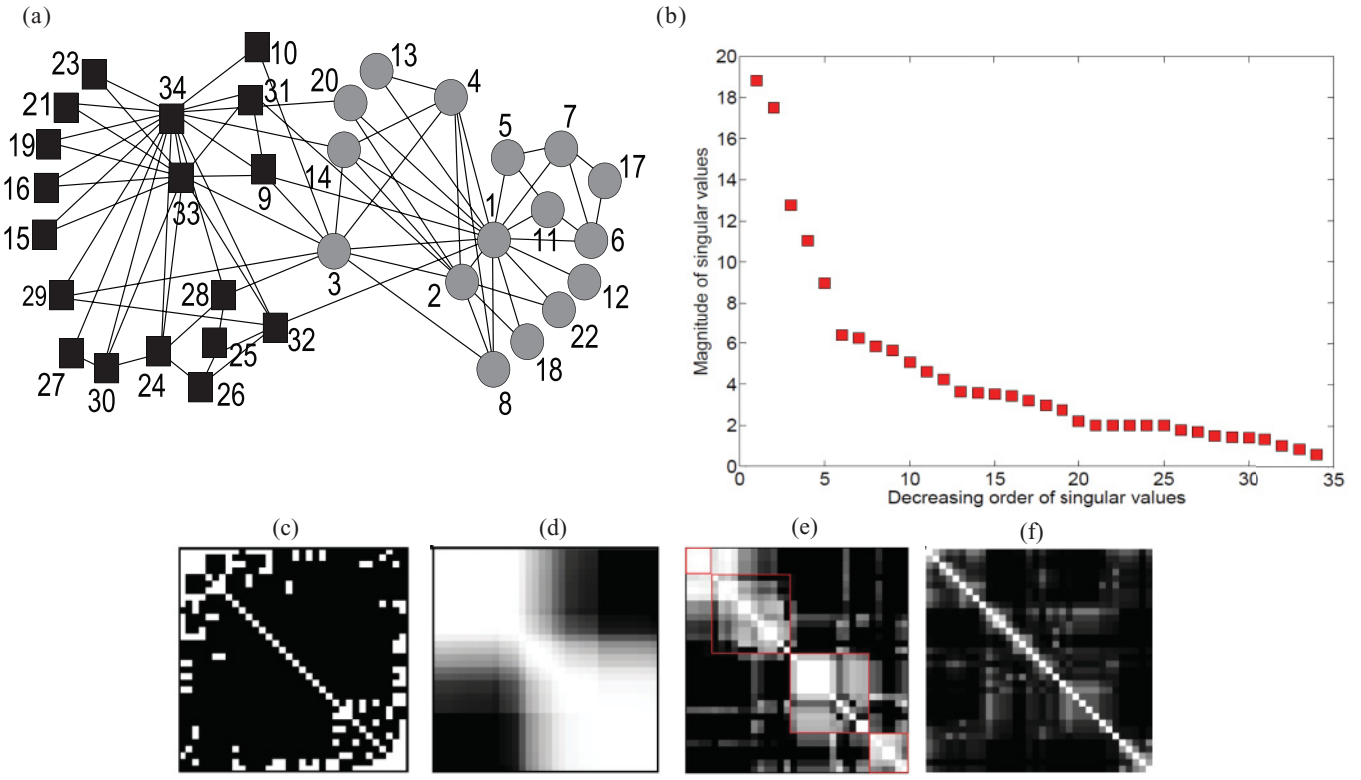


FIG. 8. (Color online) (a) The Zachary club network with the solution given by the algorithm: Gray and black nodes show the two groups into which the club broke at cosine threshold 0.7, $k = 2$. (b) Singular values decay plot. (c) Original matrix. Cosine matrices at (d) $k = 2$, (e) $k = 6$, and (f) $k = 34$.

matrix at $k = 34$, that is, the original matrix. It is obvious that considering the full rank matrix does not show any solutions. This corresponds to 34 modules, with each node in its own cluster.

The results echo "natural" patterns of organization in the data set, at least as much as they can be captured with a linearity assumption: whether there is no inherent modular structure [Figs. 5(a) and 5(b)], whether there is a clear modular structure with disjoint modules [Figs. 5(c), 5(d), and 6(b)], whether there is an overlapping structure with no heirarchy [Fig. 6(a)], whether there is a hierarchical structure but with no overlaps (Fig. 7), or whether there is an overlapping structure with heirarchy [Figs. 8(c) and 8(d)]. The overlap is revealed in terms of overlaps in the block matrices. The hierarchy is seen in the gradual breaking up of block matrices as k values are increased from 2 onward.

C. Unipartite dolphin social network

The dolphin social network [28,29] is another example well cited in the community detection literature. A group of dolphins was observed over a period of time after which the group split into two following the disappearance of a few members that were on the boundary of the group. The application of the method onto the dolphin social network shows an interesting hierarchical modularity structure that corresponds to known community partitions, but also other subcommunity partitions and overlaps in the community structure at a second hierarchical level. As seen in Fig. 9(a), the group is divided into two main groups (detected at $k = 2$, shown as squares and circles), and the larger one is further divided into three or four more overlapping clusters. Figure 9(b) shows the singular values scree plot. Figures 9(c) and 9(d) show the original matrix, and a reordered matrix using the results at $k = 5$. The results at $k = 5$ have been used to reveal the original matrix in block diagonal form. Figures 9(e) and 9(f) show the cosine matrices with solutions at $k = 3$ and $k = 5$. At $k = 3$, already visible are three subcommunities in cluster 1 and a separate cluster 2. At $k = 5$, the structure of the three (or four) subcommunities within cluster 1 have become clearer, while cluster 2 continues to appear completely separate from these three. The three subcommunities share significant overlaps as shown by the gray off-diagonal patches, while the main clusters 1 and 2 do not share overlaps, as no gray patches can be observed between the two clearly disjoint main clusters. Lusseau and Newman [29] report similar results, but their algorithm forces vertices to belong to only one group, thereby losing the interaction information held by individuals that lie on the boundaries of groups. Our results match with their results with a few differences. The decision on 3 subcommunities lies roughly according to the three main clusters shown in the $k = 5$ panel and could easily be 4 in a finer view. There are some individuals who could belong to 2 subcommunities, as they share an interaction equally with both groups. For example, while Lusseau and Newman place Kringel, Thumper, Whitetip, SN63, Hook, and TR99

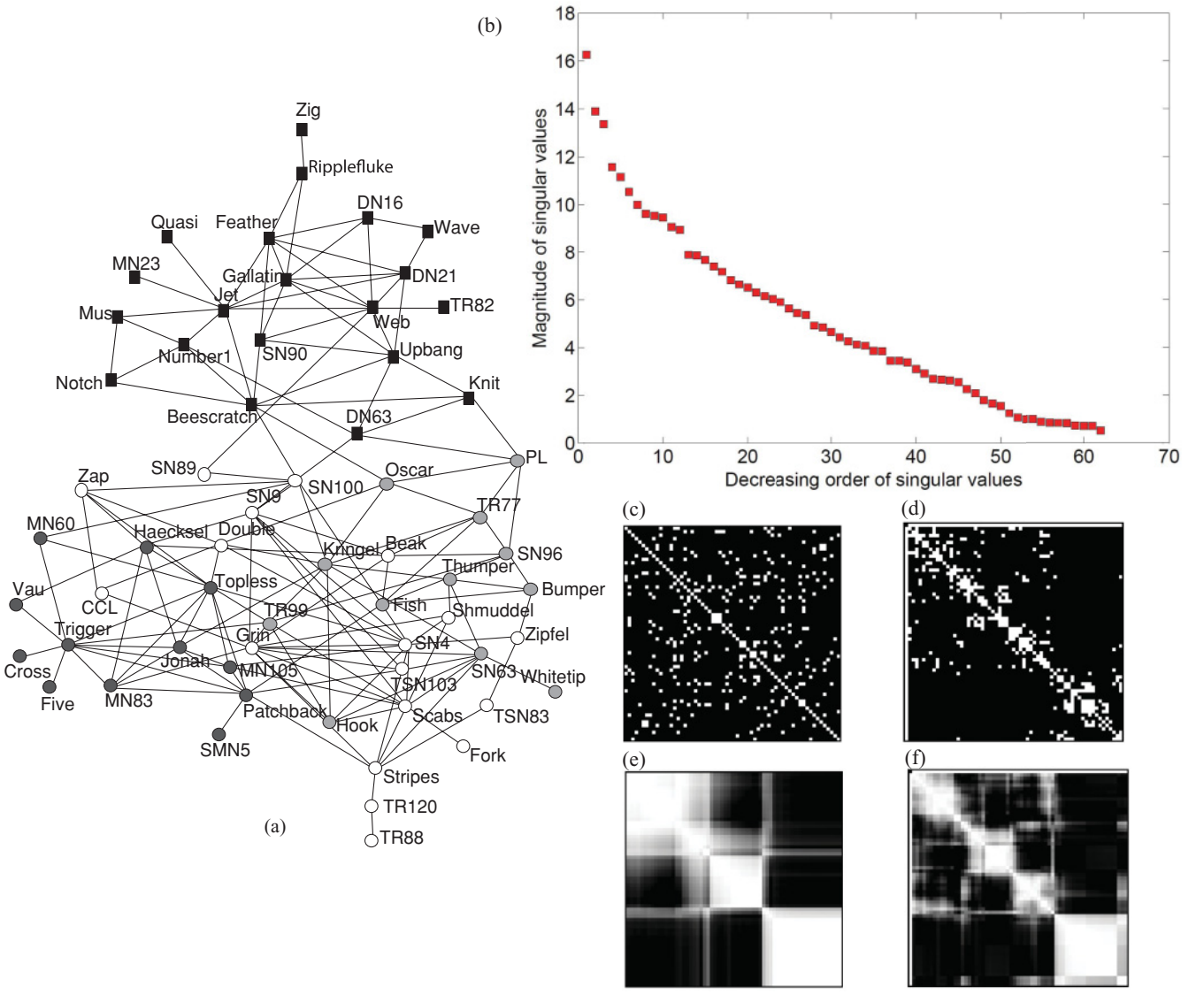
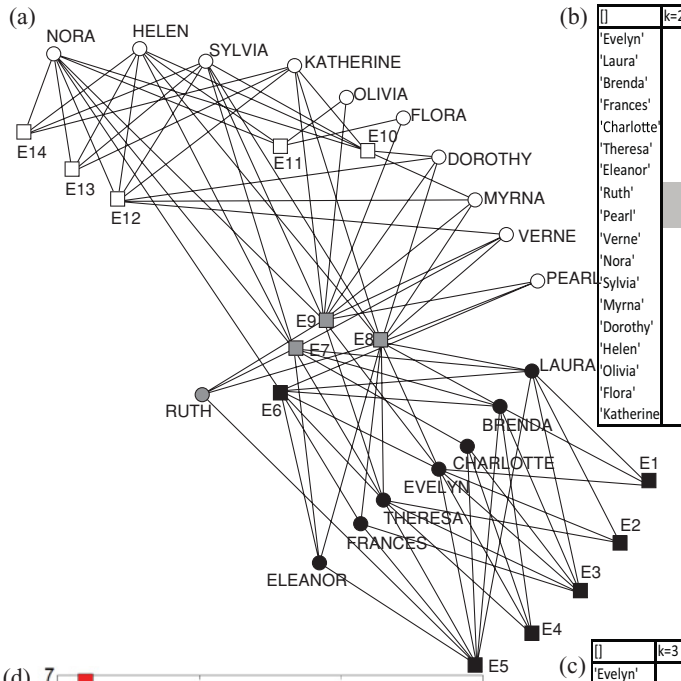


FIG. 9. (Color online) (a) The dolphin social network results. Squares and circles show the two main modules, the dark gray, light gray, and white circles show three submodules in the main module. (b) Singular values decay plot. (c) Original matrix. (d) Reordered matrix using results at $k = 5$ reveals block diagonal form. Cosine matrix at (e) $k = 3$ and (f) $k = 5$.

as part of one subcommunity (along with the first one in our analysis), they could easily form a part of both the first or the third subcommunity. The analysis provides insight into which individuals fall on boundaries of groups versus which ones were more centrally placed in the community. This example also shows that our algorithm is able to reveal communities and subcommunities in real world networks that may contain a mix of clearly disjointed (as in cluster 2) and overlapping communities (as in cluster 1).

D. Bipartite Southern Women Club Network

The Southern Women Club Network [30] is an example of a bipartite social network that maps the participation of 18 women in 14 social events. An edge exists between a woman and an event if she attended the event. The network was created in order to study class and race issues. The application of the method onto the Southern Women network shows a clear community structure. The network has been described as "... a touchstone for comparing analytic methods in social network analysis" [13], but, for our method, turned out to be quite simple to analyze. Figure 10 shows the results in graph form as well as matrix form. We consider the results at $k = 2$ and $k = 3$ since these two are shown to be the most important ones from the scree plot. The rows and columns represent the 18 women and 14 events, respectively. At $k = 2$ [Fig. 10(b)] we can clearly see two communities with a very clear overlap between these two communities. Events 7, 8, and 9 fall in the overlap as they are attended in equal measure by women from both communities. Ruth falls in the overlap, and the original study shows that she is the only woman that is a member of both groups. At $k = 2$, one other woman, Pearl, also falls in the overlap. At $k = 3$ [Fig. 10(c)], Ruth becomes the only woman that is shown as a member of both groups.



(b)

[]	k=2	1	3	4	2	5	6	7	8	9	12	10	11	13	14
'Evelyn'	1	0.973	0.975	0.978	0.986	0.988	0.999	0.883	0.849	0.566	0.173	0.141	0.136	0.131	0.131
'Laura'	2	0.985	0.986	0.988	0.994	0.995	0.996	0.855	0.818	0.519	0.118	0.085	0.081	0.075	0.075
'Brenda'	4	0.986	0.988	0.990	0.995	0.996	0.995	0.849	0.812	0.510	0.108	0.075	0.070	0.065	0.065
'Frances'	6	0.988	0.989	0.991	0.996	0.997	0.994	0.845	0.807	0.503	0.099	0.066	0.062	0.056	0.056
'Charlotte'	5	0.999	0.999	1.000	1.000	0.999	0.974	0.778	0.733	0.400	-0.016	-0.049	-0.054	-0.059	-0.059
'Theresa'	3	0.946	0.949	0.953	0.965	0.968	0.998	0.924	0.896	0.644	0.269	0.237	0.233	0.227	0.227
'Eleanor'	7	0.936	0.939	0.944	0.956	0.960	0.996	0.935	0.909	0.666	0.297	0.265	0.261	0.256	0.256
'Ruth'	9	0.779	0.784	0.793	0.817	0.824	0.918	0.999	0.994	0.867	0.582	0.554	0.551	0.546	0.546
'Pearl'	8	0.713	0.719	0.729	0.756	0.764	0.874	0.998	1.000	0.912	0.659	0.634	0.630	0.626	0.626
'Verne'	10	0.387	0.394	0.408	0.444	0.455	0.619	0.899	0.927	1.000	0.898	0.883	0.881	0.879	0.879
'Nora'	14	0.135	0.143	0.158	0.197	0.209	0.394	0.756	0.798	0.973	0.981	0.975	0.974	0.972	0.972
'Sylvia'	13	0.132	0.139	0.154	0.193	0.205	0.391	0.753	0.796	0.972	0.982	0.975	0.974	0.973	0.973
'Myrna'	11	0.131	0.139	0.153	0.193	0.205	0.390	0.753	0.796	0.972	0.982	0.975	0.975	0.973	0.973
'Dorothy'	16	0.131	0.139	0.153	0.193	0.205	0.390	0.753	0.796	0.972	0.982	0.975	0.975	0.973	0.973
'Helen'	15	0.082	0.089	0.104	0.144	0.156	0.344	0.719	0.764	0.959	0.990	0.985	0.985	0.983	0.983
'Olivia'	17	0.063	0.071	0.086	0.125	0.138	0.327	0.706	0.752	0.953	0.993	0.988	0.988	0.987	0.987
'Flora'	18	0.063	0.071	0.086	0.125	0.138	0.327	0.706	0.752	0.953	0.993	0.988	0.988	0.987	0.987
'Katherine'	12	0.019	0.027	0.042	0.081	0.094	0.285	0.674	0.723	0.939	0.997	0.994	0.994	0.993	0.993

(c)

[]	k=3	1	4	3	5	6	2	8	7	9	12	10	11	13	14
'Evelyn'	1	0.921	0.921	0.904	0.918	0.956	0.982	0.851	0.629	0.612	0.160	0.110	0.040	-0.012	-0.012
'Theresa'	3	0.937	0.942	0.931	0.951	0.993	0.961	0.895	0.778	0.603	0.265	0.226	0.192	0.161	0.161
'Frances'	6	0.985	0.987	0.981	0.989	0.993	0.983	0.800	0.739	0.452	0.098	0.063	0.049	0.035	0.035
'Laura'	2	0.979	0.984	0.984	0.993	0.985	0.934	0.768	0.824	0.367	0.120	0.097	0.122	0.134	0.134
'Brenda'	4	0.974	0.979	0.982	0.990	0.975	0.918	0.747	0.834	0.332	0.110	0.090	0.126	0.145	0.145
'Eleanor'	7	0.922	0.932	0.933	0.952	0.973	0.873	0.834	0.914	0.452	0.295	0.277	0.309	0.321	0.321
'Charlotte'	5	0.912	0.919	0.933	0.931	0.875	0.793	0.567	0.822	0.087	-0.004	-0.005	0.087	0.148	0.148
'Ruth'	9	0.758	0.768	0.750	0.790	0.902	0.823	0.994	0.797	0.828	0.568	0.527	0.463	0.403	0.403
'Pearl'	8	0.547	0.550	0.517	0.556	0.697	0.709	0.895	0.447	0.932	0.515	0.454	0.304	0.189	0.189
'Verne'	10	0.368	0.385	0.362	0.423	0.602	0.467	0.929	0.686	0.954	0.872	0.840	0.756	0.675	0.675
'Myrna'	11	0.101	0.114	0.085	0.146	0.350	0.253	0.784	0.417	0.974	0.879	0.842	0.714	0.604	0.604
'Dorothy'	16	0.101	0.114	0.085	0.146	0.350	0.253	0.784	0.417	0.974	0.879	0.842	0.714	0.604	0.604
'Olivia'	17	0.022	0.030	-0.005	0.049	0.249	0.216	0.685	0.204	0.956	0.755	0.707	0.541	0.409	0.409
'Flora'	18	0.022	0.030	-0.005	0.049	0.249	0.216	0.685	0.204	0.956	0.755	0.707	0.541	0.409	0.409
'Katherine'	12	0.016	0.038	0.020	0.087	0.284	0.092	0.721	0.572	0.844	0.993	0.982	0.927	0.862	0.862
'Sylvia'	13	0.136	0.160	0.151	0.215	0.388	0.161	0.751	0.729	0.763	0.975	0.974	0.963	0.927	0.927
'Nora'	14	0.142	0.167	0.164	0.226	0.379	0.129	0.699	0.777	0.658	0.945	0.954	0.976	0.964	0.964
'Helen'	15	0.092	0.118	0.122	0.180	0.311	0.041	0.594	0.776	0.518	0.895	0.917	0.974	0.987	0.987

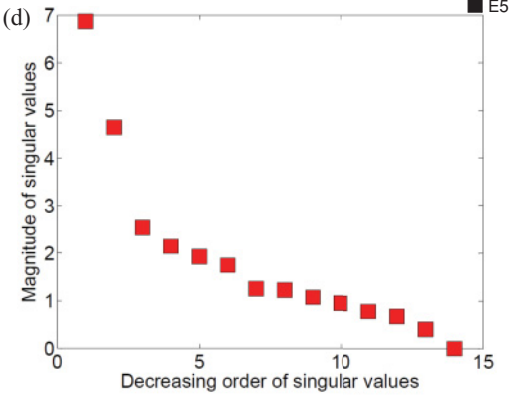


FIG. 10. (Color online) (a) The Southern Women Club Network. Circles represent women, squares represent events, black and white represent the two groups (colors have no relation to race). Cosine matrices at (b) $k = 2$ and (c) $k = 3$; gray represents the overlap between the two groups. (d) Singular values decay plot.

Events 7, 8, and 9 are still shown as the events in the overlap region. Figure 10(a) shows the solution obtained at $k = 3$ with squares representing events, circles representing women, black and white representing the two groups (colors do not represent or refer to race and have been chosen for presentation purpose only), and gray representing the overlaps.

While this network has been studied in some detail, most studies analyze it by projecting it as a one-mode network of either events or women which causes information loss. One study [13] investigates it in original form as a bipartite network. Barber's algorithm is a modified version of Newman's algorithm developed for bipartite networks. This social network presents an example where overlaps clearly exist. Hence, Barber's method, which does not cater for overlaps, forces the vertices to fall in one of the three communities he presents as his best solution. Especially, his solution forces some nodes, most especially Ruth, with group 1. This does not fit well with the actual social observation that Ruth is a member of both groups. Further, his method is not able to reveal that events 7, 8, and 9 were commonly attended by many women from both groups.

E. Bipartite Scotland Corporate Interlocks Network

We applied the method onto the bipartite data set of corporate interlocks in Scotland in 1904–1905 [31]. The full data set contains disconnected components, and we have considered the largest component comprising 131 directors and 86 firms. Barber [13] reports his best solution as identifying 20 community groups, which is considerably lower than 131 directors or 86 firms. However, he does not present a detailed report on the possible community structure of the network.

Figure 11 shows the results of the method applied to the network. The first observation is that the singular values decay gradually and there is no sharp dip obtained. In all our studies, we have found this to be a good heuristic; plotting the singular values provides a good idea of whether the graph may be

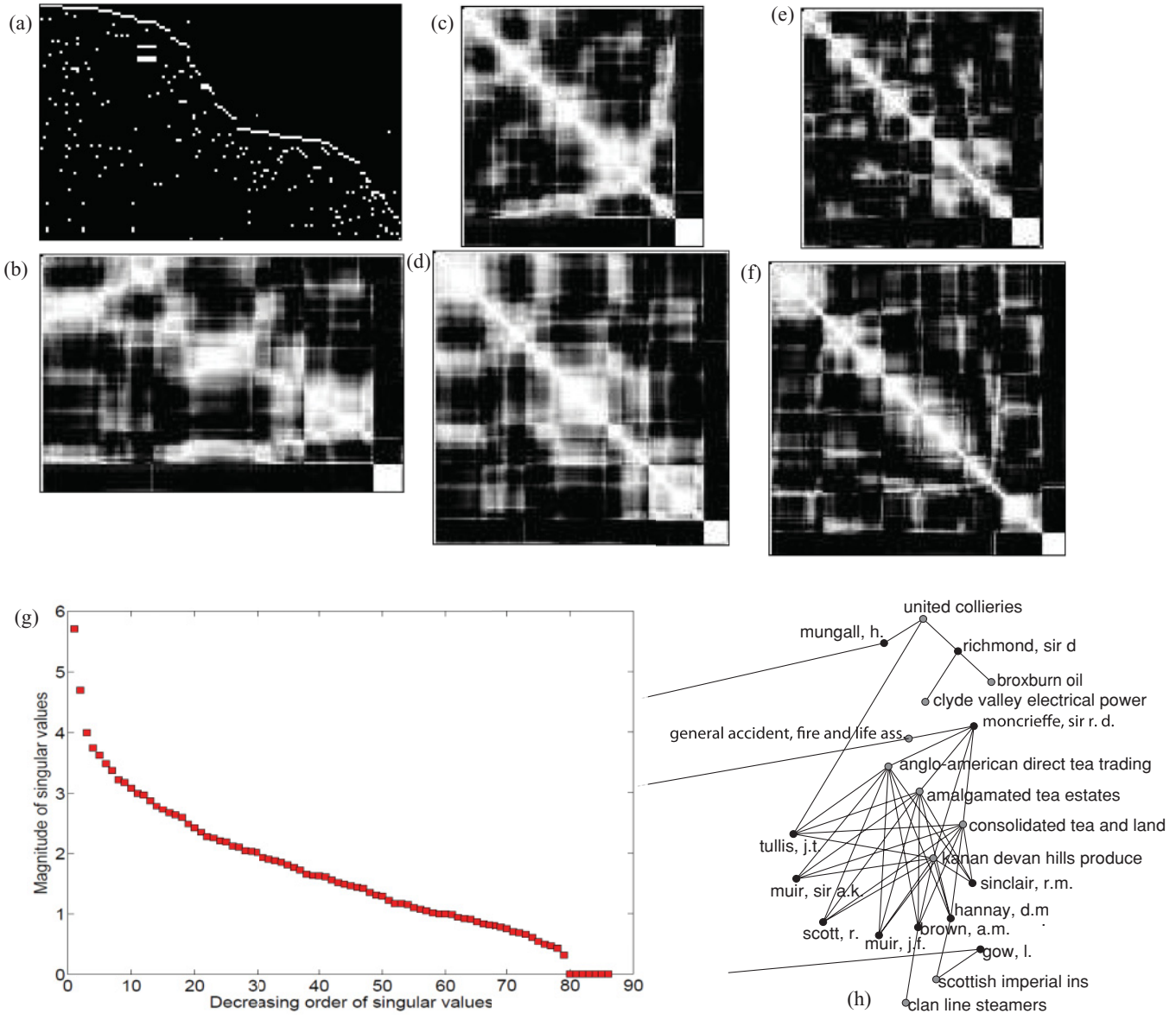


FIG. 11. (Color online) (a) The Scotland Interlocked Directorates Network matrix. Rows represent firms; columns represent directors. (b) Firm-director cosine matrix at $k = 5$; white areas represent the modules found in the bipartite network. (c),(d) Firm only and director only cosine matrices at $k = 5$. (e),(f) Firm only and director only cosine matrix at $k = 7$. (g) Singular values decay plot. (h) Details of one small module of 10 directors and 10 firms that sits clearly disjointed from the rest of the network.

partitioned into clear disjointed communities with minimal overlaps or whether there are significant overlaps that may exist between communities. If there is a sharp dip, as in the Newman-Girvan type network [Fig. 5(c)] or the Southern Womens' network case [Fig. 10(d)], it is usually possible to partition the graph, with no, or very few and clear, overlaps. However, in cases such as this directorate network, where the singular values decay more gradually, network structure turns out to be more complex.

In the case of the Scotland corporate interlocks network, we studied results from $k = 2$ to $k = 10$. The largest component (131 directors and 86 firms) seems divided into two main components, a smaller one that clearly sits as a very tight-knit community and a much larger one with shows many overlapping communities. The smaller one is shown in graph form in Fig. 11(h) with the 10 firms in gray and the 10 directors in black. Figure 11(a) shows the original matrix. The rows represent the 86 firms; the columns represent the 131 directors. Figure 11(b) shows the bipartite firm-director cosine matrix at $k = 5$. Clearly visible are about 6 communities in the larger group with significant overlaps between them and 1 separate smaller community on the bottom left [as shown in Fig. 11(h)]. Further, Figs. 11(c)–11(f) show firm-only and director-only cosine matrices at $k = 5$ and 7.

Results obtained here are very different from the 20 communities reported by [13]. However, it is difficult to compare as Barber does not present a detailed report on what the detailed structure of these 20 communities in terms of firm-director groupings. In our method, by $k = 9$ and $k = 10$ the larger groups break apart into smaller groups, but it is

difficult to identify many important relationships that are captured at lower k values of 5 or 6. Beyond $k = 10$ it is only possible to observe the smallest cliques, but no significant information can be obtained about the large-scale structure of the network, as it is visible till $k = 5$ or 6. Therefore, our results suggest 1 cluster that sits clearly disconnected from the rest of the network, and about 5–6 overlapping clusters of firms and directors.

The application of the method on this example clearly demonstrates and provides further evidence that while the lowest k values provide insight into the large-scale structure of the network, increasing the k values is like increasing the "zoom" view of the network: Progressively increasing k values to a certain threshold value reveals smaller subcommunities that lie nested in the larger ones. Therefore, the method is able to provide insight into the hierarchical structures of communities that sit inside networks. Further, as is seen from this example, the method works well for real world graphs that may be a mix of disjointed and overlapping communities. As seen in the Scotland case and in the dolphin case, the smaller community clearly sits apart from the larger one that itself is a mix of several overlapping communities.

One other advantage provided by this method is both one-mode and bipartite clustering can be observed simultaneously using the same set of computations. If a clustering of both types of elements is desired, then the cosine matrix is an $m \times n$ matrix and each entry in the matrix computes the cosine between an m-type and an n-type element. If a clustering of only one type is desired (based on the full information nonetheless), then the cosine matrix can simply be an $m \times m$ or an $n \times n$ matrix. For example, for this case, it is also possible to identify communities of only directors or only firms which derive from the bipartite information nonetheless.

VI. DISCUSSION

We have shown that SVD, combined with dimensionality reduction and unsupervised clustering can effectively reveal community structures of unweighted or weighted, unipartite or bipartite graphs in a computationally efficient manner. We tested the performance of the algorithm on various families of test benchmark networks and real world networks. In both artificially generated and real world networks, the algorithm performs successfully, and extracts the known community structure. We have shown that it outperforms other algorithms by being able to simultaneously address the issues of overlaps in community structure, the existence of hierarchy, and submodule structure within the main modules in both unipartite and bipartite networks. We have shown that it therefore provides for a powerful generalization. In addition, it is conceptually simple to implement and computationally fast. Minimal code needs to be written.

The method is defined by two main features: (1) It presents a spectral approach using SVD, which allows it to simultaneously work with unipartite as well as bipartite weighted or unweighted networks; and (2) it relaxes the strict partition assumption that is assumed in previous spectral approaches and replaces it with a continuous definition of vertex membership to multiple communities. The spectral method has been employed for solving community detection as an optimization problem. The approach presented in this paper shows that there are significant advantages if the spectral method is employed for solving community detection as a dimensionality reduction and unsupervised pattern classification problem.

One main contribution of the algorithm is that it can successfully reveal strict partitions and overlaps simultaneously in any unipartite or bipartite network. For real world large and complex data sets with a mixture of strict partitions and overlapping communities simultaneously present in the data set, this algorithm can serve as a useful tool to detect the "naturally occurring" communities.

The other main contribution of the algorithm is that it is simultaneously able to reveal that modules in large networks can be hierarchically organized with embedded sub-structures and can detect the smallest of these substructures. Networks with hierarchy show a singular values decay pattern with clustered singular values in levels, resulting in a "stepped" plot. We have shown that dimensionality reduction allows for successively revealing the finer structure in the network by increasing the k value. By considering the largest differences between successive singular values, corresponding to the number of "stepped levels" in the scree plot, one can find the modular decomposition at each hierarchical level. Each of these k values provides for one solution, with a "coarser" view of the network being revealed by the lower k approximations, and a "finer" view being revealed by the higher ones. Heuristics for deciding how many and which k values to explore for studying the network structure, and the likelihoods that a community structure is present at all, and if present, whether it is a flat one or a hierarchical one, have been thus presented.

Many ways of extending this work into the future exist. First, the method can be immediately extended for directed networks, as well as weighted networks with negative weights. Second, an existing limitation of our algorithm is that the relationship between the discovery of overlaps and hierarchical module and submodular organization is based on heuristics. Therefore, in the future, it will be interesting to extend the method and formalize a way in which a direct relationship can be established between the number and sizes of modules and submodules and the singular values and vectors. Third, while the community detection problem has mostly been studied as an optimization problem, we presented an unsupervised pattern classification and linear dimensionality reduction based interpretation. This opens up the community detection problem to a whole family of other statistical pattern classification approaches, including nonlinear (kernel based) methods. Fourth, as we have shown, the data-driven approach can work with a number of matrix forms. A good opportunity for future research could be to analyze the results from using the same method but other varied matrix formulations that capture more or different information about modules, such as a matrix of topological distances, or a matrix that captures similarity measures between nodes, etc. A theoretical comparison between different forms such as the adjacency or the signless Laplacian or other modified matrices of information using the same method is also possible. The method is data-driven, network-independent, and information-independent. Therefore, it can be applied successfully to many different matrix formulations that contain different kinds of

data on the nodes and links, with the final objective of studying the modular organization of the system. Finally, the method presented here and previous spectral approaches use the information contained in the eigenvalues and the eigenvectors in different ways. While Newman's modularity method is based on a comparison of a graph with its similar random counterpart, the method presented here makes no such comparison necessary and extracts the community information directly from the data. It will be interesting to study the relationship between the two methods, and the relationship of this method to the formal definition of modularity as given by Girvan and Newman.

Previously, we have studied system decomposition and modularity-integration for large scale engineering design problems using this approach [21,22]. In this paper, we translated our work into the domain of detecting communities in complex networks. We note here that the disciplines of genetic microarray data analysis [32,33], data compression tasks in digital image processing [34,35], statistical natural language processing tasks on large text corpora to reveal "latent" semantic meaning in the syntax of language [9,36], and ranking of web-page algorithms [23,37] also use the SVD in similar ways. Therefore, we may expect a generalized family of methods and matrix representations deriving from this approach to study patterns of regularity, modularity, and hierarchy in complex networks.

ACKNOWLEDGMENTS

The authors thank Professor Peter Robinson, School of Physics, University of Sydney for his valuable comments and sugesstions on the paper. This research was supported under Australian Research Council's Discovery Projects funding scheme (Projects No. DP0557346 and No. DP1095601).

[1] S. Fortunato, Phys. Rep. **486**, 75 (2010).
[2] M. E. J. Newman, Phys. Rev. E **74**(3), 036104 (2006).
[3] M. A. Porter, J. P. Onnela, and P. J. Mucha, Not. Am. Math. Soc. **56**, 1082 (2009).
[4] E. Ravasz, A. Somera, D. A. Mongru, Z. Oltvai, and A. Barabasi, Science **297**, 1551 (2002).
[5] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, Nature (London) **435**, 03607 (2005).
[6] M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).
[7] E. Bullmore and O. Sporns, Nat. Rev. Neurosci. **10**, 186 (2009).
[8] G. H. Golub and C. V. F. Loan, *Matrix Computations* (The John Hopkins University Press, Baltimore, 1989).
[9] T. Landauer and S. T. Dumais, Psychol. Rev. **104**, 211 (1997).
[10] A. Pothen, S. Horst, and K. P. Liou, SIAM J. Math. Anal. Appl. **11**, 430 (1990).
[11] M. E. J. Newman, *Networks: An Introduction* (Oxford University Press, Oxford, 2010).
[12] S. Fortunato and M. Barthelemy, Proc. Natl. Acad. Sci. USA. **104**, 36 (2007).
[13] M. J. Barber, Phys. Rev. E **76**, 066102 (2007).
[14] R. Guimera, M. Sales-Pardo, and L. A. N. Amaral, Phys. Rev. E **76**, 036102 (2007).
[15] A. Arenas, A. Fernandez, S. Fortunato, and S. Gomez, J. Phys. A: Math. Theor. **41**, 224001 (2008).
[16] H. Shen, X. Cheng, K. Cai, and M. Hu, Physica A **388**, 1706 (2009).
[17] X. Wang, L. Jiao, and J. Wu, Physica A **388**, 5045 (2009).
[18] A. Lancichinetti, S. Fortunato, and J. Kertesz, New J. Phys. **11**, 033015 (2009).
[19] I. A. Kovacs, R. Palotai, M. S. Szalay, and P. Csermely, PLoS One **5**, e12528 (2010).
[20] A. Arenas, J. Borge-Holthoefer, S. Gomez, and G. Zamora-Lopez, New J. Phys. **12**, 053009 (2010).
[21] S. Sarkar, A. Dong, and J. S. Gero, J. Mech. Des. **131**(8), 081006 (2009).
[22] S. Sarkar, A. Dong, and J. S. Gero, Artif. Intell. Eng. Des. Manuf. **24**, 63 (2010).
[23] G. Strang, *Introduction to Linear Algebra* (Wellesley-Cambridge Press, Wellesley, MA, 2003).
[24] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms* (Cambridge University Press, Cambridge, 2003).
[25] A. Arenas, A. Fernandez, and S. Gomez, New J. Phys. **10**, 053039 (2008).
[26] W. W. Zachary, J. Anthropol. Res. **33**, 452 (1977).
[27] A. Lancichinetti and S. Fortunato, Phys. Rev. E **80**, 016118 (2009).
[28] D. Lusseau, Proc. R. Soc. London B **270**, S186 (2003).
[29] D. Lusseau and M. E. J. Newman, Proc. R. Soc. London B **271**, S477 (2004).
[30] A. Davis, B. B. Gardner, and M. R. Gardner, *Deep South* (University of Chicago Press, Chicago, 1941).
[31] J. Scott and M. Hughes, *The Anatomy of Scottish Capital: Scottish Companies and Scottish Capital* (Croom Helm, London, 1980).
[32] O. Alter, P. O. Brown, and D. Botstein, Proc. Natl. Acad. Sci. USA. **97**, 10101 (2000).
[33] L. Liu, D. M. Hawkins, S. Ghosh, and S. Young, Proc. Natl. Acad. Sci. USA. **100**, 13167 (2003).
[34] D. Kalman, Coll. Math. J. **27**, 2 (1996).
[35] C. Long, Math. Mag. **56**, 161 (1983).
[36] A. Dong, Des. Stud. **26**, 447 (2005).
[37] J. M. Kleinberg, J. ACM **46**, 604 (1999).