FINDING OVERLAPPING COMMUNITIES FROM SUBSPACES

DAVID BINDEL , PAUL CHEW , JOHN HOPCROFT , KYU-YOUNG KIM , AND COLIN PONCE

Abstract. A *community* in a social network is a collection of unusually connected nodes. A variety of formal definitions of community have been proposed, many of which are based on optimizing some measure of the relative connectedness of a subgraph. Because exactly optimizing such measures leads to hard combinatorial problems, many researchers have considered *spectral* methods that extract approximate optima from a few leading eigenvectors of the graph Laplacian or some matrix associated with the graph. But while spectral methods have primarily been used to detect *disjoint* community structures, communities in social networks are often *overlapping*.

In this paper, we describe a framework to generalize spectral methods to the problem of detecting a community that encompasses a set of example nodes. Our approach is based on finding sparse vectors that approximately lie in some target subspace. When the target space is an invariant subspace spanned by a few eigenvectors of some graph-related matrix, our approach generalizes existing spectral methods, but we also describe how *Krylov subspaces* can be used as an effective alternative to invariant subspaces. We illustrate the effectiveness of our approach on both real data and synthetic benchmarks.

1. Introduction.

When we meet an unfamiliar group at a party, we often ask how they know each other. The usual answer is that the group is part of a larger community: college friends, co-workers, researchers in the same area, etc. Without context, it makes no sense to ask an *individual* for a single community that they belong to, but even small groups of people are often have a unique community in common. Intuitively, members of the larger community should be relatively close to all of the members of the example group. In this paper, we review some ways in which we can precisely characterize what makes a good community, and develop scalable algorithms that allow us to find a community from an example group under many of these characterizations.

Finding communities is a standard problem in network analysis, and much of the enormous literature on the subject has been devoted to mathematical characterizations of communities; see, e.g. [10, 26]. Intuitively, a community is a group that is unusually tightly connected. However, there are many possible characterizations of "unusually tight" connectivity, and different characterizations are probably relevant in different settings. Most characterizations involve the density of links or short paths *within* a community and sparsity of links or short paths *between* communities. To give the two extreme examples, we might seek to bisect a graph into two equal size communities in order to minimize the number of links between them (a min cut problem), or we might week a set with the maximum possible edge density (a maximum clique problem). Much recent work has emphasized modularity, which can be interpreted either in terms of density of edges within communities or sparsity between communities in comparison to a random graph model [23]. Different characterizations are appropriate in different settings: for example, a characterization that implicitly assumes communities are disjoint is probably inappropriate in social networks where overlapping communities are natural.

Minimizing cuts, maximizing edge density, and maximizing modularity all quickly lead to hard combinatorial optimization problems. In each of these cases, though, there are continuous relaxations that can be solved or approximated relatively quickly. The key to these methods is the observation that many interesting properties of subgraphs can be expressed in terms of quadratic forms, and the problem of finding an optimal subgraph (or collection of subgraphs) can be written as a constrained binary quadratic programming problem. Continuous relaxations of these problems frequently lead to eigenvalue problems, and to spectral approximation methods extract approximate communities from the structure of the first few eigenvectors. Spectral methods are also motivated by characterizations of communities based on the dynamics of random walks, which are well approximated using the first few eigenvectors of a (possibly scaled) transition matrix.

Spectral methods of clustering and community detection involve two choices: a choice of subspaces that will serve as the basis of approximation, and a choice of methods to extract structure from such spaces. In this paper, we propose novel choices both for the space and the method used to mine the space for community structure. Specifically, motivated by work on the dynamics of short random walks, we propose a subspace chosen by *unconverged* subspace iteration as an alternative to the invariant subspaces used in most spectral methods. We also propose an ℓ^1-penalized quadratic programming approach to finding sparse approximate indicator vectors from a given subspace. Our approach is based on standard methods from the numerical linear algebra and optimization literature, and our algorithms scale well to very large graphs.

In the next section, we review some existing spectral approaches to finding communities in graphs. In Section 3, we describe our approach to finding overlapping communities via ℓ^1 penalized quadratic programming. In Section 4, we describe a strategy for choosing subspaces that contain good approximations of indicators for communities that include a given set of example nodes, and in Section 6, we describe the performance of this strategy on a set of test cases. Finally, we conclude in Section 7.

2. Spectral approaches to community detection. Consider an undirected graph $G = (V, E)$ with n nodes and m edges. Many graph-theoretic properties of G can be connected to the linear-algebraic properties of the associated adjacency matrix $A \in \{0, 1\}^{n \times n}$. For graphs in which there is a wide distribution of node degrees, it may be more useful to consider the graph Laplacian $L = D - A$, the normalized Laplacian $\bar{L} = I - D^{-1/2}AD^{-1/2}$, the normalized adjacency matrix $\bar{A} = D^{-1/2}AD^{-1/2}$, or the transition matrix $N = D^{-1}A$, where D is the diagonal matrix of node degrees. In particular, *spectral graph theory* involves connections between the eigenvalue decompositions of these matrices and graph properties such as expansion or rapid mixing of random walks.

2.1. Spectra and quadratic forms. We think of a set of nodes $V' \subset V$ as a community if the subgraph $G' = (V', E')$ is unusually tightly connected internally or unusually sparsely connected to the rest of the graph. Suppose $V' \subset V$ is a subset of nodes and $G' = (V', E')$ is the induced subgraph. We can express many measures of relative connectivity of G' in terms of *quadratic forms*, expressions of the form $s^T Hs$ where $s \in \{0, 1\}^n$ is a binary indicator vector for the subset V' and $H \in \mathbb{R}^{n \times n}$ is a symmetric matrix. Let us write e for the vector of all ones (the indicator vector for all of G), $A \in \{0, 1\}^{n \times n}$ for the adjacency matrix, $d = Ae$ for the vector of node degrees, and $D = \operatorname{diag}(d)$ for a square matrix with node degrees on the diagonal. In terms of these primitives, we can express the following properties of G and G' via quadratic forms:

$$\begin{aligned}
\text{Nodes in } G &= e^T e \\
\text{Nodes in } G' &= s^T s \\
\text{Directed edges in } G &= e^T Ae = e^T De \\
\text{Directed edges in } G' &= s^T As \\
\text{Directed edges from } V' \text{ to } V &= s^T Ds \\
\text{Directed edges from } V' \text{ to } \bar{V}' &= s^T Ls = s^T (D - A)s.
\end{aligned}$$

Now, consider random graphs on the same nodes V drawn from the *configuration model*, in which we independently form m edges where the probability of an edge from i to j is

$d_i d_j/(2m^2)$. The expected adjacency matrix is then $\tilde{A} = \frac{dd^T}{2m}$, and

$$\begin{aligned}\text{Expected directed edges in subgraph} &= s^T \tilde{A} s \\ \text{Expected directed edges from } V' \text{ to } V &= s^T D s \\ \text{Expected directed edges from } V' \text{ to } \bar{V}' &= s^T \tilde{L} s = s^T (D - \tilde{A}) s.\end{aligned}$$

If we define the *modularity matrix* $B = A - \tilde{A} = \tilde{L} - L$, then $s^T B s$ can be interpreted as the edges in G' in excess of the expected number in a random graph, or as the count of expected edges cut between V' and $\bar{V}'$ in a random graph in excess of the cut edges in G.

The properties of a quadratic form $x^T H x$ is intimately tied to the spectra of the corresponding symmetric matrix H. For any symmetric matrix H, we can write the eigendecomposition $H = Q \Lambda Q^T$ where Q is an orthonormal matrix and $\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. This allows us to write

$$s^T H s = \sum_{j=1}^{n} \tilde{s}_j^2 \lambda_j,$$

where $\tilde{s}_j = q_j^T s$ is the projection of s onto the jth eigenvector. The *Rayleigh quotient* $\rho_H(s) \equiv s^T H s / s^T s$ is a weighted average of the eigenvalues:

$$\rho_H(s) \equiv \frac{s^T H s}{s^T s} = \sum_{j=1}^{n} w_j \lambda_j^2, \text{ where } w_j = \tilde{s}_j / \|s\|^2.$$

Note that if $\rho_H(s)$ is near to λ_1, most of the weight in the average must be on eigenvalues close to λ_1; that is, s must form an acute angle with the invariant subspace associated with the extreme eigenvalues. More precisely, if $\rho_H(s) > \lambda_1 - \delta$, then for any positive C,

$$\sum_{j : \lambda_j > \lambda_1 - C\delta} w_j > 1 - \frac{1}{C}. \tag{2.1}$$

Similarly, $\rho_H(s)$ is bounded from below by the smallest eigenvalue of H; and if $\rho_H(s)$ is close to this bound, then s must be well approximated by a linear combination of the eigenvectors associated with the smallest eigenvalues.

By relating spectral properties to Rayleigh quotients, we can bound many measures of "tightness" of a subgraph; and through (2.1), any subgraph that approaches these eigenvalue bounds must be indicated by a vector that lies nearly in the span of a few dominant eigenvectors. For example, $\rho_A(s)$ is the mean degree of G', For example, $\rho_L(s)$ is the number of edges cut relative to the number of nodes in G', and $\rho_B(s)$ is the modularity relative to the number of nodes in G'. Similarly, the generalized Rayleigh quotient $\rho_{L,D}(s) = s^T L s / s^T D s$ is the fraction of edges incident on V' that are in E'. Note that $\rho_{L,D}(s) = \rho_{\bar{L}}(D^{1/2} s)$, so we can bound this relative cut size in terms of the eigenvalues of the normalized Laplacian. Similarly, the number of edges cut relative to the expected edge cut in the configuration model is

$$\rho_{L,B}(s) = \frac{s^T L s}{s^T B s} = \frac{s^T (D - A) s}{s^T (D - \tilde{A})} = \frac{(D^{1/2} s)^T \bar{L} (D^{1/2} s)}{(D^{1/2} s)^T (I - \tilde{d}\tilde{d}^T)(D^{1/2} s)},$$

where $\tilde{d} = D^{-1/2} d / \sqrt{2m}$ is a null vector of the normalized Laplacian $\bar{L}$. Hence, for any nontrivial subgraph, this relative cut size is bounded from below by the first nonzero eigenvalue of the normalized Laplacian.

2.2. Spectra and dynamics. The probability density for a random walk of length k on G is given by the Markov chain

$$p_k = Np_{k-1} = N^k p_0,$$

where p_0 is an initial density and $N = AD^{-1}$ is the transition matrix. The stationary distribution for this Markov chain is simply $p_\infty = d/(2m)$. The matrix N is a similarity transformation of the normalized adjacency $\bar{A}$:

$$\operatorname{diag}(p_\infty)^{-1/2}\, N \,\operatorname{diag}(p_\infty)^{1/2} = D^{-1/2}AD^{-1/2} = \bar{A}.$$

Symmetry of $\bar{A}$ corresponds to *reversibility* of the Markov chain. Because the chain is reversible, all the eigenvalues of N are real, and the eigenvectors of N are related to the eigenvectors of $\bar{A}$ by a simple scaling. Further, if $\bar{A} = Q\Lambda Q^T$, we can write p_k as

$$p_k = D^{1/2}Q\Lambda^k Q^T D^{-1/2} p_0.$$

Convergence of the Markov chain is thus related to the spectrum of $\bar{A}$. In particular, the rate of convergence to the stationary distribution is typically expressed in terms of the *spectral gap* between the unit eigenvalue of $\bar{A}$ and the eigenvalue with next-largest magnitude. When the spectral gap is large, the Markov chain mixes rapidly, i.e. the distributions p_k converge very quickly to the stationary distribution.

The eigenvalues and vectors of $\bar{A}$ also give information about the *intermediate* asymptotics as the Markov chain converges to stationarity. For example, consider the case where G is partitioned into communities, with only a few inter-community links and many intra-community links. In this case, the Markov chain is nearly decoupled. According to the Simon-Ando theory for nearly-decoupled Markov chains [15], there are two phases to the dynamics of such a Markov chain: an initial transient phase associated with mixing *within* communities, and a much longer phase associated with mixing *between* communities. After the initial transient phase, the distribution is well approximated by a linear combination of *local* stationary distributions associated with the communities in the completely decoupled case. In spectral terms, the nearly-decoupled Markov chain has a cluster of eigenvalues near one, and the invariant subspace spanned by the corresponding eigenvectors is nearly identical to the invariant subspace spanned by the local stationary distributions from the decoupled chain. The remaining eigenvalues are well separated from one, and are associated with the much faster transient mixing phase.

In the scientific computing literature, spectral algorithms based on Simon-Ando theory have been applied to lumped analysis of Markov chains, with applications such as the identification of meta-stable states of biomolecules [4, 11, 16, 31]. In more recent work, Meyer has explicitly applied the theory to more general problems of data clustering [20]. The same intuition underlies the community detection work of Capocci et al [2], where correlations among eigenvectors for the transition matrix are used as the basis for community detection, though these authors appear unaware of the Simon-Ando theory. Other authors have also used the dynamics of random walks as the basis of community detection algorithm [1, 3, 25, 28, 33], though typically without any explicit exploration of the dominant eigenvalues and eigenvectors.

2.3. Spectral partitioning and clustering. Whether we want to optimize quadratic forms or find subgraphs with rapid local mixing of random walks, the eigenvalues and eigenvectors of graph-related matrices appear to hold valuable information. Spectral methods of partitioning and clustering try to mine this information from the dominant eigenvectors of a graph-related matrix such as the Laplacian L or the normalized Laplacian $\bar{L}$. Spectral bisection may be the best known example of such an algorithm. In this method, we seek to partition the nodes into equal size sets such that the number of cut edges is minimized. We can write the problem exactly as

$$\text{minimize } \tilde{s}^T L \tilde{s} \text{ s.t. } e^T \tilde{s} = 0, \tilde{s} \in \left\{ \frac{1}{2}, -\frac{1}{2} \right\}^n.$$

Note that $\tilde{s}$ is now a *sign indicator vector* rather than a binary indicator vector, though we can recover a binary indicator by adding $e/2$ to $\tilde{s}$. Every signed indicator has two-norm equal to $n/4$, which suggests we relax to the following constrained quadratic programming problem over real-valued vectors x:

$$\text{minimize } x^T L x \text{ s.t. } e^T x = 0, \|x\|_2^2 = n/4. \tag{2.2}$$

The solution to (2.2) minimizes the Rayleigh quotient $\rho_L(x) = x^T L x / x^T x$ over vectors orthogonal to the constant vector e. Because e is itself a null vector for L, minimizing $\rho_L(x)$ subject to $e^T x = 0$ is precisely the variational characterization for the smallest nonzero eigenpair of L. This smallest nonzero eigenvalue of L is then most $2/n$ times the smallest edge cut, and the signs of the corresponding eigenvector, called the *Fiedler vector*, usually provide a good partition [27]. Similarly, one can use the largest eigenvalue of B to bound the maximal modularity possible from bisecting the graph, and the sign pattern of the eigenvector of the associated eigenvector usually provides a good partition as measured by modularity [22].

It is natural to consider partitioning a network based on more than one eigenvector. This concept goes under the name of spectral clustering in the machine learning literature [19, 29], and variants on this idea have been proposed several times in the context of community detection [2, 5, 14]. The basic idea is simple: if we look across the dominant eigenvectors of various matrices — the eigenvectors of the Laplacian associated with small eigenvalues, or the eigenvectors of the transition matrix $N = D^{-1}A$ associated with large eigenvalues — then entries corresponding to the same community are likely to be highly correlated. Therefore, we associate with each node a point in a latent space whose coordinates are given by the entries of the dominant eigenvectors associated with that node. Clustering these points in latent space then reveals the communities.

2.4. Communicability and spectral clustering with overlaps.

In recent work, Estrada and co-workers have introduced definitions of community based on matrix functions [6–8]. The basic idea is to look at walks between pairs of nodes in a graph compared to walks in a complete graph. The number of walks of length k between nodes i and j in G is $e_i^T A^k e_j$, where e_i and e_j denote the ith and jth column of the identity; the matrix exponential

$$\exp(A) = I + A + \frac{1}{2}A^2 + \frac{1}{3!}A^3 + \ldots$$

therefore gives a weighted combination of the number of walks of varying lengths between any pair of nodes. Suppose $A = Q\Lambda Q^T$ is an eigendecomposition of A with $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. Then we can also write

$$\exp(A) = Q \exp(\Lambda) Q^T,$$

where $\exp(\Lambda) = \operatorname{diag}(\exp(\lambda_1), \exp(\lambda_2), \ldots)$. Note that the number of very long walks between nodes is approximately proportional to $q_1 q_1^T$, where q_1 is the dominant eigenvector of A. Thus, the matrix

$$C_A = \exp(A) - q_1 \exp(\lambda_1) q_1^T = Q \exp(\Lambda) Q^T - q_1 \exp(\lambda_1) q_1^T$$

can be interpreted as a measure of how well connected nodes are by short walks (compared to some asymptotic state). Based on this intuition, Estrada and Hatano suggest forming the "communicability graph" where i and j are connected if the (i, j) entry of C_A is positive [6]. They then define a community to be a clique in the communicability graph.

Though it is not usually described as such, the Estrada-Hatano approach is closely related to the latent-space approaches used in spectral partitioning. In most latent-space approaches, one would choose a fixed number of eigenvectors as the basis of the latent coordinate system; in contrast, the Estrada-Hatano method replaces a hard threshold separating the important vectors from the unimportant ones with an exponential weighting that assigns more importance to the eigenvectors associated with the most positive eigenvalues. The (i, j) entry of the communicability matrix C_A is given by the inner product of vectors v_i and v_j, where

$$v_i = \begin{bmatrix} q_2(i)e^{\lambda_2/2} \\ q_3(i)e^{\lambda_3/2} \\ \vdots \\ q_n(i)e^{\lambda_n/2} \end{bmatrix}.$$

The condition that the (i, j) entry of C_A is positive can then be interpreted as the condition that the latent space vectors v_i and v_j form an acute angle. Of course, because of the exponential weighting, only the first few components of these vectors are likely to matter much to the value of the entry in C_A.

2.5. The Motzkin-Straus program.

Spectral partitioning based on the graph Laplacian or the modularity matrix may be appropriate for finding a partition of a network into disjoint pieces. But what about quadratic forms associated with local properties that might be more appropriate for evaluating overlapping communities, such as relatively high edge density? For example, consider the problem of finding a subgraph that maximizes the edge density $s^T As/\|s\|_1^2$. This leads to the natural continuous relaxation

$$\text{maximize } x^T Ax \text{ s.t. } x \text{ nonnegative}, e^T x = 1.$$

According to a theorem of Motzkin and Straus [12, 21], the maximum value attainable in this program is $1 - 1/c_n$, where c_n is the maximum clique size. Furthermore, the scaled indicator vector for a maximum clique is an optimizer for the problem. Thus, in this case, the continuous relaxation of the optimization problem is exactly equivalent to the original discrete problem — and both are NP-hard. Note that the nonnegativity constraint is crucial in this problem; without the constraint, the objective function is unbounded.

It is instructive to consider the similarities and differences between clique finding using the Motzkin-Straus formulation and spectral partitioning using the Fiedler vector. In both cases, we can write down the original combinatorial problem as a discrete quadratic programming problem with constraints, which we then relax to a continuous problem. But in the case of the Laplacian or the modularity matrix, we were able to write the program in terms of a sign indicator vector rather than a binary indicator vector, and so there was no need for a nonnegativity constraint. Algebraically, we can switch from indicators to sign indicators because $Le = 0$; in terms of the graph, what matters is that the Laplacian is used to measure edges that are *cut* rather than those that are retained (and no edges are cut when we take G' to be the entirety of G). In contrast, nonnegativity plays a crucial role in the Motzkin-Straus program, and is likely to remain important in any optimization problem which emphasizes internal edge density rather than the sparsity of edges cut. The Motzkin-Straus program also involves a constraint on the ℓ^1 norm ($\|x\|_1 = e^T x$ when x is nonnegative), and ℓ^1 norm constraints and penalties are popular for encouraging sparse solutions to optimization problems. In contrast, the spectral partitioning program involves a constraint on the ℓ^2 norm.

3. Communities via ℓ^1 minimization. Suppose $U \in \mathbb{R}^{n\times k}$ is an orthonormal basis for the invariant subspace associated with a few of the largest eigenvalues of the Laplacian or the normalized Laplacian. A standard spectral approach to clustering the graph would be to cluster rows of U via k-means [19, 29], though there are also other methods to extract communities. Like k-means, though, most such methods partition the nodes into hard, non-overlapping clusters. These clusters may be recursively partitioned in order to expose a hierarchy of communities, but aside from some stochastic approaches, most spectral methods do not allow nodes to belong to multiple communities.

It is instructive to consider how spectral clustering works in a simple case when there are two disjoint communities that are loosely connected. Let us consider the two dominant eigenvectors v_1 and v_2 of the normalized Laplacian matrix $\bar{A}$. If we were to decouple the two communities, each would have an associated stationary distribution; let us call the vectors for these distribution u_1 and u_2. According to the Simon-Ando theory, both eigenvectors are well approximated by linear combinations of two local stationary distributions associated with the two communities; that is,

$$\begin{aligned} v_1 &\approx w_{11}u_1 + w_{21}u_2 \\ v_2 &\approx w_{12}u_1 + w_{22}u_2. \end{aligned}$$

In matrix terms, we can write $V \approx UW$, where

$$V = \begin{bmatrix} v_1 & v_2 \end{bmatrix}, \quad U = \begin{bmatrix} u_1 & u_2 \end{bmatrix}, \quad W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}.$$

Each row of U either has the form $\begin{bmatrix} \alpha & 0 \end{bmatrix}$ or $\begin{bmatrix} 0 & \alpha \end{bmatrix}$, depending on whether the row corresponds to a node in the first community or the second community. Therefore, each row of V is either approximately proportional to $\begin{bmatrix} w_{11} & w_{12} \end{bmatrix}$ or $\begin{bmatrix} w_{21} & w_{22} \end{bmatrix}$, depending whether the node is associated with the first or second community. Thus, each community is associated with a direction in two-dimensional space, and we can tell whether a node is in a community or not by whether or not it is almost parallel to the community's direction vector. Alternately, we could normalize each row of V to unit Euclidean length, so that two nodes correspond to the same community if the normalized rows are nearly identical.

Now, suppose we wanted to find all the nodes in the same community as node i in this two-community example. One approach would be to cluster the rows as above. Another approach is to seek a sparse vector in the span of V such that i is in the support. In general, seeking sparse vectors in a given subspace is a hard problem, but in many cases a small ℓ^1 norm is a good proxy for sparsity. Hence, we seek to solve the linear programming problem

$$\text{minimize } e^T y = \|y\|_1 \text{ s.t. } y = Vx,\ y \text{ nonnegative},\ y_i \geq 1. \tag{3.1}$$

Note that each vector in the span of V is approximately a linear combination of u_1 and u_2, so $y \approx \alpha_1 u_1 + \alpha_2 u_2$ and $e^T y \approx \alpha_1 e^T u_1 + \alpha_2 e^T u_2$. Let us assume without loss of generality that node i is in the first community; then the constraint $y_i \geq 1$ is approximately the same as $\alpha_1 u_{1,i} \geq 1$, while the nonnegativity constraint simply implies $\alpha_2 \geq 0$. The solution to (3.1) should therefore be $\alpha_1 = u_{1,i}^{-1}$ and $\alpha_2 = 0$, i.e. y is a simple scaling of u_1.

More generally, suppose we had a basis $V \in \mathbb{R}^{n\times k}$ for some k-dimensional subspace of $\mathbb{R}^n$, and that we believed $V = UW$ where the columns of U are sparse and nonnegative. Further, let us assume that the support of any given column of U (i.e. the set of indices where there are nonzero elements) is not wholly contained in the supports of the remaining columns. Under this assumption, y must be a positively-weighted linear combination of the rows of U, and the objective function will be the corresponding combination of the row sums.

A minimizer will then be $y = u_j/u_{j,i}$ where j is chosen so that $u_{j,i} \neq 0$ and $e^T u_j/u_{j,i}$ is minimal; furthermore, this minimum will generally be unique.

With these examples for motivation, let us consider the following heuristic method for community detection. Take some matrix $V \in \mathbb{R}^{n \times k}$, such as the dominant eigenvectors of the normalized Laplacian, possibly subject to row scaling. Then solve the linear programming problem

$$\text{minimize } e^T y = \|y\|_1 \text{ s.t. } y = Vx,\ y \text{ nonnegative},\ y(J_{\text{seed}}) \geq 1, \tag{3.2}$$

where J_{seed} is the set of indices for some "seed" nodes that we require to be in the community. The support of the resulting vector is then taken as a community containing all the nodes in the seed set. In practice, of course, the vector will not typically be exactly sparse, so we actually take our community to be all nodes such that $y_i \geq \delta$, where δ is some threshold value. Furthermore, we can relax the constraint that y should lie *strictly* in the span of the columns of V, and instead force y to form a small angle with the space. To do this, we solve the box-constrained quadratic programming problem

$$\text{minimize } \frac{1}{2} y^T (I - P_V) y + \tau e^T y \text{ s.t. } y \text{ nonnegative},\ y(J_{\text{seed}}) \geq 1, \tag{3.3}$$

where $P_V = V(V^T V)^{-1} V^T$ is the orthogonal projector onto the span of V. The penalty parameter τ balances the importance we assign to sparsity of the solution (as represented by small ℓ^1 norm) versus how close the solution is to the span of V.

4. Approximate invariant subspaces. One drawback to spectral clustering approaches is that they require the computation of eigenvectors. If the graph contains many small communities, we might need many eigenvectors to resolve them. The usual approach to this problem is a top-down recursion: use a few eigenvectors to partition the graph into coarse communities, then use a few eigenvectors of the associated subgraphs to further partition the coarse communities. We propose an alternate bottom-up approach: based on random walks from the seed nodes, build local approximations to invariant subspace bases (possibly with row scaling) associated with the dominant eigenvectors. We can then seek communities in the direct sum of these (scaled) approximate invariant subspaces.

To motivate our approach, first consider random walks starting from node i. Let e_i denote the probability vector which is one on node i and zero elsewhere; then the distribution of random walks of length k starting from node i is $p_k = N^k e_i$. Now, consider the span of the vectors $P_k = \begin{bmatrix} p_k & p_{k+1} & \dots & p_{k+l} \end{bmatrix}$, where k and l are some modest numbers. Note that we can compute orthonormal bases for these spans indirectly using the recurrence

$$Q_k R_k = N Q_{k-1}, \tag{4.1}$$

where $Q_0 R_0 = P_0$ and where the upper triangular matrices R are chosen so that columns of each $Q_k \in \mathbb{R}^{n \times l}$ are orthonormal. The iteration (4.1) is called *subspace iteration* []; and as $k \to \infty$, Q_k approaches an orthogonal basis for the dominant eigenspace associated with the l largest eigenvectors of N. Our interest now, though, is not in the limiting case when k is large, but for a much more modest number of steps. Intuitively, we expect rapid mixing of random walks within any communities with which i is associated. In the case of such rapid mixing, rows of Q_k associated with one of these communities will end up pointing in much the same direction. This suggests that the rows of Q_k can be used to discriminate between different mixing rates, depending on k.

Now, let $V^{(i,k,l)}$ denote an orthonormal basis for an l-dimensional subspace obtained by taking k steps of subspace iteration as described above (starting from node i) and then scaling the rows to unit length. Within the communities associated with node i, we expect to see rapid mixing; consequently, as in the Simon-Ando theory, those rows of $V^{(i,k,l)}$ should all be pointing in nearly the same direction. In contrast, rows of $V^{(i,k,l)}$ associated with nodes not closely tied to i will point in different directions. For a given seed, we now define $V^{(k,l)}$ to be an orthonormal basis for the directed sum of the spaces spanned by the $V^{(i,k,l)}$ over all i in the seed set. In $V^{(k,l)}$, we expect rows associated with a common community for all i to point in approximately the same direction, while rows associated with nodes that are not as tightly associated to most of the seed nodes will point in different directions.

5. The algorithm. Putting together the results of the previous two sections, we can detect communities around a seed set by the following procedure:

1. Generate the basis $V^{(i,k,l)}$ for each seed node in turn.
2. Generate $V^{(k,l)}$ as an orthonormal basis for (approximately) the direct sum of the $V^{(i,k,l)}$. The "approximately" comes from throwing out any nearly-linearly-dependent columns as we go.
3. Solve the quadratic program (3.3) to get a score vector.
4. Optionally, *reseed* the algorithm by finding the highest-scoring node according to setp 3, using it to augment the seed set, and re-running the algorithm from step 1.
5. Optionally, threshold the score vector and greedily refine the resulting indicator vector by adding/removing nodes from the subgraph so as to best improve the angle between the indicator vector and the subspace at each step.

6. Experimental results. We now describe the results of some experiments based on the algorithm from the previous section. In addition to some small test graphs, we demonstrate our method on benchmarks generated by the method of Lancichinetti and Fortunato [17, 18] and on Facebook data from Rice University.

6.1. Small graphs. We first test our algorithm on four sample graphs that are small enough to be easily visualized: an artificial test set due to Wang et al. [], Zachary's karate network [], a graph of games played in the NFL [], and a social network of bottlenose dolphins []. In each network, we arbitrarily chose node 11 as the sole seed node.

The test graph from the paper by Wang et al. is shown in Figure 6.1. The graph has four tightly-coupled communities that are loosely tied together in a diamond. This structure is very clearly defined, and it shows up clearly in the row-normalized adjacency spectrum as a gap between the fourth and fifth eigenvalues (Figure 6.2). We compute our score based on a four-dimensional approximate eigenspace computed from ten steps of a random walk starting at 11. The nodes in the relevant community are clearly the highest scoring. We choose our indicator vector to indicate all ndoes with score greater than 0.5, and the cosine of the angle between this indicator vector and the relevant subspace is 0.987.

The social network studied by Zachary (Figure 6.3) describes relationships between members in a karate club that split into two different clubs. Unlike in the previous example, though, there is no clear gap in the spectrum of the normalized adjacency matrix between the second and third eigenvalues (Figure 6.4). This reflects the fact that the two communities in this case are not nearly as well-defined: there are quite a few connections between the two groups, and each group contains members with degrees of only one or two. Nonetheless, we recover a good partitioning of the network into two groups using a two-dimensional subspace based on random walks of up to length ten; and the angle between the subspace and the thresholded score vector is 0.931.

The football league network (Figure 6.5) has ten or eleven distinct, well-defined modules, and these are well reflected in the spectrum of the normalized adjacency matrix (Figure 6.6). Despite the relatively large number of communities in the network, we are able to pull out the community containing node 11 using a subspace of dimension 4 based on random walks of length at most ten. This illustrates that we are able to obtain useful information about the community structure of the network *without* getting close to converging to an invariant subspace. The angle between the computed community indicator vector and the subspace is 0.976.

The final small test graph is the dolphin social network shown in Figure 6.7. This network has a small spectral gap after the first two eigenvalues, and it does indeed partition reasonably well into two subgraphs (Figure 6.8). However, like the karate club graph, there is significant variation in the node degrees. We again use a two-dimensional subspace based on walks of up to length ten to compute the score vector. Unlike the previous examples, though, greedy optimization of the angle changes the indicator vector returned by an initial thresholding of 0.5. If we simply use a threshold of 0.5, the cosine of the angle between the subspace and the indicator vector is 0.937; if we add another seven nodes, to our computed community we bring the cosine up to 0.975.

6.2. Lancichinetti-Fortunato graphs. The benchmark graph generation method due to Lancichinetti and Fortunato has been used to compare several different community detection methods [17, 18]. These random graphs are characterized by power law distributions for node degrees and community sizes and a *topological mixing parameter* μ that controls the fraction of the links for each node that cross to a community with which the node is not associated.

In Figure 6.9, we show the structure of the adjacency matrix for an LF graph with 1000 nodes, 28 communities, and a mixing parameter of 0.5. The nodes are sorted according to their community index, so the community structure is clearly visible as a block structure along the diagonal. The community structure is also clearly visible in the structure of the spectrum of the normalized adjacency matrix (Figure 6.10) in the form of a gap between eigenvalue 28 (0.4997) and eigenvalue 29 (0.4344). In Figure 6.11, we show the raw score vector returned when we start with a seed of two neighbors chosen at random from one community. For each seed, we update the space the top three vectors obtained from looking at random walks of up to length ten. Though our search subspace is only six-dimensional, the score vector clearly indicates the desired community, and simple thresholding at 0.5 yields a set with a Jaccard similarity of 0.95 to the true community 14. The computed community is wrong only in that it has two extra nodes in community 19, which respectively have two and three links into community 14.

As a second example, in Figure 6.12, we show the adjacency matrix for an LF graph with 1000 nodes, 32 communities, and a mixing parameter of 0.6. In this problem, the gap between eigenvalues 32 (0.4458) and 33 (0.4385) is much smaller than in the previous example, enough so that it is difficult to see the gap in a plot (Figure 6.13). As in the previous example, we build a six-dimensional subspace starting from a seed of size two. The resulting score vector is shown in Figure 6.14. If we threshold this vector at 0.5, 19 of the resulting 20 nodes belong to the true community (which has size 32). If we choose the 32 highest scoring nodes, 28 of them are in the true community. In this case, however, greedily changing the set to optimize the angle to the subspace is a poor strategy, since it adds many nodes that do not belong to the community.

As a final example of an LFR benchmark graph, we consider an adjacency matrix for an LF graph with 1000 nodes in 47 communities. Half the nodes belong to two different communities, and for each node 30% of the links go to a community with which the node is not directly associated. The spectrum of the normalized adjacency matrix, shown in Figure 6.15, does have a gap between eigenvalues 47 and 48, but this gap is not pronounced. We chose two nodes at random from community 18 to use as a seed; between the two of them, these nodes have six neighbors in community 18, but they also have five neighbors in com-

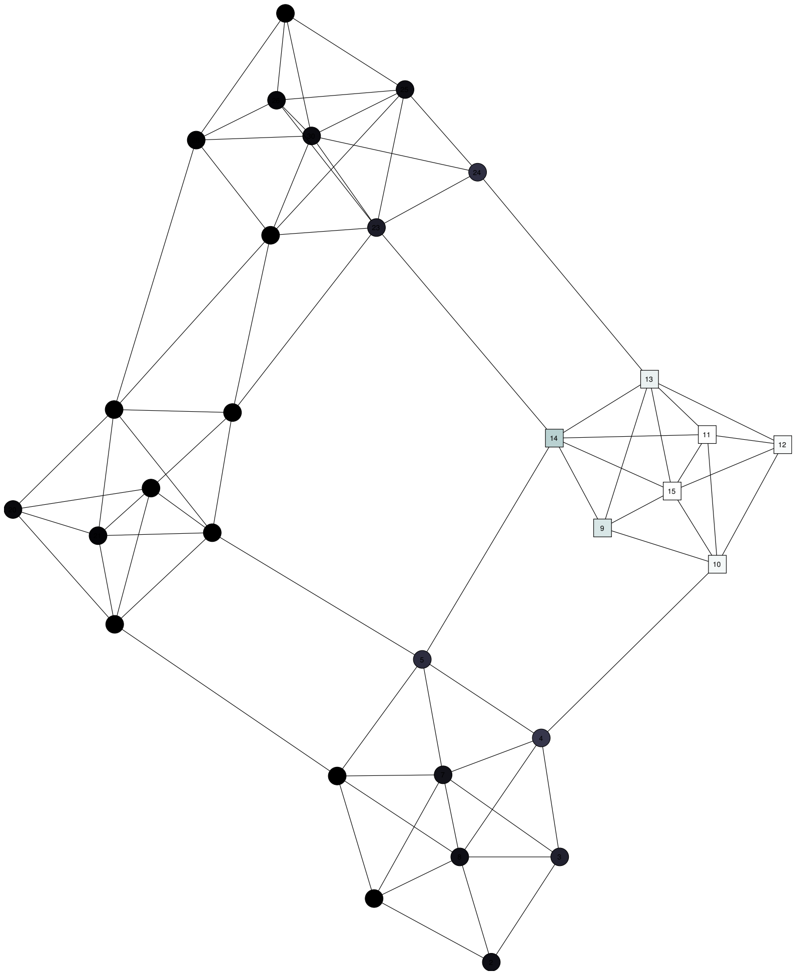

FIGURE 6.1. *Labeled test graph of Wang* et al. *The gray level indicates the raw score vector; nodes in squares are in the thresholded community.*

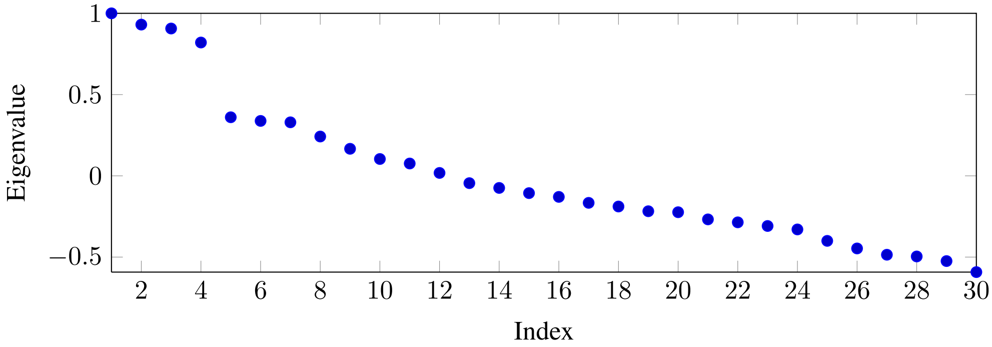

FIGURE 6.2. *Eigenvalues for the normalized adjacency matrix of the test graph of Wang* et al. *Note the clear gap between the fourth and fifth eigenvalues.*

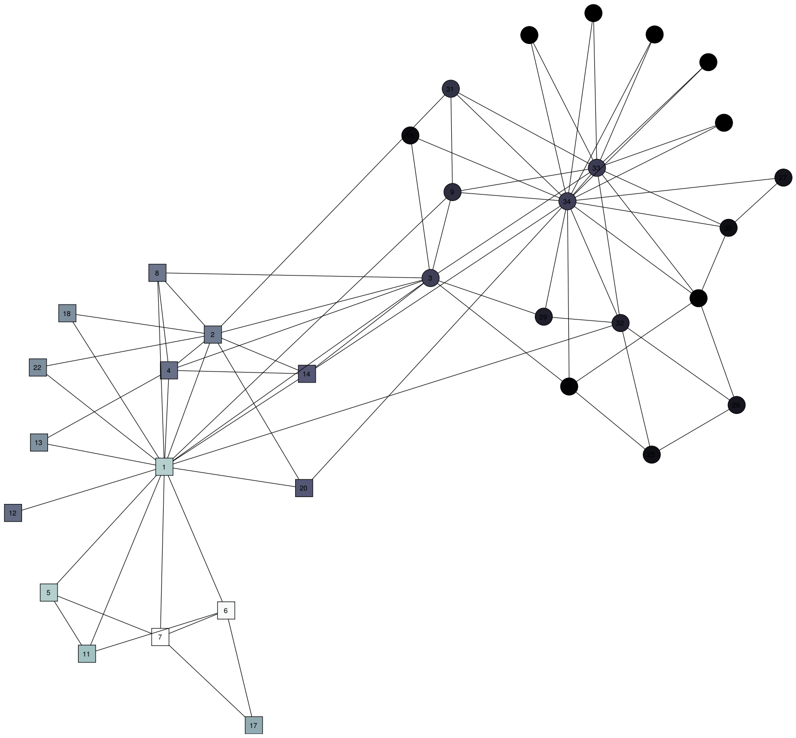

FIGURE 6.3. *Labeled graph of the Zachary karate network. The gray level indicates the raw score vector; nodes in squares are in the thresholded community.*

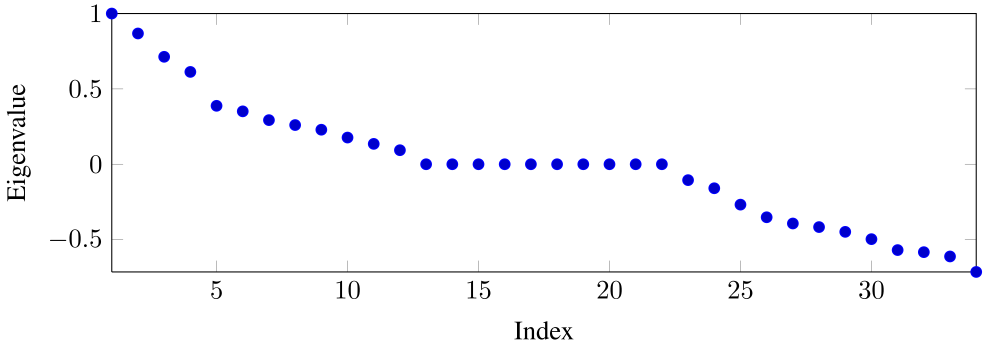

FIGURE 6.4. *Eigenvalues for the normalized adjacency matrix of Zachary's karate network. Though this graph is usually partitioned into two, note that there is not a clear spectral gap after the first two eigenvalues.*

munity 25 and four neighbors in community 4. If we try to extract the community structure as before from a six-dimensional subspace grown via random walks from the two seed nodes, we obtain the score vector shown in Figure 6.16. While this vector somewhat reflects the community structure we seek, it is also contaminated by large scores for nodes that belong to communities 4 and 25. We can surpress this undesired signal using the re-seeding approach described before, where we incrementally expand the seed set based with high-scoring nodes

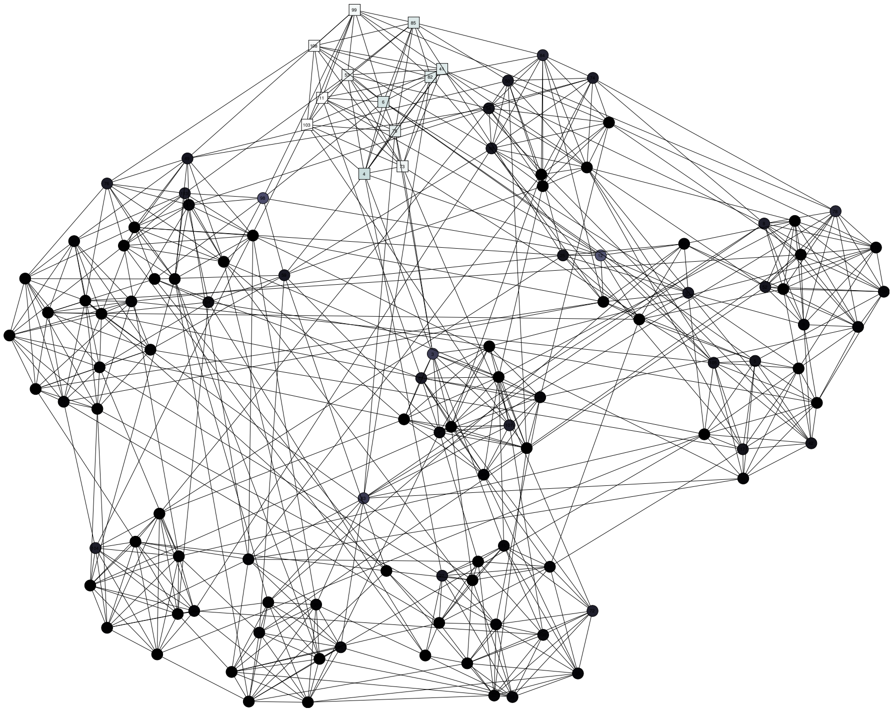

FIGURE 6.5. *Labeled graph of the football league network. The gray level indicates the raw score vector; nodes in squares are in the thresholded community.*

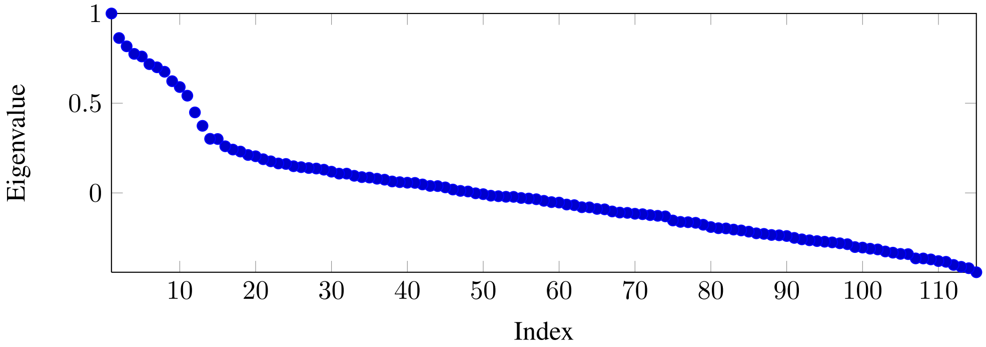

FIGURE 6.6. *Eigenvalues for the normalized adjacency matrix of the football network. There are clear gaps in the spectrum after the eleventh, twelfth, and thirteenth eigenvalues.*

from a previous computation. The results after a dozen re-seedings are shown in Figure 6.17. Simply thresholding the vector in Figure **??** with a cutoff of 0.5 gives a set of size 41 which overlaps community 18 (which has size 32) in 31 nodes. Of the 32 top-scoring nodes, 29 are in community 18.

We draw three main conclusions from our experiments with the LF benchmark graphs:

1. We can find good approximate community indicator vectors from low-dimensional subspaces (of size 6) based on random walks even when the total number of communities is much larger.

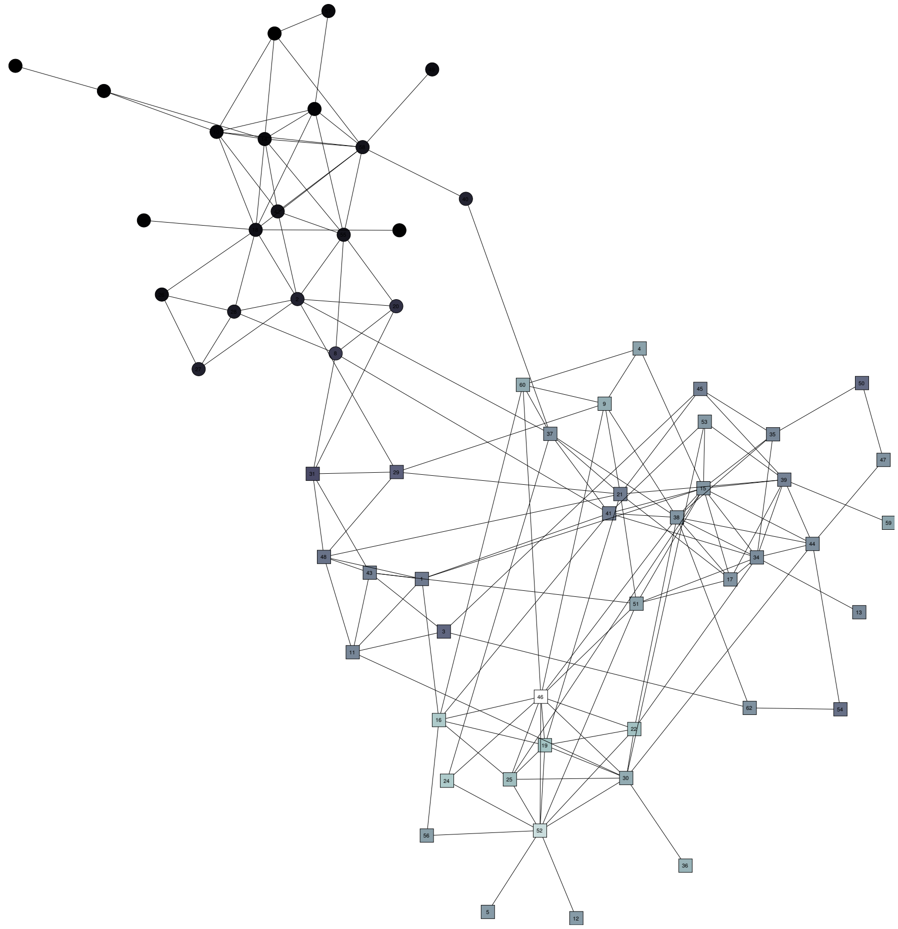

FIGURE 6.7. *Labeled graph of the dolphin social network. The gray level indicates the raw score vector; nodes in squares are in the thresholded community.*

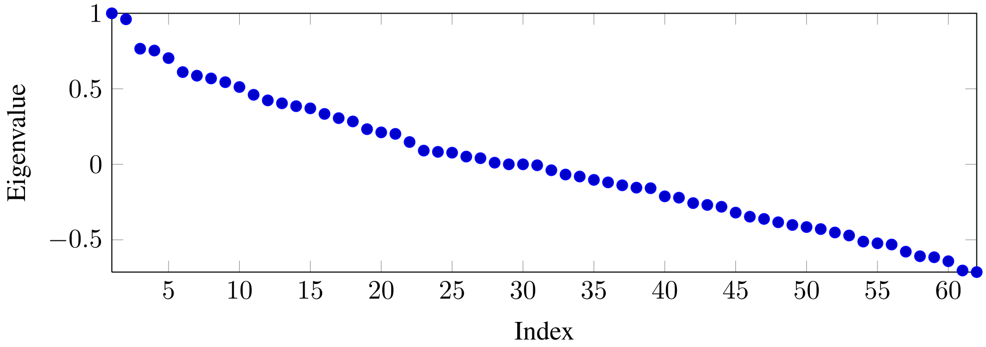

FIGURE 6.8. *Eigenvalues for the normalized adjacency matrix of the dolphin social network. There is a gap in the spectrum after the first two eigenvalues.*

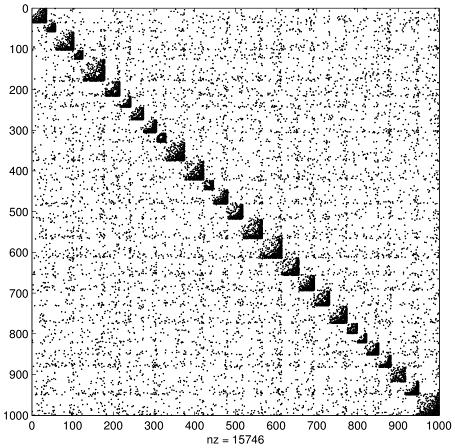

FIGURE 6.9. *Spy plot of the adjacency matrix for the non-overlapping LFR benchmark example (mixing parameter* 0.5*)*

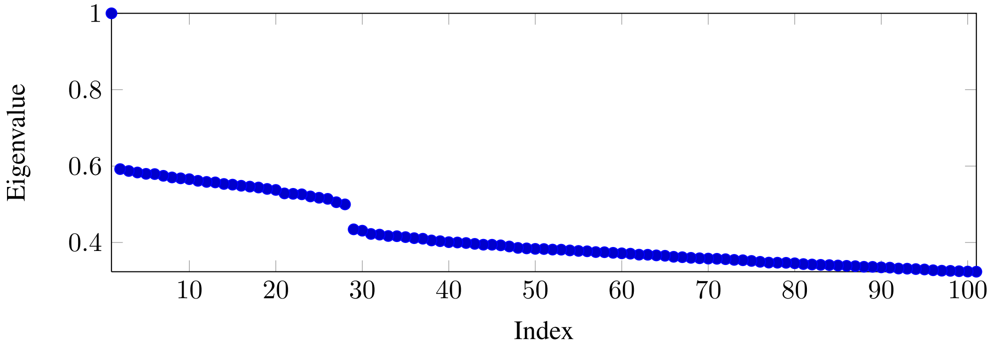

FIGURE 6.10. *Largest eigenvalues of the normalized adjacency matrix for the first LFR benchmark graph (Figure 6.9).*

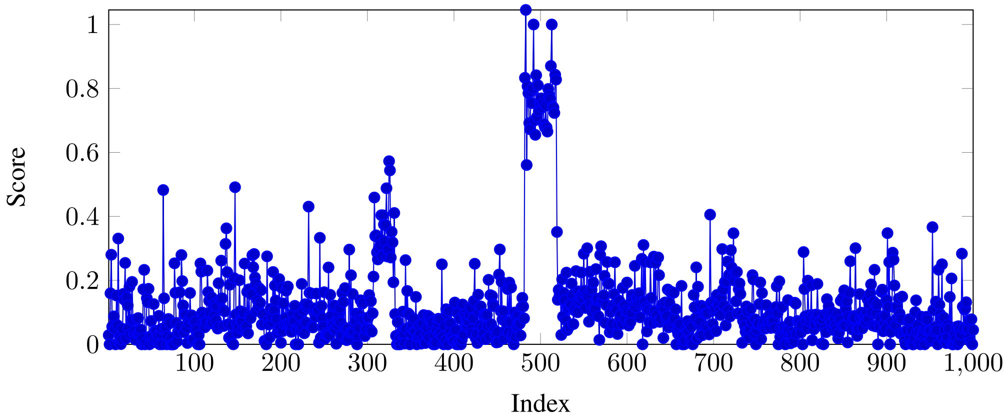

FIGURE 6.11. *Score vector for the two-node seed of 492 and 513 in the first LFR benchmark graph (Figure 6.9).*

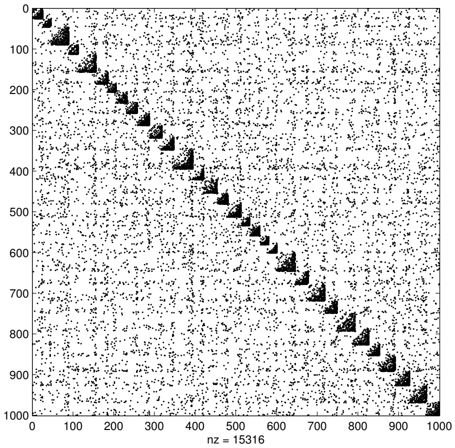

FIGURE 6.12. *Spy plot of the adjacency matrix for the non-overlapping LFR benchmark example (mixing parameter* 0.6*)*

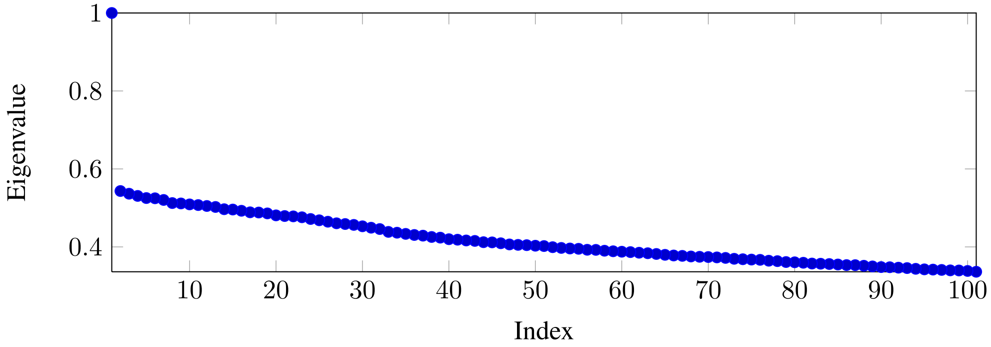

FIGURE 6.13. *Largest eigenvalues of the normalized adjacency matrix for the second LFR benchmark graph (Figure 6.12).*

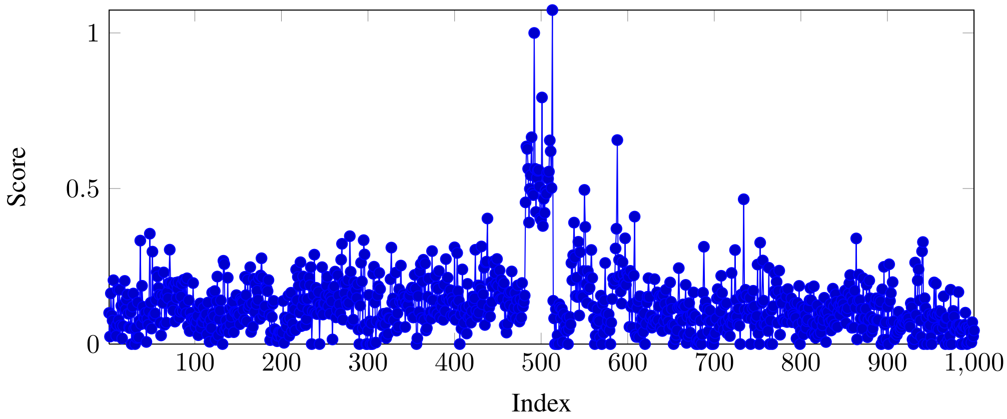

FIGURE 6.14. *Score vector for the two-node seed of 492 and 513 in the second LFR benchmark graph (Figure 6.12).*

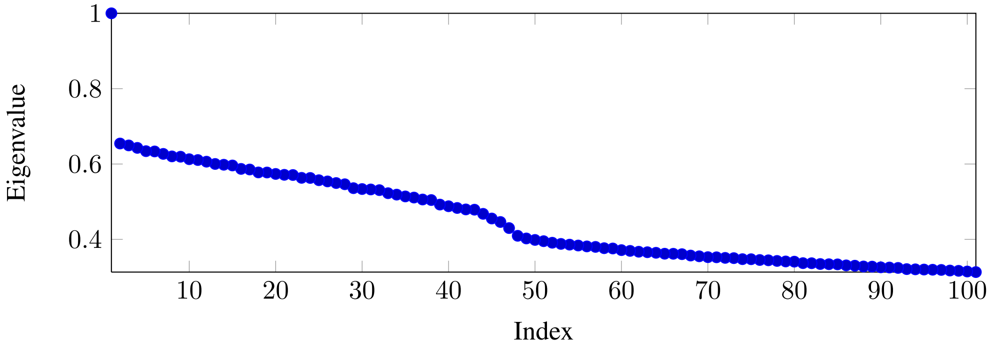

FIGURE 6.15. *Largest eigenvalues of the normalized adjacency matrix for the third (overlapping) LFR benchmark graph.*

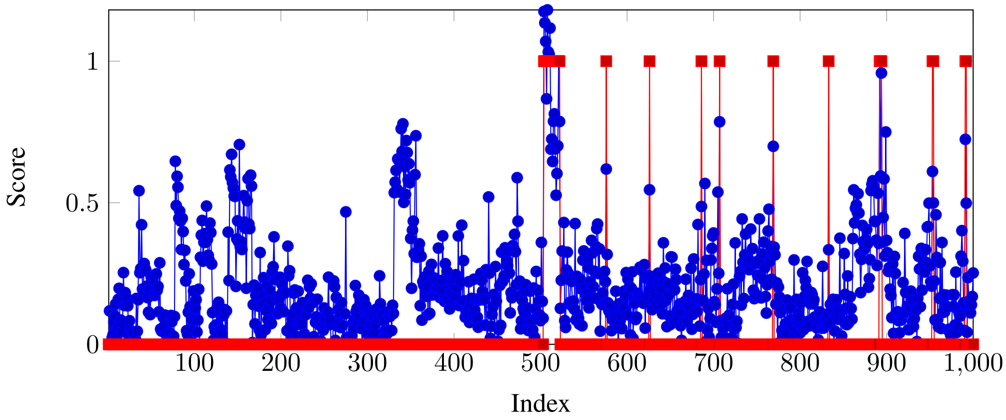

FIGURE 6.16. *Score vector for the two-node seed of 521 and 892 in the overlapping LFR benchmark graph. The indicator vector for the true community is marked with red squares.*

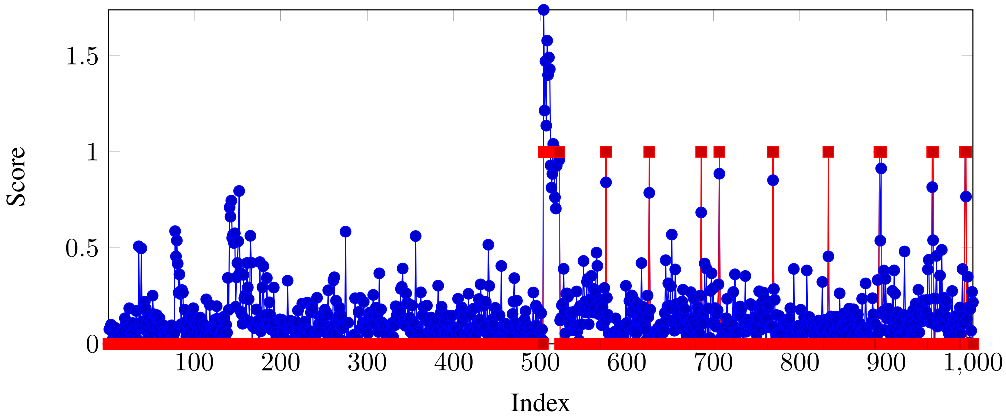

FIGURE 6.17. *Score vector for the two-node seed of 521 and 892 in the overlapping LFR benchmark graph after 12 reseeds. The indicator vector for the true community is marked with red squares.*

2. The method still provides useful information even when there is a large gap between the first and second eigenvalue and a very small gap between the λ_c and λ_{c+1}, where c is the number of communities.
3. When the communities overlap substantially, the subspaces drawn from a very small seed set may be insufficient to distinguish one of the several communities to which the seed set is closely connected. The re-seeding strategy is useful for expanding the seed set enough that it is possible to extract just one community.

7. Conclusions. In this report, we have described methods for detecting potentially overlapping communities in networks by searching for sparse vectors in subspaces. In conventional spectral approaches, we would usually look in invariant subspaces spanned by a few eigenvectors of the graph Laplacian, the normalized Laplacian, the adjacency matrix, or some related matrix. Motivated by the connection between the dominant eigenvectors of a transition matrix and the transient dynamics of mixing in an associated Markov chain, we propose an alternate method to construct subspaces based on the eigenvector approximations drawn from short random walks. We illustrate our approach on small test examples drawn from the literature and generated by a standard benchmark code. Based on these examples, we observe that our approach to generating subspaces can yield good approximations, even when we construct a subspace that is substantially smaller than the number of communities.

REFERENCES

[1] R. Andersen and K. J. Lang, *Communities from seed sets*, in Proceedings of the 15th international conference on World Wide Web, WWW '06, ACM, New York, NY, USA, 2006, pp. 223–232.

[2] A. Capocci, V. Servedio, G. Caldarelli, and F. Colaiori, *Detecting communities in large networks*, Physica A: Statistical Mechanics and its Applications, 352 (2005), pp. 669–676. http://dx.doi.org/10.1016/j.physa.2004.12.050.

[3] G. B. Davis and K. M. Carley, *Clearing the FOG: Fuzzy, overlapping groups for social networks*, Social Networks, 30 (2008), pp. 201–212. http://dx.doi.org/10.1016/j.socnet.2008.03.001.

[4] P. Deuflhard, *Identification of almost invariant aggregates in reversible nearly uncoupled markov chains*, Linear Algebra and its Applications, 315 (2000), pp. 39–59. http://dx.doi.org/10.1016/S0024-3795(00)00095-1.

[5] L. Donetti and M. A. Munoz, *Detecting network communities: a new systematic and efficient algorithm*, (2004). http://arxiv.org/abs/cond-mat/0404652.

[6] E. Estrada and N. Hatano, *Communicability in complex networks*, Physical Review E, 77 (2008), pp. 036111+. http://dx.doi.org/10.1103/PhysRevE.77.036111.

[7] ———, *Communicability graph and community structures in complex networks*, Applied Mathematics and Computation, 214 (2009), pp. 500–511. http://dx.doi.org/10.1016/j.amc.2009.04.024.

[8] E. Estrada and D. J. Higham, *Network properties revealed through matrix functions*, SIAM Rev., 52 (2010), pp. 696–714. http://dx.doi.org/10.1137/090761070.

[9] S. Fortunato, *Community detection in graphs*, (2010). http://arxiv.org/abs/0906.0612.

[10] S. Fortunato and C. Castellano, *Community structure in graphs*, (2007). http://arxiv.org/abs/0712.2716.

[11] D. Fritzsche, V. Mehrmann, D. B. Szyld, and E. Virnik, *An SVD approach to identifying metastable states of markov chains*, Electronic Transactions on Numerical Analysis, 29 (2008), pp. 46–69. http://etna.mcs.kent.edu/vol.29.2007-2008/pp46-69.dir/pp46-69.html.

[12] L. E. Gibbons, D. W. Hearn, P. M. Pardalos, and M. V. Ramana, *Continuous characterizations of the maximum clique problem*, Mathematics of Operations Research, 22 (1997). http://dx.doi.org/10.2307/3690403.

[13] M. Girvan and M. E. J. Newman, *Community structure in social and biological networks*, (2001). http://arxiv.org/abs/cond-mat/0112110.

[14] M. S. Handcock, A. E. Raftery, and J. M. Tantrum, *Model-based clustering for social networks*, Journal of the Royal Statistical Society: Series A (Statistics in Society), 170 (2007), pp. 301–354.
`http://dx.doi.org/10.1111/j.1467-985X.2007.00471.x`.

[15] D. J. Hartfiel, *Proof of the Simon-Ando theorem*, Proceedings of the American Mathematical Society, 124 (1996).
`http://dx.doi.org/10.2307/2161399`.

[16] M. N. Jacobi, *A robust spectral method for finding lumpings and meta stable states of Non-Reversible markov chains*, Electronic Transactions on Numerical Analysis, 37 (2010), pp. 296–306.
`http://etna.mcs.kent.edu/vol.37.2010/pp296-306.dir/`.

[17] A. Lancichinetti and S. Fortunato, *Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities*, Physical Review E, 80 (2009), pp. 016118+.
`http://dx.doi.org/10.1103/PhysRevE.80.016118`.

[18] ———, *Community detection algorithms: A comparative analysis*, Physical Review E, 80 (2009), pp. 056117+.
`http://dx.doi.org/10.1103/PhysRevE.80.056117`.

[19] U. Luxburg, *A tutorial on spectral clustering*, Statistics and Computing, 17 (2007), pp. 395–416.
`http://dx.doi.org/10.1007/s11222-007-9033-z`.

[20] C. D. Meyer and C. D. Wessell, *Stochastic data clustering*, (2010).
`http://arxiv.org/abs/1008.1758`.

[21] T. S. Motzkin and E. G. Straus, *Maxima for graphs and a new proof of a theorem of turán*, Canadian Journal of Mathematics, 17 (1965), pp. 533–540.
`http://math.ca/10.4153/CJM-1965-053-6#`.

[22] M. E. J. Newman, *Finding community structure in networks using the eigenvectors of matrices*, Physical Review E, 74 (2006), pp. 036104+.
`http://dx.doi.org/10.1103/PhysRevE.74.036104`.

[23] ———, *Modularity and community structure in networks*, Proceedings of the National Academy of Sciences, 103 (2006), pp. 8577–8582.
`http://dx.doi.org/10.1073/pnas.0601602103`.

[24] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek, *Uncovering the overlapping community structure of complex networks in nature and society*, Nature, 435 (2005), pp. 814–818.
`http://dx.doi.org/10.1038/nature03607`.

[25] P. Pons and M. Latapy, *Computing communities in large networks using random walks (long version)*, (2005).
`http://arxiv.org/abs/physics/0512106`.

[26] M. A. Porter, J.-P. Onnela, and P. J. Mucha, *Communities in networks*, (2009).
`http://arxiv.org/abs/0902.3788`.

[27] A. Pothen, H. D. Simon, and K. P. Liou, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 430–452.
`http://dx.doi.org/10.1137/0611030`.

[28] M. Rosvall and C. T. Bergstrom, *Maps of random walks on complex networks reveal community structure*, Proceedings of the National Academy of Sciences, 105 (2008), pp. 1118–1123.
`http://dx.doi.org/10.1073/pnas.0706851105`.

[29] S. Schaeffer, *Graph clustering*, Computer Science Review, 1 (2007), pp. 27–64.
`http://dx.doi.org/10.1016/j.cosrev.2007.05.001`.

[30] L. Tang and H. Lin, *Community Detection and Mining in Social Media*, Morgan & Claypool Publishers, 2010.

[31] R. M. Tiffenbach, *On an SVD-based algorithm for identifying meta-stable states of markov chains*, Electronic Transactions on Numerical Analysis, 38 (2011), pp. 17–33.
`http://etna.mcs.kent.edu/vol.38.2011/pp17-33.dir/`.

[32] F. Wu and B. A. Huberman, *Finding communities in linear time: a physics approach*, The European Physical Journal B - Condensed Matter and Complex Systems, 38 (2004), pp. 331–338.
`http://dx.doi.org/10.1140/epjb/e2004-00125-x`.

[33] V. Zlatić, A. Gabrielli, and G. Caldarelli, *Topologically biased random walk and community finding in networks*, Physical Review E, 82 (2010), pp. 066109+.
`http://dx.doi.org/10.1103/PhysRevE.82.066109`.