



A spectral method for bipartizing a network and detecting a large anti-community



A. Concas ^{a,*}, S. Noschese ^{b,2,4}, L. Reichel ^{c,5}, G. Rodriguez ^{a,1,2,4}

^a Department of Mathematics and Computer Science, University of Cagliari, Viale Merello, 92, 09123 Cagliari, Italy

^b Department of Mathematics "Guido Castelnuovo", Sapienza University of Rome, P.le A. Moro, 2, I-00185 Roma, Italy

^c Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA

ARTICLE INFO

Article history:

Received 12 December 2018

Received in revised form 14 June 2019

MSC:
65F15
05C50
05C82

Keywords:

Network analysis
Network approximation
Bipartition
Anti-community

ABSTRACT

Relations between discrete quantities such as people, genes, or streets can be described by networks, which consist of nodes that are connected by edges. Network analysis aims to identify important nodes in a network and to uncover structural properties of a network. A network is said to be bipartite if its nodes can be subdivided into two nonempty sets such that there are no edges between nodes in the same set. It is a difficult task to determine the closest bipartite network to a given network. This paper describes how a given network can be approximated by a bipartite one by solving a sequence of fairly simple optimization problems. The algorithm also produces a node permutation which makes the possible bipartite nature of the initial adjacency matrix evident, and identifies the two sets of nodes. We finally show how the same procedure can be used to detect the presence of a large anti-community in a network and to identify it.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Networks describe how discrete quantities such as genes, people, proteins, or streets are related. They arise in many applications, including genetics, epidemiology, energy distribution, and telecommunication; see, e.g., [1,2] for discussions on networks and their applications. Networks are represented by graphs $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$, which are determined by a set of vertices (nodes) $\mathcal{V} = \{v_i\}_{i=1}^n$, a set of edges $\mathcal{E} = \{e_k\}_{k=1}^m$, and a set of positive weights $\mathcal{W} = \{w_k\}_{k=1}^m$. Here $e_k = (i_k, j_k)$ represents an edge from vertex v_{i_k} to vertex v_{j_k} . The weight w_k is associated with the edge e_k ; a large value of $w_k > 0$ indicates that edge e_k is important. For instance, in a road network, the weight w_k may be proportional to the amount

* Corresponding author.

E-mail addresses: anna.concas@unica.it (A. Concas), noschese@mat.uniroma1.it (S. Noschese), reichel@math.kent.edu (L. Reichel), rodriguez@unica.it (G. Rodriguez).

¹ Partially supported by the Fondazione di Sardegna, Italy 2017 research project "Algorithms for Approximation with Applications (Acube)" and the Regione Autonoma della Sardegna, Italy research project "Algorithms and Models for Imaging Science (AMIS)" (intervento finanziato con risorse FSC 2014–2020 - Patto per lo Sviluppo della Regione Sardegna).

² Partially supported by the INdAM-GNCS, Italy research project "Metodi numerici per problemi mal posti".

³ Anna Concas gratefully acknowledges Sardinia Regional Government for the financial support of her Ph.D. scholarship (P.O.R. Sardegna F.S.E. Operational Programme of the Autonomous Region of Sardinia, European Social Fund 2014–2020 – Axis III Education and Formation, Objective 10.5, Line of Activity 10.5.12).

⁴ Member of the INdAM Research group GNCS.

⁵ Partially supported by National Science Foundation, USA grants DMS-1729509 and DMS-1720259.

of traffic on the road that is represented by the edge e_k . In this paper, we consider connected undirected graphs without self-loops and multiple edges. In particular, all edges represent “two-way streets”, i.e., if (i_k, j_k) is an edge, then so is (j_k, i_k) . The weights associated with these edges are assumed to be the same. In unweighted graphs all weights are set to one.

We will represent a graph \mathcal{G} with n nodes by its adjacency matrix $A = [a_{i,j}]_{i,j=1}^n$, where

$$a_{i,j} = \begin{cases} w_k, & \text{if there is an edge } e_k \text{ between the nodes } v_i \text{ and } v_j \text{ with weight } w_k, \\ 0, & \text{otherwise.} \end{cases}$$

Since \mathcal{G} is undirected and the weights associated with each direction of an edge are the same, the matrix A is symmetric. The largest possible number of edges of an undirected graph with n nodes without self-loops is $n^2 - n$, but typically the actual number of edges, m , of such graphs that arise in applications is much smaller. The adjacency matrix A , therefore, is generally very sparse.

A graph \mathcal{G} is said to be *bipartite* if the set of vertices \mathcal{V} that make up the graph can be partitioned into two disjoint nonempty subsets \mathcal{V}_1 and \mathcal{V}_2 (with $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$), such that any edge starting at a vertex in \mathcal{V}_1 points to a vertex in \mathcal{V}_2 , and vice versa. This, in particular, excludes the presence of self-loops in a bipartite graph.

Bipartivity is an important structural property. It has been studied also as the 2-coloring problem [3]. In fact determining if a graph can be colored with 2 colors is equivalent to determining whether or not the graph is bipartite, and thus testing if a network is bipartite or not is computable in linear time using breadth-first or depth-first search algorithms. It is therefore interesting to determine a bipartite approximation of a non-bipartite graph, or measure the distance of a non-bipartite graph from being bipartite. We say that a splitting of the set of vertices \mathcal{V} of a weighted undirected graph \mathcal{G} into two disjoint nonempty subsets \mathcal{V}_1 and \mathcal{V}_2 (with $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$), is a best bipartition of \mathcal{G} if the sum of the weights w_k associated with edges $e_k = (i, j)$ that point from vertices v_i in \mathcal{V}_ℓ ($\ell = 1, 2$) to vertices v_j in the same set \mathcal{V}_ℓ is minimal. Such edges e_k are called “frustrated”, and computing the minimum number of edges whose deletion makes the graph bipartite is an NP-hard optimization problem [4]. We remark that the above definition is analogous to the definition of a best bipartition of an undirected unweighted graph proposed by Estrada and Gómez–Gardeñes [5], where the *spectral bipartivity index* of a network with adjacency matrix A is defined as

$$b_s = \frac{\text{trace}(\exp(-A))}{\text{trace}(\exp(A))}. \quad (1.1)$$

This measure also can be applied to the weighted graphs considered in the present paper.

The problem of discovering approximately bipartite structures in graphs and networks has been considered by various authors. Most popular approaches are based on the eigendecomposition of the Laplacian and signless Laplacian matrices. Other spectral approaches consider the adjacency matrix associated to the graph. In the case of a symmetric bipartite adjacency matrix, the signs of the entries of an eigenvector associated with the smallest eigenvalue can be used to partition the graph, i.e., nodes that correspond to positive entries belong to one set, and nodes that correspond to negative entries belong to the other set; see [6]. In case the smallest eigenvalue is multiple, the splitting of the nodes may vary according to the considered vector in the associated eigenspace. In [7] the presence of \pm pairs in the spectrum of the adjacency matrix of a bipartite graph is exploited in order to identify approximate bipartite structures within protein–protein interaction undirected networks; see also [8] for a spectral approach that can be used to discover approximately bipartite substructures in directed graphs.

We are interested in developing a numerical method for determining a “good” bipartition $(\mathcal{V}_1, \mathcal{V}_2)$, i.e., a bipartition for which the sum of the weights w_k associated with the edges $e_k = (i, j)$ that point from a vertex v_i in \mathcal{V}_1 to a vertex v_j in \mathcal{V}_2 , or vice versa, is fairly small. The algorithm is approximate, or “heuristic”, in the sense that it does not necessarily produce the best possible bipartition.

As it will be made clear in the following, the same bipartition method may be used for the identification of a large anticomunity. A community is a group of nodes which are highly connected among themselves, but are less connected to the rest of the network, or isolated from it. Conversely, an anti-community is a node set that is loosely connected internally, but has many external connections [9]; see [10], where a spectral method is used to detect communities and anti-communities. Community and anti-community detection in networks is an important problem with applications in various fields, including physics, computer science, and social sciences [11–15]. Although the identification of communities is predominant in the investigation of meso-scale structures in networks, the detection of the so-called core–periphery structures, whose most popular notion was developed by Borgatti and Everett [16], attracts a continuing interest also in the mathematical community; see also [17]. For our purposes, the identification of a single large anti-community can be understood as that of a core–periphery structure in the given network.

This paper is organized as follows. Section 2 discusses some properties of bipartite graphs and Section 3 describes an algorithm for determining a “good” bipartition. An application of the bipartition method to the identification of a large anticomunity is discussed in Section 4. Finally, Section 5 presents computed examples and two case studies, while Section 6 contains concluding remarks.

2. Approximating the spectral structure of a bipartite graph

This section discusses some properties of the adjacency matrix for an undirected bipartite graph. Some inequalities that are useful for the design of our bipartization method also will be introduced. The discussion in the first part of the section assumes that the vertices are suitably ordered. Subsequently, we will describe how to achieve such an ordering.

Assume for the moment that the undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$ is bipartite, i.e., its vertex set \mathcal{V} can be split into two disjoint nonempty subsets \mathcal{V}_1 and \mathcal{V}_2 with n_1 and n_2 nodes, respectively, such that there are no edges between the nodes in \mathcal{V}_1 and between the nodes in \mathcal{V}_2 . We may assume that $n_1 \geq n_2$, otherwise we interchange the sets \mathcal{V}_1 and \mathcal{V}_2 .

Let the vertices in the set \mathcal{V} be ordered so that the first n_1 of them belong to the set \mathcal{V}_1 and the remaining n_2 vertices belong to \mathcal{V}_2 . Then the adjacency matrix for the graph \mathcal{G} is of the form

$$A_B = \begin{bmatrix} O_{n_1} & C \\ C^T & O_{n_2} \end{bmatrix}, \quad (2.1)$$

where O_k denotes the $k \times k$ zero matrix, and $C = [c_{i,j}] \in \mathbb{R}^{n_1 \times n_2}$ with $c_{i,j} > 0$ if the node v_i in \mathcal{V}_1 is connected to the node v_{n_1+j} in \mathcal{V}_2 ; otherwise $c_{i,j} = 0$.

We adapt to our notation a known result in graph theory; see, e.g., [18, Theorem 3.14].

Proposition 2.1. *Let \mathcal{G} be an unweighted graph with n nodes. Then \mathcal{G} is bipartite and the adjacency matrix can be partitioned as in (2.1) if and only if the spectrum of the adjacency matrix is symmetric with respect to the origin, i.e.,*

$$\sigma(A_B) = \{\lambda_1, \dots, \lambda_{n_2}, \underbrace{0, \dots, 0}_{n_1-n_2}, -\lambda_{n_2}, \dots, -\lambda_1\}, \quad (2.2)$$

for some integers $n_1 \geq n_2$ and non-negative numbers $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{n_2}$. The claim holds true also for weighted graphs, as long as the weights are positive.

Proof. For the sake of clarity, we give a quick sketch of the proof. The necessary condition is straightforward. The sufficient condition can be proved by noting that, for $k = 0, 1, \dots$, $\text{trace}(A_B^{2k+1}) = 0$ if the spectrum is symmetric. Then, the positivity of the weights implies that $(A_B^{2k+1})_{i,i} = 0$, that is, the graph is bipartite since it does not contain odd cycles. \square

Remark 2.2. Under the assumption of [Proposition 2.1](#), it is immediate to verify that if λ is a nonzero eigenvalue of A_B and $\mathbf{q} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$, with $\mathbf{x} \in \mathbb{R}^{n_1}$ and $\mathbf{y} \in \mathbb{R}^{n_2}$, is an associated eigenvector, then $(-\lambda, \begin{bmatrix} \mathbf{x} \\ -\mathbf{y} \end{bmatrix})$ is an eigenpair, too. This implies that λ is a singular value of the block C in (2.1), while \mathbf{x} and \mathbf{y} are its left and right singular vectors, respectively, if scaled to be of unit length.

Let $n = n_1 + n_2$ with $n_1 \geq n_2 \geq 1$. Then, the above observation gives us the possibility to describe the spectral structure of A_B in terms of the singular value decomposition of C ; see also [19, Section 8.6.1]. Let $C = X\tilde{D}Y^T$ be a singular value decomposition of C , where $\tilde{D} \in \mathbb{R}^{n_1 \times n_2}$ has $D = \text{diag}(\lambda_1, \dots, \lambda_{n_2})$ as its upper block, and $X = [X_1, X_2] \in \mathbb{R}^{n_1 \times n_1}$ and $Y \in \mathbb{R}^{n_2 \times n_2}$ are orthogonal matrices with $X_1 \in \mathbb{R}^{n_1 \times n_2}$. Introduce the diagonal matrix

$$\mathcal{D} = \text{diag}(D, O_{n_1-n_2}, -D),$$

and the orthogonal matrix

$$Q = \begin{bmatrix} U_1 & U_2 & U_1 \\ V & O_{n_2, n_1-n_2} & -V \end{bmatrix}, \quad (2.3)$$

where $U_1 = \frac{1}{\sqrt{2}}X_1$, $U_2 = X_2$, and $V = \frac{1}{\sqrt{2}}Y$, with $U_1^T U_1 = V^T V = \frac{1}{2}I_{n_2}$ and $U_2^T U_2 = \frac{1}{2}I_{n_1-n_2}$. Then, the spectral factorization

$$A_B = Q\mathcal{D}Q^T, \quad (2.4)$$

takes the form

$$\begin{bmatrix} U_1 & U_2 & U_1 \\ V & O_{n_2, n_1-n_2} & -V \end{bmatrix} \text{diag}(D, O_{n_1-n_2}, -D) \begin{bmatrix} U_1 & U_2 & U_1 \\ V & O_{n_2, n_1-n_2} & -V \end{bmatrix}^T. \quad (2.5)$$

In the special case when $n_1 = n_2$, the submatrices of (2.3) with n_1-n_2 columns disappear, and the spectral factorization (2.5) simplifies to

$$A_B = \begin{bmatrix} U_1 & U_1 \\ V & -V \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & -D \end{bmatrix} \begin{bmatrix} U_1 & U_1 \\ V & -V \end{bmatrix}^T,$$

with $U_1 U_1^T = VV^T = \frac{1}{2}I_{n_1}$.

Now, let A be an adjacency matrix of an undirected graph. We would like to approximate the graph by a bipartite one, and therefore seek to approximate A by a matrix of the form A_B . We do this in several steps and first show some inequalities that are applicable to diagonal eigenvalue matrices.

Proposition 2.3. Let $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_\ell$ be a nonincreasing real sequence and let $\beta_1, \beta_2, \dots, \beta_\ell$ be another real sequence. The distance between these sequences measured in the least squares sense,

$$\left(\sum_{i=1}^{\ell} (\alpha_i - \beta_i)^2 \right)^{1/2}, \quad (2.6)$$

is minimal if and only if the β_i are in nonincreasing order, i.e., if $\beta_1 \geq \beta_2 \geq \dots \geq \beta_\ell$.

Proof. Assume that both sequences are in nonincreasing order and that the distance can be reduced by changing the order of the β_i . Consider the pairs (α_1, β_1) and (α_2, β_2) . Then

$$(\alpha_1 - \beta_2)^2 + (\alpha_2 - \beta_1)^2 \leq (\alpha_1 - \beta_1)^2 + (\alpha_2 - \beta_2)^2$$

is equivalent to

$$\alpha_2(\beta_1 - \beta_2) \geq \alpha_1(\beta_1 - \beta_2).$$

Assume $\beta_1 > \beta_2$. Then $\alpha_2 \geq \alpha_1$, which is a contradiction unless $\alpha_1 = \alpha_2$. If the β_j are ordered arbitrarily, then we can reorder these coefficients pairwise until they form a nonincreasing sequence. Each pairwise swap reduces (2.6). \square

In our application of [Proposition 2.3](#), we let $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_n$ be the eigenvalues of the adjacency matrix $A \in \mathbb{R}^{n \times n}$. The graph associated with this matrix might not be bipartite. We would like the sequence of eigenvalues of the matrix $A_B \in \mathbb{R}^{n \times n}$, given by (2.1), to be close to the sequence $\alpha_1, \alpha_2, \dots, \alpha_n$ and appear in \pm pairs. By [Proposition 2.3](#), we know that the eigenvalues $\beta_1, \beta_2, \dots, \beta_n$ of A_B should be in nonincreasing order, and by [Proposition 2.1](#) they vanish or appear in \pm pairs. We know from (2.5) that at least $n_1 - n_2$ eigenvalues of A_B should be zero.

Proposition 2.4. Let $\{\alpha_j\}_{j=1}^n$, with $n = n_1 + n_2$ and $n_1 \geq n_2$, be a real nonincreasing sequence. Then the sequence $\{\beta_j\}_{j=1}^n$ with elements

$$\beta_j = \begin{cases} \frac{1}{2}(\alpha_j - \alpha_{n-j+1}), & j = 1, 2, \dots, n_2, \\ 0, & j = n_2 + 1, \dots, n_1, \\ -\beta_{n-j+1}, & j = n_1 + 1, \dots, n, \end{cases} \quad (2.7)$$

is the closest sequence to $\{\alpha_j\}_{j=1}^n$ in the least squares sense consisting of at least $n_1 - n_2$ zeros and nonvanishing entries appearing in \pm pairs.

Proof. The sequence $\{\beta_j\}_{j=1}^n$ consists of $n_1 - n_2$ zero values and $n_2 \pm$ pairs. Indeed, we have

$$\beta_j - \beta_{j+1} = \begin{cases} \frac{1}{2}(\alpha_j - \alpha_{j+1}) + \frac{1}{2}(\alpha_{n-j} - \alpha_{n-j+1}), & 1 \leq j \leq n_2 - 1, \\ \frac{1}{2}(\alpha_{n_2} - \alpha_{n_1+1}), & j = n_2, \\ 0, & n_2 + 1 \leq j \leq n_1 - 1, \\ \beta_{n_2}, & j = n_1, \\ \beta_{n-j} - \beta_{n-j+1}, & n_1 + 1 \leq j \leq n - 1, \end{cases}$$

and it follows that the sequence is nonincreasing. It remains to establish that the β_j defined by (2.7) are the best possible. Consider the minimization problems

$$\begin{cases} \min_{\beta} ((\alpha_j - \beta)^2 + (\alpha_{n-j+1} + \beta)^2), & 1 \leq j \leq n_2, \\ \min_{\beta} (\beta^2), & n_2 + 1 \leq j \leq n_1. \end{cases} \quad (2.8)$$

The solution sequence $\{\beta_j\}_{j=1}^n$ is given by (2.7). Thus, the β_j form a nonincreasing sequence consisting of $n_1 - n_2$ zero values and $n_2 \pm$ pairs. It is the closest such sequence to the sequence $\{\alpha_j\}_{j=1}^n$ in the sense that it solves the minimization problems (2.8). \square

We would like to determine an approximation of the matrix A by a matrix of the form (2.1), where we allow row and column permutations of the latter matrix. Define the spectral factorization

$$A_B = W_B \Lambda_B W_B^T, \quad \Lambda_B = \text{diag}(\lambda_1^{(B)}, \lambda_2^{(B)}, \dots, \lambda_n^{(B)}),$$

where W_B is an orthogonal matrix and the eigenvalues are ordered according to

$$\lambda_1^{(B)} \geq \lambda_2^{(B)} \geq \dots \geq \lambda_n^{(B)}.$$

We remark that only the first n_1 eigenvalues are ordered as in (2.4).

Let us initially assume that the nonzero eigenvalues are distinct. If the eigenvectors are made unique, e.g., by making their first component positive, a comparison with (2.5) shows that

$$W_B = \begin{bmatrix} U_1 & U_2 & U_1 Z \\ V & O & -VZ \end{bmatrix}, \quad A_B = \begin{bmatrix} D & O & O \\ O & O_{n_1-n_2} & O \\ O & O & -ZDZ \end{bmatrix}, \quad (2.9)$$

where Z is the *flip* matrix

$$Z = \begin{bmatrix} 0 & & 1 \\ & \ddots & \\ 1 & & 0 \end{bmatrix} \in \mathbb{R}^{n_2 \times n_2}.$$

In the presence of multiple nonzero eigenvalues, the corresponding eigenvectors are not uniquely determined, so the spectral factorization (2.9) is only one of several possible distinct factorizations.

Let

$$A = W \Lambda W^T, \quad \Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n), \quad (2.10)$$

be a spectral factorization of A with an orthogonal eigenvector matrix W and the eigenvalues ordered according to

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n. \quad (2.11)$$

Partition the eigenvector matrix W conformally with the eigenvector matrix W_B of A_B , i.e.,

$$W = \begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \end{bmatrix}.$$

We would like to approximate the eigenvector matrix W of A by the eigenvector matrix W_B of A_B . This suggests that we solve the minimization problem

$$\min_{\substack{U_1^T U_1 = V^T V = \frac{1}{2} I_{n_2} \\ U_2^T U_2 = \frac{1}{2} I_{n_1-n_2}}} \left\| \begin{bmatrix} U_1 & U_2 & U_1 \\ V & O & -V \end{bmatrix} - \begin{bmatrix} W_{11} & W_{12} & W_{13}Z \\ W_{21} & W_{22} & W_{23}Z \end{bmatrix} \right\|_F, \quad (2.12)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. This problem splits into the three independent problems

$$\min_{U_1^T U_1 = \frac{1}{2} I_{n_2}} \{ \|U_1 - W_{11}\|_F^2 + \|U_1 - W_{13}Z\|_F^2 \}, \quad (2.13)$$

$$\min_{V^T V = \frac{1}{2} I_{n_2}} \{ \|V - W_{21}\|_F^2 + \|V + W_{23}Z\|_F^2 \}, \quad (2.14)$$

$$\min_{U_2^T U_2 = \frac{1}{2} I_{n_1-n_2}} \{ \|U_2 - W_{12}\|_F^2 \}. \quad (2.15)$$

Problem (2.13) can be written as

$$\min_{X_1^T X_1 = I_{n_2}} \left\{ \left\| X_1 - \sqrt{2} W_{11} \right\|_F^2 + \left\| X_1 - \sqrt{2} W_{13}Z \right\|_F^2 \right\}. \quad (2.16)$$

The following result shows how we can easily solve this problem.

Proposition 2.5. *The solution of problem (2.16) can be determined by computing the singular value decomposition of $W_{11} + W_{13}Z$ and setting all singular values to one.*

Proof. Consider the problem

$$\min_{X^T X = I} \|X - W\|_F^2.$$

It can be written as

$$\min_{X^T X = I} \{ \text{trace}(X^T X) - 2 \text{trace}(X^T W) + \text{trace}(W^T W) \}.$$

The first and last terms are independent of X . Therefore we obtain the equivalent linear minimization problem

$$\min_{X^T X = I} \{ -\text{trace}(X^T W) \}.$$

Similarly, the linear problem associated to the minimization problem (2.16) is given by

$$\min_{X_1^T X_1 = I_{n_2}} \{ -\text{trace}(X_1^T (W_{11} + W_{13}Z)) \}. \quad (2.17)$$

Hence, the problem (2.16) is equivalent to determining the closest orthogonal matrix in the Frobenius norm to the matrix $W_{11} + W_{13}Z$. The solution is given by computing the singular value decomposition $P\Sigma Q^T$ of $W_{11} + W_{13}Z$ and setting $X_1 = PQ^T$; see [20, Theorem 4.1] for a proof of the latter statement. \square

The minimization problems (2.14) and (2.15) are solved similarly. This gives the eigenvector matrix in the spectral factorization (2.5).

Remark 2.6. We note that if $P\Sigma Q^T$ denotes the singular value decomposition of $W_{11} + W_{13}Z$, then we can express its polar decomposition by

$$W_{11} + W_{13}Z = (PQ^T)(Q\Sigma Q^T).$$

Since the first factor PQ^T is the minimizer of (2.17), the deviation of $Q\Sigma Q^T$ from the identity matrix measures the quality of the approximation.

Remark 2.7. If some of the nonzero eigenvalues of A in (2.10) are multiple, the corresponding columns of W_{11} , W_{21} , W_{13} , and W_{23} , are not uniquely determined. Anyway, when approximating $W_{11} + W_{13}Z$ by X_1 , and $W_{21} - W_{23}Z$ by Y , those columns contain linear combinations of the previous ones, and so they belong to the same space. Then, the approximations X_1 and Y will make factorization (2.9) valid.

3. A spectral bipartization method

We give here an outline of a spectral bipartization method, based on the above results. It exploits the spectral structure (2.5) of a bipartite graph to determine a node permutation that separates the two sets \mathcal{V}_1 and \mathcal{V}_2 , and to construct a bipartite approximation to a connected undirected graph \mathcal{G} , having a perturbed bipartite structure. The algorithm is exact whenever the input is the adjacency matrix of a bipartite graph, however it has to be considered “heuristic”, as we were not able to prove a complete convergence result for it, apart from the spectrum approximation theorems in Section 2.

There are three problems at hand: estimating the cardinality of the sets \mathcal{V}_1 and \mathcal{V}_2 , suitably ordering the nodes in \mathcal{G} , and, finally, approximating the adjacency matrix by a matrix of the form (2.1). Let A be the adjacency matrix of \mathcal{G} , and assume the spectral factorization

$$A = W\mathcal{D}W^T, \quad \mathcal{D} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n),$$

is available, where W is an orthogonal matrix and the eigenvalues are ordered by increasing absolute value.

1. The first step of our algorithm consists of finding the cardinality n_1 and n_2 of the two disjoint node sets \mathcal{V}_1 and \mathcal{V}_2 , unless they are known in advance. We do this by identifying the number of eigenvalues that are approximately zero. In principle, this could be done by detecting how many eigenvalues have absolute value larger than a fixed tolerance, but this process is extremely sensitive to the choice of the tolerance. In our numerical experimentation, we found it to be more reliable to detect the largest gap between “small” and “large” eigenvalues.

To do this, we compute the ratios

$$\rho_i = \frac{|\lambda_{i+1}|}{|\lambda_i|}, \quad i = 1, 2, \dots, n-1. \quad (3.1)$$

Then, for suitably chosen constants R and τ , we consider the index set

$$\mathcal{J} = \{i \in \{1, 2, \dots, n-1\} : \rho_i > R \text{ and } |\lambda_{i+1}| > \tau\}. \quad (3.2)$$

In our experiments, we set $R = 10^2$ and $\tau = 10^{-8}$. An index i is in \mathcal{J} if there is a significant gap between λ_i and λ_{i+1} ($\rho_i > R$), and λ_{i+1} is numerically nonzero ($|\lambda_{i+1}| > \tau$). If the set \mathcal{J} is empty, then we are not able to identify a partition of the nodes, and we consider the cardinality of the sets \mathcal{V}_1 and \mathcal{V}_2 to be the same. On the contrary, we let k be the index defined by

$$\rho_k = \max_{i \in \mathcal{J}} \rho_i,$$

and set

$$n_2 = \left\lceil \frac{n-k}{2} \right\rceil, \quad n_1 = n - n_2,$$

where $\lceil x \rceil$ denotes the closest integer to the real number x .

The above approach is clearly not completely robust. It is easy to trick it by constructing particular numerical examples, for example by letting C in (2.1) have singular values that decay to zero exponentially, or by introducing large gaps in the spectrum of the adjacency matrix. Nevertheless, we found the procedure quite accurate on networks stemming from real-world applications; see, e.g., Figs. 6 and 8 in Section 5.

In order to avoid overflow, it may be preferable to use the reciprocal ratios ρ_i^{-1} . This is not required in our Matlab implementation, given the features of the programming language.

2. The subsequent step is to find the sets \mathcal{V}_1 and \mathcal{V}_2 , and reorder the nodes. Assume that \mathcal{G} is bipartite, but that the adjacency matrix A corresponds to a random ordering of the nodes, so that

$$A = \Pi A_B \Pi^T,$$

for a permutation matrix Π and a matrix A_B of the form (2.1). In this case, the spectral factorization (2.4) becomes

$$A = (\Pi Q) \mathcal{D} (\Pi Q)^T,$$

i.e., the rows of the eigenvector matrix are permuted. In order to recover the structure of the eigenvectors, let us partition the eigenvector matrix as in

$$W := \Pi Q = [W_1 \quad W_2 \quad W_3], \quad (3.3)$$

with $W_1, W_3 \in \mathbb{R}^{n \times n_2}$ and $W_2 \in \mathbb{R}^{n \times (n_1 - n_2)}$.

Assume first that $n_1 > n_2 \geq 1$ and consider the matrix block W_2 . For (2.9) to be valid, the last n_2 rows of W_2 must vanish. Sorting in descending order the 1-norms of its rows concentrates the smallest entries in the lower block of W_2 . Applying the corresponding permutation σ to the rows of W brings this matrix to the form (2.9) and the adjacency matrix to the form (2.1), with the block C possibly permuted. When $n_1 = n_2$ the block W_2 is empty, so we consider the matrix $W_1 - W_3 Z$. As its first n_1 rows should be exactly zero, we sort the 1-norms of its rows in ascending order, and apply the obtained permutation σ to the rows of W . After the reordering, the first n_1 nodes are in the set \mathcal{V}_1 , and the remaining n_2 are in the set \mathcal{V}_2 . We note that applying the permutation σ to the rows and columns of the initial adjacency matrix A highlights the presence in the graph of an approximate bipartite structure.

3. To finally obtain an approximation of the matrix (2.1) by the computed spectral factorization, we first approximate the eigenvector matrix W_B by solving problem (2.12), and then approximate the eigenvalues in (2.10) by scalars that appear in \pm pairs using Proposition 2.4. Specifically, we let the α_j in the proposition be the eigenvalues (2.11). The β_j defined in the proposition are the eigenvalues of the matrix D in (2.5), in the same order.

Algorithm 1 Spectral bipartization algorithm.

Require: adjacency matrix A of size n , the user may optionally provide the cardinalities n_1 and n_2 of \mathcal{V}_1 and \mathcal{V}_2

Ensure: permutation σ which reorders the nodes, adjacency matrix A_B of the approximate bipartite graph

- 1: compute the spectral factorization $A = W \mathcal{D} W^T$, with $\lambda_1 \geq \dots \geq \lambda_n$
- 2: **if** n_1, n_2 are not provided **then**
- 3: sort the eigenvalues by increasing absolute value
- 4: compute $\rho_i, i = 1, \dots, n - 1$ by (3.1)
- 5: construct set \mathcal{J} by (3.2)
- 6: **if** $\mathcal{J} = \emptyset$ **then**
- 7: $n_1 = \lceil n/2 \rceil, n_2 = n - n_1$
- 8: **else**
- 9: $k = \arg \max_{i \in \mathcal{J}} \rho_i$
- 10: $n_2 = \lceil (n - k)/2 \rceil, n_1 = n - n_2$
- 11: **end if**
- 12: **end if**
- 13: **if** $n_1 > n_2$ **then**
- 14: find the permutation σ which sorts the 1-norms of the rows of W_2 decreasingly
- 15: **else**
- 16: find the permutation σ which sorts the 1-norms of the rows of $W_1 - W_3 Z$ increasingly
- 17: **end if**
- 18: apply the permutation σ to the rows of W
- 19: **if** $n_1 > n_2$ **then**
- 20: **Step 2 of the algorithm**
- 21: partition $W = [W_1 \quad W_2 \quad W_3]$ as in (3.3)
- 22: **Step 3 of the algorithm**
- 23: approximate the eigenvectors of A by minimizing (2.12)
- 24: approximate the eigenvalues of A by \pm pairs β_i , by Proposition 2.4
- 25: set $D = \text{diag}(\beta_1, \dots, \beta_n)$
- 26: construct the adjacency matrix $A_B = W D W^T$ of the bipartite graph

The above procedure, outlined in Algorithm 1, determines the eigenvectors and eigenvalues of a matrix A_B with the block structure

$$A_B = \begin{bmatrix} O & C \\ C^T & O \end{bmatrix}, \quad (3.4)$$

where the matrix C has real entries. The matrix A_B may have a different number of nonzero entries than A . In fact, not all nonzero entries may be positive. We can handle this issue in several ways:

- Allow A_B to be an adjacency matrix for a weighted graph with both positive and negative weights.
- Allow A_B to be an adjacency matrix for a weighted graph with positive weights. We achieve this by replacing the matrix C in (3.4) by the closest matrix, C_+ , in the Frobenius norm with nonnegative entries. The matrix C_+ is obtained from C by setting all negative entries to zero.
- Require A_B to represent an unweighted graph. The closest such matrix in the Frobenius norm to the matrix (3.4) is obtained by setting every entry of C to the closest member of the set $\{0, 1\}$.

The last procedure is the one adopted in the numerical experiments presented in Section 5.

Algorithm 1 can be applied only to small to medium sized problems, i.e., when it is possible to compute a full spectral factorization of A . For larger problems, one may reduce the complexity of the computation by renouncing the third step of the algorithm. Indeed, when $n_1 - n_2$ is not too large, a partial spectral factorization may lead to constructing a basis for the null space of A , that is, to obtaining the matrix W_2 . This would allow one to generate the permutation σ that takes the adjacency matrix to an almost bipartite form, identifying the two sets \mathcal{V}_1 and \mathcal{V}_2 .

4. Anti-communities

Let us consider a symmetric matrix A of size $n = n_1 + n_2$ with a zero leading square block of size n_1 . Then, A may be considered the adjacency matrix of a network with an anti-community of n_1 nodes. The matrix has the form

$$A = \begin{bmatrix} O_{n_1} & C \\ C^T & B \end{bmatrix}, \quad (4.1)$$

with C of size $n_1 \times n_2$ and B a square matrix of order n_2 . In the following, we denote by $\mathcal{N}(C)$ the null space of C , by $\mathcal{R}(C)$ its range, and by $B_{|\mathcal{N}(C)}$ the restriction of the submatrix B to $\mathcal{N}(C)$.

Theorem 4.1. *Let A be as in (4.1) and let $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$ be partitioned consistently with A . Then the equation*

$$A\mathbf{x} = \mathbf{0} \quad (4.2)$$

has $\nu = \dim \mathcal{N}(C^T)$ linearly independent solutions with $\mathbf{x}_2 = 0$. Moreover, if

$$d = \dim (\mathcal{R}(B_{|\mathcal{N}(C)}) \cap \mathcal{R}(C^T)) \geq 1,$$

then there are also d linearly independent solutions to $A\mathbf{x} = \mathbf{0}$ with $\mathbf{x}_2 \neq 0$, so that $\dim \mathcal{N}(A) = d + \nu$.

Proof. Let $k = \text{rank}(C)$ and consider the case $n_1 > n_2 = k$. Let us search for vectors \mathbf{x} such that $A\mathbf{x} = \mathbf{0}$. Then we have

$$A \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} O_{n_1} & C \\ C^T & B \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} C\mathbf{x}_2 \\ C^T\mathbf{x}_1 + B\mathbf{x}_2 \end{bmatrix}. \quad (4.3)$$

Since C is of full rank and $n_1 > n_2$, it follows from $C\mathbf{x}_2 = 0$ that $\mathbf{x}_2 = 0$ and, hence, $C^T\mathbf{x}_1 = 0$. The latter implies that \mathbf{x}_1 is in the null space of C^T , which has dimension $n_1 - n_2$. Thus, the matrix A admits the following linearly independent eigenvectors corresponding to the eigenvalue $\lambda = 0$,

$$\mathbf{x}^{(i)} = \begin{bmatrix} \mathbf{u}_{n_2+i} \\ \mathbf{0} \end{bmatrix}, \quad i = 1, 2, \dots, n_1 - n_2,$$

where \mathbf{u}_i , $i = 1, 2, \dots, n_1$, are the left singular vectors of C . Hence, $\lambda = 0$ has multiplicity $n_1 - n_2 = \dim \mathcal{N}(C^T)$.

Let us now assume that $k = n_1 < n_2$. Then A may or may not have zero eigenvalues. Indeed, for A to have a vanishing eigenvalue, the vector $\mathbf{x}_2 \in \mathbb{R}^{n_2}$ that appears in (4.3) has to belong to the null space of C , which has dimension $n_2 - n_1$. Then, there will be zero eigenvalues if and only if the system

$$C^T \mathbf{y} = -B\mathbf{x}_2$$

has a solution.

If instead $k = n_1 = n_2$, i.e., if C is nonsingular, then $\lambda = 0$ implies that both $\mathbf{x}_1 = 0$ and $\mathbf{x}_2 = 0$. Hence, $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = 0$, and all the eigenvalues of A are different from zero.

We finally turn to the case when the submatrix C is rank deficient, that is, $k < \min\{n_1, n_2\}$. The right-hand side of (4.3) is equivalent to

$$\mathbf{x}_2 \in \mathcal{N}(C), \quad C^T \mathbf{x}_1 = -B\mathbf{x}_2.$$

Let \mathbf{x} be a nontrivial solution of (4.2). When $\mathbf{x}_2 = \mathbf{0}$, there has to be a vector $\mathbf{x}_1 \neq \mathbf{0}$ with $C^T \mathbf{x}_1 = \mathbf{0}$. Since in this case the null space of C^T has dimension $n_1 - k$, there are $n_1 - k$ linearly independent solutions of (4.2) with $\mathbf{x}_2 = \mathbf{0}$.

The existence of a solution \mathbf{x} of (4.2) with a nonzero subvector \mathbf{x}_2 is equivalent to

$$\dim(\mathcal{R}(B_{|\mathcal{N}(C)}) \cap \mathcal{R}(C^T)) \geq 1.$$

This condition does not hold for most matrix pairs (B, C) . \square

Remark 4.2. We note that if $B = 0$, then the equation $A\mathbf{x} = \mathbf{0}$ has exactly

$$\dim \mathcal{N}(C) + \dim \mathcal{N}(C^T) = n - 2 \operatorname{rank}(C)$$

linearly independent solutions.

Theorem 4.1 shows that if a network has a large anti-community ($n_1 > n_2$), the spectral decomposition $A = WDW^T$ has the form

$$W = \begin{bmatrix} E & U_2 & F \\ G & O_{n_2, n_1 - n_2} & H \end{bmatrix}, \quad D = \begin{bmatrix} D_1 & O & O \\ O & O_{n_1 - n_2} & O \\ O & O & D_2 \end{bmatrix}.$$

The structures of W and D are very similar to those of W_B and Λ_B in (2.9), respectively. For this reason, the bipartization algorithm described in Section 3, is able to detect the presence of a large anti-community and to order the nodes so that the adjacency matrix takes the form (4.1). In case a group of nodes is only approximately an anti-community, the algorithm produces an adjacency matrix that approximates (4.1).

To summarize, when $n_1 > n_2$, if a network is either bipartite or contains a large anti-community, its adjacency matrix has zero eigenvalues; the converse is not true. If A has a multiple zero eigenvalue, then we can recognize the presence of one of the two above cases by observing the structure of the eigenvector matrix.

5. Computed examples

In the following numerical experiments, we fix the integers n_1 and n_2 , and construct a random matrix A of the form (2.1), with a sparse block C with density ξ . The matrix is first perturbed, by replacing its (1,1) and (2,2) blocks by sparse matrices of appropriate size and density η , and then “scrambled”, by applying the same random permutation to its rows and columns.

We apply the algorithm of Section 2 to the matrix A either by supplying the cardinality of the two sets \mathcal{V}_1 and \mathcal{V}_2 (this approach is referred to as `specbip-n`), or letting the method estimate n_1 and n_2 from the data; we refer to the latter approach as `specbip`. Since the block (1,2) of the matrix returned by the method is generally permuted with respect to the initial test matrix, the rows and columns are reordered according to the original sequence of the nodes. The final reordering allows us to compare the resulting matrix A_B to the test matrix A .

Our results are compared to the ones obtained by red-black ordering using the MatlabBGL library [21], a Matlab package implementing graph algorithms. A matrix has a red-black ordering if the corresponding graph is bipartite. To find a bipartite ordering, this software uses a breadth first search algorithm, starting from an arbitrary vertex. The partition of the nodes is determined by forming a group containing all the vertices having even distance from the root, and another group with the vertices at odd distance from the root. This procedure is designed to bipartite networks, not to produce an approximation when the bipartization is not exact.

Fig. 1 displays the results for a test matrix with $(n_1, n_2) = (512, 256)$, sparsity $\xi = 10^{-2}$, and perturbation $\eta = 10^{-4}$. In particular, it reports in the upper row a spy plot of the original test matrix, the perturbed version, with random arcs in the (1,1) and (2,2) blocks, and the permuted matrix that is fed to the bipartization methods. The bottom row shows the reconstructed networks. The `specbip-n` approach, which receives the information about the cardinality of the node sets, produces the matrix closest to the original. The general algorithm estimates the cardinalities $(\tilde{n}_1, \tilde{n}_2) = (492, 276)$, according to the number of “small” eigenvalues; see **Fig. 2**, where the absolute values of the eigenvalues are displayed in nondecreasing order. This algorithm produces a slightly less accurate approximation than the previous one, which is anyway much better than the matrix produced by the red-black ordering.

Fig. 3 shows the results for a test matrix similar to the previous one, but with a larger perturbation $\eta = 10^{-3}$. The estimation of (n_1, n_2) is inaccurate, but the approximation produced by the `specbip` methods is quite close to the unperturbed matrix, while the red-black ordering matrix is far from it.

Now, let

$$E = A - A_B = \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix},$$

where E_{11} and E_{22} are square matrices of size n_1 and n_2 , respectively, and let $|M|$ denote the number of nonzero elements of M . To evaluate the quality of the results, we consider the following three indices

$$\mathcal{I}_B = 1 - b_s, \quad \mathcal{E}_B = \frac{|E_{11}|}{n_1^2} + \frac{|E_{22}|}{n_2^2}, \quad \mathcal{E}_A = \frac{|E_{12}|}{|C|}.$$

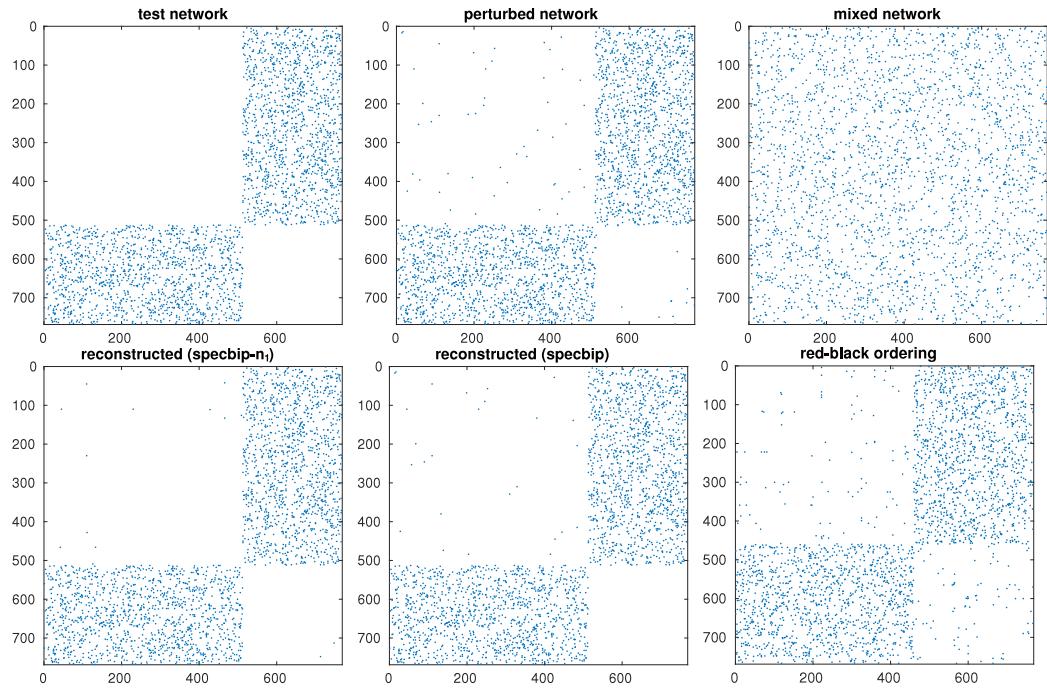


Fig. 1. $(n_1, n_2) = (512, 256)$, $(\tilde{n}_1, \tilde{n}_2) = (492, 276)$, $\xi = 10^{-2}$, $\eta = 10^{-4}$.

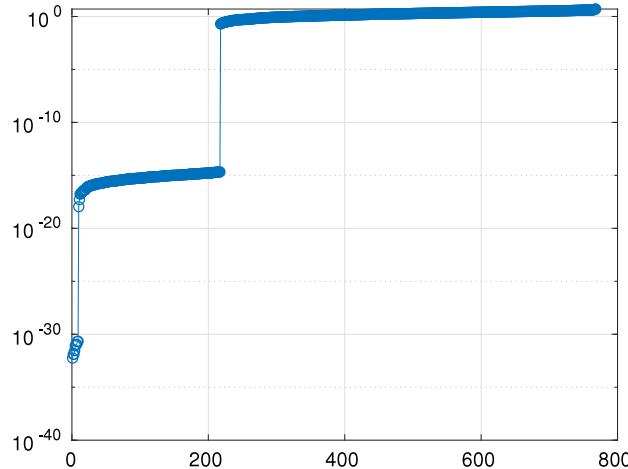


Fig. 2. $(n_1, n_2) = (512, 256)$, $(\tilde{n}_1, \tilde{n}_2) = (492, 276)$, $\xi = 10^{-2}$, $\eta = 10^{-4}$.

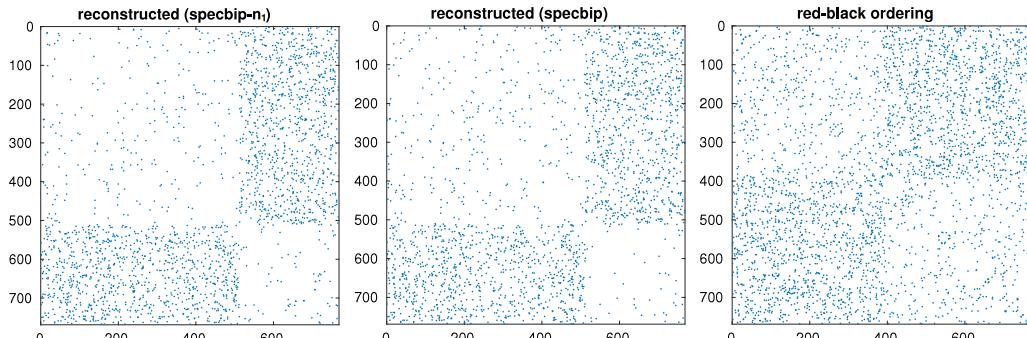


Fig. 3. $(n_1, n_2) = (512, 256)$, $(\tilde{n}_1, \tilde{n}_2) = (396, 372)$, $\xi = 10^{-2}$, $\eta = 10^{-3}$.

Table 1
Results for $\xi = 10^{-2}$, $\eta = 10^{-4}$.

(256,128)	specbip- n_1	specbip	Red-black
\mathcal{I}_B	1.22e-16	1.89e-16	2.33e-03
\mathcal{E}_B	5.46e-04	6.74e-04	3.72e-03
\mathcal{E}_A	2.80e-01	2.79e-01	–
\mathcal{E}_N	1.45e-01	1.58e-01	2.76e-01
T	4.94e-02	5.05e-02	3.15e-04
(512,256)	specbip- n_1	specbip	Red-black
\mathcal{I}_B	1.11e-17	1.11e-17	2.98e-03
\mathcal{E}_B	1.13e-04	1.50e-04	3.39e-03
\mathcal{E}_A	4.84e-02	6.27e-02	–
\mathcal{E}_N	3.36e-02	5.96e-02	2.97e-01
T	2.77e-01	2.95e-01	4.94e-04
(1024,512)	specbip- n_1	specbip	Red-black
\mathcal{I}_B	7.77e-17	0.00e+00	4.17e-02
\mathcal{E}_B	9.92e-05	2.11e-04	4.75e-03
\mathcal{E}_A	1.06e-01	1.80e-01	–
\mathcal{E}_N	3.62e-02	1.15e-01	2.75e-01
T	1.92e+00	1.94e+00	8.67e-04

Table 2
Results for $\xi = 10^{-2}$, $\eta = 10^{-5}$.

(256,128)	specbip- n_1	specbip	Red-black
\mathcal{I}_B	1.11e-17	7.77e-17	1.68e-06
\mathcal{E}_B	6.68e-04	8.79e-04	3.36e-03
\mathcal{E}_A	2.70e-01	2.68e-01	–
\mathcal{E}_N	1.23e-01	1.49e-01	2.58e-01
T	4.39e-02	4.70e-02	1.01e-03
(512,256)	specbip- n_1	specbip	Red-black
\mathcal{I}_B	0.00e+00	2.22e-17	1.40e-04
\mathcal{E}_B	3.05e-05	1.91e-05	8.81e-04
\mathcal{E}_A	3.88e-02	2.38e-02	–
\mathcal{E}_N	1.87e-02	1.93e-02	3.16e-01
T	2.72e-01	2.77e-01	5.12e-04
(1024,512)	specbip- n_1	specbip	Red-black
\mathcal{I}_B	0.00e+00	0.00e+00	4.04e-03
\mathcal{E}_B	1.91e-07	1.03e-05	1.07e-03
\mathcal{E}_A	1.73e-04	9.49e-03	–
\mathcal{E}_N	9.77e-05	9.47e-03	3.25e-01
T	1.91e+00	1.89e+00	9.52e-04

The first two indices measure the distance of A_B from the adjacency matrix of a bipartite graph; see (1.1) for the definition of b_s . The third index measures the approximation error with respect to the starting bipartite network (2.1). To better evaluate the error in the bipartition, we introduce the fourth index $\mathcal{E}_N = \tilde{\mathcal{E}}_N/n_1$, where $\tilde{\mathcal{E}}_N$ is the number of nodes from the set \mathcal{V}_1 that were incorrectly ascribed to the set \mathcal{V}_2 .

Tables 1, 2, and 3 report the average values of the above four quality indices over 10 realizations of the random test networks. Three different pairs (n_1, n_2) are considered; each table refers to different densities ξ and η ; T stands for the execution time in seconds.

A comparison of the tables shows that the spectral bipartization algorithm is always more accurate than the red-black ordering method. At the same time, it is much slower than the MatlabBGL function, as in our experiments we compute the whole spectrum of the adjacency matrix, without exploiting its sparsity. To be competitive with existing methods for large-scale problems, the spectral method should be modified in order to perform its task by suitable iterative methods, in order to take advantage of the structure of the adjacency matrix.

From the tables, it can also be observed that knowing in advance the cardinality of the two sets \mathcal{V}_1 and \mathcal{V}_2 leads in some cases to a substantial improvement in the quality of the results.

To further investigate the behavior of the bipartition error, we construct a matrix A of the form (2.1), letting $n_1 = 512$ and $n_2 = 256$, with a sparse random block C having density $\xi = 10^{-2}$. After randomly permuting the rows and columns, we apply our algorithms to this matrix, as well as to those perturbed by replacing the (1,1) and (2,2) blocks by a sparse matrix with density $\eta = 10^{-6}, 10^{-5}, \dots, 10^{-3}$. The graph on the left of Fig. 4 shows the value of the bipartition error \mathcal{E}_N obtained when the three methods are applied to an unweighted graph, the one on the right corresponds to a weighted graph. All values are averaged over 10 realizations of the random matrices. Both graphs show that the bipartition

Table 3
Results for $\xi = 10^{-1}$, $\eta = 10^{-4}$.

(256,128)	specbip- n_1	specbip	Red-black
\mathcal{I}_B	0.00e+00	0.00e+00	7.71e-02
\mathcal{E}_B	0.00e+00	5.83e-04	1.35e-02
\mathcal{E}_A	2.43e-02	4.24e-02	–
\mathcal{E}_N	0.00e+00	2.58e-02	3.18e-01
T	5.56e-02	6.05e-02	3.07e-03
(512,256)	specbip- n_1	specbip	Red-black
\mathcal{I}_B	0.00e+00	0.00e+00	1.44e-01
\mathcal{E}_B	0.00e+00	8.19e-04	8.01e-03
\mathcal{E}_A	8.02e-03	5.19e-02	–
\mathcal{E}_N	0.00e+00	4.47e-02	3.31e-01
T	2.77e-01	2.76e-01	1.08e-03
(1024,512)	specbip- n_1	specbip	Red-black
\mathcal{I}_B	0.00e+00	0.00e+00	2.60e-01
\mathcal{E}_B	0.00e+00	1.04e-03	6.54e-03
\mathcal{E}_A	2.33e-03	9.04e-02	–
\mathcal{E}_N	0.00e+00	8.71e-02	3.28e-01
T	2.02e+00	2.07e+00	3.99e-03

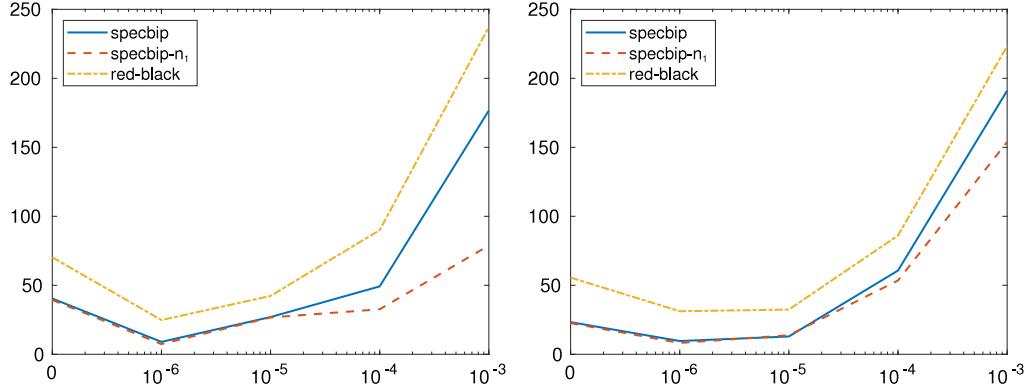


Fig. 4. Bipartition error $\tilde{\mathcal{E}}_N$ for $(n_1, n_2) = (512, 256)$; on the left unweighted random graphs, on the right weighted random graphs, both with $\xi = 10^{-2}$, as a function of $\eta = 0, 10^{-6}, 10^{-5}, \dots, 10^{-3}$.

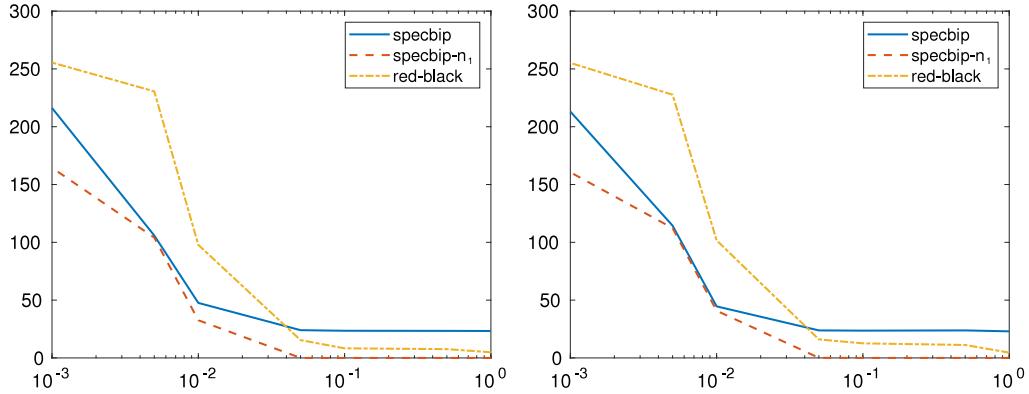


Fig. 5. Bipartition error $\tilde{\mathcal{E}}_N$ for $(n_1, n_2) = (512, 256)$; on the left unweighted random graphs, on the right weighted random graphs, both with $\eta = 10^{-2}$, as a function of $\xi = 10^{-3}, 10^{-2}, 10^{-1}, 1$.

determined by our approaches is closer to the correct one, with respect to red-black ordering, with $\text{specbip-}n_1$ producing slightly better results. The performance of all algorithms degrades as the perturbation becomes less sparse.

In Fig. 5, we display the value of \mathcal{E}_N for the same examples, for a fixed $\eta = 10^{-2}$, and letting the density ξ of the block C take values in $[10^{-3}, 1]$. The red-black ordering method is more accurate than the specbip algorithm for very sparse networks, while providing the correct cardinality of the set \mathcal{V}_1 to $\text{specbip-}n_1$ produces the best results.

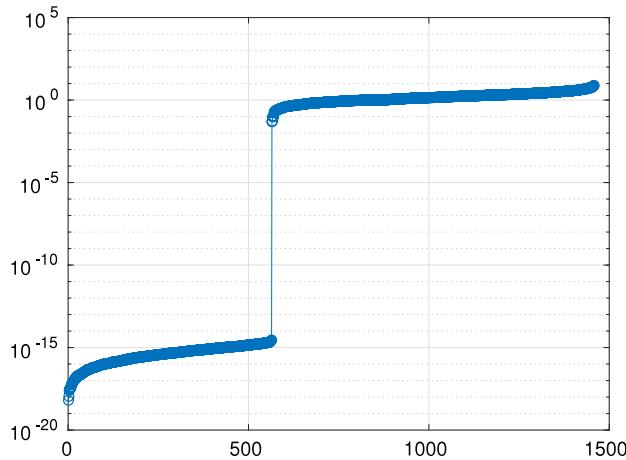


Fig. 6. Spectrum of the reduced adjacency matrix for the *NDyeast* network.

5.1. The *NDyeast* network

We illustrate the performance of the spectral bipartization algorithm when applied to the detection of anti-communities by analyzing a case study. The *NDyeast* network describes the protein interaction network for yeast, each edge representing an interaction between two proteins [22]. The data set is available at [23]. In this section we analyze this network, testing the presence of a bipartization or of a large anti-community.

The *NDyeast* network has 2114 nodes. There are 74 self-loops (nodes connected only to themselves) and 268 nodes disconnected from the network. The adjacency matrix resulting by removing both the self-loops and the isolated nodes has size $n = 1846$, and it has 149 connected components. They were identified by the `getconcomp` function from the PQser Matlab toolbox [24].

In the case of a reducible adjacency matrix, the spectral bipartization algorithm should treat each single connected component one at a time. Since most of the components in the *NDyeast* network are very small, often just 2 or 3 nodes, we consider the only component with more than 10 nodes, which has 1458 nodes. We process the reduced adjacency matrix A with our bipartization method.

The algorithm determines $n_0 = 564$ zero eigenvalues (see Fig. 6) and identifies two sets of nodes with cardinalities $n_1 = 1011$ and $n_2 = 447$.

The starting adjacency matrix is displayed in the top-left spy plot of Fig. 7. The top-right plot shows the same matrix after the ordering produced by the spectral bipartization algorithm is applied to its rows and columns. This graph clearly displays that there is a large group of nodes in the *NDyeast* network that do not communicate much among themselves, that is, an anti-community. In the same graph we show the bipartization detected by the algorithm by means of red lines.

Our algorithm can also be applied by supplying the values of (n_1, n_2) , rather than estimating them from the number of zero eigenvalues. If we do this by setting $\tilde{n}_1 = 800$ and $\tilde{n}_2 = 658$, we obtain the bottom left graph in the same figure. It shows that in the group of the first 800 proteins, only four of them directly interact.

The bottom-right graph of Fig. 7 displays the result of the red–black ordering method, which does not supply any useful information.

We remark that a data set similar to *NDyeast* (but different) is available at [23]. It is called *simply yeast*, it consists of 2361 nodes, and it refers to the paper [25]. By processing this data set with our spectral algorithm, we obtain results very similar to the ones displayed in Fig. 7.

5.2. The *geom* network

We also applied the spectral bipartization algorithm to a weighted graph, namely, the *geom* network. It is extracted from the Computational Geometry Database *geombib* by B. Jones (version 2002). Nodes represent authors; the value of the entry (i, j) of the adjacency matrix is the number of papers coauthored by authors i and j . The data set is available at [23].

The *geom* network has 7343 nodes and 11898 edges. After removing 1185 isolated nodes, the network presents 875 connected components, the largest of which has 3621 nodes. We applied the bipartization method to the adjacency matrix associated to this component.

The eigenvalues are displayed in Fig. 8: 533 of them are detected as being numerically zero, and the cardinalities of the two node sets are $n_1 = 2077$ and $n_2 = 1544$. The left graph of Fig. 9 reports the spy plot of the original adjacency matrix; the graph on the right shows the matrix reordered by the spectral bipartization algorithm. The graph highlights the presence of an anti-community of about 1000 authors, who did not collaborate with each other when writing papers.

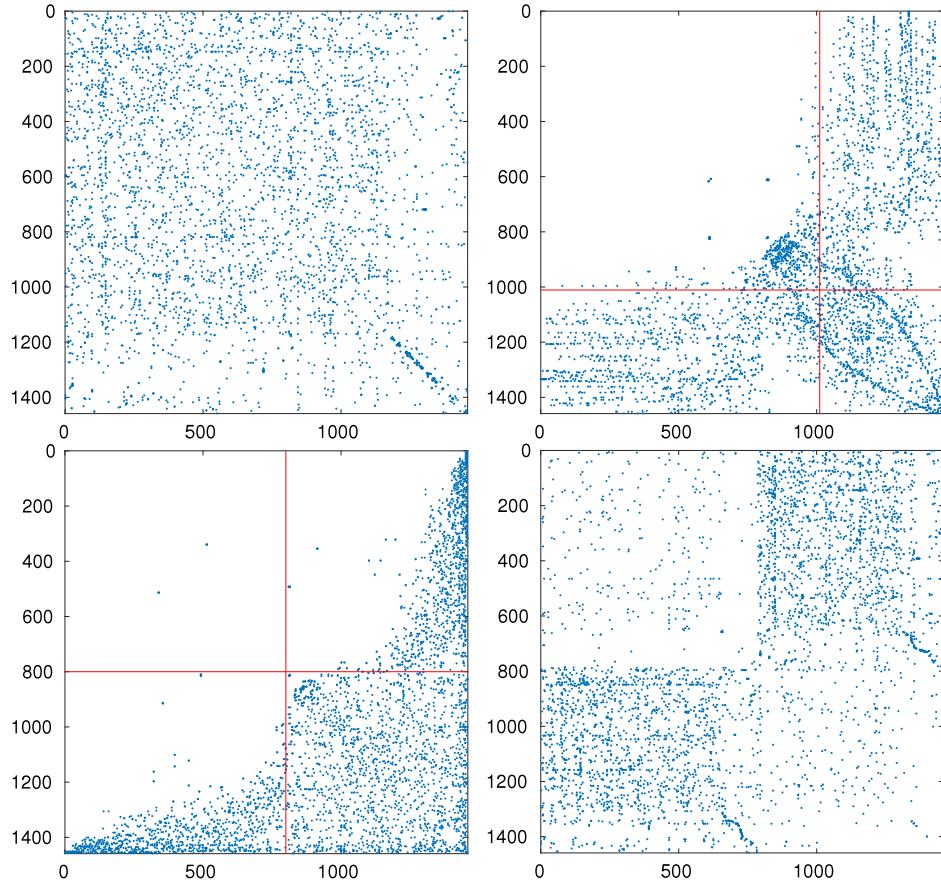


Fig. 7. Analysis of the *NDyeast* network: top-left, the starting adjacency matrix; top-right, the node reordering produced by the spectral algorithm; bottom-left, the reordering induced by the choice $(\tilde{n}_1, \tilde{n}_2) = (800, 658)$; bottom-right, the output of the red–black ordering method.

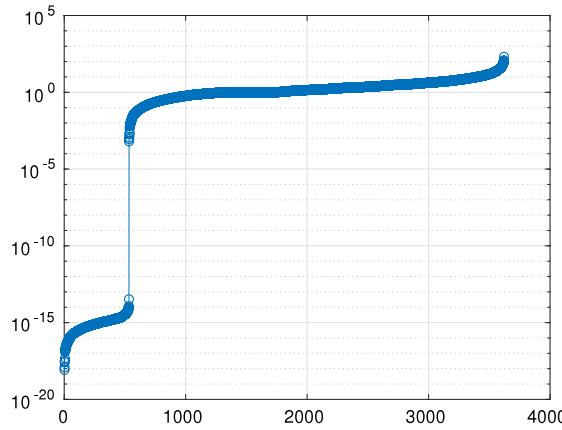


Fig. 8. Spectrum of the reduced adjacency matrix for the *geom* network.

6. Conclusion

This paper describes how an approximate bipartition of a given graph can be determined by solving a sequence of simple optimization problems. The technique can also be applied to detect anti-communities. Computed examples illustrate the performance of the method described.

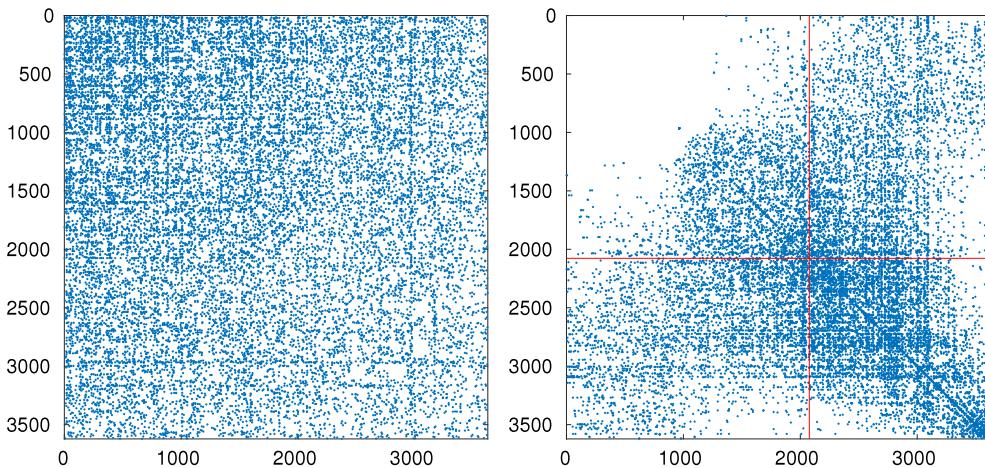


Fig. 9. Analysis of the *geom* network: on the left, the starting adjacency matrix; on the right, the node reordering produced by the spectral algorithm.

Acknowledgments

The authors would like to thank the referees for comments.

References

- [1] E. Estrada, *The Structure of Complex Networks: Theory and Applications*, Oxford University Press, Oxford, 2011.
- [2] M.E.J. Newman, *Networks: An Introduction*, Oxford University Press, Oxford, 2010.
- [3] J.A. Bondy, U.S.R. Murty, *Graph Theory with Applications*, Macmillan, London, 1976.
- [4] M. Yannakakis, Edge-deletion problems, *SIAM J. Comput.* 10 (1981) 297–309.
- [5] E. Estrada, J. Gómez-Gardeñes, Network bipartivity and the transportation efficiency of European passenger airlines, *Physica D* 323–324 (2016) 57–63.
- [6] R. Roth, On the eigenvectors belonging to the minimum eigenvalue of an essentially nonnegative symmetric matrix with bipartite graph, *Linear Algebra Appl.* 118 (1989) 1–10.
- [7] J.L. Morrison, R. Breitling, D.J. Higham, D.R. Gilbert, A lock-and-key model for protein-protein interactions, *Bioinformatics* 2 (2006) 2012–2019.
- [8] A. Taylor, J.K. Vass, D.J. Higham, Discovering bipartite substructure in directed networks, *LMS J. Comput. Math.* 14 (2011) 72–86.
- [9] E. Estrada, P.A. Knight, *A First Course in Network Theory*, Oxford University Press, Oxford, 2015.
- [10] D. Fasino, F. Tudisco, A modularity based spectral method for simultaneous community and anti-community detection, *Linear Algebra Appl.* 542 (2017) 605–623.
- [11] L. Chen, Q. Yu, B. Chen, Anti-modularity and anti-community detecting in complex networks, *Inform. Sci.* 275 (2014) 293–313.
- [12] J. Leskovec, K.J. Lang, A. Dasgupta, M.W. Mahoney, Statistical properties of community structure in large social and information networks, in: Proc. 17th Int. Conf. World Wide Web (WWW'08), 2008, pp. 695–704.
- [13] G. Palla, I. Derènyi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, *Nature* 435 (2005) 814–818.
- [14] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, *Phys. Rev. E* 76 (2007) 036106.
- [15] B. Yang, W.K. Cheung, J.M. Liu, Community mining from signed social networks, *IEEE Trans. Knowl. Data Eng.* 19 (2007) 1333–1348.
- [16] S.P. Borgatti, M.G. Everett, Models of core/periphery structures, *Social Networks* 21 (1999) 375–395.
- [17] M.P. Rombach, M.A. Porter, J.H. Fowler, P.J. Mucha, Core–periphery structure in networks, *SIAM Rev.* 59 (2017) 619–646.
- [18] R.B. Bapat, *Graphs and Matrices*, Springer, London, 2010.
- [19] G.H. Golub, C.F. Van Loan, *Matrix Computations*, fourth ed., Johns Hopkins University Press, Baltimore, 2013.
- [20] N. Higham, *Matrix nearness problems and applications*, in: M.J.C. Gover, S. Barnett (Eds.), *Applications of Matrix Theory*, Oxford University Press, Oxford, 1989, pp. 1–27.
- [21] D. Gleich, MatlabBGL - A Matlab Graph Library, https://www.cs.purdue.edu/homes/dgleich/packages/matlab_bgl/.
- [22] H. Jeong, S. Mason, A.-L. Barabási, Z.N. Oltvai, Lethality and centrality in protein networks, *Nature* 411 (2001) 41–42.
- [23] V. Batagelj, A. Mrvar, Pajek data sets (2006). Available at <http://vlado.fmf.uni-lj.si/pub/networks/data/>.
- [24] A. Concas, C. Fenu, G. Rodriguez, PQser: A Matlab package for spectral seriation, *Numer. Algorithms* 80 (2019) 879–902.
- [25] S. Sun, L. Ling, N. Zhang, G. Li, R. Chen, Topological structure analysis of the protein-protein interaction network in budding yeast, *Nucleic Acids Res.* 31 (2003) 2443–2450.