CHANGING THE RANKING IN EIGENVECTOR CENTRALITY OF A WEIGHTED GRAPH BY SMALL PERTURBATIONS

MICHELE BENZI AND NICOLA GUGLIELMI

Abstract. In this article, we consider eigenvector centrality for the nodes of a graph and study the robustness (and stability) of this popular centrality measure. For a given weighted graph $\mathcal{G}$ (both directed and undirected), we consider the associated weighted adjacency matrix A, which by definition is a non-negative matrix. The eigenvector centralities of the nodes of $\mathcal{G}$ are the entries of the Perron eigenvector of A, which is the (positive) eigenvector associated with the eigenvalue with largest modulus. They provide a ranking of the nodes according to the corresponding centralities. An indicator of the robustness of eigenvector centrality consists in looking for a nearby perturbed graph $\widetilde{\mathcal{G}}$, with the same structure as $\mathcal{G}$ (i.e., with the same vertices and edges), but with a weighted adjacency matrix $\widetilde{A}$ such that the highest m entries ($m \geq 2$) of the Perron eigenvector of $\widetilde{A}$ coalesce, making the ranking at the highest level ambiguous. To compute a solution to this matrix nearness problem, a nested iterative algorithm is proposed that makes use of a constrained gradient system of matrix differential equations in the inner iteration and a one-dimensional optimization of the perturbation size in the outer iteration.

The proposed algorithm produces the *optimal* perturbation (i.e., the one with smallest Frobenius norm) of the A which causes the looked-for coalescence, which is a measure of the sensitivity of the graph. As a by-product of our methodology we derive an explicit formula for the gradient of a suitable functional corresponding to the minimization problem, leading to a characterization of the stationary points of the associated gradient system. This in turn gives us important properties of the solutions, revealing an underlying low rank structure. Our numerical experiments indicate that the proposed strategy outperforms more standard approaches based on algorithms for constrained optimization. The methodology is formulated in terms of graphs but applies to any nonnegative matrix, with potential applications in fields like population models, consensus dynamics, economics, etc.

1991 *Mathematics Subject Classification.* Primary 15A60, 05C50, 15A42; Secondary 47A55.

Key words and phrases. Eigenvector centrality; matrix nearness problem; structured eigenvalue optimization.

Michele Benzi's work was supported in part by MUR (Ministero dell' Università e della Ricerca) through the PNRR MUR Project PE0000013-FAIR and through the PRIN Project 20227PCCKZ ("Low Rank Structures and Numerical Methods in Matrix and Tensor Computations and their Applications").

Nicola Guglielmi acknowledges support from the Italian MUR within the PRIN 2022 Project "Advanced Numerical Methods for Time Dependent Parametric Partial Differential Equations with Applications". He also thanks the support from the MIUR program for Departments of Excellence (project title of GSSI: Pattern analysis and engineering).

Both authors acknowledge the MUR Pro3 joint project entitled "STANDS: Stability of Neural Dynamical Systems". The authors are also affiliated to the INdAM-GNCS (Gruppo Nazionale di Calcolo Scientifico).

1. Introduction

In many fields, such as biology, social sciences, statistics, medicine and several others, ranking entities in a large data set is an important task. This leads to the concept of centrality measure for a network or a graph, which is the subject of this paper. High-rank (centrality) of a node in a graph indicates the importance of the node. Several centrality measures have been proposed in the literature; for a broad class of walk-based centrality measures see, e.g, [6, 4, 30]); here we consider eigencentrality, i.e., the one based on the Perron eigenvector, which was first introduced by Landau [27] and rediscovered by several authors since. For a strongly connected graph $\mathcal{G}$ with associated (possibly weighted) adjacency matrix A, this consists in ranking the nodes according to the entries of the Perron eigenvector of A, which is the positive eigenvector associated to the eigenvalue $\lambda > 0$ of maximum modulus. As is well known, one the most important ranking schemes, the PageRank algorithm [7, 28], is based on a variant of this idea.

Given a directed or undirected weighted graph, we are interested in understanding how small perturbations may affect the ranking and which edges are more sensitive in order to produce certain changes in the ranking. We can distinguish two main motivations.

(i) Perturbing the weights as a robustness tool, or sensitivity analysis. Motivation: studying the reliability of the ranking in presence of uncertainties or variations in time of the weights, which in real networks are never known exactly.

(ii) Assessing the vulnerability of a network in the presence of deliberate attempts, for example by malicious, disgruntled individuals or by other interested agents, to manipulate the rankings by slightly perturbing the weights. E.g, in PageRank, one might want to manipulate the network in order to affect the rank of a product or of a political candidate.

A robustness indicator is useful in assessing the reliability of the measure. A very cheap and intuitively reasonable indicator of robustness consists of measuring the distance between the first and the m-th largest entries of the (suitably normalized) ranking eigenvector. However such a measure may be misleading, which means that even when this distance is small, this does not imply that the ranking is not robust, because - due to the structure of the graph - even a large modification of the weights of the graph might not be sufficient to annihilate the considered distance. For this reason we pose the problem in different terms and look for a closest graph $\widetilde{\mathcal{G}}$ to $\mathcal{G}$, which has in common with $\mathcal{G}$ the same sets of vertices and edges, but different weights, such that the largest entries of the Perron eigenvector coalesce. This problem translates into the search of a perturbed weighted adjacency matrix $\widetilde{A}$ to A which preserves the sparsity pattern of A and certain of its properties, and is characterized by coalescence of the first m entries ($m \geq 2$) of the leading eigenvector.

In this way we define a structured robustness measure, whose computation will be discussed in this paper, together with the computation of the closest altered graph. It may be computationally expensive to be used as a general-purpose robustness indicator in comparison with simpler heuristic criteria, which are available without further computations, such as the above mentioned difference between the largest and m-th largest entry in the Perron eigenvector of A. But it has the advantage of

a higher reliability. Moreover, cheap measures are not able to provide the nearby graph with ambiguous or changed ranking, which is a very important byproduct of the methodology we propose. The quantity considered in this paper is related to other structured robustness measures that arise in matrix and control theory, such as stability radii, passivity distances, distance to singularity, etc. [23, 21, 22, 2, 15, 17, 26].

A main objective of this paper is to show how the robustness measure can be computed through the explicit computation of a nearby graph with ambiguous ranking. The proposed algorithm is an iterative algorithm, where in each step the leading eigenvector of the weighted adjacency matrix of a graph with perturbed weights is computed. For large sparse graphs with n vertices and $O(n)$ edges, these computations can typically be done with a more favorable complexity in the number of vertices with respect to a general purpose eigensolver. A feature of this algorithm in common with recent algorithms for eigenvalue optimization (such as the ones described in [15, 17, 1]) is a two-level methodology for matrix nearness problems, where in an inner iteration a gradient flow drives perturbations of a fixed norm to the original matrix into a (local) minimum of a nonnegative functional that depends on eigenvalues and/or eigenvectors, while in an outer iteration the perturbation size is optimized such that the functional becomes zero.

The paper is organized as follows. In Section 2 we briefly mention recent related work by other authors working on similar centrality manipulation problems. In Section 3 we recall basic concepts of eigenvector centrality measures. In Subsection 3.1 we introduce the radius of robustness (ϱ_m) for eigenvector centrality as the distance to the closest graph with coalescent highest m entries in the associated Perron eigenvector. In Subsection 3.3 we formulate a methodology to compute the ϱ_m and the nearby ambiguous graph. This requires the solution of a structured matrix nearness problem. For this we propose a two-level iterative method, which is based on a gradient system of matrix differential equations in an inner iteration and a one-dimensional optimization of the perturbation size in an outer iteration, and we discuss algorithmic aspects. In Section 4 we provide the necessary theoretical background on first order perturbation theory of eigenvalues and eigenvectors. In Section 5 we describe the complete computational procedure to compute the radius of robustness. In Section 7 we present the results of numerical experiments where the robustness measure discussed here is computed for some examples of graphs. Finally, in Section 8 we draw our conclusions.

2. Related work

Related problems have been considered in the literature by several authors. Most relevant to our work are the papers [8, 12] and [31]; see also the references therein.

In [12] the authors consider the problem of perturbing a given stochastic matrix so that it has a prescribed stationary probability distribution, with the perturbation required to be minimal with respect to the matrix 1-norm and to preserve the sparsity pattern of the original matrix (besides of course, being such that the perturbed matrix remains stochastic). The solution is obtained by formulating the problem as a constrained linear optimization problem which is then solved by a suitable algorithm.

In [8], the authors consider two popular measures of centrality, Katz and PageRank (see, respectively, [30] and [28]). Their goal is to find the "minimal" perturbation (according to a certain matrix functional) of the weights that enforce a prescribed vector of centrality scores, while at the same time preserving the sparsity pattern. In particular, the problem of coalescing the top m entries could be treated by the approach in [8], by enforcing a ranking vector in which the top m entries take the same, prescribed value. On the other hand, PageRank and especially Katz centrality are not exactly the same as eigenvector centrality and, perhaps more important, the optimality measure used in [8] is not the Frobenius norm but a weighted sum of the (squared) Frobenius norm and the matrix 1-norm. Also, the approach in [8], based on quadratic programming, is quite different from the one we propose. Nevertheless, also in view of the close relationship existing between Katz and eigenvector centrality (see [6]), it is possible that the approach in [8] may be applicable, with some modifications, to the solution of the problem considered in the present paper.

Finally, we mention the interesting paper [31], which studies the broader problem of identifying sets of nodes in a network that can be used to "control" the eigenvector centrality scores of the remaining nodes by manipulating the weights of the corresponding outgoing links. In this paper it is shown how to identify minimal sets of nodes, called "controller nodes," which can be used to obtain the desired eigenvector centrality ranking. This is clearly a different problem than the one considered in this paper, where we seek the perturbation of minimum Frobenius norm, but the approach in [31] may be combined with the one in our paper, since our method allows for the possibility of constraining the perturbations to a subset of the edges in the network (see the example in Section 6.1).

3. Robustness and vulnerability of eigenvector centrality

Let

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$$

be a weighted graph (either directed or undirected) with vertex set $\mathcal{V} = \{v_1, \ldots, v_n\}$, edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ and weighted adjacency matrix A. To each edge $(i, j) \in \mathcal{E}$ (from node v_j to node v_i: note the order) is associated a positive weight $a_{ij} > 0$; of course, for an undirected graph we have $a_{ji} = a_{ij}$ for all i, j. We set $a_{ij} = 0$ for $(i, j) \notin \mathcal{E}$. The weighted adjacency matrix of the graph is the non-negative matrix

$$A = (a_{ij}) \in \mathbb{R}^{n \times n}.$$

When G is undirected, A is symmetric. To avoid a dependence of the measure on the scaling of the weights, we assume that A is normalized with respect to the Frobenius norm,

$$\|A\| = \|A\|_F = 1 \tag{1}$$

where in this article - unless specified differently - $\|\cdot\| = \|\cdot\|_F$.

Given a graph $\widetilde{\mathcal{G}} = (\mathcal{V}, \mathcal{E}, \widetilde{A})$, with the same vertices and edges of $\mathcal{G}$, we define the distance between $\mathcal{G}$ and $\widetilde{\mathcal{G}}$ by

$$\operatorname{dist}\big(\mathcal{G}, \widetilde{\mathcal{G}}\big) = \sqrt{\sum_{(i,j)\in\mathcal{E}} (a_{ij} - \widetilde{a}_{ij})^2} = \|A - \widetilde{A}\|. \tag{2}$$

We order the eigenvalues $\{\lambda_i\}$ of A by non-increasing modulus so that λ_1 is the Perron eigenvalue of A, which is non-negative and equal to $\rho(A)$ (the spectral radius of A).

Assumption 3.1. *In this article we assume that the graph is strongly connected, which is equivalent to the irreducibility of A.*

The following theorem is classic (see e.g. [24]).

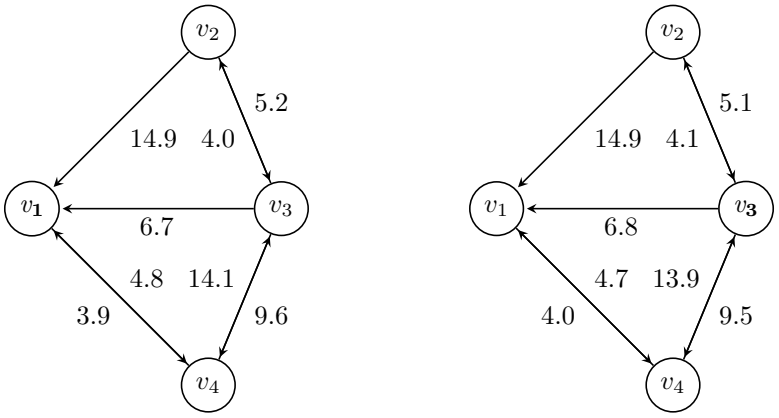
Theorem 3.1 (Perron–Frobenius)**.** *If $A \in \mathbb{R}^{n\times n}$ is nonnegative and irreducible then*

(i) *$\rho(A) > 0$ is a simple eigenvalue of A;*
(ii) *there is a positive normalized eigenvector* v *such that* $A\mathrm{v} = \rho(A)\mathrm{v}$ *(*v *is called the Perron eigenvector of A).*

Eigenvector centrality (see, e.g., [30]) measures the importance of the nodes in $\mathcal{G}$ by the magnitude of the entries of the Perron eigenvector v, allowing to rank the nodes. Note that the ranking is independent of the normalization used for v.

We indicate by $i_1, i_2, \ldots$ the indices of the entries of the Perron eigenvector v of A, ordered by non-decreasing magnitude. In the event that all the entries of v are distinct, the index i_1 corresponding to the largest entry in v identifies the most central node i_1 in $\mathcal{G}$, as measured by eigenvector centrality; i_2 identifies the second most central node, and so forth. Generally speaking, however, there could be many nodes with the same centrality score. An extreme case is when all the row sums of A are equal, in which case eigenvector centrality is unable to provide a ranking of the nodes, every node being given the same centrality score.

As an illustrative example, for the following graph $\mathcal{G}$ on the left picture,



we have

$$A = \begin{pmatrix} 0 & 14.9 & 6.7 & 3.9 \\ 0 & 0 & 4.0 & 0 \\ 0 & 5.2 & 0 & 14.1 \\ 4.8 & 0 & 9.6 & 0 \end{pmatrix}, \quad \mathrm{v} = \begin{pmatrix} 0.5665 \\ 0.1570 \\ \mathbf{0.5844} \\ 0.5594 \end{pmatrix}.$$

Hence, when ranked by (right) eigenvector centrality, v_3 is the most central node, v_1 the second most central one, and so forth.

In this case, the ranking provided by eigenvector centrality appears unambiguous. However, a relative perturbation of the weights of the graph of a few percents causes

the two entries v_1 and v_3 to flip in magnitude, indicating a lack of robustness of the ranking.

We show this exhibiting by the nearby graph $\widetilde{\mathcal{G}}$ (on the right picture) with weighted adjacency matrix $\widetilde{A}$ and corresponding Perron eigenvector $\widetilde{\mathrm{v}}$:

$$\widetilde{A} = \begin{pmatrix} 0 & 14.9 & 6.8 & 4.0 \\ 0 & 0 & 4.1 & 0 \\ 0 & 5.1 & 0 & 13.9 \\ 4.7 & 0 & 9.5 & 0 \end{pmatrix}, \qquad \widetilde{\mathrm{v}} = \begin{pmatrix} \mathbf{0.5774} \\ 0.1602 \\ 0.5772 \\ 0.5548 \end{pmatrix}.$$

Considering that in many applications the weights are known to an accuracy of only one or two digits, this example shows that indeed the ranking determined by eigenvector centrality may not always be reliable.

Generally speaking, the ranking of the vertices obtained by this approach becomes unreliable when one of the following occurrences is determined by a small perturbation of the weights:

(I) The coalescence of the entries $\mathrm{v}_{i_1}, \mathrm{v}_{i_2}, \ldots, \mathrm{v}_{i_m}$ for some index $m > 1$.
(II) The coalescence of the eigenvalues λ_1 and λ_2.

Note that (II) can be ruled out if we make Assumption 3.1, i.e. in the irreducible case.

Indeed, if a small perturbation is able to coalesce the highest m entries of the eigenvector v, it means that the ranking for the highest positions is not robust, especially in the presence of uncertainties, rounding errors, or any other form of noise in the weights; in general this is a symptom of the fact that the highest m vertices are difficult to rank using eigenvector centrality and it may therefore be better to resort to different centrality measures. A similar issue arises when a small perturbation makes the adjacency matrix A reducible (i.e., the graph is no longer strongly connected), in which case the Perron eigenvector may no longer be defined, rendering eigenvector centrality no longer usable. Note that issue (II) above falls under this scenario.

In this paper we direct our attention to issue (I), and we are interested in computing the smallest perturbation of A such that the colaescence of the highest entries in the Perron vector occurs. To this end, in the next subsection we propose a sharp measure which is aimed at characterizing the reliability of eigenvector centrality. As a by-product, the knowledge of this perturbation allows the possibility of modifying the ranking by small changes to the weights and may be used for this purpose. Issue (II) is left as a topic for future work.

3.1. **Robustness of the ranking and nearest ambiguous graph.** We consider the general case of a directed graph and look for the closest strongly connected graph $\widetilde{\mathcal{G}} = (\mathcal{V}, \mathcal{E}, \widetilde{A})$ to $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, such that the m highest entries of the Perron eigenvector coalesce. In order to preserve the strong connectivity we have to impose that the perturbed graph preserves the property $\widetilde{a}_{ij} > 0$ for $(i,j) \in \mathcal{E}$. For computational purposes we replace it with the property

$$\widetilde{a}_{ij} \geq \delta \qquad \text{for} \quad (i,j) \in \mathcal{E}, \tag{3}$$

with $1 \gg \delta > 0$ a small positive tolerance.

We consider the following measure of robustness, which we call *robustness radius*,

$$\varrho_m(\mathcal{G}) = \inf_{\widetilde{\mathcal{G}}} \operatorname{dist}\left(\mathcal{G}, \widetilde{\mathcal{G}}\right) \triangleq \inf_{\widetilde{A}} \|A - \widetilde{A}\|,$$

under the condition that the m largest entries of the Perron eigenvector of $\widetilde{A}$ coalesce and under the constraints $\widetilde{a}_{ij} \geq \delta$ for $(i,j) \in \mathcal{E}$ and $\widetilde{a}_{ij} = 0$ for $(i,j) \notin \mathcal{E}$.

Note that we may choose to only perturb a subset of the edges, $\mathcal{E}' \subset \mathcal{E}$, which would just change the constraint to $\widetilde{a}_{ij} \geq \delta$ for $(i,j) \in \mathcal{E}'$. For example, we may choose to perturb only the edges which either are subject to uncertainties or to changes in time. Another possibility would be to perturb only the nodes adjacent to a set of control nodes as considered, e.g., in [31].

Indeed it is possible (although highly unlikely) that no graph $\widetilde{\mathcal{G}}$ exists which fulfills the constraints, in which case we say that the distance is ∞. Moreover, when there exist graphs fulfilling the constraints, we have that the infimum is indeed a minimum, by compactness.

3.2. **The considered problem.** We are interested in the following problem of eigenvector-constrained minimization. We let

$$\Sigma = \left\{ B \in \mathbb{R}^{n\times n} \text{ such that } b_{ij} \geq \delta \ \forall\ (i,j) \in \mathcal{E} \ \wedge\ b_{ij} = 0 \ \forall\ (i,j) \notin \mathcal{E} \right\}$$

and (denoting with u the Perron eigenvector of $B \in \Sigma$)

$$\Omega = \{ B \in \Sigma : \mathrm{u}_{i_1} = \cdots = \mathrm{u}_{i_m} \} \tag{4}$$

where, for a given m, $i_1, i_2, \ldots, i_m$ are the indices of the m largest entries of the Perron eigenvector u of B (i.e., $B\mathrm{u} = \rho(B)\mathrm{u}$). The problem then reads

$$\min \|B - A\|_F \quad \text{s.t.} \quad B \in \Omega\,. \tag{5}$$

As we will see in the section on numerical experiments, the constraint defining the set Ω is challenging to deal with by means of classical optimization approaches.

In the next section we describe a reformulation of the problem which leads in natural way to a two-level approach, an inner eigenvector optimization problem and an outer root-finding one.

3.3. **Outline of the computational approach.** In this section we describe an approach to compute the robustness measure $\varrho_m(\mathcal{G})$ and the closest perturbed graph characterized by an ambiguous ranking of the most important m vertices. It is convenient to write

$$\widetilde{A} = A + \varepsilon E \quad \text{ with } \|E\| = 1$$

so that ε is the norm of the perturbation to A.

Based on the fact that we are addressing the coalescence of m distinct entries, we can make use of their average to obtain a concentration measure. For a given $\varepsilon > 0$ and a given $E \in \mathbb{R}^{n\times n}$ with the same pattern of A (and such that $A + \varepsilon E \geq 0$), we indicate by

$$\mathrm{v} = \mathrm{v}(A + \varepsilon E)$$

the (normalized) Perron eigenvector of $\widetilde{A}$ and define the mean of its largest m entries as

$$\Big\langle \mathrm{v}(A+\varepsilon E) \Big\rangle_m = \frac{1}{m} \sum_{j=1}^{m} \mathrm{v}_{i_j}\big(A + \varepsilon E\big), \tag{6}$$

where $i_1, i_2, \ldots, i_m$ the indices of the m largest entries of the Perron eigenvector v of A, ordered by non-decreasing magnitude.

Then we define the following functional of E :

$$F_\varepsilon(E) \quad = \quad \frac{1}{2}\sum_{k=1}^{m}\Big(\mathrm{v}_{i_k}\big(A+\varepsilon E\big) - \Big\langle \mathrm{v}(A+\varepsilon E)\Big\rangle_m\Big)^2 \tag{7}$$

under the constraints of preservation of the sparsity pattern of A, of unit Frobenius norm of E and of non-negativity $A+\varepsilon E \geq 0$. This nonnegative functional measures the dispersion of the largest m entries of the Perron eigenvector $\mathrm{v}(A+\varepsilon E)$ with respect to their mean. Clearly the functional vanishes if and only if the m highest entries of $\mathrm{v}(A+\varepsilon E)$ coalesce.

Our approach is summarized by the following two-level method:

- **Inner iteration:** Given $\varepsilon > 0$, we compute a smooth matrix flow $E(t) = (e_{ij}(t)) \in \mathbb{R}^{n\times n}$, $t \geq 0$, with the same sparsity pattern as A (i.e., $e_{ij}(t) = 0$ if $a_{ij} = 0$), with each $E(t)$ of unit Frobenius norm, with $(A+\varepsilon E(t))_{ij} \geq \delta$ (for $(i,j) \in \mathcal{E}$) and we minimize the functional $F_\varepsilon(E(t))$ over this matrix flow. The obtained minimizer is denoted by $E(\varepsilon)$.
- **Outer iteration:** We compute the smallest value of ε such that the m-th largest entries of the Perron eigenvector $\mathrm{v}\left(A+\varepsilon E(\varepsilon)\right)$ coalesce.

When there exists a solution to the problem $F_\varepsilon\left(E(\varepsilon)\right) = 0$, in the outer iteration we compute the optimal ε, denoted $\widetilde{\varepsilon}_m$ and the corresponding extremal perturbation $E(\widetilde{\varepsilon}_m)$,

$$\widetilde{\varepsilon}_m \longrightarrow \arg\min_{\varepsilon>0} F_\varepsilon\left(E(\varepsilon)\right) = 0 \tag{8}$$

by a combined Newton-bisection method.

Although in most interesting cases the equation $F_\varepsilon\left(E(\varepsilon)\right) = 0$ has solutions, in general this is not assured. If a solution to $F_\varepsilon\left(E(\varepsilon)\right) = 0$ exists then (8) yields

$$\varrho_m(\mathcal{G}) = \widetilde{\varepsilon}_m.$$

Existence of local minima. An important issue concerns the existence of minima of (7). Let $\varepsilon > 0$ be arbitrary but fixed. By the assumption of strong connectivity of the perturbed graph, we have that the eigenvalue $\lambda = \rho(A+\varepsilon E)$ is simple for all admissible perturbations E. This implies the continuity with respect to E of the Perron eigenvector v associated to λ, which implies the continuity of the functional $F_\varepsilon(E)$. Combining this and the fact that the set of considered perturbations is compact, a minimizer of (8) always exists. Since we make use of a gradient system approach to the problem, in general we cannot be sure to compute global optima, but only to globally converge to local optima, which means that the final computed matrix provides an upper bound to the solution of (8).

Two-level optimization. Due to the nested, or two-level form of problem (4)–(5), it is difficult to devise a single-level optimization approach to its solution.

The particular two-level methodology we propose has two main motivations. The first is that in this way we are able to approach the solution of the problem in a (norm) controlled way, which might be considered in analogy to a trust region method, allowing us – for example – to reach the solution by a sequence of matrices of smaller norm. The second is that in this way we obtain, as an important by-product, that the stationary points of the inner iteration have an explicit characterization in terms of the constrained gradient of the functional, therefore we are able to compute them explicitly by means of first-order eigenvector perturbation

theory, as shown in the following Theorem 5.1. In particular, this reveals an underlying low-rank property that would otherwise remain hidden, and which can be exploited at the computational level.

For the solution of the inner iteration we propose a numerical integrator for the constrained gradient system associated to the functional. This is based on an explicit formula for the gradient, which we derive by using first-order eigenvector perturbation theory. Using alternative descent methods based on the exact knowledge of the gradient is also possible, and leads to similar results.

For the solution of the outer iteration (which aims to solve a root-finding problem) we consider a Newton-bisection method, which exploits an exact formula for the derivative of the function of which we aim to compute the smallest root.

4. Derivatives of eigenvalues and eigenvectors

In this section we recall a few basic results concerning first order perturbation theory of real eigenvalues and eigenvectors of a real matrix, which is the relevant case for this paper.

4.1. First order perturbation theory for simple eigenvalues.

The following standard perturbation result for eigenvalues is well-known; see e.g. [14, Theorem 1] and [25, Section II.1.1].

Theorem 4.1 (Derivative of simple eigenvalues)**.** *Consider a continuously differentiable path of square real matrices $M(t)$ for t in an open interval I. Let $\lambda(t)$, $t \in I$, be a continuous path of real simple eigenvalues of $M(t)$. Let $x(t)$ and $y(t)$ be left and right eigenvectors, respectively, of $M(t)$ corresponding to the eigenvalue $\lambda(t)$. Then, $x(t)^\top y(t) \neq 0$ for $t \in I$ and λ is continuously differentiable on I with the derivative (we denote $\dot{} = d/dt$)*

$$\dot{\lambda} = \frac{x^\top \dot{M} y}{x^\top y}\,. \tag{9}$$

Moreover, "continuously differentiable" can be replaced with "analytic" in the assumption and the conclusion.

Since we have $x(t)^\top y(t) \neq 0$, we can apply the normalization

$$\|x(t)\| = 1, \quad \|y(t)\| = 1, \quad x(t)^\top y(t) \text{ is real and positive.} \tag{10}$$

4.2. First order perturbation theory for eigenvectors.

For the derivative of eigenvectors we need the notion of group inverse (or reduced resolvent); see [29] as well as Kato ([25, Section I.5.3]).

Definition 4.1 (Group inverse)**.** *Let $M \in \mathbb{R}^{n\times n}$ be a singular matrix with a simple zero eigenvalue. The* group inverse *(or reduced resolvent) of M is the unique matrix $M^\#$ with*

$$MM^\# = M^\# M, \qquad M^\# M M^\# = M^\#, \quad \textit{and} \quad MM^\# M = M. \tag{11}$$

It is known from Meyer and Stewart ([29]) that if M is normal, then its group inverse $M^\#$ is equal to the better known Moore–Penrose pseudoinverse $M^\dagger$. In general, the two pseudoinverses are not the same. They are, however, related by the following result, which is a special case of a more general result in [19, Appendix A] which is also verified directly.

Theorem 4.2 (Group inverse and Moore–Penrose pseudoinverse)**.** *Suppose that the real matrix M has the simple eigenvalue 0 with corresponding left and right real eigenvectors x and y of unit norm and such that $x^\top y > 0$. Let $M^\#$ be the group inverse of M, and with $\kappa = 1/(x^\top y)$ define the projection $\Pi = \mathrm{I} - \kappa x y^\top$. The group inverse $M^\#$ of M is related to the Moore–Penrose pseudoinverse $M^\dagger$ by*

$$M^\# = \Pi M^\dagger \Pi. \tag{12}$$

This result is used in the codes implemented by the authors for the computations involving the group inverse. Similar results are known in the Markov chain literature (see, e.g., [3]). The group inverse appears in the following result on the derivative of eigenvectors, which is a variant of Theorem 2 of Meyer and Stewart ([29]).

Theorem 4.3 (Derivative of eigenvectors)**.** *Consider a smooth path of square real matrices $M(t)$ for t in an open interval I. Let $\lambda(t)$, $t \in I$, be a path of simple real eigenvalues of $M(t)$. Then, there exists a continuously differentiable path of associated left and right eigenvectors $x(t)$ and $y(t)$, $t \in I$, which are of unit norm with $x^\top(t)y(t) > 0$ and satisfy the differential equations*

$$\begin{aligned} \dot{x}^\top &= -x^\top \dot{M} M^\# + \left(x^\top \dot{M} M^\# x \right) x^\top, \\ \dot{y} &= -M^\# \dot{M} y + \left(y^\top M^\# \dot{M} y \right) y, \end{aligned} \tag{13}$$

where $M^\#(t)$ is the group inverse of $M(t) - \lambda(t) I$.

If M is symmetric the previous simplify to $x = y$ and

$$\dot{x}^\top = -x^\top \dot{M} M^\dagger \tag{14}$$

where $M^\dagger(t)$ is the pseudoinverse of $M(t) - \lambda(t) I$.

Formulas (13) are interpreted as follows: the first term in the right-hand side represents the free derivative of the eigenvector, while the second term constrains the eigenvector to preserve the unit Euclidean norm with respect to t.

5. A methodology for computing the closest ambiguous graph and the robustness radius

In this section we describe the computational procedure to compute the robustness radius $\varrho_m(\mathcal{G})$. We indicate by

$$\langle X, Y \rangle = \mathrm{trace}(X^\top Y)$$

the Frobenius inner product, i.e. the inner product inducing the Frobenius norm $\|\cdot\| = \|\cdot\|$ on $\mathbb{R}^{n\times n}$.

5.1. **Inner iteration.** We denote by $i_1, i_2, \ldots, i_m$ the indices of the m largest entries of the Perron eigenvector $\mathrm{v}(A + \varepsilon E)$, ordered by non-decreasing magnitude. . Moreover, we denote the i-th unit versor by

$$\mathrm{u}_i = \Big(\underbrace{0 \ldots 0}_{i-1} \; 1 \; 0 \ldots 0 \Big)^\top$$

For a set of edges $\mathcal{E}'$, we define $P_{\mathcal{E}'}$ as the orthogonal projection from $\mathbb{R}^{n\times n}$ onto the vector subspace of all real matrices having the sparsity pattern determined by $\mathcal{E}'$:

for $A = (a_{ij})$,

$$P_{\mathcal{E}'}(A)\big|_{ij} := \begin{cases} a_{ij}\,, & \text{if } (i,j) \in \mathcal{E}'\,, \\ 0\,, & \text{otherwise.} \end{cases}$$

Further constraints. An extension to the case where only some of the weights are subject to perturbations is straightforward. For that, it is sufficient to define the subset $\mathcal{E}' \subset \mathcal{E}$ of those arcs whose weights can be perturbed, and to replace everywhere the projection $P_{\mathcal{E}}$ by the analogous projection $P_{\mathcal{E}'}$.

Similarly, if a few weights are related by a linear relationship, this can be easily included in the considered projection with respect to the Frobenius inner product. It is sufficient to determine the projection onto the considered linear subspace and combine it with the one that takes into account the sparsity structure, which is also expressed as a linear subspace constraint. For details, see subsection 5.7.

From now on, unless otherwise specified, we shall consider $\mathcal{E}' = \mathcal{E}$, that is, we perturb all edges of the graph.

Definition 5.1. *For a given weight matrix A and $\varepsilon > 0$, a matrix $E = (e_{ij}) \in \mathbb{R}^{n\times n}$ is* ε-feasible *if*

(a) *E is of unit Frobenius norm.*
(b) *$E = P_{\mathcal{E}}(E)$.*
(c) *$(A + \varepsilon E)_{ij} \geq \delta$ for $(i,j) \in \mathcal{E}$.*
(d) *E is symmetric (only for undirected graphs).*

5.2. **Structured gradient of the functional (7).** Here we compute the gradient of the functional (7) when restricted to matrices having the same sparsity pattern of A. The key idea is based on the fact that we are able to express the derivative of $F_\varepsilon(E)$ as $g(\dot{E})$ for a special function g. Using the simple fact that for vectors $b, c \in \mathbb{R}^n$ and a matrix $Z \in \mathbb{R}^{n\times n}$ one has

$$b^\top Z c = \langle bc^\top, Z\rangle,$$

we are able to write (where $\dot{E} = \frac{d}{dt}E(t)$)

$$g(\dot{E}) = \langle Z_\varepsilon(E), \dot{E}\rangle$$

with an appropriate matrix-valued function Z_ε, which – in a natural way – is interpreted as the constrained gradient of F_ε (with respect to the Frobenius inner product). Projecting it onto the set of real matrices with sparsity pattern induced by $\mathcal{E}$ (first constraint) and unit norm (second constraint), we obtain the sought fully constrained gradient.

To address the undirected case, we define for $A \in \mathbb{R}^{n\times n}$,

$$\mathrm{Sym}(A)\big|_{ij} := \frac{a_{ij} + a_{ji}}{2}$$

Recalling (7), we have that $F_\varepsilon(E) = \frac{1}{2}\sum_{k=1}^{m}\left(\mathrm{v}_{i_k}(A+\varepsilon E) - \langle \mathrm{v}(A+\varepsilon E)\rangle_m\right)^2$. After defining the auxiliary vector

$$\bar{\mathrm{u}} = \frac{1}{m}\sum_{j=1}^{m}\mathrm{u}_{i_j}$$

we are able to state the following result.

Lemma 5.1 (Structured gradient of the functional – directed graphs)**.** *Consider a smooth path $E(t)$ of ε-feasible matrices, denote by $\lambda(t) = \rho\left(A + \varepsilon E(t)\right)$ the Perron eigenvalue and $\mathrm{v}(t) = \mathrm{v}\left(A + \varepsilon E(t)\right)$ the associated Perron eigenvector. Let $i_1, i_2, \dots$ the indices of the entries of $\mathrm{v}(t)$, ordered by non-decreasing magnitude. We have (omitting the omnipresent dependence on t)*

$$\frac{d}{dt} F_\varepsilon\left(E(t)\right) = \eta \Big\langle G_\varepsilon(E), \dot{E} \Big\rangle, \tag{15}$$

where $\eta > 0$ is a suitable scaling factor, with the structured gradient $G_\varepsilon(E)$, having the sparsity pattern determined by the set of edges $\mathcal{E}$)

$$G_\varepsilon(E) = P_{\mathcal{E}}\left(R\right) \tag{16}$$

where R is the free gradient:

$$R \;\; = \sum_{k=1}^{m} \left(\mathrm{v}_{i_k} - \langle \mathrm{v} \rangle_m\right) \Big(\left(\mathrm{v}_{i_k} - \langle \mathrm{v} \rangle_m\right) \left(A_\varepsilon^{\#}\right)^\top \mathrm{v}\mathrm{v}^\top - \left(A_\varepsilon^{\#}\right)^\top \left(\mathrm{u}_{i_k} - \bar{\mathrm{u}}\right) \mathrm{v}^\top \Big).$$

Proof. By (7) we get

$$\frac{d}{dt} F_\varepsilon\left(E(t)\right) = \sum_{k=1}^{m} \big(\mathrm{v}_{i_k} - \langle \mathrm{v} \rangle_m\big) \big(\dot{\mathrm{v}}_{i_k} - \langle \dot{\mathrm{v}} \rangle_m\big).$$

Theorem 4.3 applied to $A + \varepsilon E(t)$ yields

$$\begin{aligned} \dot{\mathrm{v}}_{i_k} - \langle \dot{\mathrm{v}} \rangle_m \;\; &= \;\; \Big(\mathrm{u}_{i_k} - \frac{1}{m} \sum_{j=1}^{m} \mathrm{u}_{i_j}\Big)^\top \dot{\mathrm{v}} = \varepsilon \left(\mathrm{u}_{i_k} - \bar{\mathrm{u}}\right)^\top \Big(-A_\varepsilon^{\#} \dot{E} \mathrm{v} + \Big(\mathrm{v}^\top A_\varepsilon^{\#} \dot{E} \mathrm{v} \Big) \mathrm{v} \Big) \\ &= \;\; \varepsilon \Big(\Big\langle - \left(A_\varepsilon^{\#}\right)^\top \left(\mathrm{u}_{i_k} - \bar{\mathrm{u}}\right) \mathrm{v}^\top, \dot{E} \Big\rangle + \left(\mathrm{v}_{i_k} - \langle \mathrm{v} \rangle_m\right) \Big\langle \left(A_\varepsilon^{\#}\right)^\top \mathrm{v}\mathrm{v}^\top, \dot{E} \Big\rangle \Big) \end{aligned} \tag{17}$$

where $A_\varepsilon^{\#}(t)$ is the group inverse of $A + \varepsilon E(t) - \lambda(t) I$ and $\mathrm{v}(t)$ is the Perron eigenvector of $A + \varepsilon E(t)$, normalized with $\|\mathrm{v}(t)\| = 1$. Hence we can write

$$\frac{d}{dt} F_\varepsilon\left(E(t)\right) = \varepsilon \big\langle R, \dot{E} \big\rangle$$

with the free gradient

$$R = \sum_{k=1}^{m} \big(\mathrm{v}_{i_k} - \langle \mathrm{v} \rangle_m\big) \Bigg(\big(\mathrm{v}_{i_k} - \langle \mathrm{v} \rangle_m\big) \left(A_\varepsilon^{\#}\right)^\top \mathrm{v}\mathrm{v}^\top - \left(A_\varepsilon^{\#}\right)^\top \big(\mathrm{u}_{i_k} - \bar{\mathrm{u}}\big) \mathrm{v}^\top \Bigg).$$

Using the property that $P_{\mathcal{E}}(\dot{E}) = \dot{E}$, we simply write

$$\big\langle R, \dot{E} \big\rangle = \big\langle P_{\mathcal{E}}(R) + \left(R - P_{\mathcal{E}}(R)\right), \dot{E} \big\rangle$$

and observe that

$$\big\langle R - P_{\mathcal{E}}(R), \dot{E} \big\rangle = \big\langle R - P_{\mathcal{E}}(R), P_{\mathcal{E}}(\dot{E}) \big\rangle = 0,$$

This gives the structure-constrained gradient $G_\varepsilon(E) = P_{\mathcal{E}}(R)$ (see (16)). □

Lemma 5.2 (Fully constrained gradient of the functional – undirected graphs)**.** *Under the same assumptions of Lemma 5.1 to which is added the hypothesis that the smooth path $E(t)$ is also symmetric, we have that the fully constrained gradient is still given by* (15) *with*

$$G_\varepsilon(E) = \; - P_{\mathcal{E}} \left(\sum_{k=1}^{m} \left(\mathrm{v}_{i_k} - \langle \mathrm{v} \rangle_m\right) \mathrm{Sym}\Big(A_\varepsilon^{\dagger} \left(\mathrm{u}_{i_k} - \bar{\mathrm{u}}\right) \mathrm{v}^\top \Big) \right) \tag{18}$$

Proof. The proof is similar to that of Lemma 5.1. Here $A_\varepsilon^{\#} = A_\varepsilon^{\dagger}$; then, by the well-known property $\mathrm{v}^\top A_\varepsilon^{\dagger} = 0$, we first get

$$R = -(A_\varepsilon^{\#})^\top \big(\mathrm{u}_{i_k} - \bar{\mathrm{u}}\big)\mathrm{v}^\top.$$

Next, and including the symmetry constraint (ic) and thus the symmetry of $\dot{E}$, we get

$$\left\langle R, \dot{E}\right\rangle = \left\langle P_{\mathcal{E}}(R), \dot{E}\right\rangle = \left\langle \mathrm{Sym}\left(P_{\mathcal{E}}(R)\right) + \mathrm{Skew}\left(P_{\mathcal{E}}(R)\right), \dot{E}\right\rangle = \left\langle \mathrm{Sym}\left(P_{\mathcal{E}}(R)\right), \dot{E}\right\rangle$$

where for a matrix B, $\mathrm{Skew}(B) = (B - B^\top)/2$. Orthogonality of symmetric and skew-symmetric matrices proves the result, i.e. gradient (16) symplifies to (18). □

5.3. **Constrained gradient of the functional.** In this section we take into account the norm constraint on E and the inequality constraint on $A + \varepsilon E$. One possibility would be to use a penalization method; this, however, would have the disadvantage that the potential violation of the inequality constraints might have an impact on the Perron eigenvector and on its continuity. For this reason we opt for a Lagrangian approach.

Admissible directions. We recall the considered constraints on E:

(a) E is of unit Frobenius norm.
(b) $E = P_{\mathcal{E}}(E)$.
(c) $(A + \varepsilon E)_{ij} \geq \delta$ for $(i,j) \in \mathcal{E}$.
(d) E is symmetric (only for undirected graphs).

Since $E(t)$ is of unit Frobenius norm by condition (a), we have

$$0 = \frac{1}{2}\frac{d}{dt}\|E(t)\|^2 = \langle E(t), \dot{E}(t)\rangle.$$

To fulfill the non-negativity condition (c) of the matrix $A + \varepsilon E(t)$, we need that

$$\dot{e}_{ij}(t) \geq 0 \quad \text{for all } (i,j) \text{ with } a_{ij} + \varepsilon e_{ij}(t) = \delta \geq 0.$$

Conditions (b) and (d) are satisfied if the same condition holds for $\dot{E}$. These four conditions are in fact also sufficient for a matrix to be the time derivative of a path of ε-feasible matrices.

Hence, for every ε-feasible matrix E, a matrix $Z = (z_{ij}) \in \mathbb{R}^{n\times n}$ is the derivative of some path of ε-feasible matrices passing through E if and only if the following four conditions are satisfied:

(a') $\langle E, Z\rangle = 0$.
(b') $Z = P_{\mathcal{E}}(Z)$.
(c') $P_{\mathcal{E}_0}(Z) \geq 0$.
(d') Z is symmetric (only for undirected graphs) .

Here $\mathcal{E}_0 = \mathcal{E}_0(\varepsilon E)$ is the set

$$\mathcal{E}_0 := \{(i,j) \in \mathcal{E} : a_{ij} + \varepsilon e_{ij} = \delta\}.$$

Condition (c') implies that $z_{ij} \geq 0$ for all $(i,j) \in \mathcal{E}_0$.

Admissible direction of steepest descent. In order to determine the admissible direction $Z = \dot{E}$ of steepest descent from E, we therefore consider the following optimization problem for $G = G_\varepsilon(E)$:

$$\min_Z \langle G, Z\rangle \quad \text{subject to} \tag{19}$$

$$\begin{cases} \langle E, Z\rangle = 0\,, \\ z_{ij} \geq 0 \text{ for all } (i,j) \in \mathcal{E}_0\,, \\ z_{ij} = z_{ji} \text{ for all } (i,j) \quad \text{(for undirected graphs)}\,, \\ \langle Z, Z\rangle = 1\,. \end{cases} \tag{20}$$

The additional constraint $\|Z\| = 1$ just normalizes the descent direction. Problem (19) has a quadratic constraint.

We now formulate a quadratic optimization problem with linear constraints, which is equivalent to (19).

An equivalent quadratic optimization problem with linear constraints.

$$\min_Z \langle Z, Z\rangle \quad \text{subject to} \tag{21}$$

$$\begin{cases} \langle E, Z\rangle = 0\,, \\ z_{ij} \geq 0 \text{ for all } (i,j) \in \mathcal{E}_0\,, \\ z_{ij} = z_{ji} \text{ for all } (i,j) \quad \text{(for undirected graphs)}\,, \\ \langle G, Z\rangle = -1\,. \end{cases} \tag{22}$$

Problems (19) and (21) are equivalent in the sense that Problem (21) yields the same solution, i.e. descent direction, provided that a strict descent direction exists, i.e., satisfying $\langle G, Z\rangle < 0$ and the constraints (a')–(d'). The equivalence is a consequence of the fact that when $\langle G, Z\rangle < 0$, there exists a scaling factor $\alpha > 0$ s uch that $\langle G, \alpha Z\rangle = -1$.

Apart from the normalization, problems (19) and (21) yield the same KKT (Karush–Kuhn–Tucker) conditions. Since the objective function $\langle Z, Z\rangle$ of problem (21) is convex and all constraints are linear, the KKT conditions are not only necessary but also sufficient conditions ([10, Theorem 9.4.1]), that is, a KKT point is already a solution of the optimization problem.

The solution of (21) satisfies the KKT conditions (see, e.g., [11, Ch. 2.2])

$$\begin{cases} \kappa Z = -G - \gamma E + \sum_{(i,j)\in\mathcal{E}_0} \mu_{ij} \mathrm{u}_i \mathrm{u}_j^T\,, \\ z_{ij} \geq 0 \text{ for all } (i,j) \in \mathcal{E}_0\,, \\ \langle E, Z\rangle = 0\,, \\ \langle Z, Z\rangle = 1\,, \\ Z = Z^\top \quad \text{(for undirected graphs)}\,, \\ \mu_{ij} z_{ij} = 0 \text{ for all } (i,j) \in \mathcal{E}_0\,, \\ \mu_{ij} \geq 0 \text{ for all } (i,j) \in \mathcal{E}_0\,, \end{cases} \tag{23}$$

where u_ℓ is the ℓ-th unit versor, γ is determined by the normalization $\langle E, Z\rangle = 0$, and $\kappa > 0$ is determined by the normalization $\|Z\|^2 = \langle Z, Z\rangle = 1$.

5.4. **Constrained gradient flow.** The gradient flow of F_ε under the constraints (a)–(d) is given by

$$\dot{E}(t) = Z(t), \tag{24}$$

where $Z(t)$ solves the KKT system (23) with $G = G_\varepsilon(E(t))$ under the constraints (a')–(d'). Note that the set of edges $\mathcal{E}_0(t) = \mathcal{E}_0(\varepsilon E(t))$.

Lemma 5.3. *On an interval where $\mathcal{E}_0(t)$ does not change, the gradient system becomes, with $P^+ = P_{\mathcal{E}\setminus\mathcal{E}_0}$,*

$$\dot{E} = -P^+ G_\varepsilon(E) + \gamma P^+ E \quad \text{with} \quad \gamma = \frac{\langle G_\varepsilon(E), P^+E\rangle}{\|P^+E\|^2}. \tag{25}$$

Proof. The positive Lagrange multipliers $\mu_{ij} > 0$ just have the role to ensure that $\dot{e}_{ij} = 0$. With $G = G_\varepsilon(E)$, the gradient system therefore reads

$$\dot{E} = P^+(-G + \gamma E), \tag{26}$$

where γ is determined from the constraint $\langle E, \dot{E}\rangle = 0$. We then have

$$0 = \langle E, \dot{E}\rangle = \langle E, P^+(-G+\gamma E)\rangle = -\langle P^+E, G\rangle + \gamma\langle P^+E, P^+E\rangle,$$

and the result follows. □

5.5. **Monotonicity and stationary points.** The following result shows the monotonicity of the functional $F_\varepsilon(E(t))$ with respect to t and provides a characterization of the stationary points. We observe here that monotonicity follows directly from the construction of the gradient system (25) (for the time being, we assume that the non-negativity constraint does not activate so that penalization is not necessary, which is what we observe in all the tests we have done).

Theorem 5.1. *Let $E(t)$ of unit Frobenius norm satisfy the differential equation (25) with $G_\varepsilon(E)$ as in (16). Then, the functional $F_\varepsilon(E(t))$ decreases monotonically with respect to t:*

$$\dot{F}_\varepsilon(E(t)) \le 0.$$

Furthermore, the following statements are equivalent along solutions of (25):

1. $\dot{F}_\varepsilon = 0$.
2. $\dot{E} = 0$.
3. P^+E *is a real multiple of* $P^+G_\varepsilon(E)$.

Proof. We prove the result in three steps.

(1) implies (3): Using Lemma 5.1 and (25) we obtain, with $G = G_\varepsilon(E)$,

$$\frac{1}{\varepsilon}\dot{F}_\varepsilon = \langle G, \dot{E}\rangle = \langle G, -P^+G + \gamma P^+E\rangle = -\|P^+G\|^2 + \frac{\langle P^+G, P^+E\rangle^2}{\|P^+E\|^2}. \tag{27}$$

By means of the strong form of the Cauchy–Schwarz inequality we obtain monotonicity. Furthermore we have that

$$\frac{d}{dt}F_\varepsilon\left(E(t)\right) = 0 \qquad \Longrightarrow \qquad P^+E \text{ is a multiple of } P^+G.$$

(3) implies (2): This is a direct consequence of the vanishing of the r.h.s. of (25).

(2) implies (1): This is also a direct consequence of Lemma 5.1.

□

5.6. **Underlying low-rank property of extremizers.** For any graph, we have a number of ODEs which is given by the nonzero entries in the adjacency matrix associated to the graph. Hence, for a dense graph we may have up to n^2 ODEs. If inequality constraints are inactive, the gradient system (25) is such that P^+ is the identity and the gradient system therefore reads

$$\dot{E} = -G + \gamma E, \qquad \gamma = \langle G, E \rangle. \tag{28}$$

where $G = G_\varepsilon(E)$. In this case extremizers (stationary points) are characterized by an underlying low-rank property, being G the projection of a low-rank matrix onto the sparsity-pattern of the weighted adjacency matrix. Clearly, if $\varepsilon < \min_{ij} a_{ij}$, inequality constraints cannot be active so that we can ignore them and solve the system of ODEs (28). Moreover, in all our experiments we have never observed activation of inequality constraints associated to nonnegativity.

Following the lines of [18], we are able to exploit the underlying low-rank structure of the solutions of the optimization problem we consider. By Theorem 5.1 we have that the stationary points which we aim to compute are proportional to the gradient G, which - as we have seen - is the projection of a rank-1 matrix. Proceeding similarly to [18], by writing $E = P_{\mathcal{E}}(H)$ for a rank-1 matrix H, we are able to obtain a new system of ODEs for H, whose dynamics is constrained to the manifold of rank-1 matrices. We can prove that the stationary points of the two systems of ODEs are in a one-to-one correspondence and also that the system of ODEs for H, even if it loses the gradient system structure, converges - at least locally - to the stationary points. We omit a full derivation for sake of conciseness and refer the reader to [18] for details.

5.7. **Including a linear constraint on the entries of the adjacency matrix.** Suppose now that a few weights associated with the edge set $\mathcal{E}' \subseteq \mathcal{E}$ are related by a linear relationship, that is

$$\sum_{(i,j)\in\mathcal{E}} b_{ij} a_{ij} = c, \tag{29}$$

with $c, b_{ij} \in \mathbb{R}$ for all $(i,j) \in \mathcal{E}'$ and $b_{ij} = 0$ for all $(i,j) \in \mathcal{E} \setminus \mathcal{E}'$. Then for the perturbed matrix $A + \varepsilon E(t)$ we have to impose for all t

$$\sum_{(i,j)\in\mathcal{E}} b_{ij} e_{ij}(t) = 0 \quad \Longrightarrow \quad \sum_{(i,j)\in\mathcal{E}} b_{ij} \dot{e}_{ij}(t) = 0,$$

that is, we have to impose the scalar product $\langle b, \dot{e}(t) \rangle = 0$ for the associated vectors $b = (b_{ij})$ and $e = (e_{ij}) = \mathrm{vec}(E)$, which represents a column vector formed by the nonzero entries of E (which are defined by the sparsity pattern).

This implies that we have to modify the r.h.s. in the matrix ODE (24) in the inner iteration, which we write in short as $\dot{E} = Z$. Writing it in vectorized form as $\dot{e} = z$, with $z = \mathrm{vec}(Z)$, we modify it as

$$\dot{e} = z - \frac{1}{\|b\|^2} \langle z, b \rangle b, \tag{30}$$

that is, we orthogonally project the vector field z onto the linear space $\mathrm{Ker}(b) = \langle b, \cdot \rangle = 0$. Indeed, this implies

$$\langle \dot{e}, b \rangle = \langle z, b \rangle - \frac{1}{\|b\|^2} \langle z, b \rangle \langle b, b \rangle = 0,$$

as desired; it follows that the property (29) is conserved along the flow of the modified ODE. Equation (24) is in vector form, but it can be rewritten directly in matrix form.

6. Undirected graphs

Undirected graphs lead to some significant simplifications. In this case we have that $A_\varepsilon^\# = A_\varepsilon^\dagger$, the Moore-Penrose pseudoinverse, which is symmetric.

Recalling (18), we have

$$\begin{aligned} G_\varepsilon(E) &= -P_{\mathcal{E}}\left(\sum_{k=1}^{m}(\mathrm{v}_{i_k}-\langle \mathrm{v}\rangle_m)\,\mathrm{Sym}\Big(A_\varepsilon^\dagger\,(\mathrm{u}_{i_k}-\bar{\mathrm{u}})\,\mathrm{v}^\top\Big)\right) \\ &= -P_{\mathcal{E}}\left(\sum_{k=1}^{m} c_k \mathrm{Sym}\Big(a_k \mathrm{v}^\top\Big)\right), \end{aligned}$$

where we have defined

$$a_k = A_\varepsilon^\dagger b_k = (A+\varepsilon E-\lambda I)^\dagger b_k, \qquad \text{with} \qquad b_k = (\mathrm{u}_{i_k}-\bar{\mathrm{u}}), \qquad c_k = \big(\mathrm{v}_{i_k}-\langle \mathrm{v}\rangle_m\big),$$

which may be computed by solving the nonsingular linear system

$$\begin{pmatrix} A+\varepsilon E-\lambda I & \mathrm{v} \\ \mathrm{v}^\top & 0 \end{pmatrix}\begin{pmatrix} a_k \\ \mu \end{pmatrix} = \begin{pmatrix} b_k \\ 0 \end{pmatrix}. \tag{31}$$

Note that the submatrix $A+\varepsilon E-\lambda I$ above is negative semidefinite and singular. It is straightforward to check that the linear system (31) is nonsingular if and only if the orthogonal complement of v and the kernel of $A+\epsilon E-\lambda I$ have trivial intersection. Since $\mathrm{Ker}(A+\varepsilon E-\lambda I) = \mathrm{Span}\{\mathrm{v}\}$ and obviously $\mathrm{v}^\top\mathrm{v} \neq 0$, the condition is satisfied and the system (31) has a unique solution.

The computational cost of computing $G_\varepsilon(E)$ consists of computing the Perron eigenvector of the matrix $A+\varepsilon E(t)$ for t a suitable point on the mesh of integration and in solving – at every step – the linear system (31). The Perron eigenvector can be computed efficiently, for example using Lanczos' method; if the Perron root is well separated by the rest of the spectrum, even the power method will converge quickly. The solution of the linear system (31) merits a more detailed discussion.

Solving the linear system (31). In the considered undirected case, system (31) is symmetric and indefinite, with one positive and n negative eigenvalues. For small and moderate values of n, the solution can be obtained by means of some variant of the LDL^T factorization, see [13]. For large and sparse problems, an efficient way to solve (31) is by means of the Minimal Residual (MinRes) iteration [32], possibly combined with a suitable symmetric positive definite preconditioner [5]. Alternatively, system (31) can be replaced by the equivalent one

$$\begin{pmatrix} A+\varepsilon E-\lambda I-\mathrm{v}\mathrm{v}^\top & \mathrm{v} \\ \mathrm{v}^\top & 0 \end{pmatrix}\begin{pmatrix} a_k \\ \mu \end{pmatrix} = \begin{pmatrix} b_k \\ 0 \end{pmatrix}. \tag{32}$$

Note that the submatrix $A+\varepsilon E-\lambda I-\mathrm{v}\mathrm{v}^\top$ is negative definite (nonsingular). This matrix is completely full, since $\mathrm{v} > 0$, and it should not be formed explicitly. System (32) can be solved by means of the block LU factorization

$$\begin{pmatrix} A+\varepsilon E-\lambda I-\mathrm{v}\mathrm{v}^\top & \mathrm{v} \\ \mathrm{v}^\top & 0 \end{pmatrix} = \begin{pmatrix} A+\varepsilon E-\lambda I-\mathrm{v}\mathrm{v}^\top & 0 \\ \mathrm{v}^\top & -s \end{pmatrix}\begin{pmatrix} I & (A+\varepsilon E-\lambda I-\mathrm{v}\mathrm{v}^\top)^{-1}\mathrm{v} \\ 0 & 1 \end{pmatrix}$$

where we have set $s = \mathrm{v}^\top (A + \varepsilon E - \lambda I - \mathrm{v}\mathrm{v}^\top)^{-1}\mathrm{v}$. It follows that the solution of (32), and therefore of (31), is given by

$$\begin{aligned} a_k &= (A + \varepsilon E - \lambda I - \mathrm{v}\mathrm{v}^\top)^{-1} b_k - \mu (A + \varepsilon E - \lambda I - \mathrm{v}\mathrm{v}^\top)^{-1}\mathrm{v}\,, \\ \mu &= \frac{\mathrm{v}^\top (A + \varepsilon E - \lambda I - \mathrm{v}\mathrm{v}^\top)^{-1} b_k}{\mathrm{v}^\top (A + \varepsilon E - \lambda I - \mathrm{v}\mathrm{v}^\top)^{-1}\mathrm{v}}\,, \end{aligned}$$

showing that this approach requires the solution of two linear systems with the same coefficient matrix $A + \varepsilon E - \lambda I - \mathrm{v}\mathrm{v}^\top$ and right-hand sides b_k and v, respectively. These two linear systems can be solved by means of the conjugate gradient (CG) method, possibly combined with a preconditioner. Clearly, the CG method is well suited since matrix-vector products involving $A + \varepsilon E - \lambda I - \mathrm{v}\mathrm{v}^\top$ can be computed without forming the coefficient matrix explicitly; for sparse A, the cost of each matrix-vector product is linear in n. The choice of preconditioner is clearly problem dependent; note, however, that as the outer iteration converges, the coefficient matrix of the linear systems varies slowly from one step to the next, making it possible to reuse the same preconditioner for several steps.

Projected rank-2 properties of extremizers. Assume the inequality constraints are inactive. We let

$$L_\varepsilon(E) = \left(\sum_{k=1}^{m} (\mathrm{v}_{i_k} - \langle \mathrm{v} \rangle_m) \, \mathrm{Sym}\Big(A_\varepsilon^\dagger \, (\mathrm{u}_{i_k} - \bar{\mathrm{u}}) \, \mathrm{v}^\top \Big) \right)$$

and $G_\varepsilon(E) = -P_{\mathcal{E}}\left(L_\varepsilon(E)\right)$. Note that $A_\varepsilon^\dagger \, (\mathrm{u}_{i_k} - \bar{\mathrm{u}}) \, \mathrm{v}^\top$ is a rank-1 nilpotent matrix. We let

$$g_{i_j} = A_\varepsilon^\dagger \mathrm{u}_{i_j}, \qquad \text{the } i_j\text{-th column of } A_\varepsilon^\dagger$$

and

$$\tau_{i_k} = \mathrm{v}_{i_k} - \langle \mathrm{v} \rangle_m, \qquad a_{i_k} = g_{i_k} - \langle g \rangle_m, \quad \text{with} \quad \langle g \rangle_m = \frac{1}{m} \sum_{j=1}^{m} g_{i_j}.$$

This notation allows us to rewrite

$$L_\varepsilon(E) = \mathrm{Sym}\left(a \mathrm{v}^\top \right), \qquad a = \sum_{k=1}^{m} \tau_{i_k} a_{i_k},$$

which is a rank-2 symmetric matrix.

If $(i, j) \in \mathcal{E}$ for all $i = 1, \ldots, n$ and $j = 1, \ldots, n$ (which includes possible self-loops), we have that the pattern $\mathcal{E}$ is full, so that $P_{\mathcal{E}}$ is the identity projector. Thus $G_\varepsilon(E) = -L_\varepsilon(E)$, which means that the stationary points (if the penalty function is inactive) are rank-2.

In the general case, stationary points are projections onto $\mathcal{E}$ of rank-2 symmetric matrices.

6.1. **An illustrative example of an undirected graph.** Consider the simple undirected graph with 9 vertices of Figure 1. From a topological point of view one could argue that the vertex v_4 is characterized by the highest centrality score; however, since the graph is weighted this is not true in general.

In order to associate to the graph a scaled adjacency matrix A with unit Frobenius norm, we multiply the adjacency matrix whose weights are indicated in Figure 1 by the normalizing factor

$$\beta \approx 1/0.9974 \approx 1.0026.$$

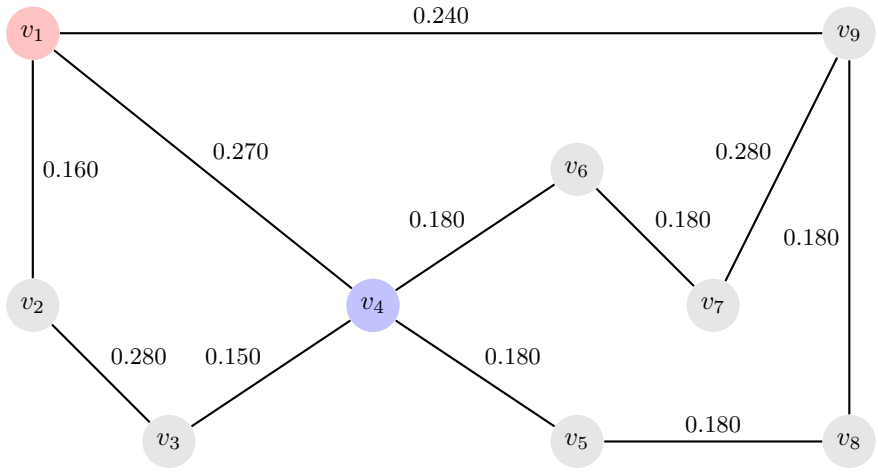


FIGURE 1. Example of symmetric weighted graph

We have $\lambda = \rho(A) \approx 0.5559\ldots$ and the associated Perron eigenvector (to 4 digits)

$$\mathrm{v} \approx \begin{pmatrix} \mathbf{0.4844} & 0.2712 & 0.2602 & \mathbf{0.4553} & 0.2154 & 0.2433 & 0.2941 & 0.2082 & 0.4259 \end{pmatrix}^\top.$$

The ranking of nodes v_1 and v_4 can be interchanged by small changes in the weights. This is illustrated in Figure 2. Choosing a suitable small perturbation (of norm 0.0280) of the normalized adjacency matrix A, we obtain the modified Perron eigenvector $\tilde{\mathrm{v}}$

$$\tilde{\mathrm{v}} = \begin{pmatrix} \mathbf{0.4754} & 0.2699 & 0.2722 & \mathbf{0.4758} & 0.2277 & 0.2543 & 0.2839 & 0.2035 & 0.4027 \end{pmatrix}^\top.$$

where now v_4 precedes v_1.

Note that, by a simple continuity argument of the Perron eigenvector, there exists a perturbation of the weights for which $\tilde{\mathrm{v}}_1$ and $\tilde{\mathrm{v}}_4$ have exactly the same value.

We are interested in the robustness of the ranking for the first $m = 2$ positions, corresponding to $i_1 = 1$ and $i_2 = 4$. The gap between the 1-st and 4-th entries is $\mathrm{v}_1 - \mathrm{v}_4 = 0.0291\ldots$

According to (18), we compute the structure-constrained gradient G_0, normalized to have Frobenius norm equal to 1; it is given by

$$G_0 = \begin{pmatrix} 0 & 0.336 & 0 & -0.113 & 0 & 0 & 0 & 0 & 0.645 \\ 0.336 & 0 & -0.029 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.029 & 0 & -0.412 & 0 & 0 & 0 & 0 & 0 \\ -0.113 & 0 & -0.412 & 0 & -0.418 & -0.438 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.418 & 0 & 0 & 0 & -0.073 & 0 \\ 0 & 0 & 0 & -0.438 & 0 & 0 & -0.074 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.074 & 0 & 0 & 0.188 \\ 0 & 0 & 0 & 0 & -0.073 & 0 & 0 & 0 & 0.109 \\ 0.645 & 0 & 0 & 0 & 0 & 0 & 0.188 & 0.109 & 0 \end{pmatrix}.$$

This indicates that a perturbation on the weights $a_{19}, a_{12}, a_{34}, a_{45}$ and a_{46} has the highest influence on the gap $\mathrm{v}_1 - \mathrm{v}_4$.

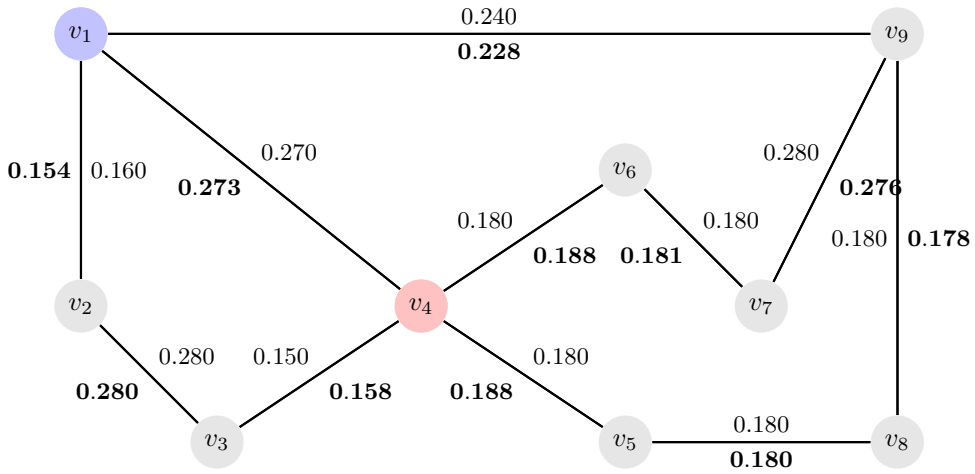


FIGURE 2. Perturbed graph of Figure 1 (weights of modified graph appear in bold on each edge).

Modifying the full pattern of A. The choice $\varepsilon = 0.029$ with the natural initial value for the ODE (25) given by $E(0) = -G_0$ turns out to give interesting results. After integration of the ODE (25), we get the Perron eigenvector of the (optimally) perturbed matrix $A + \varepsilon E(\varepsilon)$,

$$\widetilde{\mathrm{v}} = (\ \mathbf{0.4742}\quad 0.2695\quad 0.2722\quad \mathbf{0.4753}\quad 0.2284\quad 0.2551\quad 0.2848\quad 0.2044\quad 0.4029\ )^\top$$

which shows an inversion of the ranking of the leading nodes, in fact

$$\widetilde{\mathrm{v}}_1 - \widetilde{\mathrm{v}}_4 = -0.0011\ldots \qquad \Longrightarrow \qquad \varrho_2(\mathcal{G}) < 0.029, \tag{33}$$

where we recall that $\varrho_2(\mathcal{G}) = \min_{\widetilde{\mathcal{G}}} \mathrm{dist}\left(\mathcal{G}, \widetilde{\mathcal{G}}\right) = \min_{\widetilde{A}} \|A - \widetilde{A}\|$, under the condition that the 2 largest entries of the Perron eigenvector of $\widetilde{A}$ coalesce and under the constraints $\widetilde{a}_{ij} \geq \delta$ for $(i,j) \in \mathcal{E}$ and $\widetilde{a}_{ij} = 0$ for $(i,j) \notin \mathcal{E}$. It is also interesting to observe that the gap between $\widetilde{\mathrm{v}}_4$ and $\widetilde{\mathrm{v}}_9$ on the other hand has increased.

A further tuning of the parameter ε in the outer iteration provides the value $\varepsilon^* \approx 0.0279064$ as the one leading to coalescence (from a numerical point of view, coalescence was declared as soon as the condition $|\widetilde{\mathrm{v}}_1 - \widetilde{\mathrm{v}}_4| \leq \mathrm{tol}$ with $\mathrm{tol} = 10^{-5}$ was satisfied). We obtain in fact

$$\widetilde{\mathrm{v}} \approx (\ \mathbf{0.4745}\quad 0.2696\quad 0.2717\quad \mathbf{0.4745}\quad 0.2279\quad 0.2546\quad 0.2851\quad 0.2045\quad 0.4038\ )^\top.$$

Based on an heuristic argument, one may consider a perturbation E whose only nonzero entries are $e_{12} = e_{21} = e_{14} = e_{41} = e_{19} = e_{91} = -1/\sqrt{6}$ (which guarantess the unit Frobenius norm of E), aiming to decrease the weights on the edges which connect the vertex v_1, which has highest centrality score, to the rest of the graph. With a perturbation again of magnitude $\varepsilon = 0.029$, this would give the Perron eigenvector of $A + \varepsilon E$,

$$\overline{\mathrm{v}} \approx (\ \mathbf{0.4689}\quad 0.2640\quad 0.2624\quad \mathbf{0.4543}\quad 0.2229\quad 0.2523\quad 0.3047\quad 0.2161\quad 0.4270\ )^\top$$

which preserves the original unperturbed ranking and is characterized by a gap of 0.0146, significantly different (and with the opposite sign) from the one in (33).

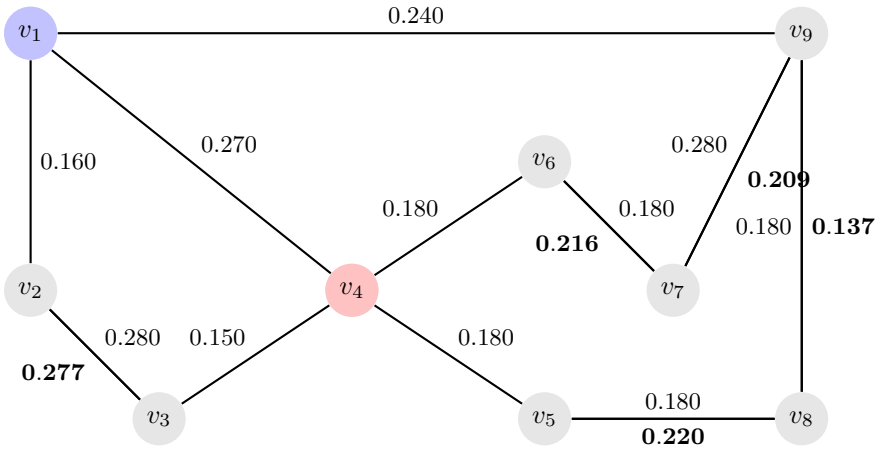


FIGURE 3. Modified graph of Figure 1 acting only on a subset of the edges (modified weights on each edge appear in bold)

Modifying only a subpattern of A. As already mentioned in the discussion preceding Definition 5.1, it is possible to consider, instead of the whole set of edges, a subset $\mathcal{E}' \subset \mathcal{E}$ and proceed exactly as discussed by simply replacing the projection $P_{\mathcal{E}}$ with $P_{\mathcal{E}'}$. As an example we consider again the simple undirected graph with 9 vertices of Figure 1 and modify only the edges not incident to the two nodes v_1 and v_4. The results so obtained are illustrated in Figure 3. We compute the Perron eigenvector v

$$\mathrm{v} \approx \begin{pmatrix} \mathbf{0.4913} & 0.2916 & 0.2856 & \mathbf{0.4913} & 0.2406 & 0.2608 & 0.2433 & 0.1894 & 0.3601 \end{pmatrix}^\top.$$

and the distance $\varepsilon^* = 0.1407018$,

6.2. **Numerical integration of the gradient system.** We describe here the k-th step of numerical integration of (25). Starting from $E_k \approx E(t_k)$ feasible matrix of unit norm, we apply a step of length h of the explicit Euler method,

$$\tilde{E}_{k+1} = E_k + hP^+\Big(-G_\varepsilon(E_k) + \gamma_k E_k\Big) \quad \text{with} \quad \gamma_k = \Big\langle P^+E_k, P^+G_\varepsilon(E_k)\Big\rangle \tag{34}$$

where G_ε has been computed according to (16).

We are not interested in following the exact solution of (25) but mainly to decrease the functional F_ε. This is equivalent to a gradient descent algorithm with line-search. According to this, for the stepsize selection we require that

$$F_\varepsilon(\tilde{E}_{k+1}) < F_\varepsilon(E_k). \tag{35}$$

Moreover, we apply an Armijo-like rule. For Equation (25) we have (recalling that P^+ is just the identity unless inequality constraints activate)

$$\frac{d}{dt}F_\varepsilon(E(t))\Big|_{t=t_k} = -\varepsilon\Big(\|P^+G_\varepsilon(E_k)\|^2 - \langle P^+G_\varepsilon(E_k), P^+E_k\rangle^2\Big) := -g_k \le 0.$$

We accept the result of the step with step size h and set $E_{k+1} = \tilde{E}_{k+1}$ if (35) holds, otherwise we reduce the stepsize to h/θ for some fixed $\theta > 1$, and repeat it. Moreover, if

$$F_\varepsilon(\tilde{E}_{k+1}) \ge F_\varepsilon(E_k) - (h/\theta)g_k, \tag{36}$$

then we reduce the step size for the next step to h/θ. If the step size has not been reduced in the previous step, we then increase the step-length to be used in $[t_{k+1}, t_{k+2}]$.

In a numerical solution of the gradient system (for example using the Euler-based scheme (34)), we have to monitor the sets of edges where $a_{ij} + \varepsilon e_{ij} = \delta$ and among those we must further track those edges where the sign of $-g_{ij} + \gamma e_{ij}$ changes (where g_{ij} is the (i,j)-th entry of the matrix $G_\varepsilon(E)$ and e_{ij} is the (i,j)-th entry of the matrix E). When the active set $\{(i,j) : a_{ij} + \varepsilon e_{ij} = \delta\}$ is changed, then γ also changes.

The computational cost of a single step requires the computation of the Perron eigenvector and the solution of the linear systems involving either the group inverse or, in the undirected case, the Moore-Penrose pseudoinverse (see (31)). In the undirected case we have to compute precisely m columns of the Moore-Penrose pseudoinverse.

6.3. **Outer iteration.** Due to the fact that in our experiments inequality constraints do not activate, we restrict our analysis of the outer iteration to this setting (where P^+ is the identical operator). A more elaborated explicit formula can be also deduced in the general case.

The case of inactive inequality constraints. Let $E(\varepsilon)$ denote the minimizer of the functional F_ε. Generically we expect that for a given perturbation size $\varepsilon < \widetilde{\varepsilon}_m$, $E(\varepsilon)$ is smooth, which means that the spectral radius of $A+\varepsilon E$ is a simple eigenvalue. If so, then $f(\varepsilon) := F_\varepsilon(E(\varepsilon))$ is a smooth function of ε and we can exploit its regularity to obtain a fast iterative method to converge to $\widetilde{\varepsilon}_m$ from the left. Otherwise we can use a bisection technique to approach $\widetilde{\varepsilon}_m$. An example is provided in Figure 4.

The following result provides an inexpensive formula for the computation of the derivative of $f(\varepsilon) = F_\varepsilon(E(\varepsilon))$, which will be useful in the construction of the outer iteration of the method.

Assumption 6.1. *We assume that $E(\varepsilon)$ is a smooth function of ε in some interval.*

We then have the following result.

Lemma 6.1. *Under Assumption* 6.1*, the function $f(\varepsilon) = F_\varepsilon(E(\varepsilon))$ is differentiable and its derivative equals (with $' = d/d\varepsilon$)*

$$f'(\varepsilon) = -\|G_\varepsilon(E(\varepsilon))\| \tag{37}$$

Proof. The conservation of $\|E(\varepsilon)\| = 1$ implies

$$\Big\langle E(\varepsilon), E'(\varepsilon) \Big\rangle = 0.$$

Differentiating $f(\varepsilon) = F_\varepsilon\big(E(\varepsilon)\big)$ with respect to ε we obtain

$$f'(\varepsilon) = \Big\langle G_\varepsilon(E(\varepsilon)), E(\varepsilon) + \varepsilon E'(\varepsilon) \Big\rangle = \Big\langle G_\varepsilon(E(\varepsilon)), E(\varepsilon) \Big\rangle. \tag{38}$$

By means of the property no. 3. of minimizers, stated in Theorem 5.1, we can conclude that $E(\varepsilon) \propto G_\varepsilon(E(\varepsilon))$, which yields formula (37). □

Newton method. For $\varepsilon < \widetilde{\varepsilon}_m$, we make use of the standard Newton iteration

$$\varepsilon_1 = \varepsilon_o - \frac{f(\varepsilon_o)}{f'(\varepsilon_o)}, \tag{39}$$

to get an updated value ε_1 from ε_o.

In a numerical code it is convenient to couple the Newton iteration (39) with the bisection method, to increase its robustness. We make use of a tolerance which allows us to distinguish numerically whether $\varepsilon < \widetilde{\varepsilon}_m$, in which case we may use the derivative formula and perform the Newton step, or $\varepsilon > \widetilde{\varepsilon}_m$, in which case we just bisect the current interval.

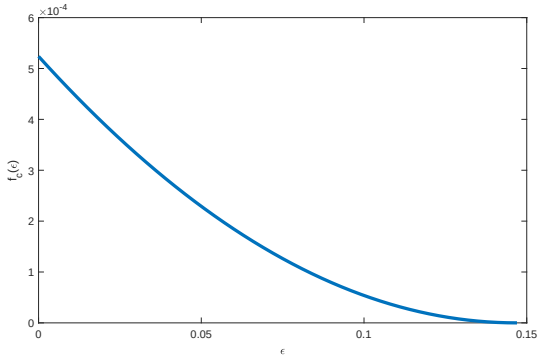


FIGURE 4. Behavior of $f(\varepsilon)$ for Example in Section 6.1.

7. Numerical tests

In this section we present numerical experiments illustrating the behavior of our algorithm on a few undirected and directed sparse weighted graphs arising in applications. We also perform a comparison with more standard approaches for solving constrained optimization problems.

All tests are made on a Surface Book 3 computer, under the MATLAB version 23.2.0.2365128 (R2023b).

7.1. **Example 1, matrix A_{27}.** The considered graph is one of a sequence of graphs arising from molecular dynamics simulations aimed at analyzing the structure of water networks under different temperature and pressure conditions (see [9]). It consists of 710 nodes (corresponding to oxygen atoms), most of which (613) have degree 4. Its pattern is shown in Figure 5. In Figure 6 we show the entries of the Perron eigenvector of the original (left picture) and perturbed (right picture), obtained for $m = 2$.

The five leading entries of the Perron eigenvector of A are the following:

$$\begin{aligned} &\mathrm{v}_{26} = 0.21705332, \quad \mathrm{v}_{29} = 0.24269500, \quad \mathrm{v}_{419} = 0.28145420, \\ &\mathrm{v}_{111} = 0.29422909, \quad \mathrm{v}_{315} = 0.37879622. \end{aligned}$$

After scaling the matrix A to have Frobenius norm 1, we apply the algorithm.

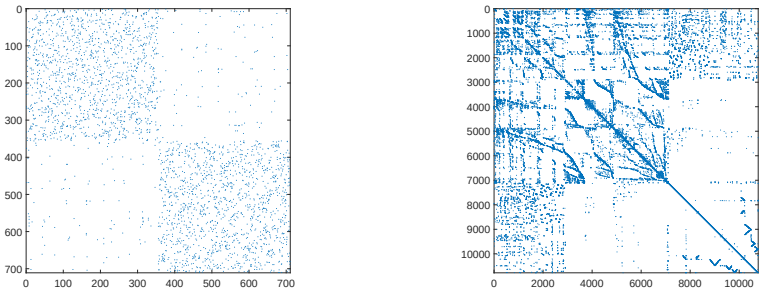


FIGURE 5. Patterns of the adjacency matrix for the graph example A_{27} (left) and for the graph example *nopoly* from Gaertner collection (right)

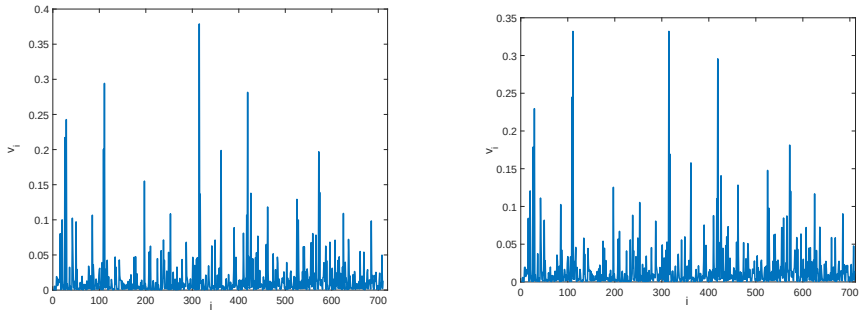


FIGURE 6. Distribution of the entries of Perron eigenvector for the example A_{27}. Left picture: original matrix. Right picture: perturbed matrix (for $m = 2$).

The norm of the perturbation which determines the coalescence of the two main entries of the Perron eigenvector is $\widetilde{\varepsilon}_2 = 0.0092654$, which is below 1%. The two leading entries after the perturbation are

$$\mathrm{v}_{111} = 0.\mathbf{331}01615, \qquad \mathrm{v}_{315} = 0.\mathbf{331}36357.$$

The number of outer iterations is 16. The CPU time is about 1.23 seconds.

Note that the relative variations of the two entries corresponding to the nodes with highest centrality are $\delta_{111} = +0.1281$ and $\delta_{315} = -0.1235$.

We also consider coalescence of the first $m = 3$ entries, for which we obtain a perturbation of norm $\widetilde{\varepsilon}_3 = 0.010718306874804$ with the 3 largest entries of Perron

k	ε_k	$f_c(\varepsilon_k)$	# eigs
0	0.005	0.000375176389506	25
1	0.005136626814311	0.000351719368954	2
2	0.008469968038737	0.000013522265277	34
3	0.008489282220273	0.000012884575588	2
6	0.009184673253577	0.000000185817345	2
10	0.009212325217551	0.000000093285281	1

TABLE 1. Computation of ε_k, $f_c(\varepsilon_k) = F_\varepsilon^c(E_k)$ and number of eigenvalue computations in the inner iterations for Example 1, matrix A_{27} with $m = 2$.

eigenvector v given by

$$\mathrm{v}_{419} = 0.\mathbf{3000}0085, \quad \mathrm{v}_{315} = 0.\mathbf{3000}2148, \quad \mathrm{v}_{111} = 0.\mathbf{3000}2174.$$

To obtain 4 digits of precision we required a smaller tolerance to step the outer iteration, which determined a CPU time of about 9.33 seconds. Finally we set $m = 4$ for which we obtain $\widetilde{\varepsilon}_4 = 0.014389966738210$, with the largest entries of the Perron eigenvector v being

$$\mathrm{v}_{109} = 0.\mathbf{2317}9527, \quad \mathrm{v}_{419} = 0.\mathbf{2318}0413, \quad \mathrm{v}_{111} = 0.\mathbf{2318}0684, \quad \mathrm{v}_{315} = 0.\mathbf{2318}0693.$$

The CPU time is about 16 seconds. It is interesting to note the the node 109 was not at the highest rank but replaced the node 29 after the perturbation.

Comparison with MATLAB *routines.* In a first experiment we have applied the MATLAB routine `fmincon` to the problem formulation (4)–(5) for the case $m = 2$. Here we use `fmincon` as a black box, without providing the analytic form of the gradient (the derivation of which is among the results of this paper) to the routine. Using standard default values we have not been able to obtain a correct solution, hence it has been necessary to increase the number of function evaluations allowed. Table 2 shows the results of the routine with an interior point algorithm, which is the one that performs best. The resulting solution is very close to the one we found, with coalescent entries #111 and #315,

$$\mathrm{v}_{111} = 0.\mathbf{33}2\ldots, \qquad \mathrm{v}_{315} = 0.\mathbf{33}2\ldots.$$

In Table 2, k stands for the iteration number, x is the vector containing the nonzero entries of the upper triangular part of the weighted adiacency matrix A, $g(x)$ is the constraint, $\mathrm{v}_1 - \mathrm{v}_2$ being $\mathrm{v}(x)$ the Perron eigenvector of the matrix $B = A - X$, where X is the perturbation associated to x. Finally, $f(x) = \sqrt{2}\|x\| = \|X\|_F$ and $\#fe$ indicates the number of function evaluations. The overall CPU time is about 80 minutes. Observing Table 2, we see that in the beginning x gets quite large with respect to its final value ($3.8 \cdot 10^1$ versus $9.27 \cdot 10^{-3}$). This very likely implies activation of inequality constraints and makes the convergence process slower. For this reason, trust region techniques would likely improve the process. The initial value x_0 provided to the algorithm is quite small (as $f(x_0)$ indicates), but the algorithm makes a large jump to a vector x_1 of much higher norm, which is due to the violation of the constraint. Starting from a consistent value would help, but requires additional computation.

k	# fe	$f(x_k)$	$g(x_k)$
0	1.4750e+03	6.9396e-04	8.4470e-02
1	2.9500e+03	2.4405e+01	7.6330e-02
6	1.0325e+04	3.8498e+01	4.3180e-04
11	1.7700e+04	7.3944e+00	2.0100e-03
21	3.2450e+04	6.7381e-01	2.5690e-03
31	4.7206e+04	6.6791e-01	1.7290e-06
41	6.1970e+04	5.1159e-02	2.3090e-02
51	7.6743e+04	4.9650e-02	2.9350e-04
61	9.1510e+04	4.0801e-02	1.0950e-02
71	1.0629e+05	1.4522e-02	4.7660e-03
81	1.2109e+05	9.5850e-03	3.9870e-05
91	1.3589e+05	9.2726e-03	3.5810e-10
96	1.4329e+05	9.2726e-03	5.8570e-10

TABLE 2. Reported behavior of the MATLAB algorithm.

Next we describe the results of a second experiment, with analytic gradients provided to the algorithm. Following the notation in Section 3.2, we have provided to the algorithm, the exact formula for the gradient of the constraint $u_{i_1} - u_{i_2} = 0$, which is obtained by the same arguments used in Lemma 5.1 and is proportional to

$$P_{\mathcal{E}}(\mathrm{Sym}(B^{\dagger}(u_{i_1} - u_{i_2})\mathrm{v}^{\top}).$$

The norm of the perturbation which determines the coalescence of the two main entries of the Perron eigenvector is $\widetilde{\varepsilon}_2 = 0.01023234$, which is about 1%. The two leading entries after the perturbation are

$$\mathrm{v}_{315} \approx \mathrm{v}_{636} = 0.20245843.$$

This indicates a different local optimum with respect to the previous experiment (which was the same as the one we found). However, they are very close and both indicate a strong sensitivity of the ranking for this graph. Table 3 shows the results of the run. The overall CPU time is about 351 seconds, which shows a significant improvement with respect to the previous experiment, although still far from the CPU time required by our approach.

7.2. **Example 2, matrix *nopoly*.** This matrix has been contributed to the TAMU Sparse Matrix Collection by Dr. Klaus Gaertner, Weierstrass Institute, Berlin[1]. The matrix, described as the adjacency matrix of a weighted undirected graph, is symmetric. However, some of the weights are negative. To make it non-negative we consider its absolute value. The corresponding graph consists of $n = 10774$ nodes. Its pattern is shown in Figure 5. We do not perform a comparison with MATLAB's optimization tools since the size of the matrix makes this prohibitively expensive.

The five leading entries of the Perron eigenvector of A are the following:

$$\mathrm{v}_{1155} = 0.12197226, \quad \mathrm{v}_{3391} = 0.12460486, \quad \mathrm{v}_{3002} = 0.12460960,$$
$$\mathrm{v}_{3007} = 0.12502517, \quad \mathrm{v}_{3397} = 0.12559705.$$

[1]See https://sparse.tamu.edu/Gaertner

k	# fe	$f(x_k)$	$g(x_k)$
0	1.475e+03	1.002577e-03	8.433e-02
1	2.950e+03	2.329900e+01	5.114e-02
21	3.2464e+04	4.892883e+00	5.230e-03
41	6.1970e+04	2.175929e-01	5.546e-03
61	9.1493e+04	1.814369e-01	6.202e-06
81	1.21020e+05	1.786729e-02	1.009e-03
101	1.50620e+05	1.025323e-02	9.777e-08
121	1.80195e+05	1.023304e-02	3.949e-07
141	2.09771e+05	1.023236e-02	1.121e-07
147	2.18661e+05	1.023233e-02	2.660e-09

TABLE 3. Reported behavior of the MATLAB algorithm provided by the analytic gradient of the constraint.

After normalizing the matrix A in the Frobenius norm, we apply the algorithm with the aim of obtaining the largest m entries of the Perron eigenvector to match to a 3-digit accuracy.

First we apply the algorithm with $m = 2$. The method starts with $\varepsilon_0 = 10^{-3}$ and sets the tolerance for the stopping criterion to 10^{-6}. It makes 3 outer iterates and the final value of the functional is $8 \cdot 10^{-12}$. The norm of the perturbation which determines the matching of the 2 leading entries of the Perron eigenvector is $\widetilde{\varepsilon}_2 = 0.00383633$, which is about 0.00008%. The computed 2 leading entries are the following:

$$\mathrm{v}_{3007} = 0.\mathbf{12558}011, \quad \mathrm{v}_{3397} = 0.\mathbf{12558}553.$$

The CPU time is about 19.2 seconds.

Then we apply the algorithm with $m = 3$. The method starts with $\varepsilon_0 = 10^{-3}$ and sets the tolerance for the stopping criterion to 10^{-10}. The method makes 8 outer iterates. The norm of the perturbation which determines the coalescence of the 3 leading entries of the Perron eigenvector is $\widetilde{\varepsilon}_3 = 0.003909011$, which is about 0.00008%. The computed 3 leading entries are the following.

$$\mathrm{v}_{3007} = 0.\mathbf{12550}017, \quad \mathrm{v}_{3002} = 0.\mathbf{12550}316, \quad \mathrm{v}_{3397} = 0.\mathbf{12550}872$$

which coincide to a 5-digit approximation. The CPU time is about 30.8 seconds.

Next we set $m = 5$. The norm of the perturbation which determines the coalescence of the 5 leading entries of the Perron eigenvector is $\widetilde{\varepsilon}_5 = 0.0161$, which is about 0.00016%. The new leading entries follow:

$$\mathrm{v}_{1155} = 0.\mathbf{124}48751, \quad \mathrm{v}_{3391} = 0.\mathbf{124}51103, \quad \mathrm{v}_{3007} = 0.\mathbf{124}53422,$$
$$\mathrm{v}_{3002} = 0.\mathbf{124}60202, \quad \mathrm{v}_{3397} = 0.\mathbf{124}63431.$$

The CPU time is about 321.8 seconds.

Finally, in Figure 7 we show the entries of the Perron eigenvector of the perturbed matrix obtained setting $m = 10$.

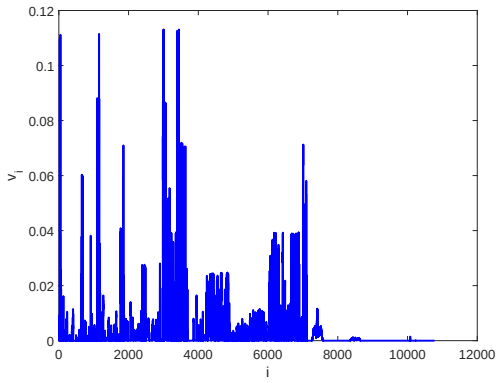


FIGURE 7. Distribution of the entries of Perron eigenvector of the matrix obtained for $m = 10$ for the example *nopoly* from Gaertner collection.

8. Conclusions

In this paper we have presented a methodology for finding a minimal perturbation to the adjacency matrix of a weighted (sparse) graph that causes the leading entries of the Perron eigenvector of the perturbed matrix to coalesce. The perturbation is constrained so as to preserve the sparsity of the original matrix. One of the motivations for this work was the question of the reliability and robustness of eigenvector centrality: if a small perturbation causes coalescence of the leading entries of the Perron eigenvector, this centrality measure cannot be considered reliable in the presence of uncertainties or noise, especially when the leading entries of the perturbed eigenvector differ from the original ones and the size of the perturbation is within the uncertainty limits. Our methodology easily allows to incorporate additional constraints; for example, only a subset of the graph's edge weights may be allowed to change. This feature can be used to identify which edges play an important role in influencing the ranking of nodes in the graph. The proposed algorithm appears to perform favorably when compared with standard constrained optimization methods on graphs arising from realistic applications, and is fairly efficient in spite of its complexity.

References

[1] E. Andreotti, D. Edelmann, N. Guglielmi and C. Lubich, *Constrained graph partitioning via matrix differential equations*, SIAM J. Matrix Anal. Appl., 40(1):1–22, 2019.

[2] P. Benner and M. Voigt, *A structured pseudospectral method for H_∞-norm computation of large-scale descriptor systems*, Math. Control Signals Systems, 26(2):303–338, 2014.

[3] M. Benzi, *A direct projection method for Markov chains*, Linear Algebra Appl., 386:27–49, 2004.
[4] M. Benzi and P. Boito, *Matrix functions in network analysis*, GAMM-Mitt., 43(3):e202000012, 2020.
[5] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, 14:1–137, 2005.
[6] M. Benzi and C. Klymko, *On the limiting behavior of parameter-dependent network centrality measures*, SIAM J. Matrix Anal. Appl., 36(2):686–706, 2015.
[7] S. Brin and L. Page, *The anatomy of a large-scale hypertextual Web search engine*, Computer Networks and ISDN Systems, 33:107–117, 1998.
[8] S. Cipolla, F. Durastante, and B. Meini, *Enforcing Katz and PageRank Centrality Measures in Complex Networks*, arXiv:2409.02524v1, 2024.
[9] C. Faccio, M. Benzi, L. Zanetti-Polzi and I. Daidone, *Low- and high-density forms of liquid water revealed by a new medium-range order descriptor*, J. Mol. Liquids, 355:118922, 2022.
[10] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, 2013.
[11] C. Geiger and C. Kanzow, *On the resolution of monotone complementarity problems*, Comput. Optim. Appl., 5(2):155–173, 1996.
[12] N. Gillis and P. Van Dooren, *Assigning Stationary Distributions to Sparse Stochastic Matrices*, SIAM J. Matrix. Anal. Appl., 45(4):2184–2210, 2024.
[13] G. H. Golub and C. F. Van Loan, *Matrix Computations. Fourth Edition*, Johns Hopkins University Press, Baltimore and London, 2013.
[14] A. Greenbaum, R.-C. Li, and M. L. Overton, *First-order perturbation theory for eigenvalues and eigenvectors*, SIAM Rev., 62(2):463–482, 2020.
[15] N. Guglielmi, D. Kressner, and C. Lubich, *Low rank differential equations for Hamiltonian matrix nearness problems*, Numer. Math., 129:279–319, 2015.
[16] N. Guglielmi and C. Lubich, *Matrix Nearness Problems and Eigenvalue Optimization*, arXiv:2503.14750, 240 pages, 2025.
[17] N. Guglielmi, C. Lubich, and V. Mehrmann, *On the nearest singular matrix pencil*, SIAM J. Matrix Anal. Appl., 38(3):776–806, 2017.
[18] N. Guglielmi, C. Lubich, and S. Sicilia, *Rank-1 matrix differential equations for structured eigenvalue optimization*, SIAM J. Numer. Anal., 61(4):1737–1762, 2023.
[19] N. Guglielmi, M. Overton, and G. W. Stewart, *An efficient algorithm for computing the generalized null space decomposition*, SIAM J. Matrix Anal. Appl., 36(1):38–54, 2015.
[20] N. Guglielmi and S. Sicilia, *A low-rank ODE for spectral clustering stabilit*, Linear Algebra Appl., 721(1):250–276, 2025.
[21] N. J. Higham, *Computing a nearest symmetric positive semidefinite matrix*, Linear Algebra Appl., 103:103–118, 1988.
[22] N. J. Higham, *Computing the nearest correlation matrix—a problem from finance*, IMA J. Numer. Anal., 22(3):329–343, 2002.
[23] D. Hinrichsen and A. J. Pritchard, *Stability radius for structured perturbations and the algebraic Riccati equation*, Systems Control Lett., 8(2):105–113, 1986.
[24] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1991.
[25] T. Kato, *Perturbation Theory for Linear Operators*, Springer Verlag, New York, N.Y., 1995.
[26] D. Kressner and M. Voigt, *Distance problems for linear dynamical systems*, in P. Benner *et al.* (Eds.). *Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory*, pages 559–583, Springer, Cham, 2015.
[27] E. Landau, *Zur relativen Wertbemessung der Turnierresultate*, Deutsches Wochenschach, 11:366–369, 1895.
[28] A. N. Langville and C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, Princeton, NJ, 2006.
[29] C. D. Meyer and G. W. Stewart, *Derivatives and perturbations of eigenvectors*, SIAM J. Numer. Anal., 25(3):679–691, 1988.
[30] M. E. J. Newman, *Networks: an Introduction*, Oxford University Press, Oxford, 2010.
[31] V. Nicosia, R. Criado, M. Romance, G. Russo, and V. Latora, *Controlling centrality in complex networks*, Sci. Rep., 2(1), Art. No. 218, 2012.
[32] C. C. Paige and M. A. Saunders, *Solutions of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12(4):617–629, 1975.

Scuola Normale Superiore, Piazza dei Cavalieri 7, Pisa, Italy.
E-mail address: michele.benzi@sns.it

Gran Sasso Science Institute (GSSI), via Crispi 7, I-67010 L' Aquila, Italy.
E-mail address: nicola.guglielmi@gssi.it