MEASURING STABILITY OF SPECTRAL CLUSTERING

ELEONORA ANDREOTTI*, DOMINIK EDELMANN†, NICOLA GUGLIELMI†, AND CHRISTIAN LUBICH‡

Abstract. As an indicator of the stability of spectral clustering of an undirected weighted graph into $k$ clusters, the $k$th spectral gap of the graph Laplacian is often considered. The $k$th spectral gap is characterized here as an unstructured distance to ambiguity, namely as the minimal distance of the Laplacian to arbitrary symmetric matrices with vanishing $k$th spectral gap. As a more appropriate measure of stability, the structured distance to ambiguity of the $k$-clustering is introduced as the minimal distance of the Laplacian to Laplacians of the same graph with weights that are perturbed such that the $k$th spectral gap vanishes. To compute a solution to this matrix nearness problem, a two-level iterative algorithm is proposed that uses 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 structured and unstructured distances to ambiguity are compared on some example graphs. The numerical experiments show, in particular, that selecting the number $k$ of clusters according to the criterion of maximal stability can lead to different results for the structured and unstructured stability indicators.

Key words. Spectral clustering; clustering stability; matrix nearness problem; structured eigenvalue optimization.

AMS subject classifications. 15A18, 65K05

1. Introduction. Clustering — identifying groups with similarities in a large data set — is a fundamental task in data analysis. In order to cluster an undirected weighted graph, several algorithms are considered in the literature. Spectral clustering (originating with Fiedler [5]; see [19] and [21] for more recent accounts) offers significant advantages over more traditional combinatorial techniques such as direct $k$-means or single linkage. Quoting [21], "Results obtained by spectral clustering often outperform the traditional approaches; spectral clustering is very simple to implement and can be solved efficiently by linear algebra methods."

The stability of spectral clustering algorithms is heuristically — and very conveniently — associated with the spectral gap in the graph Laplacian, that is the difference between the $k$th and $(k + 1)$th smallest eigenvalues of the Laplacian matrix in the case of clusters built from $k$ eigenvectors. Contrary to this, we formulate an ambiguity problem where it is asked how far the given Laplacian matrix is from the Laplacian matrix of a weighted graph with the same edges for which the $k$th and $(k + 1)$th smallest eigenvalues of the graph Laplacian coalesce and for which therefore spectral clustering with $k$ clusters becomes ambiguous. This structured distance to ambiguity is introduced here as a measure of stability of spectral $k$-clustering, for any number $k$ of clusters. On the other hand, the $k$th spectral gap divided by $\sqrt{2}$ is characterized as the distance of the Laplacian matrix to the much larger set of arbitrary, unstructured symmetric matrices with coalescing $k$th and $(k + 1)$th eigenvalues.

A stability indicator is useful in selecting an appropriate number $k$ of clusters, by choosing $k$ such that the stability indicator is maximized. The structured distance to ambiguity, whose computation will be discussed in this paper, may be too costly computationally to be used as a general-purpose stability indicator in comparison with the spectral gap, which is available without further computations in spectral clustering. Nevertheless, comparing the structured distance to ambiguity and the spectral gap for representative examples within a class of graphs of interest gives insight into the reliability (or otherwise) of the spectral gap as a stability indicator.

*Dipartimento di Fisica, Università degli Studi di Torino, Via Pietro Giuria 1, 10125 Torino, Italy. Email: eleonora.andreotti@unito.it
†Gran Sasso Science Institute, via Crispi 7, L’Aquila, Italy. Email: nicola.guglielmi@gssi.it
‡Mathematisches Institut, Universität Tübingen, Auf der Morgenstelle 10, D–72076 Tübingen, Germany. Email: lubich@ma.uni-tuebingen.de, dominik.edelmann@ma.uni-tuebingen.de
The structured distance to ambiguity considered in this paper is similar in spirit to other structured robustness measures that arise in matrix and control theory, such as stability radii, passivity distances, distance to singularity, etc. [13, 11, 12, 4, 7, 10, 17]. The notion of stability considered here is, however, different from [22], where a statistical perspective on clustering stability is developed.

A main objective of this paper is to show how the structured distance to ambiguity can be computed. The proposed algorithm is an iterative algorithm, where in each step a pair of eigenvalues and associated eigenvectors of the Laplacian of a graph with perturbed weights are computed. For a large sparse graph (where the number of edges leaving any vertex is moderately bounded), these computations can typically be done with a complexity that is linear in the number of vertices.

A feature of this algorithm in common with recent algorithms for eigenvalue optimization as given in [8, 7, 10, 9, 3] is a two-level procedure for matrix nearness problems, where in an inner iteration a gradient flow drives perturbations to the original matrix of a fixed size into a (local) minimum of a nonnegative functional that depends on eigenvalues and eigenvectors, and 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 recall basic concepts of spectral clustering. In Section 3 we introduce the structured distance to ambiguity (SDA) as a measure of stability for spectral $k$-clustering. In Section 4 we describe a numerical method to compute the SDA, which requires the solution of a structured matrix nearness problem. 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 5 we present the results of numerical experiments where the spectral gap (i.e., the unstructured distance to ambiguity) and the theoretically more appropriate measure of stability discussed here (i.e., the structured distance to ambiguity) are compared on some classes of graphs.

2. Spectral clustering. Let $G = (V, E)$ be an undirected graph with vertex set $V = \{1, \ldots, n\}$ and edge set $E \subset V \times V$. We assume that the graph is weighted, that is, to each edge $(i, j) \in E$ between two vertices $i$ and $j$ a non-negative weight $w_{ij} = w_{ji} \geq 0$ is associated. We set $w_{ij} = 0$ for $(i, j) \notin E$. The weighted adjacency matrix of the graph is the matrix

$$W = (w_{ij}) \in \mathbb{R}^{n \times n}.$$ 

The degrees $d_i = \sum_{j=1}^{n} w_{ij}$ are the elements of the diagonal matrix

$$D = \text{diag}(d_i), \quad d_i = (W \mathbb{1})_i, \quad \text{where } \mathbb{1} := (1, \ldots, 1)^T \in \mathbb{R}^n.$$ 

The (unnormalized) Laplacian of the graph is given by the matrix $L = L(W)$,

$$L = D - W,$$

i.e., $L(W) = \text{diag}(W \mathbb{1}) - W$.

The Laplacian matrix $L$ is non-negative and positive semi-definite; since - by construction - $L \mathbb{1} = 0$, $\lambda_1 = 0$ is the smallest eigenvalue of $L$. Note that the matrix $L$ does not depend on (possible) diagonal elements of the matrix $W$, which means that self-edges do not change the graph Laplacian. The graph Laplacian and its eigenvectors provide important instruments to spectral clustering, as stated by the following theorems.

**Theorem 2.1 (Bi-partition, M. Fiedler [5]).** Let $W \in \mathbb{R}^{n \times n}$ be the weight matrix of an undirected graph $G$ and $L$ the corresponding Laplacian matrix. Let $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ be the eigenvalues of $L$. Then, the graph is disconnected if and only if $\lambda_2 = 0$. Moreover, if $0 = \lambda_2 < \lambda_3$, then the entries of the corresponding eigenvector orthogonal to $\mathbb{1}$ assume only two different values, of different sign, which mark the membership to the two connected components.
If \( \lambda_2 \) is a simple eigenvalue, then the corresponding eigenvector is known as the Fiedler vector. In spectral graph theory, inspired by Theorem 2.1, it is common to compute the second smallest eigenvalue of \( L \) and label the positive components of the Fiedler vector as belonging to one subset and the negative ones to another subset, and in this way obtaining a natural partition of the graph. However, this becomes unreliable when a small perturbation of the weights yields a coalescence of the eigenvalues \( \lambda_2 \) and \( \lambda_3 \).

More generally we have the following result (see e.g. [21]). For a subset of vertices \( C \subset V \), we here denote the indicator vector \( 1_C \) as the vector whose \( i \)th entry is equal to 1 if \( v_i \in C \) and is equal to zero otherwise.

**Theorem 2.2** (\( k \)-partition). Let \( W \in \mathbb{R}^{n \times n} \) be the weight matrix of an undirected graph \( G \) and \( L \) the corresponding Laplacian matrix. Then the multiplicity \( k \) of the eigenvalue \( 0 \) (the dimension of \( \ker(L) \)) equals the number of connected components \( C_1, \ldots, C_k \) in the graph. The eigenspace of the eigenvalue \( 0 \) is spanned by the indicator vectors \( 1_{C_1}, \ldots, 1_{C_k} \).

Nonempty sets \( C_1, \ldots, C_k \) form a clustering of the graph if

\[
C_i \cap C_j = \emptyset \quad \text{for } i, j = 1, \ldots, k, \; i \neq j \quad \text{and} \quad \bigcup_{i=1}^{k} C_i = V.
\]

Similarly to the case of two clusters, this result motivates an algorithm for clustering a graph into \( k \) components, which is reviewed in Algorithm 1; see [21]. For the classical \( k \)-means algorithm we refer the reader e.g. to [16].

### Algorithm 1: Unnormalized spectral clustering algorithm

**Data:** weight matrix \( W \), number \( k \) of clusters  
**Result:** Clusters \( C_1, \ldots, C_k \)

**begin**

1. Compute unnormalized Laplacian \( L = D - W \)
2. Compute the first \( k \) eigenvectors (i.e. those associated to the smallest eigenvalues of \( L \) but 0) \( x_1, \ldots, x_k \) of \( L \)
3. Let \( X = [x_1 \mid x_2 \mid \ldots \mid x_k] \) the matrix whose columns are given by the computed eigenvectors
4. For \( i = 1, \ldots, n \) let \( r_i \in \mathbb{R}^k \) the vector given by the \( i \)-th row of \( X \)
5. Cluster the points \( (r_i)_{i=1,\ldots,n} \) by the \( k \)-means algorithm into \( k \) clusters \( C_1, \ldots, C_k \)
6. Return \( C_1, \ldots, C_k \)

**end**

Analogous results and algorithms are extended to the normalized Laplacian

\[
L_{\text{Sym}} := D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2}.
\]

### 3. Structured distance to ambiguity as a stability measure for spectral clustering.

In order to evaluate the robustness of the spectral clustering algorithm it is essential to quantify the sensitivity to perturbations of the invariant subspace associated with the \( k \) eigenvectors used by the algorithm. Suppose that a small perturbation to the weights of the graph makes the eigenvalues \( \lambda_k \) and \( \lambda_{k+1} \) of the Laplacian coalesce. Then the eigenvector associated to the computed \( k \)th smallest eigenvalue can change completely and hence can yield a different clustering. It has been suggested in the literature to use the spectral gap \( \lambda_{k+1} - \lambda_k \) as an indicator of the stability of the \( k \)-clustering. For unstructured perturbations of the graph Laplacian, this is motivated by the Davis–Kahan theorem (see, e.g., [20] and [6]), which tells us that the distance
between the eigenspaces of the Laplacian of a graph and of any perturbed symmetric matrix has a bound that is proportional to the perturbation size and inversely proportional to the spectral gap $\lambda_{k+1} - \lambda_k$ of the Laplacian $L(W)$. In another direction, which is more related to the concepts studied here, the Hoffman–Wielandt theorem [15, p. 368] yields that the spectral gap is characterized as an unstructured distance to ambiguity (up to the scaling factor $\sqrt{2}$),

$$\frac{\lambda_{k+1} - \lambda_k}{\sqrt{2}} = \min_{\hat{L}} \|L(W) - \hat{L}\|_F$$

under the condition that $\hat{L}$ is a symmetric matrix such that its $k$th and $(k + 1)$th eigenvalues coalesce.

Here, $\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$ denotes the Frobenius norm of a matrix $A = (a_{ij})$.

The interpretation is that applying Algorithm 1 is justified when the gap is relatively large, otherwise a small perturbation of the weight matrix may yield coalescence of $\lambda_k$ and $\lambda_{k+1}$ and change significantly the clustering.

The weakness of this approach is that it considers unstructured perturbations of the Laplacian, as opposed to the admissible perturbations that are Laplacians $L(\hat{W})$ of a perturbed weight matrix that preserves the symmetry and non-negativity, i.e., $\hat{w}_{ij} = \hat{w}_{ji} \geq 0$, and the sparsity pattern of $W$, i.e., $\hat{w}_{ij} = 0$ for $(i,j) \notin \mathcal{E}$. In (3.1), the minimizer is given by

$$L_* = L(W) + \frac{1}{2}(\lambda_{k+1} - \lambda_k)x_kx_k^T - \frac{1}{2}(\lambda_{k+1} - \lambda_k)x_{k+1}x_{k+1}^T,$$

where $x_k$ and $x_{k+1}$ denote normalized eigenvectors of $L(W)$ to the eigenvalues $\lambda_k$ and $\lambda_{k+1}$, respectively. Apart from exceptional cases, $L_*$ is not the Laplacian of a graph.

We therefore propose the following stability measure for $k$-clustering, which is given by a structured distance to ambiguity:

$$\delta_k(W) = \min_{\hat{W}} \|L(W) - L(\hat{W})\|_F$$

under the condition that the $k$th and $(k + 1)$th eigenvalues of $L(\hat{W})$ coalesce and under the constraints $\hat{w}_{ij} = \hat{w}_{ji} \geq 0$ for $(i,j) \in \mathcal{E}$ and $\hat{w}_{ij} = 0$ for $(i,j) \notin \mathcal{E}$.

In view of (3.1), it is clear that

$$\delta_k(W) \geq (\lambda_{k+1} - \lambda_k)/\sqrt{2},$$

and often $\delta_k(W)$ is substantially larger than the scaled spectral gap. In view of Theorem 2.1, we further note that $\delta_1(W)$ is the Frobenius-norm distance of the Laplacian $L(W)$ to that of a nearest disconnected graph.

For clustering a graph it is usually not known beforehand what the best choice of the number $k$ of clusters is. Asking for the most stable clustering of a graph (most stable with respect to admissible perturbations of the weights) determines the optimal number $k$ as

$$k_{\text{opt}}(W) = \arg \max_k \delta_k(W),$$

where $k$ can be limited to $k \leq k_{\text{max}}$, where $k_{\text{max}}$ is given a priori or chosen such that $\lambda_{k_{\text{max}}}$ is smaller than some threshold. This criterion for the selection of $k$ is considered here instead of choosing $k$ such that the spectral gap $\lambda_{k+1} - \lambda_k$ is maximized. The latter is computationally cheaper, but appears less meaningful. In our numerical experiments in Section 5 we will compare the two criteria for some families of graphs.
Remark 3.1. There are obvious alternatives to the above stability measure:

- Instead of minimizing the perturbation of the Laplacian we might minimize the perturbation of the weights, \( \| W - \tilde{W} \|_F \).
- Instead of the unnormalized Laplacian \( L(W) = D - W \) with \( D = \text{diag}(W) \) we might work with the normalized Laplacian \( I - D^{-1/2}WD^{-1/2} \).

In the following we concentrate on \( \delta_k(W) \) as given above, but our algorithm for computing \( \delta_k(W) \) is readily extended to the other two cases.

4. Computation of the structured distance to ambiguity. In this section we describe an approach to compute the stability measure \( \delta_k(W) \) defined in the previous section.

4.1. Outline of the computational approach. Our approach is summarized by the following two-level method:

- **Inner iteration**: Given \( \varepsilon > 0 \), we aim to compute a symmetric matrix \( E = (e_{ij}) \in \mathbb{R}^{n \times n} \) with the same sparsity pattern as \( W \) (i.e., \( e_{ij} = 0 \) if \( w_{ij} = 0 \)), with \( L(E) = \text{diag}(E) - E \) of unit Frobenius norm, with \( W + \varepsilon E \geq 0 \) (with componentwise inequality) such that the difference between the \( k \)th smallest and the \((k+1)\)th smallest eigenvalue of \( L(W + \varepsilon E) \) is minimal. The obtained minimizer is denoted by \( E(\varepsilon) \).

- **Outer iteration**: We compute the smallest value of \( \varepsilon \) such that the \( k \)th and \((k+1)\)th eigenvalues of \( \text{Lap}(W + \varepsilon E(\varepsilon)) \) coalesce.

In order to compute \( E(\varepsilon) \) for a given \( \varepsilon > 0 \), we make use of a constrained gradient system for the functional

\[
F_\varepsilon(E) = \lambda_{k+1}(L(W + \varepsilon E)) - \lambda_k(L(W + \varepsilon E)), \tag{4.1}
\]

under the constraints of unit Frobenius norm of \( L(E) = \text{diag}(E) - E \), the non-negativity \( W + \varepsilon E \geq 0 \) and the symmetry and the sparsity pattern of \( E \).

In the outer iteration we compute the optimal \( \varepsilon \), denoted \( \varepsilon_\lambda^* \), by a combined Newton-bisection method. We then have

\[
\delta_k(W) = \varepsilon_\lambda^*.
\]

The algorithm computes a perturbed weight matrix \( W_\varepsilon = W + \varepsilon E(\varepsilon)^* \) whose Laplacian has coalescent \( k \)th and \((k+1)\)th eigenvalues, and \( \varepsilon_\lambda^* = \|L(W) - L(W_\varepsilon)\|_F \).

4.2. Gradient of the functional \( F_\varepsilon \). We denote by \( \| \cdot \| = \| \cdot \|_F \) the Frobenius norm on \( \mathbb{R}^{n \times n} \) and by \( \langle X, Y \rangle = \text{trace}(X^T Y) \) the corresponding inner product.

For the edge set \( \mathcal{E} \) and for an arbitrary matrix \( A = (a_{ij}) \in \mathbb{R}^{n \times n} \), we define the symmetric projection onto the sparsity pattern given by \( \mathcal{E} \) as

\[
P_\mathcal{E}(A)_{ij} := \begin{cases} \frac{1}{2} (a_{ij} + a_{ji}) & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases}
\]

We denote by \( \text{Sym}(\mathcal{E}) \) the set of symmetric matrices with sparsity pattern \( \mathcal{E} \) and note that \( P_\mathcal{E} \) is the orthogonal projection from \( \mathbb{R}^{n \times n} \) to \( \text{Sym}(\mathcal{E}) \): \( \langle P_\mathcal{E}(A), W \rangle = \langle A, W \rangle \) for all \( W \in \text{Sym}(\mathcal{E}) \).

The Laplacian operator is a linear map

\[
L : \text{Sym}(\mathcal{E}) \rightarrow \mathbb{R}^{n \times n}.
\]

Its adjoint with respect to the Frobenius inner product,

\[
L^* : \mathbb{R}^{n \times n} \rightarrow \text{Sym}(\mathcal{E}),
\]
is given in the following lemma.

**Lemma 4.1.** For $V \in \mathbb{R}^{n \times n}$, let $L^*(V) \in \text{Sym}(\mathcal{E})$ be defined by $(L^*(V), W) = \langle V, L(W) \rangle$ for all $W \in \text{Sym}(\mathcal{E})$. Then,

$$
L^*(V) = P_{\mathcal{E}}(\text{diagvec}(V) \mathbb{1}^T - V),
$$

where $\text{diagvec}(V) \in \mathbb{R}^n$ is the vector of the diagonal entries of $V$. Furthermore, if $\mathcal{E}$ contains no auto-loops, i.e., $(i, i) \notin \mathcal{E}$ for all $i = 1, \ldots, n$, then we have for $W \in \text{Sym}(\mathcal{E})$,

$$
L^*(L(W)) = P_{\mathcal{E}}(d \mathbb{1}^T) + W \quad \text{with} \quad d = W \mathbb{1}.
$$

**Proof.** We have, for all $V \in \mathbb{R}^{n \times n}$ and $W \in \text{Sym}(\mathcal{E})$,

$$
\langle V, L(W) \rangle = \langle V, \text{diag}(W \mathbb{1}) - W \rangle = \langle \text{diagvec}(V), W \mathbb{1} \rangle - \langle V, W \rangle = \langle \text{diagvec}(V) \mathbb{1}^T, W \rangle - \langle V, W \rangle = \langle P_{\mathcal{E}}(\text{diagvec}(V) \mathbb{1}^T - V), W \rangle,
$$

and the result follows. \hfill \square

We make use of the following result \cite[Lemma 3.2]{Andreotti_Edelmann_Guglielmi_Lubich}, which provides a basic tool for the derivation of the gradient system we use to solve the problem.

**Lemma 4.2.** Consider a regular path $E(t)$ of feasible matrices, that is such that $W + \varepsilon E(t) \geq 0$ with $E(t)$ symmetric and with the same sparsity pattern of $W$, denote by $\lambda(t)$ a simple eigenvalue of the Laplacian matrix $L(W + \varepsilon E(t))$ and by $x(t)$ the associated normalized eigenvector, i.e. such that $\|x(t)\| = 1$. Then (omitting the argument $t$)

$$
\dot{\lambda} = \varepsilon \langle g_{\varepsilon}(E), \dot{E} \rangle, \quad \text{where} \quad g_{\varepsilon}(E) = L^*(xx^T - yy^T) = P_{\mathcal{E}}((x \bullet x) \mathbb{1}^T - xx^T),
$$

where $x \bullet x = (x_i^2) \in \mathbb{R}^n$ denotes the vector of squares of the entries of the vector $x = (x_i) \in \mathbb{R}^n$.

An immediate consequence of Lemma 4.2 is the following:

**Corollary 4.3.** Under the assumption of Lemma 4.2 we have that the gradient of the functional $F_\varepsilon(E)$ is given by

$$
G_{\varepsilon}(E) = L^*(xx^T - yy^T) = P_{\mathcal{E}}((x \bullet x - y \bullet y) \mathbb{1}^T - (xx^T - yy^T))
$$

where $x$ is the normalized eigenvector associated to $\lambda_{k+1}$ and $y$ is the normalized eigenvector associated to $\lambda_k$.

### 4.3. Gradient flow under the unit-norm constraint

We consider now the constrained gradient system associated to the functional $F_\varepsilon$ with the constraint that $L(E(t))$ has unit Frobenius norm. We then get the following matrix differential equation:

$$
\dot{E} = -G_{\varepsilon}(E) + \kappa L^*(L(E)) \quad \text{with} \quad \kappa = \frac{\langle G_{\varepsilon}(E), L^*(L(E)) \rangle}{\| L^*(L(E)) \|^2_F}.
$$

Here we remark that the denominator cannot vanish, because $L^*(L(E)) = 0$ would imply that the role of $\kappa$ is that of a Lagrange multiplier that ensures that the constraint $\|L(E)\|_F = 1$ is satisfied. The given formula for $\kappa$ is obtained as follows: Differentiating the constraint $\|L(E)\|_F^2 = \langle L(E), L(E) \rangle = 1$ gives $\langle L^*(L(E)), \dot{E} \rangle = \langle L(E), \dot{L}(E) \rangle = \frac{d}{dt}\|L(E)\|_F^2 = 0$. Taking the inner product of both sides of the differential equation with $L^*(L(E))$ yields

$$
0 = -\langle L^*(L(E)), G_{\varepsilon}(E) \rangle + \kappa \| L^*(L(E)) \|^2_F,
$$

which gives $\kappa$. 
4.4. Non-negativity constraint. It may happen that along the solution trajectory of (4.4) some positive entry of $W + \varepsilon E$ becomes negative. In our experiments we never observed such a situation although this is a potential occurrence. In this subsection we explain how to deal with the further constraint $W + \varepsilon E(t) \geq 0$ for all $t$.

One possibility would be to follow the lines of [3] and consider KKT conditions by managing active constraints and integrating a piecewise smooth system of differential equations. Another possibility is to add a penalization term such as the following:

$$Q_\varepsilon(E) = \frac{1}{2} \sum_{(i,j) \in \mathcal{E}} (w_{ij} + \varepsilon e_{ij})^2$$

where $(a)_- = \min(a, 0)$, and to minimize the functional

$$F_{\varepsilon, c}(E) = F_\varepsilon(E) + c Q_\varepsilon(E)$$

for increasing $c$ and letting $c \to \infty$. We have

$$\nabla Q_\varepsilon(E) = -\varepsilon \sum_{(i,j) \in \mathcal{E}, w_{ij} + \varepsilon e_{ij} < 0} (w_{ij} + \varepsilon e_{ij}) = -\varepsilon (W + \varepsilon E)_-,$$

giving an extra term to the gradient system (4.4).

With the notation

$$G_{\varepsilon, c}(E) = G_\varepsilon(E) + c \nabla Q_\varepsilon(E),$$

(4.5)

the differential equation (4.4) is then replaced by

$$\dot{E} = -G_{\varepsilon, c}(E) + \kappa L^*(L(E)) \quad \text{with} \quad \kappa = \frac{\langle G_{\varepsilon, c}(E), L^*(L(E)) \rangle}{\|L^*(L(E))\|_F^2}.$$  

(4.6)

4.5. Monotonicity and stationary points. The following monotonicity result follows directly from the construction of the gradient system.

**Theorem 4.4.** Let $\dot{E}(t)$ of unit Frobenius norm satisfy the differential equation (4.6) with $G_{\varepsilon, c}(E)$ of (4.5). Then, $F_{\varepsilon, c}(E(t))$ decreases monotonically with $t$. Moreover, if $W + \varepsilon E(t) \geq 0$ for all $t$, then

$$\frac{d}{dt} (\lambda_{k+1}(t) - \lambda_k(t)) \leq 0,$$

where $\lambda_k(t)$ and $\lambda_{k+1}(t)$ are the $k$th and $(k + 1)$th eigenvalues of $L(W + \varepsilon E(t))$.

**Proof.** Using (4.6) we obtain, with $G = G_{\varepsilon, c}(E)$,

$$\frac{1}{\varepsilon} \frac{d}{dt} F_{\varepsilon, c}(E(t)) = \langle G, \dot{E} \rangle = \langle G, -G - \kappa L^*(L(E)) \rangle = -\|G\|_F^2 + \frac{\langle G, L^*(L(E)) \rangle^2}{\|L^*(L(E))\|_F^2} \leq 0,$$

(4.7)

where the final inequality follows directly from the Cauchy–Schwarz inequality. This yields the monotone decay of $F_{\varepsilon, c}(E(t))$ and hence also the second statement, since $F_{\varepsilon, c}(E(t)) = \lambda_{k+1}(t) - \lambda_k(t)$ if $W + \varepsilon E(t) \geq 0$. ☐

Equilibrium points of (4.6) are characterized as follows.

**Theorem 4.5.** The following statements are equivalent along solutions of (4.6):

1. $\frac{d}{dt} F_{\varepsilon, c}(E(t)) = 0$.
2. $\dot{E} = 0$.
3. $L^*(L(E))$ is a real multiple of $G_{\varepsilon, c}(E)$.

**Proof.** By the strict Cauchy–Schwarz inequality in (4.7), $\frac{d}{dt} F_{\varepsilon, c}(E(t))$ can be zero only if $G$ is a multiple of $L^*(L(E))$, and then $\langle G, \dot{E} \rangle$ with $\dot{E} = -G + \kappa L^*(L(E))$ implies $\dot{E} = 0$. ☐
4.6. Outer iteration. Let $E(\varepsilon)$ denote the minimizer of the functional $F_{\varepsilon,c}$. generically we expect that for a given perturbation size $\varepsilon < \varepsilon_1^*$, the eigenvalues $\lambda_k(W + \varepsilon E(\varepsilon)) > 0$ and $\lambda_{k+1}(W + \varepsilon E(\varepsilon)) > 0$ are simple. If so, then $f_c(\varepsilon) = F_{\varepsilon,c}(E(\varepsilon))$ is a smooth function of $\varepsilon$ and we can exploit its regularity to obtain a fast iterative method to converge to $\varepsilon_1^*$ from the left. Otherwise we can use a bisection technique to approach $\varepsilon_1^*$.

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

**Assumption 4.1.** We assume that the $k$-th and $k+1$-th smallest eigenvalues of the Laplacian $L(W + \varepsilon E(\varepsilon))$ are simple. Moreover, $E(\varepsilon)$ is assumed to be a smooth function of $\varepsilon$ in some interval.

We then have the following result.

**Lemma 4.6.** Under Assumption 4.1, the function $f_c(\varepsilon) = F_{\varepsilon,c}(E(\varepsilon))$ is differentiable and its derivative equals (with $' = d/d\varepsilon$)

$$f'_c(\varepsilon) = -\frac{\|G_{\varepsilon,c}(E(\varepsilon))\|_F}{\|L^*(L(E(\varepsilon)))\|_F},$$

(4.8)

Proof. Differentiating $f_c(\varepsilon) = F_{\varepsilon,c}(E(\varepsilon))$ with respect to $\varepsilon$ we obtain, with $G(\varepsilon) = G_{\varepsilon,c}(E(\varepsilon))$

$$f'_c(\varepsilon) = \langle G(\varepsilon), E(\varepsilon) + \varepsilon E'(\varepsilon) \rangle,$$

(4.9)

Now we use the property of minimizers, stated by Theorem 4.5, and conclude

$$G(\varepsilon) = \kappa(\varepsilon)L^*(L(E(\varepsilon))), \quad \text{with} \quad \kappa(\varepsilon) = \frac{\langle G(\varepsilon), L^*(L(E(\varepsilon))) \rangle}{\|L^*(L(E(\varepsilon)))\|_F^2},$$

which yields that

$$\langle G(\varepsilon), E'(\varepsilon) \rangle = \kappa(\varepsilon)\langle L^*(L(E(\varepsilon))), E'(\varepsilon) \rangle = \kappa(\varepsilon)\langle L(E(\varepsilon)), L(E'(\varepsilon)) \rangle = \frac{\kappa(\varepsilon) d}{d\varepsilon}\|L(E(\varepsilon))\|^2_F = 0,$$

since $L(E(\varepsilon))$ is of unit norm for all $\varepsilon$. So we have

$$f'_c(\varepsilon) = \langle G(\varepsilon), E(\varepsilon) \rangle = \kappa(\varepsilon)\langle L^*(L(E(\varepsilon))), E(\varepsilon) \rangle = \kappa(\varepsilon)\langle L(E(\varepsilon)), L(E(\varepsilon)) \rangle = \kappa(\varepsilon).$$

Now, (4.7) shows that in an equilibrium,

$$\langle G(\varepsilon), L^*(L(E(\varepsilon))) \rangle = -\|G(\varepsilon)\|_F \|L^*(L(E(\varepsilon)))\|_F,$$

where the negative sign is a consequence of the steepest descent property. Together with the above formula for $\kappa(\varepsilon)$ we thus obtain the stated result. $\square$

For $\varepsilon < \varepsilon_1^*$, we make use of the standard Newton iteration

$$\dot{\varepsilon} = \varepsilon - \frac{f_c(\varepsilon)}{f'_c(\varepsilon)},$$

(4.10)

to get an update value $\dot{\varepsilon}$. In a practical algorithm it is useful to couple the Newton iteration (4.10) with a bisection technique, in the same way as for the method presented in [3]. We adopt a tolerance tol which allows us to distinguish whether $\varepsilon < \varepsilon^*$, in which case we may use the derivative formula and perform the Newton step, or $\varepsilon > \varepsilon^*$, so that we have to make use of bisection. The method is formulated in Algorithm 2.

The overall method is formulated in Algorithm 3. In our experience, it usually terminates already in Step 2.
Algorithm 2: Newton-bisection method

\textbf{Data:} Matrix $W$ is given, $m_{\text{max}}$ (max number of iterations), tol (tolerance) $\varepsilon_0$, $\varepsilon_{\text{lb}}$, and $\varepsilon_{\text{ub}}$ (starting values for the lower and upper bounds for $\varepsilon^*$)

\textbf{Result:} $\varepsilon^*$ (upper bound for the distance), $E(\varepsilon^*)$

\begin{algorithm}
begin
1. Compute $E(\varepsilon_0)$ by the inner iteration
2. Set $m = 0$
3. \textbf{while} $m \leq m_{\text{max}}$ \textbf{do}
   \begin{itemize}
   \item \textbf{if} $f(\varepsilon_m) < \text{tol}$ \textbf{then}
     \begin{itemize}
     \item Set $\varepsilon_{\text{ub}} = \min(\varepsilon_{\text{ub}}, \varepsilon_m)$
     \item Set $\varepsilon_{m+1} = (\varepsilon_{\text{lb}} + \varepsilon_{\text{ub}})/2$ (bisection step)
     \end{itemize}
   \item \textbf{else}
     \begin{itemize}
     \item Set $\varepsilon_{\text{lb}} = \max(\varepsilon_{\text{lb}}, \varepsilon_m)$
     \item Compute $f_\varepsilon(\varepsilon_m)$ and $f'_\varepsilon(\varepsilon_m)$
     \item Compute $\varepsilon_{m+1} = \varepsilon_m - \frac{f_\varepsilon(\varepsilon_m)}{f'_\varepsilon(\varepsilon_m)}$ (Newton step)
     \end{itemize}
   \end{itemize}
4. \textbf{if} $\varepsilon_{m+1} \not\in (\varepsilon_{\text{lb}}, \varepsilon_{\text{ub}})$ \textbf{then}
   \begin{itemize}
   \item Set $\varepsilon_{m+1} = (\varepsilon_{\text{lb}} + \varepsilon_{\text{ub}})/2$
   \end{itemize}
5. \textbf{if} $m = m_{\text{max}}$ \textbf{or} $\varepsilon_{\text{ub}} - \varepsilon_{\text{lb}} < \text{tol}$ \textbf{then}
   \begin{itemize}
   \item Return $\varepsilon_{m+1}$ and the interval $[\varepsilon_{\text{lb}}, \varepsilon_{\text{ub}}]$
   \item Stop
   \end{itemize}
6. \textbf{else}
   \begin{itemize}
   \item Set $m = m + 1$
   \item Compute $E(\varepsilon_m)$ by the inner iteration
   \end{itemize}
\end{algorithm}

4.7. Effective monotonicity with respect to $\varepsilon$. Assume we have integrated equation (4.6) with $\varepsilon = \varepsilon_1$ and we skip to a new value $\varepsilon_2 > \varepsilon_1$. In order to get continuity of the functional with respect to $\varepsilon$, we first integrate the ODE (with initial value $E(\varepsilon_1)$, i.e. the optimal matrix computed at previous step)

\[
\begin{align*}
\dot{E} &= -G_{\varepsilon_2}(E) - c \nabla Q_{\varepsilon_2}(E) \\
E(0) &= \frac{\varepsilon_1}{\varepsilon_2} E(\varepsilon_1)
\end{align*}
\]

that is (4.6) dropping the term with $\kappa$ in the right hand side accounting for norm preservation, where the norm of $L(E)$ is expected to increase. Then we check the instant $\bar{t}$ such that

\[\|L(E(\bar{t}))\| = 1\]

and continue with (4.6),

\[\dot{E} = -G_{\varepsilon_2}(E) - c \nabla Q_{\varepsilon_2}(E) - \kappa E \quad \text{with} \quad \kappa = \langle -G_{\varepsilon_2}(E) - c \nabla Q_{\varepsilon_2}(E), E \rangle\]

with initial datum $E(\bar{t})$ of unit norm.

In this way the computed functional $F_\varepsilon(E(\bar{t}))$ is made decreasing with respect to $t$ and $\varepsilon$. 

Algorithm 3: The overall algorithm

<table>
<thead>
<tr>
<th>Data:</th>
<th>Matrix $W$ and vector $c$ of increasing values $c_1, \ldots, c_2, \ldots, c_m$ of penalization values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result:</td>
<td>$\varepsilon^<em>_k$ (upper bound for the distance), $E(\varepsilon^</em>_k)$</td>
</tr>
</tbody>
</table>

begin
1 With $c = 0$, find $\varepsilon_0 = \min \varepsilon \{\varepsilon \mid F_{\varepsilon,c}(E) = 0\}$ by applying the 2-level method
2 Set $E_0$ the minimizer
   if $W + \varepsilon_0 E_0 > 0$ then
     Return $\varepsilon^*_k = \varepsilon_0$, $E^* = E_0$
3 for $\ell = 1, \ldots, m$ do
   3.1 Find $\varepsilon_\ell = \min \varepsilon \{\varepsilon \mid F_{\varepsilon,c_\ell}(E) \to \min\}$ by applying the 2-level method
   3.2 Set $E_\ell$ the minimizer
4 Set $E_m = (E_m)_+$
5 Normalize $E_m$ to unit Frobenius norm
6 Return $\varepsilon^*_k = \varepsilon_m$, $E^* = E_m$

4.8. Choice of the initial perturbation size $\varepsilon_0$ and the initial perturbation matrix.

While one might just start with a random perturbation, a more natural guess starts from the normalized free gradient $E^0 = -G_\varepsilon(0)/\|L(G_\varepsilon(0))\|_F$. We propose to choose as a first guess the scaled spectral gap $\varepsilon_0 = (\lambda_{k+1}(L(W)) - \lambda_k(L(W)))/\sqrt{2}$, which underestimates the optimal $\varepsilon^*_k$ and therefore leads us into the regime where Newton rather than bisection is used for updating $\varepsilon$ in the outer iteration.

4.9. Remark: Perturbing only selected weights.

If one is interested in the effect of perturbations only in certain weights of the graph, it is sufficient to replace the projector $P_\varepsilon$ by the following one:

$$
P_{\varepsilon_{\text{pert}}}(A)_{ij} := \begin{cases} 
\frac{1}{2}(a_{ij} + a_{ji}) & \text{if } (i,j) \in E_{\text{pert}}, \\
0 & \text{otherwise},
\end{cases}$$

where $E_{\text{pert}} = \{(i,j) \text{ subject to perturbations}\}$.

This means that when only some weights are subject to uncertainties, one could determine a more reliable stability measure by limiting the attention to such edges.

5. Numerical experiments.

In this section we compare the behavior of the spectral gap and the structured distance to ambiguity (SDA) as stability indicators. First, we determine the optimal numbers of clusters by the criterion of maximal stability with both stability indicators in a family of stochastic block models with varying edge probabilities, alongside a reduced model with similar behavior. Then we examine their performance on a model with randomly generated numbers near given centers, where the optimal number of clusters is given a priori but is here determined by the criterion of maximal stability with both stability indicators.

5.1. Stochastic block model and reduced model.

The stochastic block model (SBM) is a model of generating random graphs that tend to have communities. It is an important model in a wide range of fields ranging from sociology to physics (see [14] and, e.g., [1, 2]).

Here we consider a stochastic block model with the following parameters:

- the number $n$ of vertices,
- a partition of the vertex set $\{1, \ldots, n\}$ into $r$ disjoint subsets $C_1, \ldots, C_r$ called communities,
• a symmetric $r \times r$-matrix $P$ containing edge probabilities.

The graph is then sampled in the way that any two vertices $v_i \in C_i$ and $v_j \in C_j$ are connected with probability $p_{ij}$.

For the stability question considered in this paper, this model can be reduced to a model with $2r$ vertices. Assume that the probability matrix $P$ of the SBM is such that $p_{ii} = 1$ for all $1 \leq i \leq r$. In this case, any two vertices $v$ and $w$ belonging to the same community are connected. If $P = I$, the SBM thus generates a disconnected graph with $r$ communities, which are complete graphs of size $|C_1|, \ldots, |C_r|$. We represent this graph by a graph with vertex set $\{1, \ldots, 2r\}$ such that the vertices $2k - 1$ and $2k$ are connected with weight $|C_k|$. The edge probabilities $p_{ij}$ in case $P \neq I$ are then represented by inserting matrices $\mu_{ij}I_2$ in the respective part of the weight matrix in the reduced model, where $\mu_{ij}$ is an appropriate function of $p_{ij}$ that takes the community sizes into account.

To illustrate the construction of this reduced model, consider a SBM with $n = 300$ vertices, three communities of size 100 and the probability matrix

$$P = \begin{pmatrix}
1 & 0.2 & 0 \\
0.2 & 1 & 0.1 \\
0 & 0.1 & 1
\end{pmatrix}.$$  

Figure 5.1 shows the adjacency matrix of one sample of the corresponding SBM. This $300 \times 300$-matrix is represented by the $6 \times 6$-matrix

$$W = \begin{pmatrix}
0 & 100 & 20 & 0 & 100 & 0 \\
100 & 0 & 0 & 20 & 0 & 10 \\
20 & 0 & 0 & 100 & 0 & 10 \\
0 & 20 & 100 & 0 & 0 & 10 \\
10 & 0 & 0 & 100 & 0 & 10 \\
0 & 10 & 100 & 0 & 0 & 10
\end{pmatrix}.$$  

With regards to clustering stability, we observe a similar behavior for the full and the reduced model, as the following example shows.
Example 1. (Chain SBM) We measure the clustering stability when applied to a SBM as described above.

- We use $r = 8$ communities of size 100 in the reduced model and use $\mu_1 I_2, \ldots, \mu_{r-1} I_2$ on the off-diagonal, where

\[ \mu_k = \frac{r - k}{r - 1}\mu_1 \]

and $\mu_1 \in \{2, 4, \ldots, 100\}$ (the above example shows such a model with $r = 3$ and $\mu_1 = 20$).

For small values of $\mu_1$, we expect a clustering into $r$ communities to be most stable, whereas for increasing $\mu_1$, the optimal number of clusters should decrease. We compute the optimal number of clusters $k_{opt}(\delta)$ provided by the SDA and the optimal number of clusters $k_{opt}(g)$ as provided by the spectral gaps. Figure 5.2 shows the results. We observe the expected behavior in both robustness measures, but the SDA tends to opt for a lower number of clusters for smaller values of $\mu_1$. Figure 5.3 shows the measures $\delta_6, \delta_7, \delta_8$ and $g_6, g_7, g_8$ for different values of $\mu_1$. As we expect, $\delta_8$ and $g_8$ are decreasing, $\delta_7$ and $g_7$ are increasing up to a certain point and then decreasing again.

- To compare the behavior of the above reduced model to the full SBM, we compute the same values for a SBM with $r = 6$ communities of size 30, edge probabilities $p_k = \mu_k/100$, where $\mu_1 \in [0, 50]$. Figure 5.4 shows the resulting optimal number of clusters, figure 5.5 shows the distances to ambiguity and the spectral gaps for different values of $p_1$. We see how the results are affected by randomness in the graph generation but conclude that the behavior of both models is similar. Thus, the reduced model described above is a reasonable representation of the SBM.

It is remarkable that the optimal number of communities provided by both stability indicators does not differ by more than one. This suggests that the spectral gap is a reasonable stability indicator in this case, even though it does not take the special structure of weight matrices into account. We observe a similar behavior in different stochastic block models with a similar structure.

5.2. Randomly distributed numbers around centers. In the previous section we have compared the spectral gap to the distance to ambiguity as a stability measure. We have seen a similar behavior yet we cannot conclude which measure works better or worse. We now present an example where we can measure a success rate of the stability indicators.

Consider a given set of centers $m_1, \ldots, m_r$ with the intention to create clusters $C_1, \ldots, C_r$ around them. We generate a set of $n$ random numbers $x_i$ such that for each $i \in \{1, \ldots, n\}$,
we first choose an index \( j \in \{1, \ldots, r\} \) randomly (uniformly distributed) and then the random number \( x_i \) which is normally distributed with expectation value \( m_j \) and variance 1. If the centers \( m_1, \ldots, m_r \) are well separated, we expect that the numbers \( x_1, \ldots, x_n \) are naturally clustered into \( r \) communities.

For a graph with weights \( w_{ij} \) related to the distances \( |x_i - x_j| \), we then compute the SDA \( \delta_k \) and the spectral gap \( g_k \) for \( k \in \{r_{\text{min}}, r_{\text{max}}\} \). Since the data set is constructed such that it should naturally have \( r \) communities, we expect that \( \arg \max_k \delta_k = r \) in most cases.

**Example 2.** We generate 250 samples of \( n = 120 \) random numbers and \( r = 6 \) groups. For each \( i = 1, \ldots, n \), we first choose a group \( j \in \{1, \ldots, r\} \) randomly and then generate a random number \( x_i \), normally distributed around \( m_j \), where

\[
m = (0 \ 8 \ 16 \ 24 \ 32 \ 40) .
\]

Figure 5.6 shows one sorted random sample \((x_1, \ldots, x_n)\).

In order to represent this data set by a graph, we set, following [18]

\[
f(x_i, x_j) = \exp(-\alpha(x_i - x_j)^2),
\]

and then use

\[
w_{ij} = \begin{cases} 
  f(x_i, x_j), & \text{if } f(x_i, x_j) \geq \text{tol}, \\
  0, & \text{otherwise}
\end{cases}
\]
to avoid a complete graph. We denote by $g_{\text{opt}}$ and $\delta_{\text{opt}}$ the optimal number of clusters obtained by taking the spectral gaps and the SDA, respectively, as stability indicators in the criterion of maximal stability. We used $r_{\text{min}} = 4$ and $r_{\text{max}} = 8$ as bounds for the number of communities. Table 5.1 reports the frequency of occurrence of $k$ optimal clusters for $4 \leq k \leq 8$ with both stability indicators for $\alpha = \frac{1}{2}$, and in Table 5.2 for $\alpha = \frac{1}{4}$. We conclude that the success of recognizing the number of communities strongly depends on the graph representation of the data set, but in all cases considered the SDA is outperforming the spectral gap in finding the correct number of clusters.

Acknowledgments. We thank Ulrike von Luxburg and Matthias Hein for a helpful and encouraging discussion.

Part of this work was developed during some visits to Gran Sasso Science Institute in L’Aquila and to the University of Tübingen. The authors thank both institutions for the very kind hospitality.

N. Guglielmi thanks the INdAM GNCS (Gruppo Nazionale di Calcolo Scientifico) and the Italian M.I.U.R. for financial support.

REFERENCES

Table 5.1: Optimal number of clusters ($\alpha = 1/2$)

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_{opt}$</td>
<td>0.0%</td>
<td>0.0%</td>
<td>69.6%</td>
<td>20.4%</td>
<td>10.0%</td>
</tr>
<tr>
<td>$\delta_{opt}$</td>
<td>0.0%</td>
<td>0.0%</td>
<td>78.0%</td>
<td>15.2%</td>
<td>6.8%</td>
</tr>
</tbody>
</table>

Table 5.2: Optimal number of clusters ($\alpha = 1/4$)

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$g_{opt}$</td>
<td>0.0%</td>
<td>0.0%</td>
<td>89.6%</td>
<td>7.6%</td>
<td>2.8%</td>
</tr>
<tr>
<td>$\delta_{opt}$</td>
<td>0.0%</td>
<td>0.0%</td>
<td>95.6%</td>
<td>3.2%</td>
<td>1.2%</td>
</tr>
</tbody>
</table>


