Citation for published version:
Cucuringu, M, Pizzoferrato, A & Van Gennip, Y 2021, 'An MBO scheme for clustering and semi-supervised clustering of signed networks', *Communications in Mathematical Sciences*, vol. 19, no. 1, pp. 73-109. https://doi.org/10.4310/CMS.2021.v19.n1.a4

DOI:
10.4310/CMS.2021.v19.n1.a4

Publication date:
2021

Document Version
Peer reviewed version

Link to publication

(C) International Press of Boston, 2021.

University of Bath

Alternative formats
If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 22. Jun. 2021

AN MBO SCHEME FOR CLUSTERING AND SEMI-SUPERVISED CLUSTERING OF SIGNED NETWORKS*

MIHAI CUCURINGU†, ANDREA PIZZOFERRATO‡, AND YVES VAN GENNIP§

Abstract. We introduce a principled method for the signed clustering problem, where the goal is to partition a weighted undirected graph whose edge weights take both positive and negative values, such that edges within the same cluster are mostly positive, while edges spanning across clusters are mostly negative. Our method relies on a graph-based diffuse interface model formulation utilizing the Ginzburg–Landau functional, based on an adaptation of the classic numerical Merriman–Bence–Osher (MBO) scheme for minimizing such graph-based functionals. The proposed objective function aims to minimize the total weight of inter-cluster positively-weighted edges, while maximizing the total weight of the inter-cluster negatively-weighted edges. Our method scales to large sparse networks, and can be easily adjusted to incorporate labelled data information, as is often the case in the context of semi-supervised learning. We tested our method on a number of both synthetic stochastic block models and real-world data sets (including financial correlation matrices), and obtained promising results that compare favourably against a number of state-of-the-art approaches from the recent literature.

Keywords. Merriman–Bence–Osher scheme, threshold dynamics, clustering, signed networks, graph Laplacians, spectral methods, time series.

AMS subject classifications. 15A99, 05C85, 35R02, 35Q56, 49K15, 68R10, 62H30 , 05C22.

1. Introduction

Clustering is one of the most widely used techniques in data analysis, and aims to identify groups of nodes in a network that have similar behaviour or features. It is arguably one of the first tools used when understanding the structure of a network.

Spectral clustering methods have become a fundamental tool with a broad range of applications in applied mathematics and computer science, and tremendous impact on areas including scientific computing, network science, computer vision, machine learning and data mining, such as ranking and web search algorithms. On the theoretical side, understanding the properties of the spectrum (both eigenvalues and eigenvectors) of the adjacency matrix and its various corresponding Laplacians, is crucial for the development of efficient algorithms with performance guarantees, and leads to a very mathematically rich set of open problems.

Variational methods have a long history in mathematics and physics. In a variational model, the aim is to find minimizers of a given functional (which is often called an 'energy' in recognition of the field's origins, which are rooted in physics, even in situations where this terminology is not physically justified). When formulated in a continuum setting, such models are closely related to partial differential equations (PDEs) which arise as the Euler–Lagrange equations or gradient flow equations in the model. These equations are often nonlinear and can have other interesting properties, such as nonlocality. In recent years it has been recognized that such properties can be useful for clustering problems as well [1, 2]. When attempting a hard clustering problem, i.e. a clustering problem in which each node is to be assigned to exactly one cluster, the approach is usually to relax the exact combinatorial problem. For example, in the case of

*Submitted to the editors August 19, 2020- Received date, and accepted date (The correct dates will be entered by the editor).

†Department of Statistics and Mathematical Institute, University of Oxford & The Alan Turing Institute(mihai.cucuringu@stats.ox.ac.uk).

‡Department of Mathematical Sciences, University of Bath & The Alan Turing Institute(ap@andreapizzoferrato.com). This is the corresponding author.

§Delft Institute of Applied Mathematics, Delft University of Technology(Y.vanGennip@tudelft.nl)

spectral relaxations, one typically leverages the spectrum of certain graph-based matrix operators, in particular their eigenvalues and especially eigenvectors. This introduces the question of how to translate the real-valued outcomes of the relaxed problem into a discrete cluster label. To this end, non-linearities can prove useful as they can be used to drive the real-valued approximate solution away from the decision boundaries. Moreover, what is non-local in a continuum setting, can be made local in a graph setting, by adding edges between nodes that are located far away in continuum space. Furthermore, since the equations which result from graph-based variational methods tend to resemble discretized PDEs, the large existing literature on numerical methods for solving PDEs often provides fast algorithms for solving such equations. Considerations of similar nature have led to a growing interest in the use of variational methods in a graph setting [3–6].

In certain applications, negative weights can be used to denote dissimilarity or distance between a pair of nodes in a network. The analysis of such signed networks has become an increasingly important research topic in the past decade, and examples include social networks that contain both friend and foe links, shopping bipartite networks that encode like and dislike relationships between users and products [7], and online news and review websites such as Epinions [8] and Slashdot [9] that allow users to approve or denounce others [10]. Such networks may be regarded as signed graphs, where a positive edge weight captures the degree of trust or similarity, while a negative edge weight captures the degree of distrust or dissimilarity. Further efforts for the analysis of signed graphs have lead to a wide range of applications including edge prediction [10,11], node classification [12,13], node embeddings [14–17], and node ranking [18,19]. We refer the reader to [20] for a recent survey on the topic.

Finally, another main motivation for our proposed line of work arises in the area of time series analysis, where clustering time series data is a task of particular importance [21]. It arises in a number of applications in a diverse set of fields ranging from biology (analysis of gene expression data [22]), to economics (time series that capture macroeconomic variables [23]). A particularly widely used application originates from finance, where one aims to cluster the large number of time series corresponding to financial instruments in the stock market [24,25]. In all the above application scenarios, a popular approach in the literature is to consider the similarity measure given by the Pearson correlation coefficient (and related distances), that measures the linear dependence between pairs of variables and takes values in $[-1,1]$. Furthermore, the resulting empirical correlation matrix can be construed as a weighted network, where the signed edge weights capture pairwise correlations, which reduces the task of clustering the multivariate time series to that of clustering the underlying signed network. For scalability and robustness reasons, one can further consider thresholding on the p-value associated to each individual sample correlation entry [26], following tests of statistical significance, which renders the network sparse, and thus amenable to larger scale computations. We refer the reader to the seminal work of Smith et al. [27] for a detailed survey and comparison of various methodologies for casting time series data into networks, in the context of fMRI time series. Their overall conclusion that correlation-based approaches are typically quite successful at estimating the connectivity of brain networks from fMRI time series data, provides sound motivation for the development of algorithms for analysis of signed networks and correlation clustering.

Contribution. Our main contribution is to extend the Ginzburg–Landau functional to the setting of clustering signed networks, and show the applicability of the MBO scheme in this context, for different types of constraints that arise in the semi-

supervised setting. As a side contribution, we provide a more computationally efficient way to determine node-cluster membership (as an alternative to the traditional projection/thresholding step). In addition, we show extensive numerical experiments on both synthetic and real data, showcasing the competitivity of our approach when compared to state-of-the-art methods, across various noise and sparsity regimes.

Paper outline. The remainder of this paper is organized as follows. Section 2 is a summary of related work from the signed clustering literature. Section 3 extends the Ginzburg–Landau framework to the setting of signed graphs, by explicitly handling negative weights and modifying the MBO algorithm accordingly. Section 4 pertains to the semi-supervised learning setting, and incorporates various types of a priori information available to the user. Section 5 summarizes the structure of the algorithm we propose. Section 6 contains numerical experiments under a signed stochastic block model, and includes various scenarios in the semi-supervised setting. Section 7 details the outcome of numerical experiments under the Barabási–Albert model. Section 8 to clustering correlation matrices arising from financial time-series data, and Section 9 to computational social science using a network resulting from migration patterns between US counties. Section 10 summarizes our findings and details future research directions. Finally, we defer to the Appendix A.3 the technical details that pertain to the projection/thresholding step.

Notation. We summarize our notation in the Appendix A.1 Table A.1.

2. Related literature

The problem of clustering signed graphs traces its roots to the work of Cartwright and Harary from the 1950s from social balance theory [28,29]. Consider, for example, three mutually connected nodes from a signed graph, and define the relationship between two nodes to be *positive* if the edge in-between is $+1$ (they are "friends"), and *negative* if the edge in-between is -1 (they are "enemies"). We say that such a triad is *balanced* if the product of its signs is positive, and that the signed graph is balanced if all of its cycles are positive [29], i.e., each cycle contains an even number of negative edges. Cartwright and Harary proved the so-called *structure theorem* that if a signed graph $G=(V,E)$ is *balanced*, then the set of vertices, V can be partitioned into two subsets, called plus-sets, such that all of the positive edges are between vertices within a plus-set and all negative edges are between vertices in different plus-sets.

Davis [30] proposed a weaker notion of balance, *weak balance theory*, which relaxed the balanced relationship by allowing an enemy of one's enemy to also be an enemy, and showed that a network with both positive and negative edges is k-weakly balanced if and only if its nodes can be clustered into k groups such that edges within groups are positive and edges between groups are negative. For a k-balanced network, the problem of finding the partition of the node set can be viewed as an optimization problem: find the k-partition[1] C which minimizes $P_\alpha:\Phi\to\mathbb{R}$ given by

$$P_\alpha(C):=\alpha\sum\nolimits_n+(1-\alpha)\sum\nolimits_p, \tag{2.1}$$

over Φ, the set of all possible k-partitions. Here $\sum_p$ is the number of positive arcs between plus-sets, $\sum_n$ is the number of negative arcs within plus-sets, and $\alpha\in(0,1)$ is a weighting constant [31]. If the graph is k-balanced the set of minimizers of P will not change when α varies; if the graph is not k-balanced, the parameter α allows control over which kind of error is penalized more strongly, positive ties between groups or negative ties within groups.

[1] We define a k-partition of the graph vertices to be a collection of subsets $(\mathcal{P}_i)_{i=1}^k$ such that $V=\bigcup_{i=1}^k\mathcal{P}_i$, and $\forall i\neq j\ \mathcal{P}_i\cap\mathcal{P}_j=\emptyset$.

Motivated by this characterization, the k-way clustering problem in signed networks amounts to finding a partition into k clusters such that most edges within clusters are positive and most edges across clusters are negative. As an alternative formulation, one may seek a partition such that the number of **violations** is minimized, i.e., negative edges within the cluster and positive edges across clusters. To avoid partitions where clusters contain only a few nodes, one often prefers to incentivize clusters of similar large size or volume, leading to balancing factors such as the denominators $x_c^T x_c$ in (2.5) or $x_c^T \bar{D} x_c$ in (2.6), discussed below.

Weak balance theory has motivated the development of algorithms for clustering signed networks. Doreian and Mrvar [32] proposed a local search approach in the spirit of the Kernighan–Lin algorithm [33]. Yang et al. [34] introduced an agent-based approach by considering a certain random walk on the graph. Anchuri et al. [35] proposed a spectral approach to optimize modularity and other objective functions in signed networks, and demonstrated its effectiveness in detecting communities from real world networks. In a different line of work, known as *correlation clustering*, Bansal et al. [36] considered the problem of clustering signed complete graphs, proved that it is NP-complete, and proposed two approximation algorithms with theoretical guarantees on their performance. On a related note, Demaine and Immorlica [37] studied the same problem but for arbitrary weighted graphs, and proposed an $\mathrm{O}(\log n)$ approximation algorithm based on linear programming. For correlation clustering, in contrast to k-way clustering, the number of clusters is not given in advance, and there is no normalization with respect to size or volume.

Kunegis et al. [38] proposed new spectral tools for clustering, link prediction, and visualization of signed graphs, by solving a signed version of the 2-way ratio-cut problem via the signed graph Laplacian [39]

$$\bar{L} := \bar{D} - A, \tag{2.2}$$

where A is the signed adjacency matrix of the graph whose elements are $A_{ij} \in \mathbb{R}$ (in [39] they are limited to $A_{ij} \in \{-1,0,+1\}$) and $\bar{D}$ is a diagonal degree matrix with elements $\bar{D}_{ii} = \sum_{j=1}^n A_{ij}$. The authors proposed a unified approach to handle graphs with both positive and negative edge weights, motivated by a number of tools and applications, including random walks, graph clustering, graph visualization and electrical networks. The signed extensions of the combinatorial Laplacian also exist for the random-walk normalized Laplacian and the symmetric graph Laplacian

$$\bar{L}_{\mathrm{rw}} := I - \bar{D}^{-1} A, \tag{2.3}$$

$$\bar{L}_{\mathrm{sym}} := I - \bar{D}^{-1/2} A \bar{D}^{-1/2}, \tag{2.4}$$

the latter of which is particularly suitable for skewed degree distributions. Here I denotes the identity matrix. Recent work, by a subset of the authors in our present paper, provided the first theoretical guarantees, under a signed stochastic block model, for $\bar{L}$ [40].

In related work, Dhillon et al. [41] proposed the *Balanced Ratio Cut* objective

$$\min_{\{x_1,\ldots,x_k\} \in \mathcal{I}_k} \left(\sum_{c=1}^k \frac{x_c^T (D^+ - A) x_c}{x_c^T x_c} \right), \tag{2.5}$$

where D^+ is the degree matrix of the subgraph induced by the strictly positively weighted edges, and the closely related *Balanced Normalized Cut* (BNC) objective for which the balance factors $x_c^T x_c$ in the denominators are replaced by $x_c^T \bar{D} x_c$,

$$\min_{\{x_1,\ldots,x_k\} \in \mathcal{I}_k} \left(\sum_{c=1}^k \frac{x_c^T (D^+ - A) x_c}{x_c^T \bar{D} x_c} \right). \tag{2.6}$$

Here $\mathcal{I}_k$ denotes the collection of all sets containing the indicator vectors of a k-partition of the node set V, i.e. if $\{x_1,\ldots,x_k\}\in\mathcal{I}_k$, then each of the k vectors has entries

$$x_t(i):=\begin{cases}1, & \text{if node } i\in C_t,\\ 0, & \text{otherwise},\end{cases} \tag{2.7}$$

for a collection of disjoint clusters C_t which partitions the node set V. The same authors claimed that the approach of Kunegis et al. [38] via the Signed Laplacian faces a fundamental weakness when directly extending it to k-way clustering, and obtained better results on several real data sets via their proposed algorithm which optimizes the BNC objective function. On the other hand, [40] reported good numerical performance of the Signed Laplacian even for higher values of k, aside from providing a robustness analysis of it for $k=2$.

A rather different approach for signed graph clustering has been investigated by Hsieh et al. [42], who proposed a low-rank model, based on the observation that the problem of clustering a signed network can be interpreted as recovering the missing structure of the network. The missing relationships are recovered via state-of-the-art matrix completion algorithms (MC-SVP), which first complete the network and then recover the clusterings via the k-means algorithm using the top k eigenvectors of the completed matrix. An alternative method (MC-MF), scalable to very large signed networks, is based on a low-rank matrix factorization approach, which first completes the network using matrix factorization and derives the two low-rank factors $U,H\in\mathbb{R}^{n\times k}$, runs k-means on both U and V, and selects the clustering which achieves a lower BNC objective function (2.6).

Both strong and weak balance are defined in terms of local structure at the level of triangles. As observed early in the social balance theory literature, local patterns between the nodes of a signed graph induce certain global characteristics. Chiang et al. [43] exploited both local and global aspects of social balance theory, and proposed algorithms for sign prediction and clustering of signed networks. They defined more general measures of social imbalance based on cycles in the graph, and showed that the classic Katz measure, ubiquitous for unsigned link prediction, also has a balance theoretic interpretation in the context of signed networks. The global structure of balanced networks leads to a low-rank matrix completion approach via convex relaxation, which scales well to very large problems. This approach is made possible by the fact that the adjacency matrix of a complete k-weakly balanced network has rank 1 if $k\leq 2$, and rank k for all $k>2$ [43].

The recent work of Mercado et al. [44] presented an extended spectral method based on the geometric mean of Laplacians. In another recent work, a subset of the authors showed in [45] that, for $k=2$, signed clustering can be formulated as an instance of the group synchronization problem over $\mathbb{Z}_2$, for which spectral, semidefinite programming relaxations, and message passing algorithms have been studied; extensions to the multiplex setting and incorporation of constraints have also been considered. Finally, we refer the reader to [46,47] for recent surveys on clustering signed and unsigned graphs.

3. Ginzburg–Landau functional on signed graphs

An introduction to the Ginzburg–Landau (GL) method for the unsigned case is given in Appendix A.2. In this section, we define a new affinity matrix that also incorporates negative weights, and investigate how the degree and Laplacian matrices can be suitably modified. In this way, we also show how negative weights can be interpreted in the context of the GL functional (GLF). Moreover, we discuss how the standard MBO algorithm can be simplified.

3.1. Signed Laplacians In the following, we will consider connected weighted undirected simple graphs (hence with symmetric affinity matrices and no self-loops). The corresponding affinity matrix A is a symmetric matrix whose entries A_{ij} are zero if $i=j$ or $i \not\sim j$[2] and nonzero real otherwise. This can be decomposed in two matrices $A^+ \in \mathbb{R}_{+0}^{V\times V}$ and $A^- \in \mathbb{R}_{+0}^{V\times V}$ whose components are defined as

$$A_{ij}^+ := \max\{A_{ij},0\} \in \mathbb{R}_{+0} \text{ and } A_{ij}^- := -\min\{A_{ij},0\} \in \mathbb{R}_{+0}, \tag{3.1}$$

respectively, so that

$$A = A^+ - A^- \in \mathbb{R}^{V\times V}. \tag{3.2}$$

In this decomposition, note that A^+, respectively A^-, takes into account the affinity, respectively antipathy, between nodes. Similarly to [38], we define the *signed degree matrix* as

$$\bar{D}_{ij} := \begin{cases} \sum_{k=1}^{V} |A_{ik}| \in \mathbb{R}_{+0} & \text{if } i=j \\ 0 & \text{if } i\neq j \end{cases}, \tag{3.3}$$

where $|A_{ik}|$ is the absolute value of A_{ik}. As for the affinity matrix, $\bar{D}$ can be decomposed into $D^+ \in \mathbb{R}_{+0}^{V\times V}$ and $D^- \in \mathbb{R}_{+0}^{V\times V}$, with components

$$D_{ij}^+ := \sum_{k=1}^{V} A_{ik}^+ \in \mathbb{R}_{+0} \text{ and } \quad D_{ij}^- := \sum_{k=1}^{V} A_{ik}^- \in \mathbb{R}_{+0}, \tag{3.4}$$

so that

$$\bar{D} = D^+ + D^- \in \mathbb{R}_{+0}^{V\times V}. \tag{3.5}$$

Following [38], the *signed Laplacian* is defined as

$$\bar{L} := \bar{D} - A \in \mathbb{R}^{V\times V}, \tag{3.6}$$

together with its variants

$$\bar{L}_{\text{rw}} := \bar{D}^{-1}\bar{L} = I_V - \bar{D}^{-1}A \in \mathbb{R}^{V\times V} \tag{3.7}$$

and

$$\bar{L}_{\text{sym}} := \bar{D}^{-\frac{1}{2}}\bar{L}\bar{D}^{-\frac{1}{2}} = \bar{D}^{\frac{1}{2}}\bar{L}_{\text{rw}}\bar{D}^{-\frac{1}{2}} = I_V - \bar{D}^{-\frac{1}{2}}A\bar{D}^{-\frac{1}{2}} \in \mathbb{R}^{V\times V}. \tag{3.8}$$

Kunegis et al. [38] showed that the signed Laplacian $\bar{L}$ is positive semidefinite, and furthermore, unlike the ordinary Laplacian L, the signed Laplacian $\bar{L}$ is *strictly positive-definite* for certain graphs, including most real-world networks. It is known that (see for instance [63, 64]) if a matrix Y_1 is positive semidefinite then $Y_2 := XY_1X^T$ is also positive semidefinite provided that X is invertible. We can apply this to show from the definition (3.8) that $\bar{L}_{\text{sym}}$ is also positive semidefinite since $\bar{D}$ is diagonal with rank V. Positive semidefiniteness of $\bar{L}_{\text{rw}}$ follows from the second equality in (3.8) since it shows that $\bar{L}_{\text{rw}}$ and $\bar{L}_{\text{sym}}$ are similar matrices.

In particular, using (3.2) and (3.5), we can decompose (3.6) as

$$\bar{L} = \bar{D} - A = \left(D^+ - A^+\right) + \left(D^- + A^-\right) =: L^+ + Q^-, \tag{3.9}$$

where we stress that $L^+ \in \mathbb{R}^{V\times V}$ and $Q^- \in \mathbb{R}_{+0}^{V\times V}$. The latter is known in the literature as *signless Laplacian* [65] and, to the best of our knowledge, first appeared, although without this nomenclature, in [66].

To understand the meaning of the different terms contributing to the signed Laplacian, first consider the following quadratic form

$$\underline{x}^T Q^- \underline{x} = \underline{x}^T D^- \underline{x} + \underline{x}^T A^- \underline{x}. \tag{3.10}$$

[2] The symbols $i \sim j$ and $i \not\sim j$ denote that nodes i and j are connected and not connected, respectively.

for some $\underline{x} \in \mathbb{R}^V$. Using the definition of D^- together with the fact that A^- is symmetric with null diagonal, we have for the first term[3]

$$\begin{aligned}\underline{x}^T D^- \underline{x} &= \sum_{i=1}^{V} \left(A^-_{ii} + \sum_{j \neq i} A^-_{ij} \right) x_i^2 = \sum_{i=1}^{V} \left(\sum_{j<i} A^-_{ij} + \sum_{j>i} A^-_{ij} \right) x_i^2 \\ &= \sum_{i=1}^{V} \sum_{j<i} A^-_{ij} x_i^2 + \sum_{i=1}^{V} \sum_{j>i} A^-_{ij} x_i^2 \\ &= \sum_{i=1}^{V} \sum_{j<i} A^-_{ji} x_i^2 + \sum_{i=1}^{V} \sum_{j>i} A^-_{ij} x_i^2 = \sum_{i=1}^{V} \sum_{j>i} A^-_{ij} x_j^2 + \sum_{i=1}^{V} \sum_{j>i} A^-_{ij} x_i^2,\end{aligned}$$

and for the second

$$\underline{x}^T A^- \underline{x} = \sum_{i=1}^{V} A^-_{ii} x_i^2 + \sum_{i=1}^{V} \sum_{j>i} A^-_{ij} x_i x_j + \sum_{i=1}^{V} \sum_{j<i} A^-_{ij} x_i x_j = 2 \sum_{i=1}^{V} \sum_{j>i} A^-_{ij} x_i x_j. \tag{3.11}$$

Putting all the contributions together gives for $\underline{x}^T Q^- \underline{x}$

$$\sum_{i=1}^{V} \sum_{j>i} A^-_{ij} x_i^2 + \sum_{i=1}^{V} \sum_{j>i} A^-_{ij} x_j^2 + 2 \sum_{i=1}^{V} \sum_{j>i} A^-_{ij} x_i x_j = \sum_{i=1}^{V} \sum_{j>i} A^-_{ij} (x_i + x_j)^2 \geqslant 0. \tag{3.12}$$

The minimum of (3.12) is achieved when $x_i = -x_j$. In the simplified case in which the node vector is limited to dichotomic values such as $\underline{x} \in \{-1,1\}^V$, one can envision the system to be made up of $K=2$ clusters and the value of x_i (being only $+1$ or -1) specifies the node-cluster association. In addition, with this particular choice, (3.12) counts the number of times, weighted by A^-_{ij}, that two nodes were placed incorrectly in the same cluster (which we want to minimize). We note that choosing as dichotomic values $\{0,1\}$, we would reach a different conclusion since the result of the minimization would be $x_i = x_j = 0$, that is, the nodes would belong to the same cluster. See also [58]. The Laplacian L^+ corresponding to the affinity matrix A^+ is indifferent to the choice of numerical labels. Both $\{-1,1\}$ and $\{0,1\}$ (or any other choice) lead to the same interpretation; see for instance (A.10).

The elements of the canonical basis, which correspond to pure node-cluster associations, can be suitably transformed to take values $\{-1,1\}$, via the following expression[4]

$$\underline{e}_k^{\pm} := 2\underline{e}_k - \underline{1}_K \in \{-1,1\}^K, \tag{3.13}$$

where the associated simplex is

$$\Sigma_K^{\pm} := \left\{ \left(\sigma_1^{\pm}, \sigma_2^{\pm}, \ldots, \sigma_K^{\pm} \right) \in [-1,1]^K : \sum_{k=1}^{K} \sigma_k^{\pm} = 2 - K \right\}. \tag{3.14}$$

As shown in Appendix A.2.4, each row of the node-cluster association matrix U of the unsigned case is a linear combination of $\underline{e}_k$ whose coefficients sum up to 1. The characteristic matrix in the new representation can be obtained as

[3]Here, we use the following shorthand notation $\sum_{j<i} \equiv \sum_{\substack{j=1 \\ j<i}}^{V}$, together with $\sum_{j>i} \equiv \sum_{\substack{j=1 \\ j>i}}^{V}$ and $\sum_{j \neq i} \equiv \sum_{j<i} + \sum_{j>i}$.

[4]Here and in the rest of the paper, the superscript $\pm$ does not indicate two different cases as is the usual convention in the literature. Instead, it only marks that the relevant quantity depends on the basis whose components take values in $\{-1,1\}$.

$$U^{\pm} := 2U - 1_V \in \left(\Sigma_K^{\pm}\right)^V. \tag{3.15}$$

Again in reference to the unsigned case, by plugging $\bar{L}$ in place of L in (A.19), we have that the *signed GL* functional is given by

$$\begin{aligned}\bar{F}_\varepsilon(U^{\pm}) &:= \frac{\varepsilon}{8}\langle U^{\pm}, \bar{L}U^{\pm}\rangle + \frac{1}{2\varepsilon}\sum_{i=1}^{V}\left(\prod_{k=1}^{K}\frac{1}{16}\left\|\underline{u}_i^{\pm} - \underline{e}_k^{\pm}\right\|_1^2\right) \\ &= \frac{\varepsilon}{8}\langle U^{\pm}, L^{+}U^{\pm}\rangle + \frac{\varepsilon}{8}\langle U^{\pm}, Q^{-}U^{\pm}\rangle + \frac{1}{2\varepsilon}\sum_{i=1}^{V}\left(\prod_{k=1}^{K}\frac{1}{16}\left\|\underline{u}_i^{\pm} - \underline{e}_k^{\pm}\right\|_1^2\right),\end{aligned} \tag{3.16}$$

where the different normalization of the terms comes from (3.13) and (3.15). In other words, through the signed GL functional we want to minimize the presence of positive links between clusters as well as negative links within each one. We can verify that the following equality holds $\bar{F}_\varepsilon(U^{\pm}) - \frac{\varepsilon}{8}\langle U^{\pm}, Q^{-}U^{\pm}\rangle = F_\varepsilon(U)$, where, on the right-hand side, the Laplacian of the GL functional is L^{+}, and we used the fact that $L^{+}1_V = 0$ together with known properties of the trace of the product of two matrices.

3.2. MBO scheme for signed graphs The MBO scheme arising from the signed GL has the same structure as in the unsigned case, see (A.21), that is

$$\left(U^{\pm}\right)^{n+\frac{1}{2}} = \bar{B}\left(U^{\pm}\right)^n \quad \text{where} \quad \bar{B} := \left[\frac{1}{4}\bar{X}\left(I_V + \frac{d\tau}{N_\tau}\bar{\Lambda}\right)^{-1}\bar{X}^T\right]^{N_\tau} \in \mathbb{R}^{V\times V}, \tag{3.17}$$

together with $\bar{X}$ and $\bar{\Lambda}$ which are the eigenvector and eigenvalue matrices of $\bar{L}$, respectively[5]. As explained in the Appendix A.3, the succession of projection (which in [61] is performed using the algorithm developed in [67]) and thresholding steps can be reduced to determine the entry of the largest component of each row of $(U^{\pm})^{n+\frac{1}{2}}$, so that[6]

$$\left(\underline{u}_i^{\pm}\right)^{n+1} = \underline{e}_k^{\pm} \quad \text{with} \quad k = \operatorname*{argmax}_{j=1,\dots,K}\left\{\left(u_{ij}^{\pm}\right)^{n+\frac{1}{2}}\right\}. \tag{3.18}$$

However, we note that the calculation of the argmax in (3.18) is independent of the representation chosen for the characteristic matrix. In fact, using (3.15) we can see that

$$\left(U^{\pm}\right)^{n+\frac{1}{2}} = 2\bar{B}U^n - \bar{B}1_V. \tag{3.19}$$

Since $\bar{B}1_V$ has constant rows, the location of the largest entry in a row of $(U^{\pm})^{n+\frac{1}{2}}$ is the same as in $\bar{B}U^n$, that is from (3.18)

$$k = \operatorname*{argmax}_{j=1,\dots,K}\left\{2\sum_{h=1}^{V}\bar{b}_{ih}u_{hj}^n - \sum_{h=1}^{V}\bar{b}_{ih}\right\} = \operatorname*{argmax}_{j=1,\dots,K}\left\{2\sum_{h=1}^{V}\bar{b}_{ih}u_{hj}^n\right\}. \tag{3.20}$$

[5] Here we have used another definition for $\bar{B}$ which is different from the definition of B in (A.21). In fact, since $\bar{L}$ is real and symmetric its eigenvector matrix is orthogonal, that is $\bar{X}^T = \bar{X}^{-1}$. So, we have that $\left[\bar{X}\left(I_V + \frac{d\tau}{N_\tau}\bar{\Lambda}\right)\bar{X}^T\right]^{-N_\tau} = \left[\left(I_V + \frac{d\tau}{N_\tau}\bar{\Lambda}\right)\bar{X}^T\right]^{-N_\tau}\bar{X}^{-N_\tau} = \left(\bar{X}^T\right)^{-N_\tau}\left(I_V + \frac{d\tau}{N_\tau}\bar{\Lambda}\right)^{-N_\tau}\bar{X}^{-N_\tau} = \left[\bar{X}\left(I_V + \frac{d\tau}{N_\tau}\bar{\Lambda}\right)^{-1}\bar{X}^T\right]^{N_\tau}$.

[6] As before, when $\left(\underline{u}_i^{\pm}\right)^{n+1}$ has the same distance to multiple vertices, one of those is picked at random with uniform probability.

since $\sum_{h=1}^{V} \bar{b}_{ih}$ is independent of j by definition. The factor 2 can also be neglected since it does not change the node-cluster association. We indicate this equivalence with the symbol

$$\left(U^{\pm}\right)^{n+\frac{1}{2}} \approx \bar{B}U^{n}. \tag{3.21}$$

In other words, after applying $\bar{B}$, the position of the largest entry of $\left(\underline{u}_i^{\pm}\right)^{n+1}$, resulting from the argmax in (3.18), is the same as the one of $(\underline{u}_i)^{n+1}$. In this way, we can keep using a sparse node-cluster association matrix by picking U instead of $U^{\pm}$.

4. GLF with constraints Now, we extend the GLF to include different kinds of a priori information in the form of constraints. We distinguish between two different types of constraints: *soft*, when the final result of the algorithm does not necessarily satisfy the original constraint, and *hard*, when the output matches the constraint. In Sections 4.1 and 4.2 we focus on the former, while leaving the discussion of the latter to Appendix A.6.

4.1. Must-link and cannot-link In practice, the affinity matrix usually results from measurements of a system's properties (see Appendix A.7 for example). However, in the same spirit as in [68], additional information in the form of both positive (must-link) and negative (cannot-link) edge weights, can also be known and could have a different relative strength with respect to measurements. In quantitative terms, we will embed must-links in A^{+} and cannot-links in A^{-} and generate the new degree matrices D^{+} and D^{-} accordingly. To this end, we define

$$\mathcal{A}^{+} := A^{+} + \lambda^{+} M \in \mathbb{R}_{+0}^{V\times V}, \tag{4.1}$$

where $M \in \mathbb{R}_{+0}^{V\times V}$ is the *must-link matrix*, and $\lambda^{+} \in \mathbb{R}_{+}$ is a trade-off parameter. The role of M consists in adding links to nodes which should be belong to the same cluster or to increase the weight of a link to reinforce node-cluster associations. In general, it is a symmetric sparse matrix with positive weights (whose values can also be different among each couple of nodes). We indicate the resulting degree matrix for $\mathcal{A}^{+}$, defined in the same fashion as in (3.3), with $\mathcal{D}^{+} \in \mathbb{R}_{+0}^{V\times V}$. Similarly

$$\mathcal{A}^{-} := A^{-} + \lambda^{-} C \in \mathbb{R}_{+0}^{V\times V}, \tag{4.2}$$

where $C \in \mathbb{R}_{+0}^{V\times V}$ is the *cannot-link matrix*, and $\lambda^{-} \in \mathbb{R}_{+}$ the trade-off parameter. In the same fashion as M, the matrix C can include links to nodes which should not belong to the same cluster or to decrease even more the weight of a link pushing node couples to belong to different clusters. In this case as well, C is a symmetric sparse matrix with positive weights, where the greater the weight the more the two nodes are incentivised to belong to different clusters. Also here we have $\mathcal{D}^{-} \in \mathbb{R}_{+0}^{V\times V}$ as the degree matrix of $\mathcal{A}^{-}$. In the end, we have that the GLF in the presence of must- and cannot-links is given as

$$\bar{\mathcal{F}}_{\varepsilon}(U^{\pm}) = \frac{\varepsilon}{8}\langle U^{\pm}, \mathcal{L}^{+}U^{\pm}\rangle + \frac{\varepsilon}{8}\langle U^{\pm}, \mathcal{Q}^{-}U^{\pm}\rangle + \frac{1}{2\varepsilon}\sum_{i=1}^{V}\left(\prod_{k=1}^{K}\frac{1}{16}\left\|\underline{u}_i^{\pm} - \underline{e}_k^{\pm}\right\|_1^2\right), \tag{4.3}$$

where $\mathcal{L}^{+} := \mathcal{D}^{+} - \mathcal{A}^{+}$, $\mathcal{Q}^{-} := \mathcal{D}^{-} + \mathcal{A}^{-}$ so that $\mathcal{L} := \mathcal{L}^{+} + \mathcal{Q}^{-}$. As in (3.17), we now need to determine the matrix $\bar{\mathcal{B}}$ resulting from the eigendecomposition of $\mathcal{L}$.

4.2. Fidelity and avoidance terms Soft constraints can also be implemented at the level of the characteristic matrix mainly in two ways. First, as a *fidelity term* (see for instance [61]) which attracts the dynamics of the ACE towards a particular characteristic matrix $\hat{U}$. At the level of the GLF this translates into a term which increases the overall energy when the node-cluster association in $\hat{U}$ is not matched. Second, as an *avoidance term* which pushes the ACE dynamics to avoid particular

node-cluster combinations expressed in a characteristic matrix $\tilde{U}$. Again, for the GLF this means an energy increase when the wrong node-cluster association is met. We notice that while the first constraint points the system to a particular combination and increasing the energy every time this condition is not met, the second does the opposite, that is only one combination increases the energy and leaves all the others as equally possible. The GLF for an affinity matrix with positive edges only reads as

$$G_\varepsilon^R(U,\hat{U},\tilde{U}) := F_\varepsilon(U) + \frac{1}{2}\sum_{i=1}^{V} R_{ii}^{fi} \|\underline{u}_i - \underline{\hat{u}}_i\|_2^2 + \sum_{i=1}^{V} R_{ii}^{av} \|\underline{u}_i \circ \underline{\tilde{u}}_i\|_1 \tag{4.4}$$

where $\circ$ is the Hadamard product and R^{fi} (and R^{av}) is a diagonal matrix whose element R_{ii}^{fi} (R_{ii}^{av}) is positive if and only if there exists a $j \in \{1,\dots,V\}$ such that $\hat{U}_{ij} \neq 0$ ($\tilde{U}_{ij} \neq 0$) and is zero otherwise. At the level of the MBOE, following [61] the first step in the procedure is

$$U^{n+\frac{1}{2}} = B\left[U^n - d\tau R^{fi}\left(U^n - \hat{U}\right) - d\tau R^{av}\tilde{U}\right]. \tag{4.5}$$

We can obtain an equivalent expression for the signed GLF using (3.15). The result is

$$\begin{aligned}\bar{G}_\varepsilon^R(U^\pm,\hat{U}^\pm,\tilde{U}^\pm) := &\bar{F}_\varepsilon(U^\pm)\\ &+ \frac{1}{8}\sum_{i=1}^{V} R_{ii}^{fi} \left\|\underline{u}_i^\pm - \underline{\hat{u}}_i^\pm\right\|_2^2 + \frac{1}{4}\sum_{i=1}^{V} R_{ii}^{av} \left\|\underline{u}_i^\pm \circ \underline{\tilde{u}}_i^\pm + \underline{u}_i^\pm + \underline{\tilde{u}}_i^\pm + \underline{1}_V\right\|_1.\end{aligned}$$

Now the MBOE (3.17) becomes

$$\left(U^\pm\right)^{n+\frac{1}{2}} = \bar{B}\left\{\left(U^\pm\right)^n - \frac{d\tau}{4}\left[R^{fi}\left(\left(U^\pm\right)^n - \hat{U}^\pm\right) + R^{av}\left(\tilde{U}^\pm + 1_V\right)\right]\right\}, \tag{4.6}$$

and after the transformation (3.15) becomes

$$\left(U^\pm\right)^{n+\frac{1}{2}} = 2\left\{\bar{B}U^n - \frac{1}{2}\bar{B}1_V - \frac{d\tau}{4}\bar{B}\left[R^{fi}\left(U^n - \hat{U}\right) + R^{av}\tilde{U}\right]\right\}, \tag{4.7}$$

The factor $\frac{1}{4}$ can be embedded in the definitions of R^{fi} and R^{av}. As explained in Section 3.2, since 1_V has constant rows, the term $\bar{B}1_V$ does not modify the result of the argmax problem. Moreover, since the global factor is positive, the presence of the coefficient 2 in front of the brackets in (4.7) does not affect which entry is the largest and it can be neglected. So, we can consider only the following terms

$$\left(U^\pm\right)^{n+\frac{1}{2}} \approx \bar{B}\left\{U^n - d\tau\left[R^{fi}\left(U^n - \hat{U}\right) + R^{av}\tilde{U}\right]\right\}. \tag{4.8}$$

5. The algorithm This section is devoted to describing the general structure of the MBO scheme. First, we will show the basic implementation and then we will explain how to modify the relevant structures to incorporate different types of constraints.

The MBO scheme for signed Laplacians we propose is given in Algorithm 1. In addition to the graph variables (i.e., the number of nodes, clusters and the signed Laplacian), we also need to specify

- $d\tau \in \mathbb{R}_+$ which is the length of the interval of the time discretization procedure as in (A.12),
- $N_\tau \in \mathbb{N}$ which is the number of times we repeat the diffusion step in (A.13),
- $m \in \mathbb{N}$ which is the number of eigenvectors and eigenvalues considered in the eigendecomposition of $\bar{B}$ in (3.17) (we refer the reader to the next paragraph for a theoretical motivation of this choice).

Require
V, K, $d\tau$, N_τ, m, $\bar{L}$, ϵ;

Initialize
$\bar{B}_m \leftarrow \bar{X}_m \left(I_m + d\tau \bar{\Lambda}_m\right)^{-1} \bar{X}_m^T$;
$U^0 = 0_{V\times K}$;
for $i \leftarrow 1$ **to** V **do**
  $k = randperm(K,1)$;
  $u_{ik}^0 = 1$;
end

MBO Scheme
$n \leftarrow 0$;
while *Stop criterion not satisfied* **do**
  for $s \leftarrow 1$ **to** N_τ **do**
    $U^{n+\frac{1}{2}} \leftarrow \bar{B}_m U^n$;
    $U^n \leftarrow U^{n+\frac{1}{2}}$;
  end
  for $i \leftarrow 1$ **to** V **do**
    $\underline{u}_i^{n+1} = \underline{0}_K$;
    $k = \arg\max_{j=1,\dots,K} \left\{ u_{ij}^{n+\frac{1}{2}} \right\}$;
    $u_{ik}^{n+1} = 1$;
  end
  $U^n \leftarrow U^{n+1}$
end

Output $\underline{v}^* \in \{1;\dots;K\}^V$
for $i \leftarrow 1$ **to** V **do**
  $v_i^* = \arg\max_{j=1,\dots,K} \{u_{ij}^n\}$;
end

Algorithm 1: Structure of the MBO scheme for signed Laplacians in the absence of constraints. The algorithm requires values specifying the affinity matrix and some parameters for best code-tuning which are explained in the main text of the present section. The output of the code is a vector $\underline{v}^*$ which indicates the cluster (numbered from 1 to K) each node belongs to. The command $randperm(K,1)$ generates a vector of length K whose entries are natural numbers uniformly distributed in the interval $[1,K]$, i.e., it associates to each node a random cluster membership.

The corresponding eigenvector and eigenvalue matrices (thus having m eigenvectors and eigenvalues) are indicated as $\bar{X}_m$ and $\bar{\Lambda}_m$, respectively. The parameter ϵ is defined later in (5.1). In regard to $d\tau$, in Section 6.1 we show, empirically on a synthetic data set, that the outcomes of Algorithm 1 are independent of the chosen value as long as it remains small. In [61], it is reported that the optimal value for the number of diffusion step iterations is $N_\tau = 3$. Finally, the number of eigenvectors considered is chosen to be equal to the number of clusters set a priori since, at a heuristic level, the eigenvalues empirically exhibit a big jump after the K-th term (although this may not always hold

true, see [71]). On a related note, we refer the reader to the celebrated result of Lee, Gharan, and Trevisan on higher-order Cheeger inequalities for undirected graphs with $k \geq 2$ clusters [72], which motivates the use of the top k eigenvectors and generalizes initial results on $k=2$ [73]. Moreover, the version of the algorithm we propose does takes into account the projection/thresholding step in a different fashion, as explained in Section 3.2, which allows us to shorten the running time. To include various constraints, it suffices to modify only a few relevant parts of the code in Algorithm 1. In particular, we consider

- **Must-link and cannot-link soft constraints** - here, we just need to use $\bar{\mathcal{B}}$ (see below (4.3)) in place of $\bar{B}$ in (3.21),
- **Fidelity and avoidance terms** - the evolution equation given by $\bar{B}$ now includes the additional terms as in (4.5),
- **Anchor nodes** - the matrix $\bar{B}$ is modified, as explained in Section A.6, to include the fixed nodes in the MBO dynamics.

In all cases, as a stop criterion we use, as in [61],

$$\frac{\max_i \left\| \underline{u}_i^{n+1} - \underline{u}_i^n \right\|_2^2}{\max_i \left\| \underline{u}_i^{n+1} \right\|_2^2} < \epsilon \in \mathbb{R}_+. \tag{5.1}$$

All the constraints can also be combined together since they act at different levels in the scheme. However, we notice that including anchors automatically rules out possible soft constraints acting on the same node.

6. Numerical experiments for a signed stochastic block model

This section is devoted to present the performance of the MBO scheme for signed Laplacians on synthetic data sets. This approach is useful since it allows a direct comparison of the results with the ground truth. Here, we consider a *signed stochastic block model* (SSBM) followed by the definition of the cost function used to compare the output of the MBO algorithm with the ground truth.

Given a set of numbers $\{c_1, c_2, \ldots, c_K\} \in \mathbb{N}$ representing the size of K clusters of V nodes, the corresponding affinity matrix for the fully connected case is given in block form as

$$S := \begin{pmatrix} 1_{c_1} & -1_{c_1 \times c_2} & \ldots & -1_{c_1 \times c_K} \\ -1_{c_2 \times c_1} & 1_{c_2} & \ldots & -1_{c_2 \times c_K} \\ \vdots & \vdots & \ddots & \vdots \\ -1_{c_K \times c_1} & -1_{c_K \times c_2} & \ldots & 1_{c_K} \end{pmatrix} \tag{6.1}$$

where 1_{c_i} is a square matrix of size c_i whose entries are all 1 and $-1_{c_i \times c_j} \in \{-1\}^{c_i \times c_j}$ is a matrix with c_i rows and c_j columns whose elements are all -1 (for consistency, we have that $\sum_{k=1}^{K} c_k = V$). Here, we see that all the nodes belonging to the same (respectively different) cluster have a $+1$ (respectively -1) link. The corresponding output vector (see Algorithm 1) for the ground truth is given as

$$\underline{s} = \left(c_1 \underline{1}_{c_1}^T, c_2 \underline{1}_{c_2}^T, \ldots, c_K \underline{1}_{c_K}^T \right)^T \in \{1, \ldots, K\}^V, \tag{6.2}$$

where each entry corresponds to a node and its value to the cluster (indexed from 1 to K) the node belongs to, and its characteristic matrix is

$$U_s = \begin{pmatrix} \underline{1}_{c_1} & 0 & \ldots & 0 \\ 0 & \underline{1}_{c_2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \underline{1}_{c_K} \end{pmatrix} \in \{0,1\}^{V \times K}. \tag{6.3}$$

An instance of the SSBM is obtained from (6.1) by introducing *sparsity* (understood as null entries) and *noise* (which corresponds to a change of the sign of the elements S_{ij}) with probabilities λ and η, respectively. More specifically, an edge is present independently with probability λ, while the sign of an existing edge is flipped with probability η. In quantitative terms, an SSBM matrix A, which we fix to be symmetric and null-diagonal, is generated from S via the following probabilistic mixture model. Consider a row index $i=1,\dots,V$ and a column index $j=i,\dots,V$ (the j index starts from i so to consider only the upper triangular part in the definition of the elements of A), we define the affinity matrix as

$$A_{ij}=\begin{cases} S_{ij} & \text{if } i<j, \text{ with probability } (1-\eta)\lambda \\ -S_{ij} & \text{if } i<j, \text{ with probability } \eta\lambda \\ 0 & \text{if } i<j, \text{ with probability } 1-\lambda \\ 0 & \text{if } i=j, \text{ with probability } 1 \end{cases} \qquad \text{with } A_{ji}=A_{ij}. \tag{6.4}$$

Then, the corresponding Laplacian can be computed as detailed in Section 3.1. As we can see, the smaller λ is, the sparser the network is on average, and the larger η is, the noisier the edges are on average.

As explained in Section 5, the output of the MBO algorithm is a vector $\underline{v}^*$ which holds the information related to the node-cluster association in the same spirit as (6.2). Therefore, it is sensible to compare $\underline{v}^*$ with $\underline{s}$ to measure the ability of MBO to retrieve the ground truth clustering. To quantify success in terms of robustness to noise and sampling sparsity, we use the popular *adjusted rand index* (ARI) [74] whose definition is given in Appendix A.4.

For the numerical tests, the different types of constraints described in Section 4 can be included by extracting, at random, information from the ground truth. More specifically, M and C can be obtained from S for the must/cannot-link case, and $\hat{U}$ and $\tilde{U}$ from U_s for the fidelity/avoidance term etc. We indicate the percentage of information overlapping with the ground truth using the parameter $\alpha\in[0,1]$. Since we are incorporating constraints using ground truth information, it is clear that the results will be closer to the ground truth for increasing α. The purpose of this set of experiments is simply to provide a proof of concept for our approach. In summary, the numerical experiments we consider are generated using the set of parameters summarized in Table 6.1, some of which are also explained in Section 5. In the following, we generate a new

Parameter	Value
Nodes	$V=1200$
Clusters	$K=2,\ 3,\ 4,\ 5,\ 8,\ 10$
Size	Equal clusters size V/K
$d\tau$	0.1
N_τ	3
m	Equal to no. of clusters K
ϵ	10^{-7}
No. SSBMs per sample	100
Sparsity	$\lambda\in[0.02,0.1]$
Noise	$\eta\in[0.05,0.45]$
Known Information	$\alpha\in[0.05,0.9]$

TABLE 6.1. *Parameter values used to test the MBO scheme in the SSBM experiments.*

sample of SSBMs for each plot in a Figure. The resulting set of SSBMs (having size specified in Table 6.1) is then used by each algorithm whose average ARI we show in a given plot. This protocol allows for better performance comparisons of the algorithms within the same plot.

Unless stated otherwise, the initial condition we use is the clustering induced by a segmentation based on a bottom eigenvector, that is an eigenvector associated with the smallest nonzero positive eigenvalue[7], of the Laplacian matrix (each experiment is specified by a different Laplacian matrix which will be used for its initial condition as well as the MBO evolution). In particular, we consider the interval given by the smallest and largest entries in the aforementioned eigenvector, and create K equally-sized bins spanning this interval. Finally, each node will be allocated in its corresponding bin according to its value in the eigenvector. The resulting characteristic matrix is used as the initial condition.

6.1. Best practice for MBO As explained in Section 3.1, one may use different versions of the Laplacians given by (3.6), (3.7) and (3.8). In particular, in the context of spectral clustering of (unsigned) graphs, L_{rw} and L_{sym} typically provide better results compared to L, especially in the context of skewed degree distributions [71, 75]. By referring to Figure A.2 in Appendix A.5, we observe a similar behaviour of their respective signed counterparts $\bar{L}_{\mathrm{rw}}$, $\bar{L}_{\mathrm{sym}}$ and L. Here, $\bar{L}_{\mathrm{sym}}$ achieves the highest accuracy and will be used in subsequent experiments. Moreover, we have tested the additional degree of freedom resulting from the time discretization procedure by running MBO with different orders of magnitude for the parameter $d\tau$. Results are detailed in Figure A.3 in Appendix A.5 and highlight that the accuracy is insensitive to different choices of the parameter $d\tau$.

6.2. Comparison to other methods The performance of our MBO algorithm is compared to that of the Balanced Normalized Cut (BNC) [41], and with the results achieved by standard Kmeans++ clustering [76, 77]. In this respect, as shown in Figure 6.1, MBO always provides better performance than Kmeans++, and improves BNC when sparsity and noise are high, which correspond to small λ and large η, respectively.

6.3. Must-link and cannot-link In this section, we detail how MBO can be applied in the context of constrained clustering. We have run the MBO algorithm for $\mathcal{L}=\mathcal{L}^{+}$, corresponding to the must-link scenario, and $\mathcal{L}=\bar{L}+\mathcal{Q}^{-}$, corresponding to the case in which only cannot-links are included as additional information in the definition of the Laplacian. In practice, we have considered the symmetric version of $\mathcal{L}$ in the same fashion as in (3.8). In the plots shown in Figure 6.2, we see that, for high sparsity and noise, the constrained clustering improves upon the standard signed Laplacian in both cases. In addition, the two approaches can be combined to obtain better results. In this case, we have included the known information up to $\alpha=0.1$ to not modify considerably the sparsity of the affinity matrix.

6.4. Fidelity and avoidance terms The second soft constraint we include in the MBO scheme is at the level of the characteristic matrix. The fidelity term pushes the characteristic matrix to be similar to $\hat{U}$ and increases the energy for every row U which is different from $\hat{U}$. Instead, for the avoidance term the energy of the GL functional is increased only when there is an exact match between two rows of U and $\tilde{U}$. This means that, in general, the effect of the avoidance term will be less influential since it provides an energy increase fewer times. This can be seen in Figure 6.3, where the black curve is

[7] Since the (signed) Laplacians we use are all positive semidefinite (and not identically zero) [38], a nonzero positive eigenvalue can always be found. When the geometric multiplicity of this eigenvalue is greater than one, we pick at random one of the elements of its eigenspace.

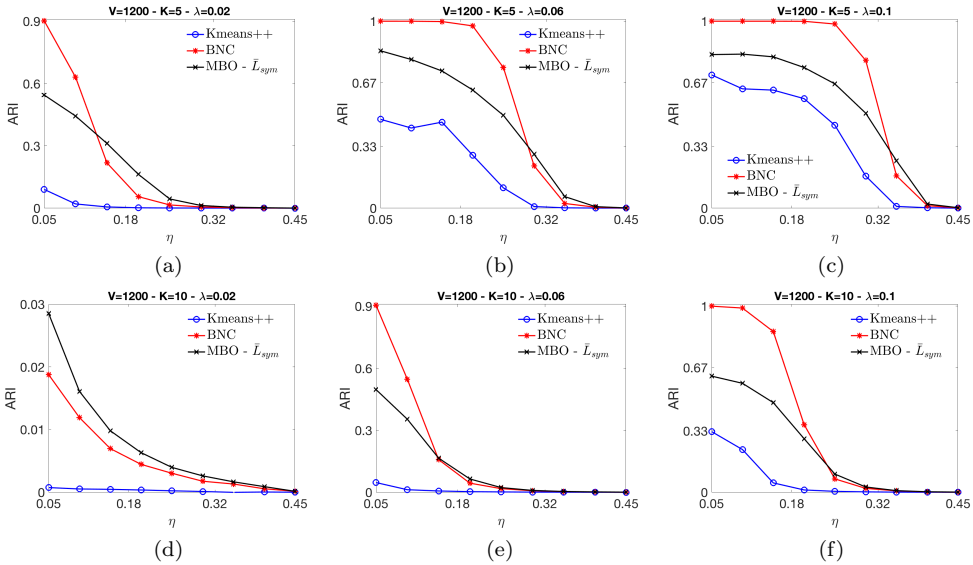


FIG. 6.1. *We have compared the node-cluster association recovery ability of MBO for signed Laplacians with two popular methods from the literature, Kmeans++ [76] and BNC [41]. From the plots, it is possible to see that Kmeans++ always performs worse than the other methods. MBO delivers better results than BNC for high noise level in the sparsity regime considered.*

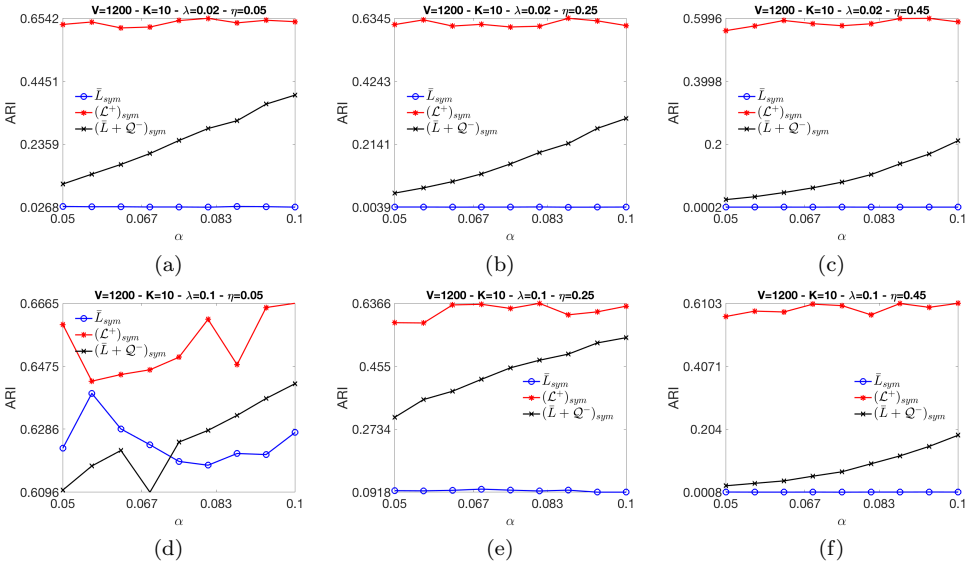


FIG. 6.2. *We compare MBO run over an instance of the affinity matrix generated from (6.4), against the same matrix where additional information extracted from the ground truth has been included in the form of must-links and cannot-links. This corresponds to adding soft constraints to the system in the form of edge weights. The parameter α is the fraction of links of the ground truth used to define the must- and cannot-link matrices. In this case, we have considered α up to 0.1 to avoid large variations in the sparsity of the matrix compared to the original one. For each Laplacian, we have considered its symmetric version (3.8). The algorithms are initialized using different eigenvectors since they are specified by different Laplacians. At any value of α, adding must-links and cannot-links significantly improves the recovery of the regular MBO for most sparsity and noise levels.*

always below the red one. We add that, in order to make the effect of the fidelity and avoidance terms more evident, we fix the nonzero diagonal elements of the matrices R and Q in (4.5) equal to 30.

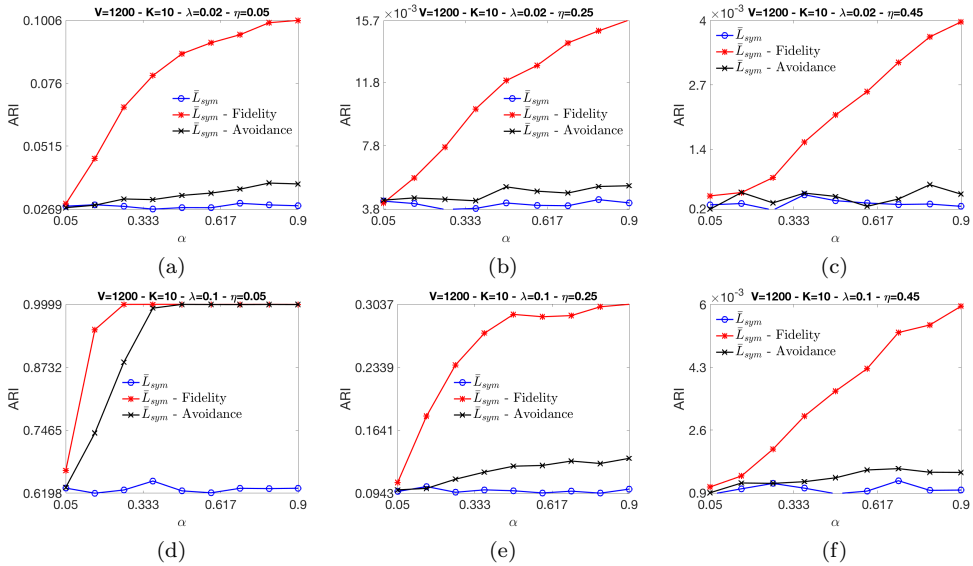


FIG. 6.3. *We consider two soft constraints at the level of the characteristic matrix U as expressed in (4.5). The fidelity/avoidance constraints are extracted at random from the ground truth for each experiment. For the fidelity/avoidance case, the relevant entries of the initial characteristic matrix are modified to be consistent with the constraints. As observed, adding the two constraints improves the results compared to those of the regular MBO procedure for all values of α. We remind that the matrices R and Q in (4.5) are multiplied by a coefficient equal to 30 in order to magnify the effect of the constraint (without varying the relative order of position of the three curves).*

7. Numerical experiments for the Barabási–Albert model In this section, we apply the MBO scheme to signed networks for which the underlying measurement graph (indicating the presence or absence of an edge) is generated from the *Barabási–Albert* (BA) model [78]. In a nutshell, the algorithm to generate the BA network begins with a connected network with V_0 nodes. At each time step, a new node is added to the graph and is connected to $\nu \leqslant V_0$ pre-existing nodes. The probability that a link from the new node connects to a generic node i is given by $p_i = \frac{D_{ii}}{\sum_j D_{jj}}$, where the sum indexed by j runs over all pre-existing nodes. We call $H \in \{0,1\}^{V\times V}$ the resulting BA network which can be used to build the relevant signed affinity matrix in the following way. As before, consider a row index $i = 1,\ldots,V$ and a column index $j = i,\ldots,V$ (the j index starts from i so to consider only the upper triangular part in the definition of the elements of A), we define the signed BA network as

$$A_{ij} = \begin{cases} S_{ij}H_{ij} & \text{if } i<j, \text{ with probability } (1-\eta) \\ -S_{ij}H_{ij} & \text{if } i<j, \text{ with probability } \eta \\ 0 & \text{if } i=j, \text{ with probability } 1 \end{cases} \qquad \text{with } A_{ji} = A_{ij}, \tag{7.1}$$

where the sparsity of the BA network is included via H_{ij} (the noise probability η is understood as in (6.4)) and S_{ij} are the elements of S in equation (6.1). We compared the numerical performance of MBO with that of BNC and Kmeans++, in the same spirit as in Section 6.1. The results are given in Figure 7.1, and are consistent with those obtained for the SSBM model.

8. Time series correlation clustering

This section details the outcomes of our numerical experiments on two time series data sets. Since ground truth data is not available, we compare the output of the algorithms by plotting the adjacency matrix sorted by cluster membership.

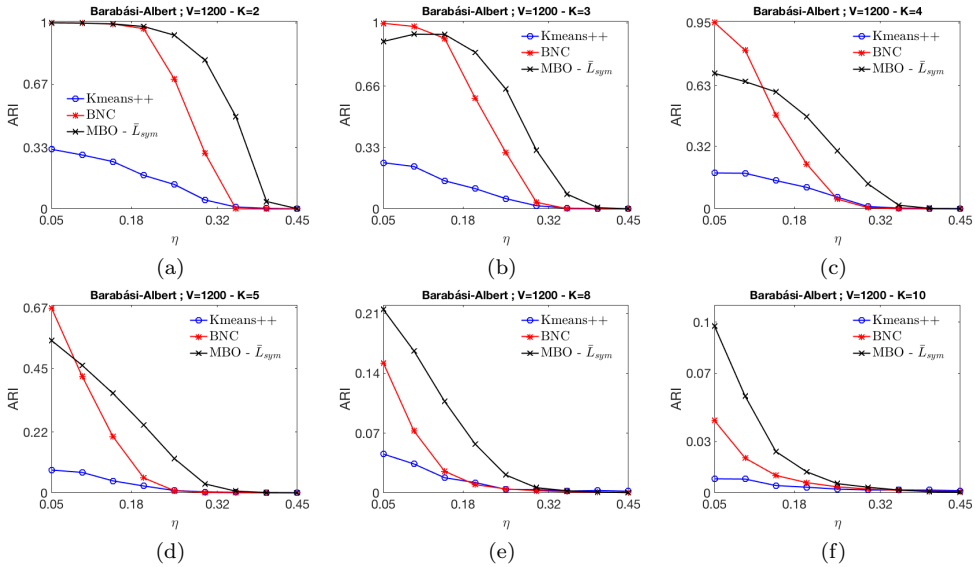


FIG. 7.1. *These plots compare the ground truth retrieval capabilities of different algorithms using as affinity matrix a signed version of the Barabási–Albert model, as explained in* (7.1). *As we can see, MBO dominates in the high noise regime for every sparsity level and number of clusters* $K \in \{2,3,4\}$ *we consider. In addition, we observe that for larger number of clusters, MBO performs better than the BNC.*

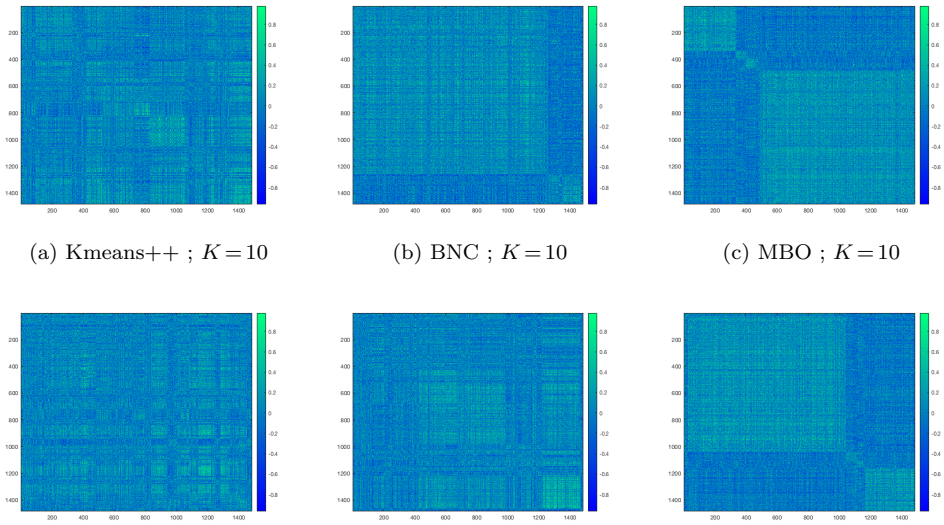
8.1. S&P 1500 Index We use time series daily prices for $n = 1500$ financial instruments, constituents of the S&P 1500 Composite Index, during 2003-2015, containing approximately $n_d = 3000$ trading days [83]. We consider daily log returns of the prices $R_{i,t} = \log \frac{P_{i,t}}{P_{i,t-1}}$, where $P_{i,t}$ denotes the market close price of instrument i on day t. In order to promote both positive and negative weights in the final correlation matrix we aim to cluster, we also compute the daily market excess return for each instrument,

$$\tilde{R}_{i,t} = R_{i,t} - R_{\mathrm{SPY},t}, \forall i = 1,\ldots,n,\ t = 1,\ldots,n_d, \tag{8.1}$$

where $R_{\mathrm{SPY},t}$ denotes the daily log return of SPY, the S&P 500 index ETF which is used as a proxy (weighted average) for the market. Next, we calculate the Pearson correlation between historical returns for each pair of instruments (i,j), which ultimately provides the edge weight A_{ij} in our signed graph. Figure 8.1 shows a segmentation of the US market into $K \in \{10,30\}$ clusters. The cluster structure is more obvious to the naked eye for the BNC and MBO methods, as opposed to the plain Kmeans++ clustering. The large cluster recovered by MBO could potentially correspond to the 1000 instruments which are within the S&P 1500 Index and outside of the S&P 500, but we defer this investigation to future work.

8.2. Foreign exchange This experiment pertains to clustering a foreign exchange FX currency correlation matrix, based on daily special drawing rights (SDR) exchange rates [84]. As shown in Figure 8.2, only BNC and MBO are able to recover four clusters, associated respectively to the EURO €, US Dollar $, UK Pound Sterling £, and Japanese Yen ¥. These are precisely the four currencies that, in certain percentage weights, define the value of the SDR reserve. To the naked eye, the adjacency matrix (sorted by cluster membership) recovered by MBO exhibits sharper transitions between clusters, compared to BNC and Kmeans++.

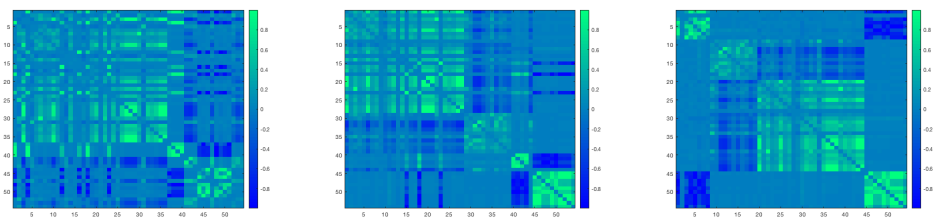
9. US migration As the last test bed for the proposed MBO scheme, we consider a migration network based on a data set in the 2000 US Census report [85, 86] which captures migration patterns between all pairs of counties in the US during the 1995-2000



(a) Kmeans++ ; $K=10$ (b) BNC ; $K=10$ (c) MBO ; $K=10$

(d) Kmeans++ ; $K=30$ (e) BNC ; $K=30$ (f) MBO ; $K=30$

FIG. 8.1. *The adjacency (correlation) matrix of the S&P 1500 time series data set, sorted by cluster membership, for $K \in \{10,30\}$ clusters.*



(a) Kmeans++ (b) BNC (c) MBO

FIG. 8.2. *The adjacency (correlation) matrix of the Foreign Exchange time series data set, sorted by cluster membership, for $K=4$. We remark that MBO exhibits sharper transitions between the four clusters (as highlighted by the green submatrices on the diagonal blocks), when compared to BNC and especially Kmeans++.*

time frame[8]. Here, we do not intend to capture the exact modelling of this sociological phenomenon but we aim only to compare in a real world context the different clustering capabilities of the Kmeans++, BNC, and the MBO. The network has $V=3107$ nodes, where each node corresponds to a county in mainland US. The adjacency matrix of the given network is asymmetric, with $M_{ij} \neq M_{ji}, 1 \leq i,j \leq V$. Similar to the approach in [87], we first symmetrize the input matrix, and let $\tilde{M}_{ij} = M_{ij} + M_{ji}$ denote the total number of people that migrated between county i and county j (thus $\tilde{M}_{ij} = \tilde{M}_{ji}$) during the five-year period. Finally, we turn the network into a signed one by approximately centering the distribution of edge weights at 0. More specifically, from each nonzero

[8]This publicly available data set was originally downloaded from https://www.census.gov/population/www.cen2000/ctytoctyflow/index.html. The website at that address is currently no longer available (last attempt to access on 10/08/2020). A version in a new format of the data set is now available at [85]. The reader is welcome to contact the corresponding author for questions about the data set. This product uses the Census Bureau Data API but is not endorsed or certified by the Census Bureau.

entry we remove the median value $\tilde{m}$ of the distribution of nonzero edge weights in $\tilde{M}$, and denote the resulting symmetric matrix by A, with $A_{ij} = \tilde{M}_{ij} - \tilde{m}$. In Table 9.1, we compare the outputs of plain Kmeans++, BNC, and our MBO approach, for a varying number of clusters $K \in \{5,10,15,20,30,40,50\}$. It can be seen that the plain Kmeans++ approach is not able to return any meaningful results across all values of k, and a similar statement holds true for BNC for low values of $K \in \{5,10,15,20\}$: in both cases, a single cluster captures almost all the nodes in the network. BNC returns meaningful results —in the sense that there are no clusters which contain only a few nodes— for larger numbers of clusters $K \in \{30,40,50\}$, aligning well with state boundaries. On the other hand, our proposed MBO algorithm returns meaningful results across all values of K, and the clusters align well with the true boundaries between the states, as in [87].

10. Summary and future directions

We introduced graph-based diffuse interface models utilizing the Ginzburg–Landau functionals, using an adaptation of the classic numerical Merriman–Bence–Osher (MBO) scheme for minimizing such graph-based functionals. Our approach has the advantage that it can easily incorporate various types of labelled data available to the user, in the context of semi-supervised clustering.

We have provided extensive numerical experiments that demonstrate the robustness to noise and sampling sparsity of the MBO scheme, under a signed stochastic block model. In particular, for sparse graphs and larger number of clusters, we often obtain more accurate results compared to state-of-the-art methods such as the BNC and the signed Laplacian.

Our work opens up a number of interesting directions for future investigation. It would be insightful to investigate the performance of the MBO scheme in the setting of very sparse graphs in the stochastic block model with $p = \Theta(\frac{1}{n})$, thus below the connectivity threshold, and potentially incorporate regularization terms considered in the recent literature [88, 89].

As a by-product of our research, we expect that the low-dimensional embeddings of signed networks (e.g., correlation matrices or related similarity/dissimilarity measures) arising from high-dimensional time series data obtained via the MBO scheme, could be of independent interest in the context of robust dimensionality reduction in multivariate time series analysis. Such embeddings could be used to learn nonlinear mappings from multivariate time series data, which has broad applications ranging from financial data analysis to biomedical research. Such mappings could prove useful for exploiting weak temporal correlations inherent in sequential data, and can be subsequently augmented with a regression or classification step, with the end goal of improving out-of-sample prediction. In a financial setting, this corresponds to building nonlinear risk/factor models for financial instruments, a task of paramount importance when it comes to understanding the risks that govern a financial system [90, 91].

An interesting potential future direction is the development of an MBO scheme for detecting structures in directed graphs. The vast majority of methods for structural analysis of graphs, for tasks such as clustering and community detection, have been developed for undirected graphs, and a number of challenges arise when considering directed graphs, complicating the immediate extension of existing algorithms for the undirected case. Various directed extensions typically consider a symmetrized version of the adjacency matrix [92–95]. In ongoing work by a subset of the authors [96], we consider the problem of clustering directed graphs, where the clustering structure arises due to the existence of certain pairs of imbalanced cuts in the graph. More specifically, an imbalanced cut between two sets of vertices X and Y is one where the

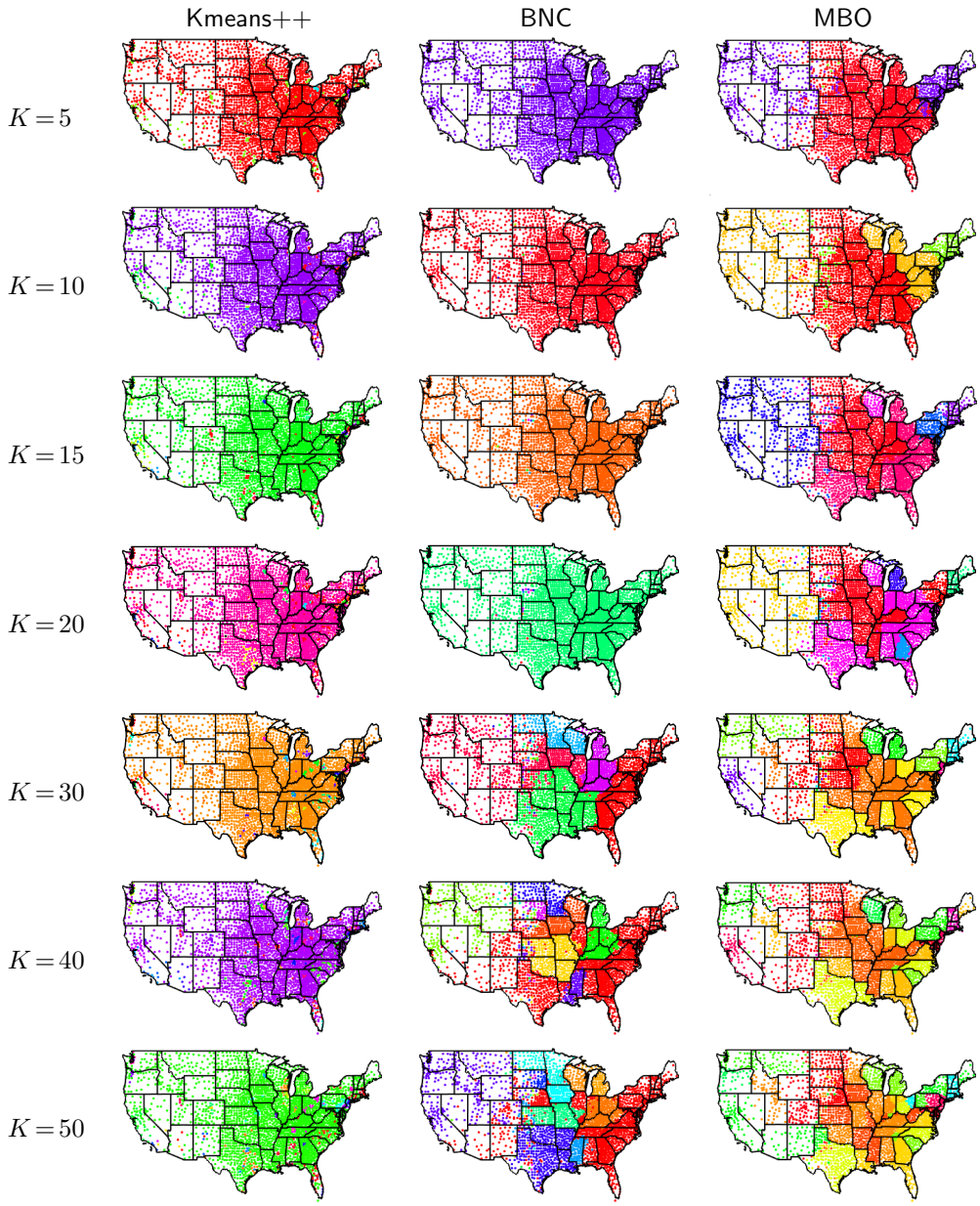


TABLE 9.1. *Clusterings computed by Kmeans++, BNC, and MBO for the signed US Migration network, for varying number of clusters K. Each color, extracted from a continuous color map, corresponds to a different cluster.*

majority of the edges are directed from X to Y, with few edges being directed in the opposite direction from Y to X. The goal is to find k clusters such that the sum of all (or many of) the $\binom{k}{2}$ pairwise imbalanced cuts is maximized, while also penalizing the existence of small clusters in order to obtain a more balanced partition. In [96], we consider a Hermitian matrix derived directly from the skew-symmetric adjacency

matrix, and recover an embedding from the spectrum of its corresponding (Hermitian) graph connection Laplacian, potentially after normalization [69, 97]. Extending the Ginzburg–Landau functional and the MBO scheme to the setting of directed graphs and Hermitian Laplacians is an appealing future research direction.

Acknowledgements. Francesco Cosentino is acknowledged for his contribution in the derivation of the proof of Lemma A.2.

Funding. MC and AP acknowledge support from The Alan Turing Institute EPSRC grant EP/N510129/1 and seed funding project SF029 *"Predictive graph analytics and propagation of information in networks"*. AP also acknowledges support from the National Group of Mathematical Physics (GNFM-INdAM), by Imperial College together with the Data Science Institute and Thomson-Reuters Grant No. 4500902397-3408 and EPSRC grant EP/P002625/1. YvG did a substantial part of the work which has contributed to this paper at the University of Nottingham. YvG acknowledges that this project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 777826.

Appendix A.

A.1. Notation Table The notation used in the present work can be found in Table A.1.

A.2. Ginzburg–Landau functional

The *Ginzburg–Landau functional* (GLF) was originally introduced in the materials science literature to study phase separation, and has since also found many uses in image processing and image analysis due to its connections with the total variation functional [48, 49]. In this section, we briefly review the main idea behind the GLF together with a description of the (local) minimizer(s) of the functional in terms of solutions of the *Allen–Cahn equation* (ACE). Then, we will show how this approach can be adapted to the discrete case of clustering over signed graphs.

A.2.1. GLF as potential energy In this introductory presentation, which is based on [50, 51], we restrict to the one dimensional case for simplicity. Consider the function ρ in the Sobolev space $\mathscr{W}^{1,2}([0,1])$. The GLF $F_\varepsilon : \mathscr{W}^{1,2}([0,1]) \mapsto [0,\infty)$ is defined as

$$F_\varepsilon[\rho] := \frac{\varepsilon}{2}\int_0^1 \left(\frac{d\rho}{ds}\right)^2 ds + \frac{1}{4\varepsilon}\int_0^1 \left(1-\rho^2\right)^2 ds, \tag{A.1}$$

for some $\varepsilon > 0$. The second term has the shape of a double-well and is often labelled $W(\rho) := \left(1-\rho^2\right)^2 \in \mathbb{R}_+$. The GLF is interpreted as the *potential energy* for the physical observable ρ. At this stage, ρ is also understood as time dependent and we want to find solutions of

$$\frac{d\rho}{dt}(t,x) = -\frac{\delta F_\varepsilon}{\delta \rho}[\rho](t), \tag{A.2}$$

where $\frac{\delta F}{\delta \rho}$ is the functional derivative of F, which lead in their time evolution to the local minima of the GLF. A common choice for the boundary values of the problem (A.2) is the *von Neumann condition*

$$\frac{d\rho}{dx}(t,1) = \frac{d\rho}{dx}(t,0) = 0 \quad \text{for} \quad t > 0. \tag{A.3}$$

Calculating the functional derivative of the right-hand side in (A.2) gives rise to the ACE

$$\frac{d\rho}{dt}(t,x) = \varepsilon \frac{d^2\rho}{dx^2}(t,x) - \frac{1}{\varepsilon}\rho(t,x)\left(\rho^2(t,x)-1\right), \tag{A.4}$$

Symb.	Description
α	percentage of known information
A	affinity matrix of the signed case
A^+	positive part of A
A^-	negative part of A
B	time evolution matrix in the MBO scheme with affinity matrix N
$\bar{B}$	time evolution matrix in the MBO scheme with affinity matrix A
c_i	size of clusters of the Signed Stochastic Block Model
$d\tau$	time discretization parameter
D	degree matrix of N
$\bar{D}$	degree matrix of A
$\underline{e}$	element of the vertices of the probability simplex
$\underline{e}^{\pm}$	element of the vertices of $\Sigma_K^{\pm}$
ε	parameter of the GL functional
F_ε	GL functional
$\bar{F}_\varepsilon$	signed GL functional
I_V	Identity matrix with size V
K	number of clusters
Λ	eigenvalue matrix
L	Laplacian of N
L^+	Laplacian of A^+
L_{rw}	random walk Laplacian of N
L_{sym}	symmetric Laplacian of N
$\bar{L}$	signed Laplacian
$\bar{L}_{\text{rw}}$	signed random walk Laplacian
$\bar{L}_{\text{sym}}$	signed symmetric Laplacian
λ	sparsity probability
λ^+, λ^-	trade-off parameters for must- and cannot-links, respectively
m	number of eigenvector considered in the expansion of $\bar{B}$
n	step number in the MBO algorithm
N	affinity matrix of the positive edges case
N_τ	number of repetitions of the diffusion step
$\mathbb{N}_+$	natural numbers excluding 0.
η	noise probability
Q^-	signless Laplacian

Symb.	Description
ρ	density for $1d$ continuous version of the GL functional
$\underline{\rho}$	vertex set values of the discrete bi-dimensional case
R^{fi}, R^{av}	weight matrix for the fidelity and avoidance contributions, respectively
$\mathbb{R}$	real numbers
$\mathbb{R}_+$	positive real numbers excluding 0
$\mathbb{R}_{+0}$	positive real numbers including 0
S	ground truth affinity matrix for the Stochastic Signed Block Model
σ	component of the probability simplex
$\sigma^{\pm}$	component of the $\Sigma_K^{\pm}$ simplex
Σ_K	probability simplex
$\Sigma_K^{\pm}$	'$2-K$' simplex (see Equation (3.14))
t	time variable
$\underline{u}$	column vector carrying the weight of a node on various clusters
$\underline{u}^{\pm}$	column vector carrying the weight of a node on various clusters in the signed case
U	node-cluster association matrix
$U^{\pm}$	node-cluster association matrix in the signed case
V	number of nodes
x	space coordinate
X	eigenvectors matrix
$\underline{0}_K$	column vector of length K whose elements are all 0
$0_{V\times K}$	matrix with V rows and K columns whose elements are all 0
$\underline{1}_V$	column vector of length V whose elements are all 1
1_V	square matrix of size V whose elements are all 1
$\circ$	Hadamard product
$\lvert\cdot\rvert$	absolute value
$\lVert\cdot\rVert_1$	taxicab metric
$\lVert\cdot\rVert_2$	Euclidean distance

TABLE A.1. *Summary of frequently used symbols in the paper.*

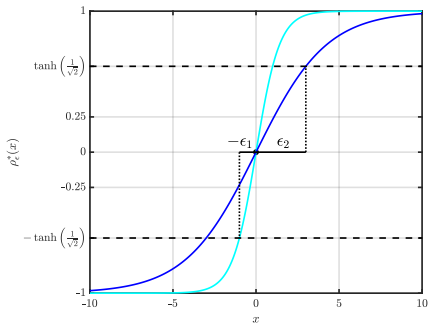


FIG. A.1. *Plot of the stationary solution* (A.6) *of the ACE* (A.4) *for two parameter values* ε_2 *(blue curve) and* ε_1 *(cyan curve) with* $\varepsilon_2 > \varepsilon_1$. *We can see that bigger values of* ε *reduce sharpness of the curve* $\rho_\varepsilon^*(x)$. *Moreover,* $\rho_{\varepsilon_1}^*(\varepsilon_1) = \rho_{\varepsilon_2}^*(\varepsilon_2) = \tanh\left(\frac{1}{\sqrt{2}}\right) \approx 0.609$.

where the last term is often called $W'(\rho) := \rho\left(\rho^2 - 1\right) \in \mathbb{R}$. Moreover, it is possible to verify by direct calculation that the GLF is decreasing in time on the solution of the ACE, that is

$$\frac{dF_\varepsilon[\rho]}{dt}(t) = -\int_0^1 \left(\frac{d\rho}{dt}(t,s)\right)^2 ds < 0, \tag{A.5}$$

as is expected for the evolution of a physical observable along its energy potential. The meaning of the coefficient ε can be understood as follows. A stationary solution for the ACE, such that $\frac{d\rho}{dt}(t,x) = 0 \forall t \in \mathbb{R}_+$ in (A.4), is given by

$$\rho_\varepsilon^*(x) := \tanh\left(\frac{x}{\varepsilon\sqrt{2}}\right). \tag{A.6}$$

By taking the first derivative with respect to the spatial coordinate, we can see that the parameter ε is responsible for the sharpness of the solution (see Figure A.1). Thus, given the values of the function ρ_ε^* at the boundary of its domain, the ACE can be understood as the evolution of the layer profile connecting the two boundary values of the system.

A.2.2. GLF and clustering Using tools from nonlocal calculus, see Section 2.3 in [1], it is possible to write a version of the GLF where the density function is defined on a network. To fix the notation, we denote as $G(V,E,N)$ a positive-weighted undirected graph with *vertex set* V, *edge set* E and *affinity matrix* N which is a symmetric matrix whose entries N_{ij} are zero if $i = j$ or $i \not\sim j$ and nonnegative otherwise. From the affinity matrix, ones defines the diagonal *degree matrix* as

$$D_{ij} := \begin{cases} \sum_{k=1}^V N_{ik} \in \mathbb{R}_{+0} & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \tag{A.7}$$

where V is the cardinality of the vertex set. When the entries of the affinity matrix are positive, the associated network is often called an *unsigned graph*. In the same way, the discrete equivalent of the density function is now a column vector on the vertex set, that is $\varrho : \mathbb{N} \times [0,T] \mapsto \mathbb{R}^V$, written as

$$\varrho(t) := \{\rho_0(t), \rho_1(t), \ldots, \rho_V(t)\}. \tag{A.8}$$

Following [3], it turns out that the network version of (A.1) for unsigned graphs, also called *graph GLF* (GGLF), is given by

$$F_\varepsilon(\varrho) := \frac{\varepsilon}{2}\langle \varrho, L\varrho \rangle + \frac{1}{4\varepsilon}\sum_{i=1}^V (\rho_i^2 - 1)^2, \tag{A.9}$$

where $L := D - N \in \mathbb{R}^{V \times V}$ is the (combinatorial) graph Laplacian, and $\langle \cdot, \cdot \rangle$ denotes the usual inner product on $\mathbb{R}^V$. It is important to notice that the following expression holds true

$$\langle \varrho, L\varrho \rangle = \sum_{i=1}^{V} N_{ij} (\rho_i - \rho_j)^2 \geq 0. \tag{A.10}$$

The inequality in (A.10) shows that L is positive semidefinite (i.e., L has non-negative, real-valued eigenvalues, including 0) and this assures the non-negativity of the GGLF (A.9). Again, the minimization of the GGLF can shed light on the interpretation of the terms composing the potential. The second term in F_ε will force minimizers $\rho \in \mathbb{R}^V$ to have entries which are close to -1 or 1 in order to keep $(\rho_i^2 - 1)^2$ small. The first term in F_ε penalizes the assignment to different clusters of nodes that are connected by highly weighted edges. This means that minimizers of F_ε are approximate indicator vectors of clusters that have few highly weighted edges between them. We interpret nodes having different sign as belonging to different clusters.

As before, minimizing the GGLF using the method of the gradient descent, we obtain the ACE for networks which we write component-wise as

$$\frac{d\rho_i}{dt}(t) = -\varepsilon [L\varrho]_i (t) - \frac{1}{\varepsilon} \rho_i (t) \left(\rho_i^2 (t) - 1 \right), \tag{A.11}$$

where $[L\varrho]_i$ indicates the i-th component of the vector $L\varrho$. Instead of the combinatorial graph Laplacian L, it is possible to use the random walk Laplacian $L_{\mathrm{rw}} := I - D^{-1}N$ (which is positive semidefinite), or the symmetric graph Laplacian $L_{\mathrm{sym}} := I - D^{-1/2} N D^{-1/2}$ (symmetric positive semidefinite). In fact, [52] argues for the use of L_{sym}, but no rigorous results are known that confirm this is necessarily the best choice for each application.

A.2.3. Numerical solution of the ACE

Various methods have been employed to numerically compute (approximate) minimizers of F_ε. In [52] a gradient flow approach is used and (A.11), interpreted as a system of ordinary differential equations (ODEs), is solved. The hope is that the gradient flow dynamics will converge to a local minimizer of F_ε. Under some assumptions [53], this hope is theoretically justified [54], but numerically the nonconvexity of F_ε can cause some issues. In [52] such issues were tackled with a convex splitting scheme.

Another strategy which has been developed to approximately solve the ACE for networks consists in using a variation of the *Merriman–Bence–Osher* (MBO) scheme (also known as *threshold dynamics*). This was first introduced in [55,56] in a continuum setting as a proposed algorithm to approximate flow by mean curvature. Following [1], we first neglect the terms $W'(\rho_i)$ and discretize (A.11) in the following way

$$\frac{\varrho^{n+\frac{1}{2}} - \varrho^n}{d\tau} = -\varepsilon L \varrho^{n+\frac{1}{2}}, \tag{A.12}$$

where $d\tau \in \mathbb{R}_+$ is the length of a small time interval to be tuned and the superscript n indicates the step of the iterative procedure ($n + \frac{1}{2}$ simply specifies an intermediate step between the n-th step and the subsequent one which will be labelled as $n+1$). The above leads to the *MBO equation* (MBOE)

$$\varrho^{n+\frac{1}{2}} = (I_V + \varepsilon d\tau L)^{-1} \varrho^n, \tag{A.13}$$

where $I_V \in \{0,1\}^{V \times V}$ is the identity matrix. After performing the eigendecomposition of the Laplacian, (A.13) becomes

$$\varrho^{n+\frac{1}{2}} = \left[X (I_V + d\tau\varepsilon\Lambda) X^T \right]^{-1} \varrho^n, \tag{A.14}$$

where X is a matrix whose columns are the normalized eigenvectors of L and Λ is the relevant eigenvalue diagonal matrix. Since the double-well potential pushes the components of $\underline{\rho}$ to be either $+1$ or -1, its role can be approximated via a thresholding step, in the following way

$$\rho_i^{n+1} = T\left(\rho_i^{n+\frac{1}{2}}\right) := \begin{cases} +1 \text{ if } \rho_i^{n+\frac{1}{2}} \geqslant 0 \\ -1 \text{ if } \rho_i^{n+\frac{1}{2}} < 0 \end{cases}. \tag{A.15}$$

This procedure can be iterated until a convergence criterion is met. In practice, the diffusion step (A.13) is repeated $N_\tau \in \mathbb{N}_+$ times before the thresholding step (A.15). This corresponds to multiplying $\underline{\rho}^n$ in (A.13) by $\left[X\left(I_V + \frac{d\tau}{N_\tau}\varepsilon\Lambda\right)X^T\right]^{-N_\tau}$ (note the division of $d\tau$ by N_τ).

The graph version of MBO has been used successfully in recent years in various applications, both applied to F_ε [1], its multi-class extension [57], and its signless Laplacian variant [58]. In general, we notice that the choice of ε also poses a problem. Smaller values of ε are expected to lead to minimizers $\underline{\rho}$ that are closer to being indicator vectors, however the discrete length scales of the problem impose a lower bound on the ε values that can be resolved. The MBO scheme avoids this problem since the $W'(\rho_i)$ are substituted with a thresholding step. In this way, we can fix $\varepsilon = 1$. Using the MBO scheme instead of the gradient flow does not solve all problems: It eliminates the terms in the ODEs that originated with the non-convex term in F_ε and it removes ε from consideration, but instead a value for $d\tau$ has to be chosen. The discrete length scales in the problem disallow values that are too small, yet $d\tau$ has to be small enough to prevent oversmoothing by the diffusion term $[L\underline{\rho}]_i$.

A.2.4. Generalization to multiple clusters

The Ginzburg–Landau method has been adapted to allow for clustering or classification into $K > 2$ clusters [59–62]. We briefly review this line of work here, since our proposed approach for clustering signed graphs is for general K as well. This construction relies upon a *characteristic matrix* whose entries correspond to the weight for a node i to belong to a cluster k. More precisely, consider the simplex

$$\Sigma_K := \left\{ \left(\sigma_1, \sigma_2, \ldots, \sigma_K\right) \in [0,1]^K : \sum_{k=1}^{K} \sigma_k = 1 \right\}, \tag{A.16}$$

which corresponds to row vectors whose components sum up to 1. Then, we can associate to each node a *characteristic vector*[9], written as a column vector

$$\underline{u}_i := \left(u_{i1}, u_{i2}, \ldots, u_{iK}\right)^T \in \Sigma_K, \tag{A.17}$$

which specifies the weight for the node i to belong to each of the clusters in the network. So, the characteristic matrix of all the nodes in the system is defined as

$$U := \left(\underline{u}_1, \underline{u}_2, \ldots, \underline{u}_V\right)^T \in (\Sigma_K)^V. \tag{A.18}$$

In the end, as explained in [61], the generalization of the GLF reads as

$$F_\varepsilon(U) := \frac{\varepsilon}{2}\langle U, LU\rangle + \frac{1}{2\varepsilon}\sum_{i=1}^{V}\left(\prod_{k=1}^{K}\frac{1}{4}\|\underline{u}_i - \underline{e}_k\|_1^2\right), \tag{A.19}$$

[9] We generally refer to the vector whose entries are the "intensity" of belongingness to a cluster as *characteristic vector* also if these coefficients are not only 0 or 1.

where $\|\cdot\|_1$ is the taxicab metric, $\langle U, LU\rangle = \mathrm{Tr}\left(U^T LU\right)$ and $\underline{e}_k$ correspond to the canonical basis vector defined as

$$\underline{e}_k := \left(e_1, e_2, \ldots, e_K\right)^T \in \{0,1\}^K, \tag{A.20}$$

whose components satisfy $e_{i=k} = 1$ and $e_{i\neq k} = 0$. In this context, the MBO scheme for the characteristic matrix reads as

$$U^{n+\frac{1}{2}} = BU^n, \quad \text{where} \quad B := \left[X\left(I_V + \frac{d\tau}{N_\tau}\Lambda\right)X^T\right]^{-N_\tau} \in \mathbb{R}^{V\times V} \tag{A.21}$$

where we have fixed $\varepsilon = 1$, as explained at the end of Appendix A.2.3, and we have included the contribution from the diffusion iteration directly in B as explained below in (A.15). Following [61], we project each row of $U^{n+\frac{1}{2}}$ on the simplex Σ_K. We call the corresponding vector

$$\underline{v}_i := \underset{\underline{y}\in\Sigma_K}{\mathrm{argmin}} \left\|\underline{u}_i^{n+\frac{1}{2}} - \underline{y}\right\|_2 \in [0,1]^K. \tag{A.22}$$

It is important to notice that in the intermediate step $\underline{u}_i^{n+\frac{1}{2}} \in \mathbb{R}^K$. The thresholding step consists in determining the closest vertex to $\underline{v}_i$ of Σ_K, that is[10]

$$\underline{u}_i^{n+1} = \underline{e}_k \quad \text{with} \quad k = \underset{j=1,\ldots,K}{\mathrm{argmin}} \left\|\underline{v}_i - \underline{e}_j\right\|_2. \tag{A.23}$$

In the end, we have $U^{n+1} = \left(\underline{u}_1^{n+1}, \underline{u}_2^{n+1}, \ldots, \underline{u}_V^{n+1}\right)^T$. In Section 3.2, we will use a simplified version of the MBO scheme which reduces the projection (A.22) and thresholding (A.23) to a single step (the equivalence of these two versions is proved in the Appendix A.3).

A.3. Projection and Thresholding on Σ_K and $\Sigma_K^\pm$ Let $K \in \mathbb{N}_+$. In this section, we show that computation of the projection on Σ_K (or $\Sigma_K^\pm$) followed by the thresholding step in the MBO scheme in (A.22)–(A.23) is equivalent to determining a maximal component of every vector $\underline{u}_i^{n+\frac{1}{2}}$. To fix notation, the vertices of Σ_K are the elements of the canonical basis in $\mathbb{R}^{\mathbb{K}}$ (for instance, for $K=3$ we have $\underline{e}_1 = (1,0,0)^T$, $\underline{e}_2 = (0,1,0)^T$ and $\underline{e}_3 = (0,0,1)^T$), while those of $\Sigma_K^\pm$ are defined as $\underline{e}_k^\pm := 2\underline{e}_k - \underline{1}_K \in \{-1,1\}^V$ (that is, for the $K=3$ case, we have $\underline{e}_1^\pm = (1,-1,-1)^T$, $\underline{e}_2^\pm = (-1,1,-1)^T$ and $\underline{e}_3^\pm = (-1,-1,1)^T$). We first remind the reader of a useful equality for vectors $\underline{a}$, $\underline{b} \in \mathbb{R}^K$:

$$\begin{aligned}\|\underline{a}-\underline{b}\|_2^2 - \|\underline{a}-\underline{c}\|_2^2 &= \|\underline{a}\|_2^2 + \|\underline{b}\|_2^2 - 2\langle \underline{a},\underline{b}\rangle_2 - \|\underline{a}\|_2^2 - \|\underline{c}\|_2^2 + 2\langle \underline{a},\underline{c}\rangle_2 \\ &= \|\underline{b}\|_2^2 - \|\underline{c}\|_2^2 + 2\langle \underline{a}, \underline{c}-\underline{b}\rangle_2,\end{aligned} \tag{A.24}$$

where we have used the standard inner product in $\mathbb{R}^K$: $\langle \underline{a},\underline{b}\rangle_2 := \sum_{k=1}^K a_k b_k$. The following lemma lists three direct consequences of (A.24).

Lemma A.1.

1. *Given $\underline{a}$, $\underline{b} \in \mathbb{R}^K$ and $i,j \in \{1,\ldots,K\}$, define $\underline{b}^* \in \mathbb{R}^K$ by*

$$b_k^* := \begin{cases} b_k, & \text{if } i \neq k \neq j, \\ b_j, & \text{if } k = i, \\ b_i, & \text{if } k = j. \end{cases} \tag{A.25}$$

Then

$$\|\underline{a}-\underline{b}^*\|_2^2 = \|\underline{a}-\underline{b}\|_2^2 + 2(a_i - a_j)(b_i - b_j). \tag{A.26}$$

[10] When $\underline{v}_i$ has the same distance to multiple vertices, one of those is picked at random with uniform probability.

2. *Let* $\underline{x}\in\mathbb{R}^K$ *and let* $\underline{e}_i,\underline{e}_j\in\mathbb{R}^K$ *be two vertices of the simplex* Σ_K. *Then* $\|\underline{x}-\underline{e}_i\|_2\geqslant\|\underline{x}-\underline{e}_j\|_2$ *iff* $x_j\geqslant x_i$, *with equality between the norms iff* $x_i=x_j$.
3. *Let* $\underline{x}\in\mathbb{R}^K$ *and let* $\underline{e}_i^{\pm},\underline{e}_j^{\pm}\in\mathbb{R}^K$ *be two vertices of the simplex* $\Sigma_K^{\pm}$. *Then* $\|\underline{x}-\underline{e}_i^{\pm}\|_2\geqslant\|\underline{x}-\underline{e}_j^{\pm}\|_2$ *iff* $x_j\geqslant x_i$, *with equality between the norms iff* $x_i=x_j$.

Proof.

1. We apply (A.24) with $\underline{c}=\underline{b}^*$. Since $\|\underline{b}\|_2=\|\underline{b}^*\|_2$ and
$$\langle\underline{a},\underline{c}-\underline{b}\rangle_2=a_i(b_j-b_i)+a_j(b_i-b_j)=(a_i-a_j)(b_j-b_i), \tag{A.27}$$
the result follows.
2. We apply (A.24) with $\underline{a}=\underline{x}$, $\underline{b}=\underline{e}_i$, and $\underline{c}=\underline{e}_j$. Since $\|\underline{e}_i\|_2=1=\|\underline{e}_j\|_2$ and $\langle\underline{x},\underline{e}_j-\underline{e}_i\rangle_2=x_j-x_i$, it follows that $\|\underline{x}-\underline{e}_i\|_2^2-\|\underline{x}-\underline{e}_j\|_2^2=2(x_j-x_i)$, and thus the result follows.
3. This follows from a proof completely analogous to the previous one, once we note that $\|\underline{e}_i^{\pm}\|_2=\sqrt{K}=\|\underline{e}_j^{\pm}\|_2$ and $\langle\underline{x},\underline{e}_j^{\pm}-\underline{e}_i^{\pm}\rangle_2=2(x_j-x_i)$ where $\underline{x}\in\mathbb{R}^K$.

□

Lemma A.2. *Let* $\underline{x}\in\mathbb{R}^K$ *and*
$$\underline{x}^{\perp}:=\operatorname*{argmin}_{\underline{y}\in\Sigma_K}\|\underline{x}-\underline{y}\|_2 \tag{A.28}$$
be its projection on Σ_K[11]. *Given two components* x_j *and* x_i *of* $\underline{x}$, *if* $x_j\geqslant x_i$ *then* $x_j^{\perp}\geqslant x_i^{\perp}$. *The analogous result holds for the projection of* $\underline{x}$ *on* $\Sigma_K^{\pm}$.

Proof. We will prove the result for the projection on Σ_K. A completely analogous proof gives the result for $\Sigma_K^{\pm}$. First note that the result is trivially true if $i=j$, hence we assume that $i\neq j$. For a proof by contradiction, assume that $x_j\geq x_i$ and $x_j^{\perp}<x_i^{\perp}$. Define $\underline{x}^{\perp *}$ by
$$x_k^{\perp *}:=\begin{cases}x_k^{\perp}, & \text{if } i\neq k\neq j,\\ x_j^{\perp}, & \text{if } k=i,\\ x_i^{\perp}, & \text{if } k=j.\end{cases} \tag{A.29}$$
Then, by the first result of Lemma A.1 (with $\underline{a}=\underline{x}$ and $\underline{b}=\underline{x}^{\perp}$), we have that
$$\|\underline{x}-\underline{x}^{\perp *}\|_2^2-\|\underline{x}-\underline{x}^{\perp}\|_2^2=2(x_i-x_j)(x_i^{\perp}-x_j^{\perp})\leq 0, \tag{A.30}$$
The inequality follows from our assumptions on x_i, x_j, $x_i^{\perp}$, and $x_j^{\perp}$. Since $\underline{x}^{\perp}\in\Sigma_K$, we have $\sum_{k=1}^K x_k^{\perp *}=\sum_{k=1}^K x_k^{\perp}=1$, hence $\underline{x}^{\perp *}\in\Sigma_K$. Thus, per definition (A.28), we have $\|\underline{x}-\underline{x}^{\perp *}\|_2^2-\|\underline{x}-\underline{x}^{\perp}\|_2^2\geq 0$. It follows that $\|\underline{x}-\underline{x}^{\perp *}\|_2=\|\underline{x}-\underline{x}^{\perp}\|_2$. By uniqueness of the minimizer in (A.28) we deduce that $\underline{x}^{\perp *}=\underline{x}^{\perp}$. In particular that means that $x_j^{\perp}=x_j^{\perp *}=x_i^{\perp}$, which is a contradiction. □ Applying Lemma A.1 (result 2) to the thresholding step in (A.23), we find that k in (A.23) is such that the kth entry of $\underline{v}_i$ is not smaller than all its other entries. Furthermore, Lemma A.2 applied to (A.22) tells us that $\underline{v}_i$ and $(\underline{u}_i^{n+\frac{1}{2}})$ have their largest entries at the same positions. Thus (A.22) and (A.23) are equivalent to $\underline{u}_i^{n+1}=\underline{e}_k$, where k is such that the kth entry of $(\underline{u}_i^{n+\frac{1}{2}})$ is not smaller than all its other entries.

A.4. Adjusted Rand Index Consider the vector $\underline{v}^*$ and its associated nodes partition
$$\mathscr{P}(\underline{v}^*)=\{\mathscr{P}_1(\underline{v}^*),\mathscr{P}_2(\underline{v}^*),\ldots,\mathscr{P}_K(\underline{v}^*)\}$$
into K clusters, where $\mathscr{P}_i(\underline{v}^*)$ is a set whose elements are those nodes belonging to cluster i. Also for the ground truth vector (6.2), it is possible to construct a partition $\mathscr{P}(\underline{s})$ which is in general different from $\mathscr{P}(\underline{v}^*)$. Then, the ARI is defined as

[11] Existence and uniqueness of $\underline{x}^{\perp}$ is a well-known result, see for instance Lemma 2.2 in [98]

$$\mathrm{ARI}(\underline{v}^*,\underline{s}) := \frac{\sum_{ij}\binom{p_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j\binom{b_j}{2}\right]/\binom{V}{2}}{\frac{1}{2}\left[\sum_i\binom{a_i}{2}+\sum_j\binom{b_j}{2}\right] - \left[\sum_i\binom{a_i}{2}\sum_j\binom{b_j}{2}\right]/\binom{V}{2}} \in \mathbb{R}, \tag{A.31}$$

where a_i, b_j and p_{ij} are extracted from the *contingency table*

$$\begin{array}{c|cccccc|c}
\underline{v}^* \diagdown \underline{s} & \mathscr{P}_1(\underline{s}) & \mathscr{P}_2(\underline{s}) & \dots & \mathscr{P}_j(\underline{s}) & \dots & \mathscr{P}_K(\underline{s}) & \sum_{j=1}^K p_{ij} \\
\hline
\mathscr{P}_1(\underline{v}^*) & p_{11} & p_{12} & \cdots & p_{1j} & \cdots & p_{1K} & a_1 \\
\mathscr{P}_2(\underline{v}^*) & p_{21} & p_{22} & \cdots & p_{2j} & \cdots & p_{2K} & a_2 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\
\mathscr{P}_i(\underline{v}^*) & p_{i1} & p_{i2} & \cdots & p_{ij} & \cdots & p_{iK} & a_i \\
\vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\
\mathscr{P}_K(\underline{v}^*) & p_{K1} & p_{K2} & \cdots & p_{Kj} & \cdots & p_{KK} & a_K \\
\hline
\sum_{i=1}^K p_{ij} & b_1 & b_2 & \dots & b_j & \dots & b_K &
\end{array}, \tag{A.32}$$

given $p_{ij} := |\mathscr{P}_i(\underline{v}^*) \cap \mathscr{P}_j(\underline{s})| \in \mathbb{N}$ the number of common nodes between $\mathscr{P}_i(\underline{v}^*)$ and $\mathscr{P}_j(\underline{s})$.

A.5. Results for Section 6.1 In this section of the Appendix, we show the results discussed in Section 6.1 using the values in Table 6.1. In Figure A.2, we can see that $\bar{L}_{\mathrm{sym}}$ provides better results compared to $\bar{L}$ and $\bar{L}_{\mathrm{rw}}$. In Figure A.3, we show that different order of magnitudes of the diffusion time do not change the ARI for different values of the parameters, as is clear from the curves overlap.

A.6. Anchors In general, *anchors* correspond to a subset of the network nodes whose values are fixed once and for all. For example, in the context of sensor network localization (that aims to find coordinates of sensors in $\mathbb{R}^d$ from given pairwise distance measurements; typically $d=\{2,3\}$), anchors are sensors that are aware of their location, and anchor-based algorithms make use of their existence when computing the coordinates of the remaining sensors [69]. In the context of the *molecule problem* from structural biology applications [70] (motivated by NMR), there are molecular fragments (i.e., embeddings of certain subgraphs) whose local configuration is known in advance up to an element of the special Euclidean group SE(3), rather than the Euclidean group E(3). In other words, in the terminology of signed clustering for $k=2$, there exists a subset of nodes (anchors) whose corresponding ground truth assigned value $\{-1,1\} \in \mathbb{Z}_2$ (i.e., cluster membership) is known a priori, and can be leveraged to find the values of the remaining non-anchor nodes.

In the context of multiple clusters, anchors are understood as those nodes whose cluster membership is known. From the point of view of the MBO, we can consider anchors as fixed points in the dynamics, that is from (A.12)[12] $\frac{\underline{u}_i^{n+\frac{1}{2}} - \underline{u}_i^n}{d\tau} = 0$. Anchors are implemented in the following way. Suppose we know that node z always belongs to cluster k, that is

$$U^n = \left(\underline{u}_1^n, \underline{u}_2^n, \dots, \underline{u}_{z-1}^n, \underline{e}_k, \underline{u}_{z+1}^n, \dots, \underline{u}_V^n\right)^T \text{ for all } n, \tag{A.33}$$

including $n=0$ and the intermediate steps $n+\frac{1}{2}$. To include this a priori information in the MBO scheme, it is sufficient to use at the level of (A.21) an evolution matrix

[12] For this presentation we use the notation for the unsigned case, but the arguments apply to $\bar{L}$ as well.

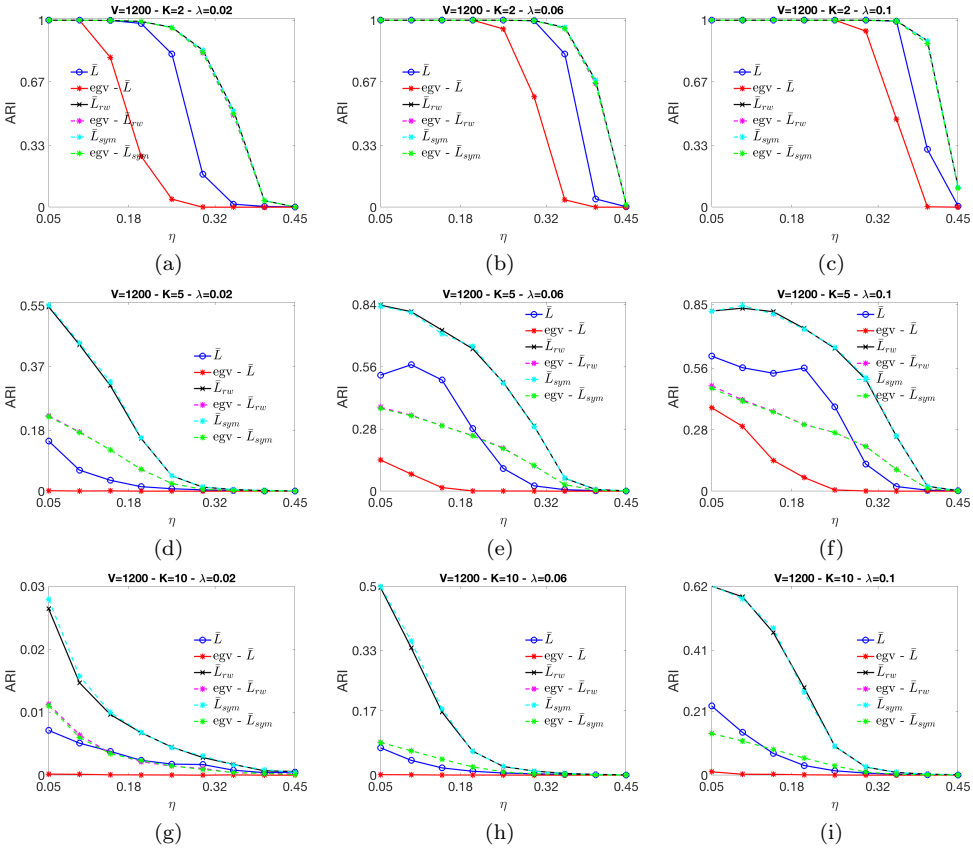


FIG. A.2. *These plots show the capability of MBO to recover the ground truth node-cluster association using the Laplacians* (3.6), (3.7) *and* (3.8) *at different regimes of noise, sparsity and number of clusters. As explained in the main text in Section 6, the algorithms are initialized by a partition into K sets based on an eigenvector associated with the smallest positive eigenvalue. For this reason, we have included the ARI value of the initial condition to see how MBO improves on its starting point. The first row shows results for $K=2$. As we can see, MBO gives better results than the initial condition for $\bar{L}$, while $\bar{L}_{rw}$ and $\bar{L}_{sym}$ do not improve over their eigenvectors but they give better results than $\bar{L}$. For $K=5$ and $K=10$, MBO always gives better results than the initial condition for all values of λ and η.*

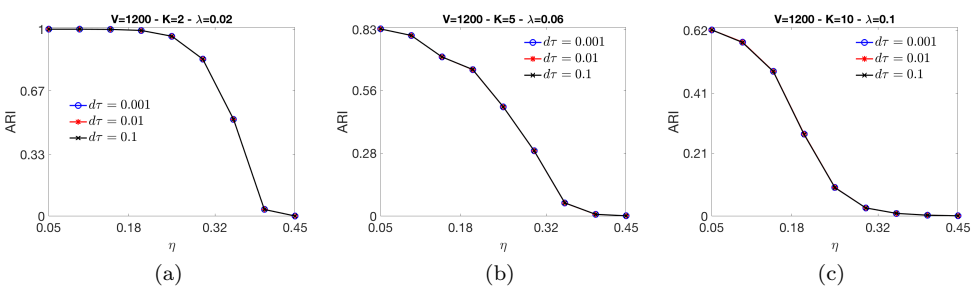


FIG. A.3. *The plots show different instances of MBO performed using different choices for the time discretization but with the same Laplacian L_{sym}. Here we can see that there is no difference in the choice of $d\tau$ in the whole range of values considered in Table 6.1.*

(to be used in place of the original B) which leaves the anchor node unaffected by the dynamics. The latter can be defined as

$$B^a_{ij} := \begin{cases} B_{ij} & \text{if } i \neq z \\ 0 & \text{if } i = z \text{ and } j \neq z \\ 1 & \text{if } i = j = z \end{cases} . \tag{A.34}$$

In general, to take into account the presence of a second anchor node it is sufficient to modify all the relevant entries of B^a_{ij} in the same fashion as in equation (A.34). This procedure can be repeated for all the anchor nodes. As shown in Figure A.4, anchors always improve upon the unconstrained signed Laplacian, as expected.

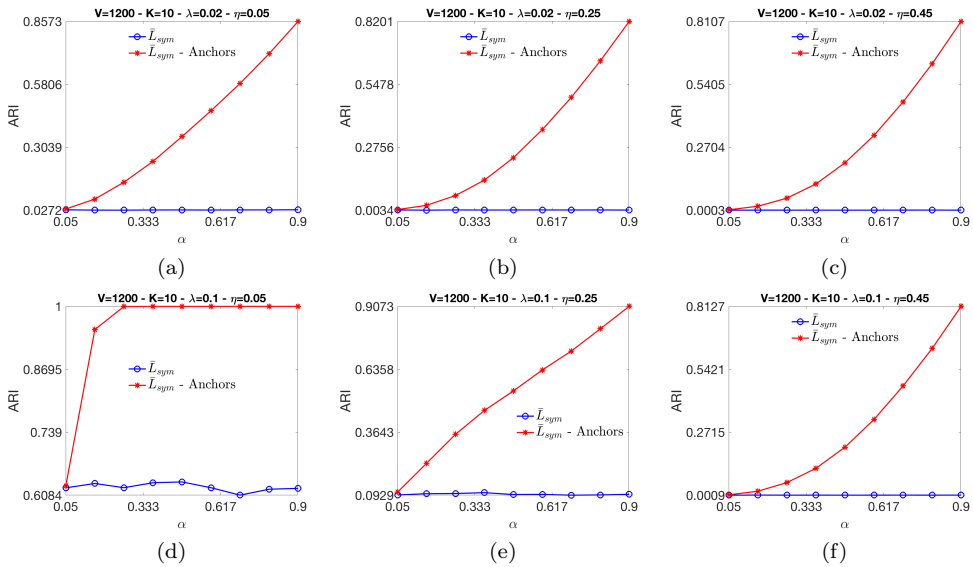


FIG. A.4. *The plots show the behaviour of the MBO scheme in the presence of hard constraints in the form of anchors as specified in Appendix A.6. The anchors are extracted at random from the ground truth for each experiment. For the anchor case, the relevant entries of the initial characteristic matrix are modified to be consistent with the constraints. As expected, for any value of α, adding fixed points in the dynamics improves the recovery of the regular MBO for most sparsity and noise levels.*

A.7. Image segmentation

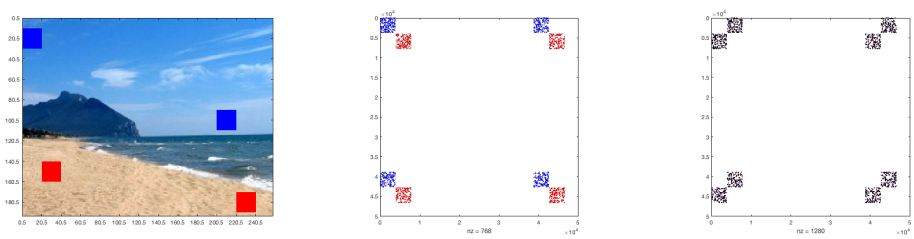
This section considers a proof-of-concept application of the Ginzburg–Landau functional under different types of constraints, as introduced in Section 4, in the context of image segmentation. As an example, we consider a picture of Cape Circeo in Sabaudia (LT, Italy) [79], Figure A.5 (a), in which it is possible to distinguish four main clusters which are the sand, sea, mountain, and sky.

The network associated to the picture has been obtained in the following way. First, each pixel has a one-to-one correspondence with a node in the resulting graph. Second, each pixel is linked in the network only to a subset of its nearest-neighbouring pixels (in particular, we have considered each pixel to be connected to all the others which are present within a radius of 5, as specified in the MATLAB Package *The graph analysis toolbox: Image processing on arbitrary graphs*, see [80, 81]). Third, the weights of the affinity matrix corresponding to the picture can be obtained by considering the weight of the link between two pixels as a function of their color difference. In fact, considering the colors red, green, and blue (RGB) as the basis of a 3-dimensional vector space, each

color can be described in terms of its RGB components in the same way as a vector can be characterized as the vector sum of the canonical basis elements weighted by some coefficients. In this way, we can naturally define the distance of two colors as a function of the Euclidean distance of their vectors in the RGB space. We used the distance function

$$d(i,j) := \begin{cases} e^{-b\|col(i)-col(j)\|_2} \in (0,1] & i \neq j \\ 0 & i = j \end{cases}, \tag{A.35}$$

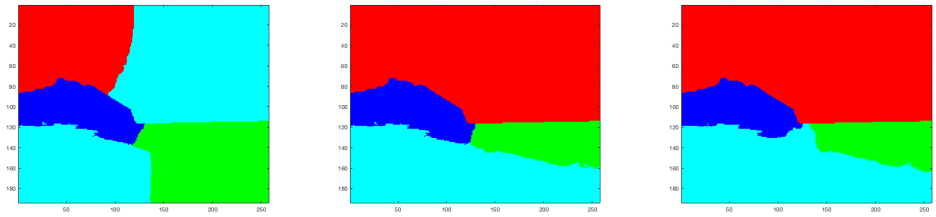
where b is an external parameter (which we fix to $b=14$) and $col(i)$ is the corresponding color vector of pixel i (note that the norm in (A.35) is not squared). The set of all the differences $\|col(i)-col(j)\|$ in a picture is normalized according to their *z-score* using the MATLAB command *normalize* [82]. The MBO schemes used for this image segmentation problem run using $K=4$ clusters as an input parameter. For constrained clustering, we have considered both must-links and cannot-links, as highlighted in Figure A.5. One iteration (multiple iterations do not change the result) of the MBO scheme yields the clustering highlighted in Figure A.6 (a). Afterwards, the resulting U has been used as the initial condition for two different constrained clustering optimizations, whose results are shown in Figure A.6 (b) and (c) for the must-links and cannot-links, respectively. It can be inferred that the cluster recovery works well, except for a small area in the middle of the picture; however, we remark that even in the original picture (that is Figure A.5 (a) without blue and red square) it is difficult to the naked eye to understand whether that patch (that is the bit of sea to the bottom right of the mountain) corresponds to sea or mountain, just by looking at its color.



(a) (b) (c)

FIG. A.5. *From a picture of Cape Circeo in Sabaudia (LT, Italy) (a) With the same color we highlight the patches in the image which we would like to belong to the same cluster as a result of the MBO scheme. In terms of must-links, this means that all the nodes corresponding to pixels in the blue squares should be pairwise positively connected, and so should those corresponding to the red squares. For example, the top left blue patch in (a) corresponds to the top left blue patch in (b); the other blue patch in (a) corresponds to the bottom right patch in (b); the remaining blue patches in (b) pertain to the cross interactions between the pixels in the blue patches in (a). Similar correspondences hold for the red patches. This can be seen in the affinity matrix in picture (b) where we have used the same color scheme as in (a). Instead, each blue/red node pair should be negatively connected since they belong to different clusters, as shown in figure (c). For the present case, we have weighted both must- and cannot-links with a prefactor of* 2. *The reason why squares in (b) are not fully filled with color as in (a) is because, to reduce computational time when calculating eigenvectors, we have removed half of the must-link edges at random and half of the cannot-link edges at random (white color corresponds to missing edges).*

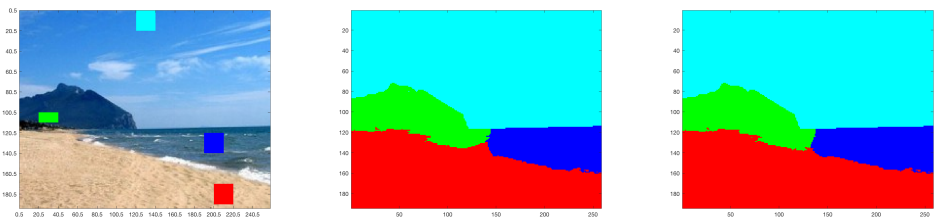
In Figure A.7, we have taken into account fidelity and avoidance terms as highlighted in plot (a). The same information has been used for both the fidelity and the avoidance parts. For the former, the rows of the matrix $\hat{U}$ associated to the nodes in the colored patches have entries equal to 1 at the corresponding cluster column (we have picked



(a) (b) (c)

FIG. A.6. *Segmentation results of the MBO scheme for the setting of (a) no constraints, (b) must-links, and (c) cannot-links, as chosen in Figure A.5.*

column 1 for red, 2 for blue, 3 for cyan, and 4 for green). All the remaining entries of $\hat{U}$ are equal to 0. For the latter, we have considered an energy increase every time nodes are attributed to clusters to which they should not belong. This results in a constraint of the form $\tilde{U} = \hat{U}(1_K - I_K)$. Starting from a random configuration of U, a single application of the MBO scheme using $\hat{U}$ yields the segmentation shown in Figure A.7 (b) and a separate run of the algorithm using $\tilde{U}$ yields to (c). We note that the node-cluster association recovery results agree with the expected boundaries present within the image, with a minor uncertainty in the central area (which is a feature present also in the constrained clustering case).



(a) (b) (c)

FIG. A.7. *In plot (a), we color those pixels whose cluster membership is considered to be known a priori, with each color corresponding to a different cluster. Contrary to the constrained clustering case, the cluster assignment of these pixels is chosen a priori since it results from inserting the known information directly at the level of the characteristic matrix.*

We have also considered the clusters highlighted in Figure A.7 (a) as anchors, and the results are identical to those shown in Figure A.7 (b).

REFERENCES

[1] E. Merkurjev, T. Kostic, and A.L. Bertozzi. An MBO scheme on graphs for classification and image processing. *SIAM Journal on Imaging Sciences*, 6(4):1903–1930, 2013. 1, A.2.2, A.2.3, A.2.3

[2] H. Huiyi, L. Thomas, A.P. Mason, and A.L. Bertozzi. A method based on total variation for network modularity optimization using the MBO scheme. *SIAM Journal on Applied Mathematics*, 73(6):2224–2246, 2013. 1

[3] A.L. Bertozzi and A. Flenner. Diffuse interface models on graphs for classification of high dimensional data. *SIAM Review*, 58(2):293–328, 2016. 1, A.2.2

[4] E. Merkurjev. Variational and PDE-based methods for big data analysis, classification and image processing using graphs. Technical report, Ph.D. Thesis, Department of Mathematics, University of California at Los Angeles, 2015. 1

[5] Y. van Gennip, N. Guillen, B. Osting, and A.L. Bertozzi. Mean curvature, threshold dynamics, and phase field theory on finite graphs. *Milan Journal of Mathematics*, 82(1):3–65, 2014. 1
[6] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999. 1
[7] S. Banerjee, K. Sarkar, S. Gokalp, A. Sen, and H. Davulcu. Partitioning signed bipartite graphs for classification of individuals and organizations. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, pages 196–204. Springer, 2012. 1
[8] Epinions data set. http://www.epinions.com. 1
[9] Slashdot data set. http://www.slashdot.com. 1
[10] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 641–650, New York, NY, USA, 2010. ACM. 1
[11] S. Kumar, F. Spezzano, V.S. Subrahmanian, and C. Faloutsos. Edge weight prediction in weighted signed networks. In *ICDM*, 2016. 1
[12] J. Bosch, P. Mercado, and M. Stoll. Node classification for signed networks using diffuse interface methods. *arXiv e-prints*, September 2018. 1
[13] J. Tang, C. Aggarwal, and H. Liu. Node classification in signed social networks. In *SDM*, 2016. 1
[14] KY. Chiang, N. Natarajan, A. Tewari, and I.S. Dhillon. Exploiting longer cycles for link prediction in signed networks. In *CIKM*, 2011. 1
[15] T. Derr, Y. Ma, and J. Tang. Signed Graph Convolutional Network. In *IEEE International Conference on Data Mining (ICDM)*, pages 929–934. IEEE, 2018. 1
[16] J. Kim, H. Park, JE. Lee, and U. Kang. Side: Representation learning in signed directed networks. In *WWW*, 2018. 1
[17] S. Wang, J. Tang, C. Aggarwal, Y. Chang, and H. Liu. *Signed Network Embedding in Social Media*. 2017. 1
[18] F. Chung, A. Tsiatas, and W. Xu. Dirichlet PageRank and Ranking Algorithms Based on Trust and Distrust. *Internet Mathematics*, 9(1):113–134, 2013. 1
[19] M. Shahriari and M. Jalili. Ranking nodes in signed social networks. *Social Network Analysis and Mining*, 4(1):172, Jan 2014. 1
[20] J. Tang, Y. Chang, C. Aggarwal, and H. Liu. A survey of signed network mining in social media. *ACM Comput. Surv.*, 49(3):42:1–42:37, 2016. 1
[21] S. Aghabozorgi, A.S. Shirkhorshidi, and T.Y. Wah. Time-series clustering–a decade review. *Information Systems*, 53:16–38, 2015. 1
[22] A. Fujita, P. Severino, K. Kojima, J.R. Sato, A.G. Patriota, and S. Miyano. Functional clustering of time series gene expression data by granger causality. *BMC systems biology*, 6(1):137, 2012. 1
[23] S.M. Focardi. Clustering economic and financial time series: Exploring the existence of stable correlation conditions. *The Intertek Group*, 2005. 1
[24] H. Ziegler, M. Jenny, T. Gruse, and D.A. Keim. Visual market sector analysis for financial time series data. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pages 83–90. IEEE, 2010. 1
[25] N.G. Pavlidis, V.P. Plagianakos, D.K. Tasoulis, and M.N. Vrahatis. Financial forecasting through unsupervised clustering and neural networks. *Operational Research*, 6(2):103–127, 2006. 1
[26] GG. Ha, JW. Lee, and A. Nobi. Threshold network of a financial market using the P-value of correlation coefficients. *Journal of the Korean Physical Society*, 66(12):1802–1808, Jun 2015. 1
[27] S.M. Smith, K.L. Miller, G. Salimi-Khorshidi, M. Webster, C.F. Beckmann, T.E. Nichols, J.D. Ramsey, and M.W. Woolrich. Network modelling methods for fmri. *NeuroImage*, 54(2):875 – 891, 2011. 1
[28] F. Harary. On the notion of balance of a signed graph. *Michigan Math. J.*, 2(2):143–146, 1953. 2
[29] D. Cartwright and F. Harary. Structural balance: a generalization of Heider's theory. *Psychological review*, 63(5):277, 1956. 2
[30] J.A. Davis. Clustering and structural balance in graphs. *Human Relations*, 20(2):181–187, 1967. 2
[31] P. Doreian and A. Mrvar. A partitioning approach to structural balance. *Social networks*, 18(2):149–168, 1996. 2
[32] P. Doreian and A. Mrvar. Partitioning signed social networks. *Social Networks*, 31(1):1–11, 2009. 2
[33] B.W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49(2):291–307, 1970. 2
[34] B. Yang, W.K. Cheung, and J. Liu. Community mining from signed social networks. *IEEE Trans Knowl Data Eng*, 19(10):1333–1348, 2007. 2
[35] P. Anchuri and M. Magdon-Ismail. Communities and balance in signed networks: A spectral

approach. In *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*, pages 235–242. IEEE, 2012. 2

[36] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Mach. Learn.*, 56(1-3):89–113, June 2004. 2

[37] E.D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica. Correlation clustering in general weighted graphs. *Theor. Comput. Sci.*, 361(2):172–187, September 2006. 2

[38] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E.Q. De Luca, and S. Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA*, pages 559–570, 2010. 2, 2, 3.1, 3.1, 3.1, 7

[39] J.P. Hou. Bounds for the least Laplacian eigenvalue of a signed graph. *Acta Mathematica Sinica*, 21(4):955?–960, 2005. 2, 2

[40] M. Cucuringu, P. Davies, A. Glielmo, and H. Tyagi. SPONGE: A generalized eigenproblem for clustering signed networks. *to appear in AISTATS 2019, available at* http://www.stats.ox.ac.uk/~cucuring/signedClustering.pdf, 2018. 2, 2

[41] KY. Chiang, J. Whang, and I.S. Dhillon. Scalable Clustering of Signed Networks using Balance Normalized Cut. In *ACM Conference on Information and Knowledge Management (CIKM)*, oct 2012. 2, 6.2, 6.1

[42] CJ. Hsieh, KY. Chiang, and I.S. Dhillon. Low-Rank Modeling of Signed Networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2012. 2

[43] K.Y. Chiang, C.J. Hsieh, N. Natarajan, I.S. Dhillon, and A. Tewari. Prediction and clustering in signed networks: A local to global perspective. *Journal of Machine Learning Research*, 15:1177–1213, 2014. 2

[44] P. Mercado, F. Tudisco, and M. Hein. Clustering signed networks with the geometric mean of Laplacians. In *NIPS*. 2016. 2

[45] M. Cucuringu. Synchronization over Z_2 and community detection in multiplex networks with constraints. *Journal of Complex Networks*, 3:469–506, 2015. 2

[46] J.H. Gallier. Notes on elementary spectral graph theory. Applications to graph clustering using normalized cuts. *CoRR*, abs/1311.2492, 2013. 2

[47] A. Fox, G. Sanders, and A. Knyazev. Investigation of spectral clustering for signed graph matrix representations. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pages 1–7. IEEE, 2018. 2

[48] L. Modica and S. Mortola. Un esempio di Γ^--convergenza. *Bollettino U.M.I.*, 5(14-B):285–299, 1977. A.2

[49] L. Modica. The gradient theory of phase transitions and the minimal interface criterion. *Arch. Rational Mech. Anal.*, 98(2):123–142, 1987. A.2

[50] B. Weizhu and L. Jian-guo. *Dynamics in Models of Coarsening, Coagulation, Condensation and Quantization*, volume 9. World Scientific, 2007. A.2.1

[51] S. Bartels. *Numerical methods for nonlinear partial differential equations*, volume 47. Springer, 2015. A.2.1

[52] A.L. Bertozzi and A. Flenner. Diffuse interface models on graphs for analysis of high dimensional data. *Multiscale Modeling and Simulation*, 10(3):1090–1118, 2012. A.2.2, A.2.3

[53] PA. Absil and K. Kurdyka. On the stable equilibrium points of gradient systems. *Systems & control letters*, 55(7):573–577, 2006. A.2.3

[54] X. Luo and A.L. Bertozzi. Convergence of the graph Allen–Cahn scheme. *Journal of Statistical Physics*, 167(3):934–958, May 2017. A.2.3

[55] B. Merriman, J.K. Bence, and S. Osher. Diffusion generated motion by mean curvature. *UCLA Department of Mathematics CAM report CAM 06–32*, 1992. A.2.3

[56] B. Merriman, J.K. Bence, and S. Osher. Diffusion generated motion by mean curvature. *AMS Selected Letters, Crystal Grower's Workshop*, pages 73–83, 1993. A.2.3

[57] E. Merkurjev, J. Sunu, and A.L. Bertozzi. Graph MBO method for multiclass segmentation of hyperspectral stand-off detection video. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 689–693. IEEE, 2014. A.2.3

[58] B. Keetch and Y. van Gennip. A Max-Cut approximation using a graph based MBO scheme. *to appear in DCDS-B, arXiv preprint arXiv:1711.02419*, 2017. 3.1, A.2.3

[59] C. Garcia-Cardona, A. Flenner, and A.G. Percus. Multiclass diffuse interface models for semi-supervised learning on graphs. In *Proceedings of the 2nd International Conference on Pattern Recognition Applications and Methods (ICPRAM 2013)*, pages 78–86, 2013. A.2.4

[60] E. Merkurjev, C. Garcia-Cardona, A.L. Bertozzi, A. Flenner, and A.G. Percus. Diffuse interface methods for multiclass segmentation of high-dimensional data. *Applied Mathematics Letters*, 33:29–34, 2014. A.2.4

[61] C. Garcia-Cardona, E. Merkurjev, A.L. Bertozzi, A. Flenner, and A.G. Percus. Multiclass data

segmentation using diffuse interface methods on graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36:1600–1613, 2014. 3.2, 4.2, 4.2, 5, A.2.4, A.2.4, A.2.4

[62] C. Garcia-Cardona, A. Flenner, and A. G. Percus. Multiclass semi-supervised learning on graphs using Ginzburg-Landau functional minimization. *Advances in Intelligent Systems and Computing*, 318:119–135, 2015. A.2.4

[63] G.H. Golub and C.F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012. 3.1

[64] J. Gallier. Spectral theory of unsigned and signed graphs. applications to graph clustering: a survey. *CoRR*, abs/1601.04692, 2016. 3.1

[65] W. H. Haemers and E. Spence. Enumeration of cospectral graphs. *European Journal of Combinatorics*, 25(2):199–211, 2004. 3.1

[66] M. Desai and V. Rao. A characterization of the smallest eigenvalue of a graph. *Journal of Graph Theory*, 18(2):181–194, 1994. 3.1

[67] Y. Chen and X. Ye. Projection onto a simplex. *arXiv preprint arXiv:1101.6081*, 2011. 3.2

[68] M. Cucuringu, I. Koutis, S. Chawla, G. Miller, and R. Peng. Simple and scalable constrained clustering: a generalized spectral method. In *Artificial Intelligence and Statistics*, pages 445–454, 2016. 4.1

[69] M. Cucuringu, Y. Lipman, and A. Singer. Sensor network localization by eigenvector synchronization over the Euclidean group. *ACM Trans. Sen. Netw.*, 8(3):19:1–19:42, August 2012. 10, A.6

[70] M. Cucuringu, A. Singer, and D. Cowburn. Eigenvector synchronization, graph rigidity and the molecule problem. *Information and Inference*, 1(1):21–67, 2012. A.6

[71] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007. 5, 6.1

[72] J.R. Lee, S.O. Gharan, and L. Trevisan. Multiway spectral partitioning and higher-order Cheeger inequalities. *Journal of the ACM (JACM)*, 61(6):37, 2014. 5

[73] L. Trevisan. Max cut and the smallest eigenvalue. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 263–272, 2009. 5

[74] A.J. Gates and YY. Ahn. The impact of random models on clustering similarity. *Journal of Machine Learning Research*, 18(87):1–28, 2017. 6

[75] A. Dasgupta, J.E. Hopcroft, and F. McSherry. Spectral analysis of random graphs with skewed degree distributions. pages 602–610. IEEE, 2004. 6.1

[76] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007. 6.2, 6.1

[77] K-means++. https://uk.mathworks.com/matlabcentral/fileexchange/28804-k-means. 6.2

[78] AL. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. 7

[79] Picture extracted from the newspaper LatinaToday http://www.latinatoday.it/green/sabaudia-spiagge-fondali-puliti-2019.html. A.7

[80] L. Grady and E. Schwartz. The graph analysis toolbox: Image processing on arbitrary graphs. Technical report, Boston University Center for Adaptive Systems and Department of Cognitive and Neural Systems, 2003. A.7

[81] L. Grady. *Space-Variant Computer Vision: A Graph-Theoretic Approach*. PhD thesis, Boston University, 2003. A.7

[82] MATLAB Documentation for *normalize* https://uk.mathworks.com/help/matlab/ref/double.normalize.html?s_tid=doc_ta. A.7

[83] Yahoo Finance, 2015. https://uk.finance.yahoo.com. 8.1

[84] Foreign exchange SDR valuation. https://www.imf.org/external/np/fin/data/rms_sdrv.aspx. 8.2

[85] U.S. Census Bureau, 2002. https://www.census.gov/data/tables/2000/demo/geographic-mobility/county-to-county-migration-flows.html 9, 8

[86] M.J. Perry. State-to-State Migration Flows: 1995 to 2000. *Census 2000 Special Reports*, 2003. 9

[87] M. Cucuringu, V. Blondel, and P. Van Dooren. Extracting spatial information from networks with low order eigenvectors. *Physical Review E*, 87, 2013. 9

[88] A. Joseph, B. Yu, et al. Impact of regularization on spectral clustering. *The Annals of Statistics*, 44(4):1765–1791, 2016. 10

[89] C.M. Le, E. Levina, and R. Vershynin. Sparse random graphs: regularization and concentration of the Laplacian. *arXiv preprint arXiv:1502.03049*, 2015. 10

[90] L. Laloux, P. Cizeau, M. Potters, and JP. Bouchaud. Random matrix theory and financial correlations. *International Journal of Theoretical and Applied Finance*, 3(03):391–397, 2000. 10

[91] G. Connor and R.A. Korajczyk. The Arbitrage Pricing Theory and Multifactor Models of Asset

Returns. *Handbooks in Operations Research and Management Science*, 9, 1993. 10

[92] W. Pentney and M. Meila. Spectral clustering of biological sequence data. In *AAAI*, volume 5, pages 845–850, 2005. 10

[93] D. Gleich. Hierarchical directed spectral graph partitioning. *Information Networks*, 2006. 10

[94] M. Meilă and W. Pentney. Clustering by weighted cuts in directed graphs. In *Proceedings of the 2007 SIAM international conference on data mining*, pages 135–144. SIAM, 2007. 10

[95] D.L. Sussman, M. Tang, D.E. Fishkind, and C.E. Priebe. A consistent adjacency spectral embedding for stochastic blockmodel graphs. *Journal of the American Statistical Association*, 107(499):1119–1128, 2012. 10

[96] M. Cucuringu, L. Huan, H. Sun, and Zanetti L. Spectral clustering of directed graphs. *In preparation*, 2019. 10

[97] A. Singer and H.T. Wu. Vector diffusion maps and the connection Laplacian. *Comm. Pure Appl. Math.*, page n/a, 2012. 10

[98] A. Bagirov, N. Karmitsa, and M.M. Mäkelä. *Introduction to Nonsmooth Optimization: theory, practice and software*. Springer, 2014.

11