Stochastic Block Models are a Discrete Surface Tension

Zachary M. Boyd, Mason A. Porter & Andrea L. Bertozzi

Journal of Nonlinear Science

ISSN 0938-8974
Volume 30
Number 5

J Nonlinear Sci (2020) 30:2429-2462
DOI 10.1007/s00332-019-09541-8

Springer

Journal of Nonlinear Science (2020) 30:2429–2462
https://doi.org/10.1007/s00332-019-09541-8

Journal of Nonlinear Science

Stochastic Block Models are a Discrete Surface Tension

Zachary M. Boyd[1,2] · Mason A. Porter[1] · Andrea L. Bertozzi[1]

Received: 6 June 2018 / Accepted: 19 March 2019 / Published online: 22 April 2019

Abstract
Networks, which represent agents and interactions between them, arise in myriad applications throughout the sciences, engineering, and even the humanities. To understand large-scale structure in a network, a common task is to cluster a network's nodes into sets called "communities," such that there are dense connections within communities but sparse connections between them. A popular and statistically principled method to perform such clustering is to use a family of generative models known as stochastic block models (SBMs). In this paper, we show that maximum-likelihood estimation in an SBM is a network analog of a well-known continuum surface-tension problem that arises from an application in metallurgy. To illustrate the utility of this relationship, we implement network analogs of three surface-tension algorithms, with which we successfully recover planted community structure in synthetic networks and which yield fascinating insights on empirical networks that we construct from hyperspectral videos.

Keywords Networks · Community structure · Data clustering · Stochastic block models (SBMs) · Merriman–Bence–Osher (MBO) scheme · Geometric partial differential equations

Mathematics Subject Classification 65K10 · 49M20 · 35Q56 · 62H30 · 91C20 · 91D30 · 94C15

Communicated by Paul Newton.

✉ Mason A. Porter
mason@math.ucla.edu

Zachary M. Boyd
zachboyd@email.unc.edu

Andrea L. Bertozzi
bertozzi@math.ucla.edu

[1] Department of Mathematics, UCLA, Los Angeles, CA, USA

[2] Department of Mathematics, University of North Carolina, Chapel Hill, USA

1 Introduction

The study of networks, in which nodes represent entities and edges encode interactions between entities (Newman 2018), can provide useful insights into a wide variety of complex systems in myriad fields, such as granular materials (Papadopoulos et al. 2018), disease spreading (Pastor-Satorras et al. 2015), criminology (Hegemann et al. 2011), and more. In the study of such applications, the analysis of large data sets—from diverse sources and applications—continues to grow ever more important.

The simplest type of network is a graph, and empirical networks often appear to exhibit a complicated mixture of regular and seemingly random features (Newman 2018). Additionally, it is increasingly important to study networks with more complicated features, such as time dependence (Holme 2015), multiplexity (Kivelä et al. 2014), annotations (Newman and Clauset 2016), and connections that go beyond a pairwise paradigm (Otter et al. 2017). One also has to worry about "features" such as missing information and false positives (Kim and Leskovec 2011). Nevertheless, it is convenient in the present paper to restrict our attention to undirected, unweighted graphs for simplicity.

To try to understand the large-scale structure of a network, it can be very insightful to coarse-grain it in various ways (Fortunato and Hric 2016; Peixoto 2015b; Porter et al. 2009; Rombach et al. 2017; Rossi and Ahmed 2015). The most popular type of clustering is the detection of assortative "communities," in which dense sets of nodes are connected sparsely to other dense sets of nodes (Fortunato and Hric 2016; Porter et al. 2009). A statistically principled approach is to treat community detection as a statistical inference problem using a model such as a stochastic block model (SBM) (Peixoto 2018). The detection of communities has given fascinating insights into a variety of applications, including brain networks (Betzel and Bassett 2017), social networks (Traud et al. 2012), granular networks (Bassett et al. 2015), protein-interaction networks (Ayati et al. 2015), political networks (Porter et al. 2005), and many others.

One of the most popular frameworks for detecting communities is to use an SBM, a generative model that can produce networks with community structure (Fortunato and Hric 2016; Peixoto 2018).[1] One uses an SBM for community detection by fitting an observed graph to a statistical model to attempt to infer the most probable community assignment for each node. SBMs can incorporate a variety of features, including degree heterogeneity (Karrer and Newman 2011), hierarchical structure (Peixoto 2014), and metadata (Newman and Clauset 2016). The benefits of an SBM approach include statistical defensibility, theoretical tractability, asymptotic consistency under certain conditions, definable transitions between solvable and unsolvable regimes, and theoretically optimal algorithms (Moore 2017; Peixoto 2018). As reviewed in Fortunato and Hric (2016), there are numerous other approaches for community detection, and statistical inference using SBMs is a method of choice among many people in the network-science community. A recent empirical study compared several

[1] Networks that are generated from an SBM can also have other types of block structures, depending on the choice of parameters; see Sect. 2.1 for details.

types of SBMs and other community-detection approaches on a variety of examples (Ghasemian et al. 2018).

Recently, Newman showed that one can interpret modularity maximization (Newman 2006; Newman and Girvan 2004), which is still among the most popular approaches for community detection, as a special case of an SBM (Newman 2016). In another paper (Hu et al. 2013), it was shown that one can also interpret modularity maximization in terms of graph cuts and total-variation (TV) minimization. The latter connection allows the application of methods from geometric partial differential equations (PDEs) and ℓ^1 minimization to community detection. This relationship also raises the possibility of formulating SBM maximum-likelihood estimation (MLE) in terms of TV.[2] In this paper, we develop such a formulation, and we also incorporate substantial new ingredients to do so. The principal one is the notion of surface tension as a generalization of total variation. Additionally, we need to examine an energy landscape that requires a novel splitting–merging heuristic to navigate it, whereas previous graph-TV methods have been able to rely on gradient descent to discover satisfactory optima. Moreover, the dynamical systems that arise in the present work differ from those in Bertozzi and Flenner (2012) and Hu et al. (2013), in that our modified Allen–Cahn (AC) and Merriman–Bence–Osher (MBO) schemes involve diffusion with all-to-all coupling in addition to coupling that arises from a potential well, balance terms, or thresholding.

The main result of the present work is the establishment of an equivalence between SBMs and surface-tension models from the literature on PDEs that model crystal growth. Crystal growth is an important aspect of certain annealing processes in metallurgy (Kinderlehrer et al. 2006; Mullins 1956). It is a consolidation process, wherein the many crystals in a metal grow and absorb each other to reduce the surface-tension energy that is associated with the interfaces between them. The various processes involved have been modeled from many perspectives, including molecular dynamics (Cleri et al. 1999), front tracking (Frost et al. 1990), vertex models (Weaire and Kermode 1983), and many others. (See Kinderlehrer et al. (2006) for a much more extensive set of references.) It has been observed experimentally that the interface between any two grains evolves according to motion by mean curvature (Smith 1952). Because mean-curvature flow is related to gradient descent (in the L^2 inner product) of the TV energy (Rudin et al. 1992), this leads naturally to formulations in terms of level sets (Osher and Sethian 1988), phase fields (Boettinger et al. 2002), and threshold dynamics (Merriman et al. 1992). Although the interfaces follow mean-curvature flow, each different interface can evolve at a different rate, as there are different surface-tension densities between each pair of crystals. In realistic cases, surface tensions are both inhomogeneous and anisotropic, and they require careful adaptation of standard mean-curvature-flow approaches (Esedoglu and Otto 2015; Jacobs 2017), especially for dealing with the topological challenges that arise at crystal junctions, which routinely form and disappear.

Recently, Jacobs showed how to apply techniques from models of crystal growth to graph-cut problems from semisupervised learning (Jacobs 2017). (See Jacobs et al.

[2] Another recent paper (Tudisco et al. 2018) used total variation for maximizing modularity, although the paper's exposition was not phrased in those terms.

(2018) for additional related work.) Several other recent papers, which do not directly involve surface tension, have used ideas from perimeter minimization and/or TV minimization for graph cuts and clustering in machine learning (Bertozzi and Flenner 2016). Three of those papers are concerned explicitly with ideas from network science (Boyd et al. 2018; Hu et al. 2013; Tudisco et al. 2018).

Each community in a network is analogous to a crystal, and the set of edges between nodes from a pair of communities is akin to the topological boundary between a pair of crystals. The surface-tension densities correspond to the differing affinities between each pair of communities. To demonstrate the relevance of this viewpoint, we develop and test discrete analogs of surface-tension numerical schemes on several real and synthetic networks, and we find that straightforward analogs of the continuum techniques successfully recover planted community structure in synthetic networks and reveal meaningful structure in the real networks. We also prove a theoretical result, in terms of Γ-convergence, that one can meaningfully approximate the SBM MLE problem by smoother energies. Finally, we introduce three algorithms, which are inspired by work on crystal growth, that we test on synthetic and real-world networks.

Our paper proceeds as follows. In Sect. 2, we present background information about stochastic block models, total variation, and surface tension. In Sect. 3, we state and prove our main result, which establishes an equivalence between discrete surface tension and maximum-likelihood estimation via an SBM. In Sect. 4, we discuss three numerical approaches for performing SBM MLE: mean-curvature flow, Γ-convergence, and threshold dynamics. We discuss our results on both synthetic and real-world networks in Sect. 5. In Sect. 6, we conclude and discuss our results. We give additional technical details in appendices.

2 Background

2.1 Stochastic Block Models (SBMs)

The most basic type of SBM has N nodes and an assignment $g{:}1, \ldots, N \longrightarrow 1, \ldots, \hat{n}$ that associates each node with one of $\hat{n}$ sets. It also has an associated $\hat{n} \times \hat{n}$ symmetric, nonnegative matrix ω that encodes the affinities between pairs of communities. One generates an undirected, unweighted graph as follows: For each pair of nodes, i and j, we place an edge between them with probability $\omega_{\alpha\beta}$, where α and β denote the community assignments of nodes i and j, respectively. Similar models have been studied and rediscovered many times (Condon and Karp 2001; Fienberg and Wasserman 1981; Fortunato and Hric 2016; Frank and Harary 1982; Holland et al. 1983; Peixoto 2018; Snijders and Nowicki 1997). In the present paper, we use the SBM from Newman (2016).

There is considerable flexibility in the choice of ω, which leads in turn to flexibility in the SBMs themselves (Fortunato and Hric 2016; Peixoto 2018). Three examples of ω, using $\hat{n} = 2$, will help illustrate the diversity of possible block structures.

1. If $\omega_{11} = \omega_{22} > \omega_{12}$, one obtains traditional assortative community structure, in which nodes have a larger probability to be adjacent to nodes in the same community, instead of ones in different communities.
2. If $\omega_{11} = \omega_{22} < \omega_{12}$, nodes tend to associate more with nodes that are in other communities. As $\omega_{12} \to 1$, the graph becomes increasingly bipartite.
3. If $\omega_{11} > \omega_{12} > \omega_{22}$, there is a core–periphery (CP) structure: Nodes from set 1 are connected densely to many nodes, but nodes from set 2 are connected sparsely to other nodes (Csermely et al. 2013; Rombach et al. 2017).

We illustrate these three examples in Fig. 1. To simplify our presentation, we refer to latent block structures as "community structure," regardless of the form of the matrix ω.

The above SBM is not realistic enough for many applications, largely because each node has the same expected degree (Karrer and Newman 2011). To address this issue, one can suppose that one knows a network's degree sequence $\{k_i\}$ (with k denoting the associated vector of degrees) and then define connection probabilities to take this information into account. The easiest approach [see the discussion in Karrer and Newman (2011)] is to model the adjacency-matrix elements A_{ij} (which is generated by the SBM) as Poisson-distributed with the parameter $\tilde{\omega}_{g_i g_j} := \omega_{g_i g_j} \frac{k_i k_j}{2m}$, where m is the number of edges in the associated network and $\omega_{\alpha\beta}$ is now allowed to take any value in $[0, \infty)$. This allows both multi-edges and self-edges. Such edges can have important effects, including in configuration models (Fosdick et al. 2018). Observe that the parameters ω, k, and g are necessary and sufficient to specify A as a random variable. In the present paper, we focus on the SBM that we described in this paragraph; it is known as a "degree-corrected" SBM (Karrer and Newman 2011).

Given an observed network, one can attempt to infer some sort of underlying community structure by statistical fitting methods. There are several ways to do this, including maximum-likelihood estimation (MLE), maximum a posteriori (MAP) esti-

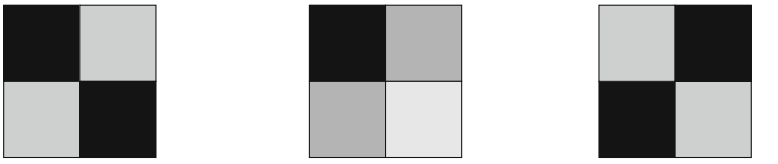

Fig. 1 Examples of different connectivity patterns that one can generate using stochastic block models. Each panel corresponds to a different type of structure. In each panel, the upper-left and lower-right squares represent the density of connections between nodes in the same set, and the upper-right and lower-left squares represent the density of connections between nodes in different sets. Darker squares represent more densely connected sets of nodes. In (assortative) community structure, nodes are densely connected to other nodes in the same community but sparsely connected to nodes in other communities. In core–periphery structure, core nodes (as illustrated by the dark square in the upper left) are densely connected to other core nodes and somewhat densely connected to peripheral nodes, but the latter predominantly have connections only to core nodes. In bipartite block structures, nodes in a set are more densely connected to nodes in other sets than to nodes in their own set. One can also model other structures, such as hierarchical and role-based structures, using SBMs. See Sect. 2.1 for additional discussion. [This figure is inspired by a figure from Jeub et al. (2015).]

mation, and maximum marginal likelihood (MML) estimation. In MLE, one chooses the parameters g and ω under which an observed network is most probable (without using a prior), MAP estimation yields the most probable parameter configuration under a Bayesian prior, and MML estimation yields the best community assignment for each node individually by integrating out all of the other variables (Moore 2017; Peixoto 2018). We use MLE, which is the simplest approach. In mathematical terms, the problem is to determine

$$\operatorname{argmax}_{g,\omega} P(A|g,\omega)\,, \tag{1}$$

where P is the probability density function. Because we determine the edges independently, P is given by

$$P(A|g,\omega) = \prod_{i\le j} P(A_{ij}|g,\omega) = \prod_{i\le j} P\left(A_{ij}\,\middle|\,w_{g_i g_j}\frac{k_i k_j}{2m}\right).$$

We use a Poisson distribution, so

$$P(A_{ij}|\lambda) = \begin{cases} \frac{\lambda^{A_{ij}}}{A_{ij}!}e^{-\lambda}\,, & i \neq j\,, \\ \frac{\lambda^{A_{ij}/2}}{(A_{ij}/2)!}e^{-\lambda}\,, & i = j\,, \end{cases}$$

where the need for cases arises from our convention that $A_{ii} = 2$ if a self-edge is present. To solve (1), one can equivalently maximize the logarithm of $P(A|g,\omega)$. Conveniently, this changes the multiplicative structure into additive structure and allows us to drop irrelevant constants. The resulting problem is

$$\operatorname{argmax}_{g,\omega} \sum_{i,j}\left[A_{ij}\log(\omega_{g_i g_j}) - \omega_{g_i g_j}\frac{k_i k_j}{2m}\right]. \tag{2}$$

If $\omega_{g_i g_j} = 0$, the quantity $A_{ij}\log(\omega_{g_i g_j})$ is understood to be 0 if $A_{ij} = 0$ and $-\infty$ otherwise.

Common optimization heuristics for solving (2) include greedy ascent (Karrer and Newman 2011), Kernighan–Lin (KL) node swapping (Karrer and Newman 2011; Kernighan and Lin 1970), and coordinate descent (Newman 2016). As far as we are aware, the theory of these approaches has not received much attention.

In light of the extreme non-convexity of the modularity objective function (Good et al. 2010) [which is known to be related to the planted-partition form of SBMs (Newman 2016)], we expect that it is necessary to use multiple random initializations for any local algorithm. Ideas from consensus clustering may also be helpful (Fortunato and Hric 2016).

Ways to elaborate SBMs include incorporating overlapping and hierarchical communities (Peixoto 2014, 2015b), generalizing to structures such as time-dependent and multilayer networks (Peixoto 2015a), and incorporating metadata (Newman and Clauset 2016). There are also Bayesian models and pseudo-likelihood-based methods (Amini et al. 2013; Peixoto 2018). We do not consider such embellishments in this

paper, although we conjecture that it is possible to generalize our approach to some (and perhaps all) of these settings.

2.2 Total Variation

Consider a smooth function $f : \Omega \subset \mathbb{R}^d \to \mathbb{R}$ for some d. The total variation (TV) of f is

$$|f|_{\mathrm{TV}} = \int_\Omega |\nabla f| \mathrm{d}x \,. \tag{3}$$

For $d = 1$, Eq. (3) describes the total amount of increase and decrease of the function f. If f is smooth except for jump discontinuities along a smooth hypersurface Γ, one can interpret the derivative of f in a generalized sense, yielding

$$|f|_{\mathrm{TV}} = \int_{\mathbb{R}^d - \Gamma} |\nabla f| \mathrm{d}x + \int_\Gamma |[f]| \, \mathrm{d}x \,,$$

where $[f]$ is the height of the jump across the discontinuity. The first integral uses a d-dimensional measure, and the second one uses a $(d-1)$-dimensional measure. In the particular case in which $d = 2$ and f is the characteristic function of some set S, we see that $|f|_{\mathrm{TV}}$ is the perimeter of S. Similarly, when $d = 3$, we obtain surface area.

Total variation is an important regularizer in machine learning. It is worth contrasting it with the Dirichlet energy $\int_\Omega |\nabla f|^2 \mathrm{d}x$, which has minimizers that satisfy $\Delta f = 0$, a condition that guarantees smoothness. However, minimizers of TV need not be smooth, as they can admit jump discontinuities. In image denoising, for instance, regularization using Dirichlet energy tends to blur edges to remove discontinuities, whereas a TV regularizer preserves the edges (Candès et al. 2006; Rudin et al. 1992).

Another use of TV energy is in relaxations, in which one can transform a non-convex problem involving piecewise-constant constraints into a convex problem with the same minimizers (Candès et al. 2006; Merkurjev et al. 2015). A common heuristic explanation for this phenomenon (see Fig. 2) uses the shape of the 1-norm unit ball. The simplest case is in two dimensions, where the 1-norm ball is diamond-shaped, and minimizing the 1-norm over certain domains (e.g., a line) gives a sparse solution, in the sense that most components of the solution vector are 0. In this case, minimizing the 1-norm, constrained to a line, is the same as minimizing the number of nonzero elements of the vector, subject to the same constraint.

In the context of TV minimization, we take the 1-norm of a function's gradient, rather than of the function itself. Therefore, instead of promoting sparsity of the function values, we promote sparse gradients, thereby incentivizing piecewise-constant minimizers for TV. Although our discussion is heuristic, note that the ideas therein can be treated rigorously (Candès et al. 2006).

Algorithmically, one can minimize TV using approaches such as phase-field models (Boettinger et al. 2002) or threshold dynamics (Merriman et al. 1992), both of which rely on the fact that the gradient descent (in the L^2 inner product) of TV is related to mean-curvature flow (Rudin et al. 1992). The alternating-directions method

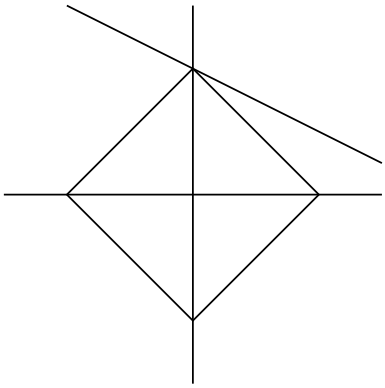

Fig. 2 Image of the 1-norm unit ball and a line in the plane. The point on the line with the smallest 1-norm is almost always on one of the axes

of multipliers (ADMM) (Goldstein and Osher 2009) and graph-cut methods, such as the one in Boykov et al. (2006), are also very effective at solving such problems.

Thus far, we have restricted our discussion of TV to a continuum setting. There exist graph analogs of the mathematical objects—gradients, measures, integrals, tangent spaces, divergences, and so on—that one uses to define TV in a continuum setting. For instance, for any function f on the nodes of a graph and for any edge between nodes i and j, the discrete derivative at i in the direction j is

$$\nabla f(i,j) = f(j) - f(j)\,.$$

Using the inner products

$$\langle f, g\rangle = \sum_{i=1}^{N} f_i g_i\,,$$

$$\langle \phi, \psi\rangle = \sum_{i,j} A_{ij}\phi_{ij}\psi_{ij}$$

on the spaces of functions on the nodes and edges, respectively, gives the divergence as the adjoint of the gradient:

$$(\operatorname{div}\phi)_i = \sum_j A_{ij}\phi_{ji}\,.$$

In a continuum setting, an alternative definition of TV is

$$|f|_{\mathrm{TV}} = \sup\langle \operatorname{div}\phi,\, f\rangle\,, \tag{4}$$

where the supremum is over an appropriate set of test functions. For a graph, (4) is equivalent to

$$|f|_{\mathrm{TV}} = \frac{1}{2}\sum_{i,j} A_{ij}|f(i) - f(j)|.$$

See Gilboa and Osher (2008) and van Gennip et al. (2014) for a detailed justification of these definitions.

Some methods for graph clustering (e.g., see von Luxborg (2007)) rely on the combinatorial graph Laplacian $L = \mathrm{diag}(k) - A$, which is a discrete analog of the continuum Laplacian Δ. The continuum Laplacian arises in solutions to constrained optimization problems that involve the Dirichlet energy, so it is reasonable to expect minimizers of energies that involve the combinatorial graph Laplacian to have analogous properties to minimizers of the Dirichlet energy. Indeed, minimizers that arise from graph spectral methods are usually smooth[3], instead of having sharp interfaces, so one needs to threshold them in some way. Such thresholding is a major source of difficulties for attempts to obtain theoretical guarantees about the nature of minimizers after thresholding. By contrast, methods that use graph TV can directly accommodate piecewise-constant solutions (Merkurjev et al. 2015), which do not require thresholding to give classification information. Several previous papers have exploited this property of TV on graphs (Bertozzi and Flenner 2012; Hu et al. 2013; Trillos et al. 2016; Zhu et al. 2017).

2.3 Surface Tension

Very roughly, one can consider a metal object as being composed of a large number of crystals that range in size from microscopic to macroscopic (Ashcroft and Mermin 1976). Each crystal is a highly ordered lattice; and there is a thin, disordered interface between crystals. The sizes and orientations of these crystals affect material properties, and one goal of annealing processes is to allow crystals to reorganize to produce a useful metal (see Fig. 3).

The potential energy of a crystal configuration is roughly

$$\sum_{\alpha,\beta} \sigma_{\alpha\beta}\mathrm{Area}(\Gamma_{\alpha\beta})\,, \tag{5}$$

where $\Gamma_{\alpha\beta}$ is the interface between crystals α and β, and $\sigma_{\alpha\beta}$ is the surface-tension energy density between these crystals. Each $\sigma_{\alpha\beta}$ is different, based on physical considerations that involve the exact offset between the orientations of the lattices in each pair of crystals. When prepared and heated appropriately, the individual crystals decrease (5) by growing to consume their neighboring crystals. See Esedoglu and Otto (2015), Jacobs (2017), and Kinderlehrer et al. (2006) for further background information.

[3] In this context, "smooth" entails that a value varies gradually along edges in a graph (i.e., adjacent nodes have similar values), although this notion of smoothness is not one with a strict mathematical meaning. Minimizers of Dirichlet-type energies on graphs normally have this type of smoothness property, whereas minimizers of graph-TV energies often have sharp interfaces between sets of nodes. (Such an interface occurs, for example, if a solution must have a value of either 0 or 1.)

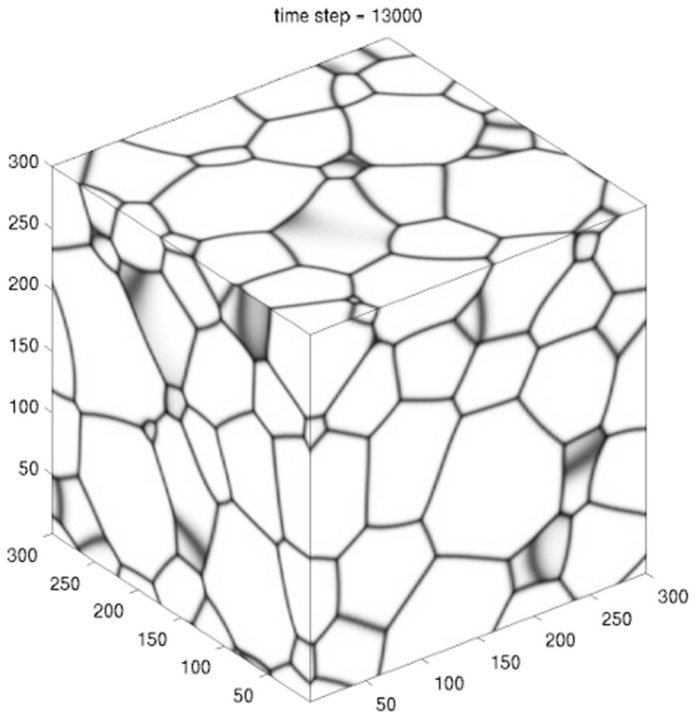

Fig. 3 An example arrangement of crystals. The interfaces between pairs of crystals grow into each other according to motion by mean curvature. [This image is from Cenna/Wikimedia Commons/Public Domain (Cenna 2012)]

In the study of SBMs, one can use TV to express (2), but we find a more natural formulation in terms of surface-tension energy (a related notion). Specifically, we exploit the appearance of surface area in (5) to cast it as a TV problem. Mathematically, we model the metal as a region of space that is partitioned into $\hat{n}$ regions, corresponding to the crystals in the metal. Let u^α and u^β, respectively, denote the characteristic functions of the regions α and β. Therefore,

$$\text{Area}_{\alpha\beta} = |u^\alpha|_{\text{TV}} + |u^\beta|_{\text{TV}} - |u^\alpha + u^\beta|_{\text{TV}} .$$

Each interface between two regions evolves according to mean-curvature flow. Consequently, the surface-tension flow is locally mean-curvature flow, except at the junction of three or more crystals (Esedoglu and Otto 2015; Jacobs 2017). Because of this connection, one can use some of the ideas [such as phase-field and threshold-dynamics methods (Esedoglu and Otto 2015)] from TV minimization to perform surface-tension minimization. When using threshold dynamics, it is possible to do theoretical analysis in the form of Lyapunov functionals, Γ-convergence, and descent conditions (Jacobs 2017).

3 An Equivalence Between SBM MLE and Discrete Surface Tension

We now present a mathematical result that connects SBM MLE and discrete surface tension.

Proposition 3.1 *Maximizing the likelihood of the parameters $g \in \{1, \hat{n}\}^N$ (i.e., node assignments) and ω (i.e., affinities) in the degree-corrected SBM (see Sect. 2.1) is equivalent to minimizing*

$$\sum_{\alpha,\beta} \left[W_{\alpha\beta} \mathrm{Cut}_{g,A}(\alpha, \beta) + e^{-W_{\alpha\beta}} \frac{\mathrm{vol}_{g,A}(\alpha)\, \mathrm{vol}_{g,A}(\beta)}{2m} \right], \tag{6}$$

where $\mathrm{Cut}_{g,A}(\alpha, \beta) = \sum_{\substack{g_i=\alpha \\ g_j=\beta}} A_{ij}$, *the volume term is* $\mathrm{vol}_{g,A}(\alpha) = \sum_{g_i=\alpha} k_i$, *and* $W_{\alpha\beta} = -\log \omega_{\alpha\beta}$ *(so* $W \in (-\infty, \infty]^{\hat{n}\times\hat{n}}$*).*

One immediately has the following well-posedness results. For a fixed W, the expression (6) has a solution, because the state space over which one is minimizing is of finite cardinality. Furthermore, for fixed g, one can find the optimal $W = W(g)$ in closed form by differentiating with respect to each component of W and setting the result to 0 (Karrer and Newman 2011). (We obtain a minimum of (6) because it is concave up.) Therefore, the full problem, in which we allow W to vary, also has a solution, because there are a finite number of candidate pairs $(g, W(g))$. Uniqueness is not guaranteed, because one can permute the community labels (and the corresponding entries in W) to obtain another minimizer. Another source of non-uniqueness is the possibility of symmetries in the underlying graph, which allows any optimizer to be converted to another optimizer by permuting the node labels. Continuous dependence of the minimizer on A is automatic, because the set of possible values for A is discrete.

The analogy with continuum surface tension is as follows. Graph cuts are analogous to surface area. Given a domain in $\mathbb{R}^3$, one can superimpose a fine grid on space and count the number of edges that cross the boundary to estimate its surface area. In the limit of an infinitely fine grid, this estimate converges to the surface area under appropriate conditions (Boykov and Kolmogorov 2003). Similarly, graph volumes are analogous to continuum volumes.[4] The quantities $W_{\alpha\beta}$ play the role of surface tensions $\sigma_{\alpha\beta}$, so the first set of terms is analogous to (5). One can view the second set of terms as a soft volume constraint. A constraint is "soft" if violating it adds a finite penalty to an objective function, so minimizers usually approximately satisfy the constraint. Volume-constrained versions of (5) have received a great deal of attention (Jacobs et al. 2018; Kinderlehrer et al. 2006).[5]

[4] For example, in a uniform square grid (ignoring boundaries), the sum of degrees for a set of nodes is proportional to the number of nodes, which in turn is roughly proportional to the area encompassed by filling in the squares that are associated with the selected grid nodes.

[5] As far as we are aware, our formulation of SBM MLE in terms of graph cuts and volumes is novel, although similar formulas have appeared previously in the literature (see, e.g., Peixoto 2018).

We now prove Proposition 3.1.

Proof (Proposition 3.1) In Newman (2016), it was shown that maximizing the log-likelihood of the parameters g and ω for a particular version of the degree-corrected SBM amounts to maximizing (2). Let $\Pi(G, \hat{n})$ be the set of partitions of the nodes of a graph G (associated with an adjacency matrix A) into at most $\hat{n}$ sets. Substituting $W_{\alpha\beta} = -\log \omega_{\alpha\beta}$ into (2) gives

$$\underset{\substack{W_{\alpha\beta}\in(-\infty,\infty]\\ g\in\Pi(G,\hat{n})}}{\operatorname{argmin}} \sum_{i,j}\left[A_{ij}W_{g_ig_j} + \frac{k_ik_j}{2m}e^{-W_{g_ig_j}}\right].$$

Rearranging the summations gives

$$\underset{\substack{W_{\alpha\beta}\in(-\infty,\infty]\\ g\in\Pi(G,\hat{n})}}{\operatorname{argmin}} \left[\sum_{\alpha,\beta}\sum_{\substack{g_i=\alpha\\ g_j=\beta}} A_{ij}W_{\alpha\beta} + \sum_{\alpha,\beta}\sum_{\substack{g_i=\alpha\\ g_j=\beta}} \frac{k_ik_j}{2m}e^{-W_{\alpha\beta}}\right],$$

where the inner sums are over all nodes i and j such that $g_i = \alpha$ and $g_j = \beta$. Rearranging again gives

$$\underset{\substack{W_{\alpha\beta}\in(-\infty,\infty]\\ g\in\Pi(G,\hat{n})}}{\operatorname{argmin}} \left[\sum_{\alpha,\beta} W_{\alpha\beta}\sum_{\substack{g_i=\alpha\\ g_j=\beta}} A_{ij} + \sum e^{-W_{\alpha\beta}}\sum_{\substack{g_i=\alpha\\ g_j=\beta}} \frac{k_ik_j}{2m}\right].$$

Using the definition of $\mathrm{Cut}_{g,A}$ in the first set of terms and summing over the j index independently in the second set of terms gives

$$\underset{\substack{W_{\alpha\beta}\in(-\infty,\infty]\\ g\in\Pi(G,\hat{n})}}{\operatorname{argmin}} \left[\sum_{\alpha,\beta} W_{\alpha\beta}\mathrm{Cut}_{g,A}(\alpha,\beta) + \sum_{\alpha,\beta} e^{-W_{\alpha\beta}}\sum_{g_i=\alpha}\frac{k_i}{2m}\mathrm{vol}_{g,A}(\beta)\right].$$

Finally, we sum over the i index in the second set of terms to obtain

$$\underset{\substack{W_{\alpha\beta}\in(-\infty,\infty]\\ g\in\Pi(G,\hat{n})}}{\operatorname{argmin}} \sum_{\alpha,\beta}\left[W_{\alpha\beta}\mathrm{Cut}_{g,A}(\alpha,\beta) + e^{-W_{\alpha\beta}}\frac{\mathrm{vol}_{g,A}(\alpha)\,\mathrm{vol}_{g,A}(\beta)}{2m}\right]. \tag{7}$$

One difference between the graph setting (6) and the continuum setting (5) is that in (6), one performs optimization over the $W_{\alpha\beta}$, whereas in (5) (i.e., in a continuum), one ordinarily treats the surface-tension densities as fixed by the choice of material that one is modeling. Another difference is that the surface-tension coefficients in the graph setting can be any element of $(-\infty, \infty]$,

subject only to the symmetry condition $W_{\alpha\beta} = W_{\beta\alpha}$ (see Sect. 2). By contrast, for a continuum, one needs further restrictions to ensure well-posedness. Esedoglu and Otto (2015) proved the following sufficient conditions for well-posedness:

(1) $\sigma_{\alpha\beta} \geq 0$,
(2) $\sigma_{\alpha,\alpha} = 0$,
(3) $\sigma_{\alpha\gamma} + \sigma_{\gamma\beta} \geq \sigma_{\alpha\beta}$.

In a graph setting, one can use a straightforward change of variables, $W_{\alpha\beta} \to W_{\alpha\beta} - \frac{1}{2}W_{\alpha\alpha} - W_{\beta\beta}$, to make W satisfy requirement (2).[6] In general, however, at least one of requirements (1) and (3) is not necessarily satisfied for a graph. Requirement (1) is false whenever some component of W is negative; this occurs exactly when ω has a component that is larger than 1. In the continuum, requirement (3) has the interpretation of preventing "wetting," where one phase can spontaneously appear between two others. In the graph case, such a restriction is unnecessary, because the number of points is fixed and finite, with no possibility of inserting points of another phase between two nodes.

The analogy of (6) with continuum surface tension is simplest for the case of assortative communities, although it is also relevant for other types of block structure. For disassortative blocks, rather than an energy cost from surface area, particles in one phase can achieve a lower energy by interacting with particles in a different phase. This leads to solutions that maximize surface area, and the evolutions that we will consider in Sect. 4 then involve backward diffusion. In the continuum case, this is ill-posed; however, a graph does not include arbitrarily small length scales, so ill-posedness does not cause a problem. Backward diffusion on graphs also appeared recently in Welk et al. (2018) in the context of image processing, and it would be interesting to see if their techniques would be insightful in our context. For core–periphery structure, one phase has an energy penalty from interacting with itself but lowers its energy by interacting with the other phase. The other phase, however, prefers to interact with itself. As far as we are aware, such structures have not been studied previously in the literature on surface-tension models of crystal growth. For more complicated block structures, it is concomitantly more complicated to interpret the analogy with continuum surface tension. In a sense, one should view the SBM in (6) as a generalization of discrete surface tension, rather than as an analog. In the present paper, we emphasize applications to assortative community structure.

4 Mean-Curvature Flow (MCF), Γ-Convergence, and Threshold Dynamics

We now outline three algorithmic approaches that illustrate how one can use tools from surface-tension theory to solve SBM MLE problems. Our three algorithms

[6] See Appendix A for the change of variables, which causes the sum in (6) to instead be over all $\alpha \neq \beta$, so that there are no "internal" surface tensions.

are graph versions of mean-curvature flow (MCF), Allen–Cahn (AC) evolution, and Merriman–Bence–Osher (MBO) dynamics. In Sect. 5, we will conduct several numerical experiments to demonstrate that these algorithms can effectively solve (2). We expect the performance of these algorithms to be good relative to other algorithms for SBM MLE, although a full evaluation of this claim is beyond the scope of our paper. We have posted our code at https://github.com/zboyd2/SBM-surface-tension, and we encourage readers to experiment with it.

In the next three subsections, we describe how we infer g when ω is fixed and finite. We then describe how to jointly infer ω and g.

4.1 Mean-Curvature Flow

Surface-tension dynamics are governed by mean-curvature flow except at junctions. Intuitively, each point on a surface moves in the direction normal to the surface at a speed given by the mean curvature at that point. In the two-phase case, such dynamics have been well studied, and there exist notions of viscosity solutions and regularity theory (Mantegazza 2011). In the multi-phase case, the situation is much more complicated, especially because of the topological changes that can occur and the issue of defining the behavior at the junction of three or more phases. In two-phase surface-tension dynamics, it was shown in Boykov et al. (2006) that one can approximate the flow by solving a discrete-time minimizing-movements problem. Let C_n be one of the two regions at time $n\,dt$, where dt denotes the time step. To update C_n, one calculates

$$C_{n+1} = \text{argmin}_C \left[\text{SurfaceArea}(C) + \frac{1}{dt} \int_{C_n \Delta C} \hat{\rho}(p, C_n) \mathrm{d}p \right], \tag{8}$$

where

$$\hat{\rho}(p, C_n) = \inf_{x \in \partial C_n} \|x - p\|,$$

the operation Δ denotes the symmetric difference, C is the generic notation for a region, and ∂ is the topological boundary operator. The idea behind this approach is, at each time step, to shrink the region as much as possible without straying too far from the region location at the previous time step.

In the setting of graphs, a similar approach was developed in van Gennip et al. (2014), where the mean-curvature flow was given by

$$C_{n+1} = \text{argmin}_C \left[\text{Cut}_{g,A}(C, C^c) + \frac{1}{dt} \sum_{i \in C_n \Delta C} \rho(i, \partial(C_n)) \right], \tag{9}$$

the operation Δ is again the symmetric difference, and $\rho(i, \partial(C_n))$ is the shortest-path distance from node i to the boundary $\partial(C_n)$ of C_n. In this context, the boundary of a set of nodes is the set of nodes in C_n with at least one neighbor in

C_n^c along with the nodes in C_n^c that have at least one neighbor in C_n. We use the term *boundary node* for any node that lies on the boundary. In the limit of a small time step dt, (9) may still evolve, as opposed to the MBO scheme (which we use later), which becomes "stuck" when the time step is too small. Such evolution can still occur, because the penalty (associated with moving any node in $\partial(C_n)$) induced by the second set of terms in (9) is 0, regardless of the value of dt. Conveniently, this implies for sufficiently small dt that the only acceptable moves at each time step are ones that are allowed to change only the boundary nodes themselves. This makes it possible to drastically reduce the search space when solving (9).

Because careful studies in the spirit of van Gennip et al. (2014) are not yet available for multi-way graph partitioning, we resort to a heuristic approach based on what is known for bipartitioning. Specifically, we are motivated by the situation in which time steps are sufficiently small that only boundary nodes can change their community assignment. Ideally, we wish to compute an optimal reassignment of all boundary nodes jointly to minimize (6). To save computation time and facilitate implementation, we instead decouple the computations in the following manner. During a single time step, for each boundary node, we determine an optimal community assignment of that node, assuming that all other nodes keep their assignment from the beginning of the time step. After this (but before the end of the time step), we assign each boundary node to its community, as determined previously in the time step. Because most nodes are boundary nodes[7] in our SBM-generated graphs, we find it both more efficient and easier to consider reassigning all nodes in each time step, rather than maintaining and referencing a separate data structure to track the boundary. For other networks and initialization techniques, such as in networks that arise from nearest-neighbor graphs with initialization from spectral clustering, it may be more efficient to loop over only the boundary nodes. (This idea aligns particularly with the spirit of mean-curvature flow.[8]) In Algorithm 1, we give pseudocode for this graph MCF procedure.

[7] Recall that a node is a boundary node if it shares an edge with a node that lies outside of its own community, so most reasonable partitions of many real graphs have many boundary nodes. Additionally, because we initialize g with nodes assigned to communities uniformly at random, most nodes are initially boundary nodes for most graphs.

[8] There are similarities between gradient-descent methods and greedy approaches, because both attempt to make locally optimal moves. Our decisions to move nodes such that each move is conditionally independent and to not track boundary nodes are also reminiscent of greedy approaches. Ultimately, which nodes are considered at each time step is an implementation detail, because only boundary nodes change assignment (at least in the assortative case that we emphasize in this paper). For larger graphs with more communities and fewer boundary nodes, it may be possible to increase efficiency by considering moves of boundary nodes only to neighboring communities, rather than employing our present approach of considering moves of nodes to any community. At the scale of our examples (up to millions of edges), this implementation choice is not necessary.

Algorithm 1. Modified graph mean-curvature flow (MCF) for SBM MLE (2).

Input $A, W, \hat{n}$.
Initialize g so that nodes are assigned to communities uniformly at random.
Let $\mathrm{eW} = e^{-W}$ denote the entry-wise exponential of W.
while not converged **do**
  Let $U_{i\alpha} = \delta_{g_i \alpha}$ for each i and α, where δ is the Kronecker delta.
  Let $X = AU$. // Counts the number of neighbors that each node has in each community
  Let $\mathrm{vol}_{g,A} = (k^T U)$.
  for $a' = 1, \ldots, \hat{n}$ **do**
    Let $I_{a'}$ be the set of nodes that are currently assigned to community a'.
    for $a = 1, \ldots, \hat{n}$ **do**
      Let I be the indices $1, \ldots, \hat{n}$ aside from a and a'.
      Let $\mathrm{Delta}(I_{a'}, a)$ be given by the following formula:

$$\begin{aligned}\mathrm{Delta}(I_{a'}, a) = {} & 2X(I_{a'}, I)W(I, a) \\ & - 2X(I_{a'}, I)W(I, a') \\ & + 2X(I_{a'}, a)W(a, a) \\ & - 2X(I_{a'}, a)W(a, a') + 2X(I_{a'}, a')W(a, a') \\ & - 2X(I_{a'}, a')W(a', a') \\ & + \frac{1}{2m}\Big(2k(I_{a'})\mathrm{vol}_{g,A}(\mathrm{eW}(:, a) - \mathrm{eW}(:, a')) \\ & + k(I_{a'})^2\left(\mathrm{eW}(a, a) + \mathrm{eW}(a', a') - 2\mathrm{eW}(a, a')\right)\Big).\end{aligned}$$

    end for
  end for
  for $i = 1, \ldots, N$ **do**
    $g_i = \mathrm{argmin}(\mathrm{Delta}(i, :))$. // Choose uniformly at random in case of a tie.
  end for
end while
Output g.

4.2 Allen–Cahn (AC) Evolution

Another approach for studying MCF is approximation by a Ginzburg–Landau (GL) functional. This approach is popular due to its simple implementation and the existence of unconditionally stable numerical methods (Bertozzi and Flenner 2012).

In the two-phase case, the GL functional is

$$\int_\Omega \left[\epsilon\,|\nabla u|^2 + \frac{1}{2\epsilon}u^2(1-u)^2\right]\mathrm{d}x\,, \tag{10}$$

where $u : \Omega \subset \mathbb{R}^N \to \mathbb{R}$ is a smooth function and ϵ is a small parameter.

The L^2 gradient descent of the GL functional is

$$u_t = \epsilon\Delta u - \frac{1}{2\epsilon}\frac{\mathrm{d}}{\mathrm{d}u}\left[u^2(1-u)^2\right],$$

which is the Allen–Cahn (AC) equation. The minimizers of the GL energy are predominantly piecewise constant, with $O(\epsilon)$-width transition layers between the constant

regions. One can show that the GL energy Γ-converges to the TV energy as $\epsilon \to 0$, assuming that $\int_\Omega u \, dx = \text{const}$ (Modica 1987). Consequently, if u_ϵ is a minimizer of the constrained GL energy with parameter ϵ and the minimizers converge in L^1 as $\epsilon \to 0$, then the accumulation point is a minimizer of the TV energy.

In the setting of graphs, the first use of AC schemes for TV minimization was in Bertozzi and Flenner (2012). One can invoke the combinatorial graph Laplacian $L = \text{diag}(k) - A$ to obtain a *graph GL functional*

$$U^T L U + \frac{1}{\epsilon} U^2 (1 - U^2) \,, \tag{11}$$

where U is a function on the graph nodes (so it is an N-element vector) and ϵ is again a positive number. Expression (11) Γ-converges to graph TV (van Gennip and Bertozzi 2012).

In the multi-phase case, we represent the community assignments g in terms of an $N \times \hat{n}$ matrix whose i, α entry is $\delta_{g_i \alpha}$, where δ is the Kronecker delta. Instead of a double-well potential, we use a multi-well potential on $\mathbb{R}^{N \times \hat{n}}$ whose value is minimized by arguments with exactly one nonzero entry in each row. For example, Garcia-Cardona et al. (2014) proposed the following potential:

$$T(U) = \sum_{i=1}^{N} \left(\prod_{\alpha=1}^{\hat{n}} \frac{1}{4} \| U_i - e_k \|_{\ell^1}^2 \right) ,$$

where U_i is the ith row of the $N \times \hat{n}$ matrix U and e_k is an $\hat{n}$-element vector that is equal to 0 except for a 1 in the kth entry.

For the particular case of surface-tension dynamics, we proceed as follows. Additionally, we assume in this subsection and the next that we have already eliminated the diagonal of W (see Appendix A).

Given community assignments (and hence a partition of a network), if U is the corresponding $N \times \hat{n}$ matrix, one can show that $W .* (U^T L U) = -W .* (U^T A U)$, where $.*$ is the entry-wise product.[9] Therefore, an appropriate GL functional for our problem is

$$\sum_{\alpha,\beta} \left[-W_{\alpha\beta} U_\alpha^T L U_\beta + \frac{\text{vol}_{g,A}(\alpha) \; e_{\alpha\beta}^{-W} \; \text{vol}_{g,A}(\beta)}{2m} \right] + \sum_\alpha W_{\alpha\alpha} \text{vol}_{g,A}(\alpha) + \frac{1}{2\epsilon} T(U) \,. \tag{12}$$

[9] As a proof, we note that $[W .* (U^T \text{diag}(k) U)]_{\alpha\beta} = \sum_i W_{\alpha\beta} U_{i\alpha} k_i U_{i\beta} = 0$, because $U_{i\alpha} U_{i\beta} = 0$ if $\alpha \neq \beta$ and $W_{\alpha\alpha} = 0$.

Because $k^T U$ gives the vector of volumes, one can rewrite (12) as

$$\sum_{\alpha,\beta}\left[-W_{\alpha\beta}U_\alpha^T L U_\beta + \frac{k^T U_\alpha e_{\alpha\beta}^{-W} U_\beta^T k}{2m}\right] + \sum_\alpha W_{\alpha\alpha}\mathrm{vol}_{g,A}(\alpha) + \frac{1}{2\epsilon}T(U)\,, \tag{13}$$

where e^{-W} is the entry-wise exponential.

As in a continuum setting, one can prove Γ-convergence.

Theorem 4.1 *Let $W \in \mathbb{R}^{\hat{n}\times\hat{n}}$. The functionals in (13) Γ-converge (as functions on $\mathbb{R}^{N\times\hat{n}}$) to (6) as $\epsilon \to 0$.*

See Appendix B for a proof. As far as we are aware, this is the first Γ-convergence result for a multi-phase graph energy on arbitrary graphs. However, see Osting and Reeb (2017), Trillos and Slepčev (2018), and Trillos et al. (2016) for Γ-convergence applied to consistency of multi-phase geometric graph energies.

The resulting AC equation is

$$U_t = LUW - \frac{1}{2m}kk^T U e^{-W} - k.*\,\mathrm{diag}(W) - \frac{1}{\epsilon}T'(U)\,. \tag{14}$$

See Appendix C for further details on the numerical solution of (14).

4.3 MBO Iteration

Algorithm 2. A two-phase, continuum MBO scheme.

Input the initial domain.
Initialize u as the characteristic function of the initial domain.
for $i = 1, \ldots$ **do**
  $u^{i+1/2}$ is the solution at time dt of $u_t = \Delta u$ with initial condition u^i.
  $u^{i+1} = \lfloor u^{i+1/2} + 0.5 \rfloor$, where $\lfloor\cdot\rfloor$ is the floor function.
end for
Output the set of points for which $u = 1$.

In Algorithm 2, one needs a stopping condition for the 'for' loop. The simplest choice is to execute its contents for a fixed number of iterations (e.g., 500). Note that we do not use this algorithm in our numerical experiments

In Merriman et al. (1992), Merriman, Bence, and Osher showed that continuum MCF is well-approximated by the simple iteration in Algorithm 2. In a rectangular domain, the iteration is extremely efficient, as one can use a fast Fourier transform when solving the heat equation. Esedoglu and Otto (2015) developed a generalized version of the MBO scheme (see Algorithm 3) for computing the evolution of multi-phase systems that are modeled by (5).

Algorithm 3. A multiphase, continuum MBO scheme.

Input the initial state of the domain.
Initialize $u_1, \ldots, u_{\hat{n}}$ as the characteristic functions of the initial domains.
for $i = 1, \ldots$ **do**
  for $\alpha = 1, \ldots, \hat{n}$ **do**
    $u_\alpha^{i+1/2}$ is the solution at time dt to $u_{\alpha,t} = \Delta u_\alpha$ with initial condition u_α^i.
  end for
  for each point x **do**
    $\hat{\alpha} = \operatorname{argmin}_\alpha \sum_\beta \sigma_{\alpha\beta} u_\beta(x)$. // Choose uniformly at random in case of a tie
    $u_{\hat{\alpha}}(x) = 1$ and $u_\beta(x) = 0$ if $\beta \neq \hat{\alpha}$.
  end for
end for
Output u.

We stop the parent 'for' loop in Algorithm 3 either after $i = 100$ (which almost never occurs in practice) or when the community-assignment vector after step i changes sufficiently little from that after step $i - 1$. (Our tolerance is that the norm of the difference between these consecutive community-assignment vectors is less than 0.001 multiplied by the norm of the ith community-assignment vector.)

One can apply the MBO idea to community detection in networks by replacing the continuum Laplacian with the (negative) combinatorial graph Laplacian, replacing σ with W, changing u to U, and adding appropriate forcing terms for the gradient descent of the volume-balance terms. See Appendix C for additional implementation details.

4.4 Learning *W*

The MCF, AC, and MBO algorithms are able to produce a good partition of a network, given W, but they do not include a way to find W. A simple way to address this issue is to use an expectation-maximization (EM) algorithm, in which one alternates between solving for g with fixed W (using MCF, AC, or MBO) and solving for W with fixed g. Given g, one can find a closed-form expression for the optimal W by differentiating (6) with respect to any component of W and setting the result to 0 (Karrer and Newman 2011).

One must be careful, however, because the optimal $W_{\alpha\beta}$ is infinite when $\mathrm{Cut}_{g,A}(\alpha, \beta) = 0$. This is problematic, because once one of the entries in W is infinite, it prevents g in subsequent iterations from taking any nonzero value of $\mathrm{Cut}_{g,A}(\alpha, \beta)$; this gives bad results in our test examples. (See Sect. 5 for a discussion of these examples.) We address this issue by modifying the EM algorithm to reset all infinite values of W to $1.1 \times W_{\max}$, where $W_{\max}$ is the largest non-infinite element of W and 1.1 is a (hand-tuned) parameter that allows moderate growth in W.

We also need to address another practical issue for an EM approach to work. Specifically, the algorithm that we have described thus far in this section often finds bad local minima in which communities are merged erroneously or a single community is split inappropriately.[10]

To overcome this issue, we implement a wrapper function (see Algorithm 4) that checks each community that is returned by MCF, AC, or MBO for further possible

[10] In other words, one can improve these local minima either by merging or by splitting existing communities. This is a special (and convenient, in this case) form of non-convexity.

splitting or merging with other communities. Whenever we call MCF, AC, or MBO on a subgraph, we use the values of k and m for the whole graph rather than those for a subgraph. A similar idea was used in Hu et al. (2013) and Newman (2006) for their recursive partitioning procedures.

There is also a danger of overfitting by setting $\hat{n} = N$, which gives a likelihood of 1 in (2). The proper selection of $\hat{n}$ is a complicated problem, both algorithmically and theoretically (Newman and Reinert 2016; Riolo et al. 2017). For our tests, we were very successful by using a simple heuristic approach. (Our framework is also compatible with more sophisticated methods for selecting $\hat{n}$.) For each data set, we supply an expected value of $\hat{n}$ for that data set, and we then add a quadratic penalty to the objective-function value whenever $\hat{n}$ differs from its expected value. This helps curtail overfitting, while still allowing our algorithms to perform merges and splits to escape bad local minima. Notably, this penalty does not alter the MCF, AC, or MBO procedures; instead, it is part of Algorithm 4.

Algorithm 4. Our splitting–merging wrapper for escaping from bad local minima. In this algorithm, Q is the objective function from (6) multiplied by $[1 + 0.1\left(\hat{n} - \hat{n}_{\text{expected}}\right)^2]$, where we chose the value 0.1 based on hand-tuning.

Input A, $\hat{n}_{\text{expected}}$.
Place all nodes in the same community, and add this community to a queue.
while the queue is not empty **do**
  Save $g_{\text{old}} = g$ and $W_{\text{old}} = W$.
  Save the current objective-function value as Q_{old}.
  Partition the next community (as an induced subgraph, as we include all associated edges*) in the queue into $\min\left\{\hat{n}_{\text{expected}}, \sqrt{N}\right\}$ communities using MCF, AC, or MBO with $w_{\alpha\beta} = \begin{cases} 1, & \alpha = \beta, \\ 0.1, & \alpha \neq \beta. \end{cases}$
  while it is possible to improve the objective-function value by merging two partition elements **do**
    Perform the merge that most improves the objective function.
  end while
  if the objective-function value is larger than Q_{old} **then**
    Add any newly created communities to the queue.
  else
    Set $g = g_{\text{old}}$ and $W = W_{\text{old}}$.
    Remove the current community from the queue.
  end if
end while
Output g, W.

*For this step, note that we use the degrees and number of edges from the entire graph, rather than from the induced subgraph, when computing volumes. See Boyd et al. (2018) and Hu et al. (2013), which make an analogous adjustment in the associated recursive step of their algorithms

5 Empirical Results

We now discuss our results from several numerical experiments to (1) confirm that our algorithms can successfully recover g and ω from networks that we generate using SBMs and (2) explore their applicability to real-world networks. In our experiments, we use three different families of SBMs, three Facebook networks [whose community structure is partly understood (Traud et al. 2011, 2012)], and an example

Table 1 Results of several tests on several synthetic and empirical networks

	PP	LFR	MS	Caltech	Princeton	Penn. State	Plume
Nodes	16,000	1000	10,230	762	6575	41,536	284,481
Edges	2.9×10^5	9.8×10^3	1.0×10^5	16,651	293,307	1,362,220	2,723,840
Communities	10	40	10	8	4	8	5
Score							
MCF	0	0	0	−0.16	−0.02	−0.56	−1.41
AC	0	0	0	0.21	0.58	−0.04	−1.23
MBO	0	0	0	0.53	1.12	0.40	−1.21
KL	0.28	0.03	0.04	−0.16	0.11	−0.55	−1.38
Reference	0	0	0	0	0	0	0

We use three surface-tension-based methods (mean-curvature flow, Allen–Cahn, and Merriman–Bence–Osher) and the Kernighan–Lin algorithm from Karrer and Newman (2011) to partition three synthetic networks (planted partition, LFR, and multiscale SBM) and the largest connected components of three empirical networks (Caltech36, Princeton12, and Penn94) from the FACEBOOK100 data set (Traud et al. 2012). The score is the recovered surface-tension energy (6) minus the corresponding energy of a reference partition, divided by the absolute value of the energy of the reference partition. Smaller values indicate better performance, and 0 corresponds to a partition that is of comparable quality as the reference partition. For the synthetic networks, we use the planted (and hence ground-truth) community structure as the reference partition. For the Facebook networks, we use metadata that are positively correlated with community structure (specifically, House affiliation for Caltech and graduation year for the other two networks). For the plume video, our reference partition is to assign all nodes to the same community, because no pixel-level metadata are associated with the images. The edge counts on the synthetic networks constitute an order-of-magnitude approximation, because the exact number differs across the three instantiations of these models. (This table gives the best result among three tests for each example; for the random-graph models, each such example is a different graph from the same model.)

related to hyperspectral video segmentation. Because of the random initialization in our approach, we perform three trials on each of the networks for each algorithm; we report the best result in each case.[11] For comparison, we also report the results of a Kernighan–Lin (KL) algorithm, which was reported in Karrer and Newman (2011) to be effective. We summarize our results in Table 1, and we highlight that we consistently recover the underlying structure in the synthetic examples. For the real networks, we compare our results with a reference partition based on metadata that is thought to be correlated with the community structure. We find that the MCF scheme performs the best among our three schemes on these networks, and we note that it finds partitions with a larger likelihood than the reference partition.[12] We implement our methods in MATLAB, so one should interpret computation times in Table 2 as indicative that the run time is reasonable for networks with millions of edges. Given a careful implementation in a compiled language, it is possible to study even larger networks.[13] For an example of code for a similar problem that was solved by an MBO scheme at large scale (including a weighted graph with almost 14 million nodes and 1.8×10^{14} edges), see Meng et al. (2017).

We briefly describe the three families of SBM-related networks that we use in our numerical experiments.

[11] We use three trials (using three different networks drawn from the random-graph models) to illustrate that our algorithms do not require a large number of attempts to reach a good optimum. In most of our trials, even a single run of a solver is likely to give good results. In Table 1, we report our best scores. Our worst scores for MCF are 0.00, 0.00, 0.00, -0.14, and 0.01 for the PP, MS, LFR, Caltech, and Princeton networks, respectively. (We did not record the worst score for Penn. State or the plume network.) Our corresponding worst scores for AC and MBO are 0.00, 0.00, 0.01, 0.22, and 0.86 (for AC) and 0.15, 0.00, 0.02, 0.53, and 1.12 (for MBO). Comparing these results with those in Table 1, we see that our best and worst scores are often similar to each other.

[12] In synthetic networks, the reference partitions represent a "ground truth," in the sense that they reflect the principle upon which we constructed the network. For these networks, finding a partition with higher likelihood than the reference partition reflects the fact that the data is stochastic, and a maximum-likelihood partition may differ slightly from the ground-truth one. An algorithm that achieves a likelihood that is higher than ground truth is more successful at optimizing the likelihood function than one that does not. In real networks, the reference partition is not a "ground truth." Instead, it is a point of reference that is based on a "natural" grouping of the nodes when one is available. In most applications of community detection, there is no ground truth (Peel et al. 2017). Real networks can have many different organizing principles and multiple insightful partitions, including both (1) partitions that are slight variations of each other that yield similar values of objective functions and (2) partitions that are very different from each other that yield similar values of such functions (Good et al. 2010). In particular, the fact that some of our methods find partitions with higher likelihood scores than the reference partition is not indicative of a failure of the maximum-likelihood approach, because there is no reason for a reference partition to be the best possible partition of a network. [For the Facebook networks, for example, it is known that this is not the case (Hric et al. 2016).] To the extent that SBM MLE is appropriate for the data and application, our partitions are sometimes better than the reference partitions.

[13] One may perhaps construe from Table 2 that the MCF method's speed on the large plume example indicates that its scaling is better than linear with the number of nodes; this is of course not the case, because we alter each node during each iteration in our implementation. Several factors influence the observed computation times. For example, using a larger data set improves the effectiveness of parallelization and vectorization of operations (which MATLAB does automatically for certain operations). Furthermore, the operation count can be quadratic in the number of communities, which is heterogeneous across the data sets that we examined and is largest in the LFR example. Finally, the different networks are very different structurally, which affects the number of iterations that are necessary for convergence.

Table 2 Median computation times (in seconds) for our example networks

	PP	LFR	MS	Caltech	Princeton	Penn. State	Plume
MCF	5.36	17.71	3.47	1.39	1.46	38.91	77.91
AC	5.37	26.27	7.28	8.84	480.4	3853	268.7
MBO	4.27	11.05	1.73	0.67	7.43	382.31	270.0
KL	16,566	176	5117	20	662	95,603	980,520

- Planted partition (PP) is a 16,000-node graph that consists of 10 equal-sized communities. It is produced by the method that was described in Karrer and Newman (2011). It builds a degree-corrected SBM with a truncated power-law degree distribution with exponent 2. The parameter λ from Equation (27) in Karrer and Newman (2011) is 0.001, indicating a fairly clear separation between communities.
- Lancichinetti–Fortunato–Radicchi (LFR) is a standard benchmark SBM network (Lancichinetti et al. 2008). We construct 1000-node LFR graphs with a power-law degree distribution (with exponent 2), mean degree 20, maximum degree 50, power-law-distributed community sizes (with exponent 1), community sizes between 10 and 50 nodes, and mixing parameter 0.1.
- Multiscale SBM (MS). To construct such a graph, we take a sequence of disjoint components; in order, these are a 10-clique, a 20-clique, and a sequence of Erdős–Rényi (ER) graphs (drawn from the $G(n, p)$ model with n nodes and $np = 20$) of sizes 40, 80, 160, …, 5120. Each of these graphs has a total of 10,230 nodes. In each such graph, we connect the components to each other by adding a single edge, from nodes chosen uniformly at random, between each consecutive clique or ER graph. This construction tests whether an algorithm can find communities of widely varying sizes in the same graph (Arenas et al. 2008; Fortunato and Barthélemy 2007).

The hyperspectral video is a recording of a gas plume as it was released at the Dugway Proving Ground (Gerhart et al. 2013; Manolakis et al. 2001; Merkurjev et al. 2014). A hyperspectral video is different from an RGB video, in that each pixel in the former encodes the intensity of light at a large number (e.g., 129, in this case) of different wavelengths rather than at only 3, with each channel corresponding to a wavelength. We consider the classification problem of identifying pixels that include similar materials (such as dirt, road, grass, and so on). This problem is difficult because of the diffuse nature of the gas, which leads to a faint signal that spreads out among many wavelengths and with boundaries that are difficult to determine. We construct a graph representation of this video using "non-local means," as described in Buades et al. (2005). Specifically, we use the following construction. For each pixel p and in each of 7 frames, we construct a vector v_p by concatenating the data in a 3×3 window that is centered at p. We then use a weighted cosine similarity measure (which is a common choice for hyperspectral imaging applications) on these $(3 \times 3 \times 129)$-component vectors, where we give the most weight to the components from the center

Fig. 4 Segmentation of a hyperspectral video using graph MCF. The gas plume is clearly represented in the yellow and orange pixels. The two blue communities on the bottom are the ground, and the other two communities are the sky. This image is frame 3 of 7. (It is best to view this plot in color.) (Color figure online)

Table 3 Optimal surface tensions for the MS SBM example

0	5.22	∞	∞	∞	∞	∞	∞	∞	∞
5.22	0	6.1817	∞	∞	∞	∞	∞	∞	∞
∞	6.1817	0	6.8471	∞	∞	∞	∞	∞	∞
∞	∞	6.8471	0	7.6316	∞	∞	∞	∞	∞
∞	∞	∞	7.6316	0	8.362	∞	∞	∞	∞
∞	∞	∞	∞	8.362	0	9.0869	∞	∞	∞
∞	∞	∞	∞	∞	9.0869	0	9.7926	∞	∞
∞	∞	∞	∞	∞	∞	9.7926	0	10.4911	∞
∞	∞	∞	∞	∞	∞	∞	10.4911	0	11.1869
∞	∞	∞	∞	∞	∞	∞	∞	11.1869	0

The entries are heterogeneous, so there are different surface tensions between different pairs of communities. The infinite entries correspond to pairs of communities with no observed edge between them

of the window.[14] Finally, using the VLFeat software package (Vedaldi and Fulkerson 2008), we build an unweighted 10-nearest-neighbor graph using the similarity measure and a k-dimensional tree (with $k = 10$) (Bentley 1975). We see from Fig. 4 that partitions with small values of (6) correspond to meaningful segmentations of the image.

In Table 3, we show an example of a W matrix that we obtain from an MS network to illustrate that we recover different surface tensions between different pairs of communities.[15]

[14] We weight the center pixel components by 1, the components from adjacent pixels by 0.5, and the components from corner pixels by 0.25. That is, we let v_{ij} be the 129-element vector at pixel (i, j), and we define w_{ij} as the concatenation of v_{ij}, $.5v_{i+1,j}$, $.5v_{i-1,j}$, $.5v_{i,j+1}$, $.5v_{i,j-1}$, $.25v_{i+1,j+1}$, $.25v_{i+1,j-1}$, $.25v_{i-1,j+1}$, and $.25v_{i-1,j-1}$. We then calculate the cosine similarity between each pair of w_{ij} vectors.

[15] For this example, we used the change of variables from Appendix A to eliminate the diagonal elements.

6 Conclusions and Discussion

We have shown that a particular stochastic block model (SBM) maximum-likelihood estimation (MLE) problem is equivalent to a discrete version of a well-known surface-tension problem. This equivalence, which associates graph cuts to surface areas and SBM parameters to physical surface tensions, gives new geometric and physical interpretations to SBM MLE problems, which are traditionally viewed from a statistical perspective. We used the new connection to adapt three well-known surface-tension-minimization algorithms to community detection in graphs. Our subsequent computations suggest that the resulting algorithms are able to successfully find underlying community structure in SBM-related graphs. When applied to graphs that are constructed from empirical data, our mean-curvature-flow (MCF) method performs very well, but the other two methods face some issues (which will be interesting to explore in future studies).[16] We also proved a Γ-convergence result that gives theoretical justification for our algorithms.

Although our paper has focused on a specific form of an SBM and an associated MLE problem, our techniques should also be insightful for other studies of SBMs and their applications. One straightforward adaption is to consider SBMs without degree correction, although that is more interesting for theoretical work than for applications. Additionally, it seems promising to incorporate priors on the values of g and ω as regularizers in the surface-tension energy [perhaps in a way that is similar to the procedure in Bertozzi et al. (2018)]. Another viable extension is to incorporate a small amount of supervision into the community-inference process using techniques (such as quadratic fidelity terms) from image processing. A similar idea was used for modularity maximization in Hu et al. (2013) and was tested further in Boyd et al. (2018).

Introducing supervision helps alleviate severe non-convexity by penalizing local minima that are inconsistent with the (ideally) ground-truth classifications from which one draws the supervision. It is also important to generalize our approach to more complicated types of networks, such as multilayer (Kivelä et al. 2014) and temporal networks (Holme and Saramäki 2012), and to incorporate metadata (Newman and Clauset 2016) into our inference methodology. For example, given our successful results on the hyperspectral video, it may be particularly interesting to use temporal network clustering to analyze time-dependent communities in the video.

[16] Several factors seem to contribute to the performance difference, and it is impossible to disentangle them without extensive additional testing. The following are some possible contributing factors. The Facebook networks are not generated from an SBM, and the former have more complicated structures than those in synthetic networks. Our particular solution method involving eigenvectors may thus be less appropriate for solving diffusion equations in the Facebook networks than in synthetic networks. It is empirically clear that the Facebook networks have more eigenvector localization (e.g., as measured using inverse participation ratio) in the combinatorial graph Laplacian than the synthetic networks. The AC and MBO methods deal with the non-convexity differently than MCF. (The former two use pseudospectral methods to jump to better regions, whereas MCF allows all boundary nodes to move independently and simultaneously.) It would be interesting to conduct a detailed study of these various factors using a large variety of networks, as it will likely improve scientific understanding of the geometry and associated flows for different families of networks.

Approaches such as inference using SBMs and modularity maximization are also related to other approaches for community detection, and the results in the present paper may help further illuminate those connections. These include recent work that relates SBMs to local methods for community detection that are based on personalized PageRank (Kloumann et al. 2017) and very recent work that established new connections between modularity maximization and several other approaches (Veldt et al. 2017). We expect that further investigations of the relations between the diverse available perspectives on community detection (and other problems in network clustering) will yield many new insights for network theory, algorithms, and applications.

Acknowledgements ZMB and ALB were funded by NSF Grants DMS-1737770 and DMS-1417674, as well as ONR Grant N00014-16-1-2119. ZMB was also supported by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program and the Eunice Kennedy Shriver National Institute of Child Health & Human Development of the National Institutes of Health under Award Number R01HD075712. All three authors were supported by DARPA award number FA8750-18-2-0066. We thank Brent Edmunds, Robert Hannah, and Kevin Miller for helpful discussions. We also thank two anonymous referees for many helpful comments. The content is solely the responsibility of the authors and does not necessarily represent the official views of any of the agencies that supported this work.

A. Eliminating the Diagonal Elements of *W*

It is difficult to interpret the parameters $W_{\alpha\alpha}$ in the context of (6) and our surface-tension analogy, because they correspond to "internal" surface tensions of a single crystal. In this appendix, we use a change of variables to eliminate these diagonal terms and replace them with additional volume terms, which are much easier to interpret.

We begin with the identity

$$\sum_{\alpha,\beta} W_{\alpha\beta}\mathrm{Cut}_{g,A}(\alpha,\beta) = \sum_{\alpha}\sum_{\beta\neq\alpha} W_{\alpha\beta}\mathrm{Cut}_{g,A}(\alpha,\beta) + \sum_{\alpha} W_{\alpha\alpha}\mathrm{Cut}_{g,A}(\alpha,\alpha)\,, \tag{15}$$

and we compute

$$\begin{aligned}
\sum_{\alpha} W_{\alpha\alpha}\mathrm{Cut}_{g,A}(\alpha,\alpha) &= \sum_{\alpha} W_{\alpha,\alpha} \sum_{g_i=\alpha, g_j=\alpha} w_{ij} \\
&= \sum_{\alpha} W_{\alpha,\alpha} \left(\sum_{g_i=\alpha, j=1,\ldots,N} w_{ij} - \sum_{g_i=\alpha, g_j\neq\alpha} w_{ij} \right) \\
&= \sum_{\alpha} W_{\alpha,\alpha} \left(\sum_{g_i=\alpha} k_i - \sum_{\beta\neq\alpha, g_i=\alpha, g_j=\beta} w_{ij} \right) \\
&= \sum_{\alpha} W_{\alpha,\alpha} \left(\mathrm{vol}_{g,A}(\alpha) - \sum_{\beta\neq\alpha} \mathrm{Cut}_{g,A}(\alpha,\beta) \right). \qquad (16)
\end{aligned}$$

Combining (16) with (15) yields

$$\sum_{\alpha,\beta} W_{\alpha\beta}\mathrm{Cut}_{g,A}(\alpha,\beta) = \sum_{\alpha\neq\beta} \left(W_{\alpha\beta} - W_{\alpha\alpha}\right)\mathrm{Cut}_{g,A}(\alpha,\beta) + \sum_{\alpha} W_{\alpha\alpha}\mathrm{vol}_{g,A}(\alpha)\,, \quad (17)$$

assuming[17] that $W_{\alpha\alpha}$ is finite for each α. This formulation removes the diagonal from the double sum at the cost of introducing asymmetry into the subscripts of the coefficients. We can fix this new issue by replacing (17) with

$$\begin{aligned}\sum_{\alpha,\beta} W_{\alpha\beta}\mathrm{Cut}_{g,A}(\alpha,\beta) &= \sum_{\alpha\neq\beta}\left(W_{\alpha\beta} - \frac{1}{2}W_{\alpha\alpha} - \frac{1}{2}W_{\beta\beta}\right)\mathrm{Cut}_{g,A}(\alpha,\beta)\\ &\quad + \sum_{\alpha} W_{\alpha\alpha}\mathrm{vol}_{g,A}(\alpha)\\ &= \sum_{\alpha\neq\beta}\hat{\sigma}_{\alpha\beta}\mathrm{Cut}_{g,A}(\alpha,\beta) + \sum_{\alpha} W_{\alpha\alpha}\mathrm{vol}_{g,A}(\alpha)\,, \quad (18)\end{aligned}$$

where $\hat{\sigma}_{\alpha\beta} = W_{\alpha\beta} - \frac{1}{2}W_{\alpha\alpha} - \frac{1}{2}W_{\beta\beta}$. The matrix $\hat{\sigma}$ is symmetric and has 0 values on the diagonal.

Finally, we expand a bit on the role of the volume terms in (6). The term

$$\sum_{\alpha} W_{\alpha\alpha}\mathrm{vol}_{g,A}(\alpha) \quad (19)$$

is the inner product of the vector of volumes with the diagonal of W. We minimize (19), subject to the constraints $\sum_{\alpha}\mathrm{vol}_{g,A}(\alpha) = 2m$ and $\mathrm{vol}_{g,A}(\alpha) \geq 0$, by placing all of the nodes in the community that corresponds to the smallest[18] entry in the diagonal of W. Therefore, these terms incentivize placing more mass in the communities that have the smallest volume penalties.

B. Γ-Convergence of the Ginzburg–Landau Approximation of (6)

The notion of Γ-convergence is defined as follows.

Definition B.1 Let Y be a metric space, and let F_n be a sequence of functionals that take values in $\mathbb{R}\cup\{\infty\}\cup\{-\infty\}$. We say that F_n Γ-*converges* to another functional F if for all $x \in Y$, the following bounds hold:

1. (Lower bound) For every sequence $x_n \to x$, we have $F(x) \leq \liminf_{n\to\infty} F_n(x_n)$.
2. (Upper bound) For every $x \in Y$, there is a sequence $x_n \to x$ such that $F(x) \geq \limsup_{n\to\infty} F_n(x_n)$.

[17] The case in which $W_{\alpha\alpha} = \infty$ does not occur in our methods.

[18] When referring to "smallest" eigenvalues in the appendices, we mean the smallest positive or most negative values, rather than those that are smallest in magnitude.

We now prove Theorem 4.1.

Proof We largely follow van Gennip and Bertozzi (2012), although we generalize to account for the multi-phase nature of our problem. The terms that do not involve the potential T are continuous and independent of ϵ, so they cannot interfere with Γ-convergence (Dal Maso 1993).[19] Consequently, it suffices to prove that $\frac{1}{\epsilon}T : \mathbb{R}^{N\times\hat{n}} \to \mathbb{R}$ Γ-converges to

$$\chi(U) = \begin{cases} 0\,, & \text{if } U \text{ corresponds to a partition}\,, \\ +\infty\,, & \text{otherwise}\,. \end{cases}$$

To prove the lower bound, let $U_n \to U$ and $\epsilon_n \to 0$. (In this proof, the subscript n indexes the sequence, rather than the matrix columns.) If U corresponds to a partition, $\chi(U) = 0$, which is automatically less than or equal to $\frac{1}{\epsilon_n}T(U_n)$ for each n. If U does not correspond to a partition, $\chi(U) = +\infty$. There exists a constant $c > 0$ such that the distance (in, for example, the Frobenius norm) from U_n to the nearest feasible point (i.e., a point corresponding to a partition) is at least c as $n \to \infty$. Let T_c be the infimum of T on all of $\mathbb{R}^{N\times\hat{n}}$ except for the balls of radius c that surround each feasible point (so, in particular, $T_0 > 0$). It follows that $\liminf_{n\to\infty} \frac{1}{\epsilon_n}T(U_n) \geq \lim_{n\to\infty} \frac{1}{\epsilon_n}T_0 = +\infty$. Therefore, the lower bound always holds.

To prove the upper bound, let U be any $N \times \hat{n}$ matrix. If U corresponds to a partition, then letting $U_n = U$ for all n gives the required sequence. If u does not correspond to a partition, then $U_n = U$ for all n still satisfies the upper bound.

Therefore, both the upper and lower bound requirements hold, and we have proven Γ-convergence. □

C. Additional Notes on the AC and MBO Schemes

In this appendix, we discuss some practical details about our implementation of the AC and MBO solvers.

The choice of ϵ in AC is important, because it selects a characteristic scale of the transition between the $U_\alpha \approx 1$ and $U_\alpha \approx 0$ regions. If ϵ is too small, the barrier to transition is large, and no evolution occurs. If it is too large, the transition layer includes so many nodes that U does not approximately correspond to a partition of a graph. Furthermore, Theorem 4.1 asserts only that the minimizers of (6) and (13) are related when ϵ is sufficiently small. In our numerical experiments, we set $\epsilon = 0.004$, a choice that we selected by hand-tuning using our synthetic networks. There is no reason to believe that the same value should work for all networks. For example, for the well-known Zachary Karate Club network (Zachary 1977), we obtain much better results for $\epsilon = 0.04$. A very interesting problem is to determine a correct notion of distance and accompanying quantitative estimates to allow an automated selection

[19] The graph-TV term is a composition of addition, subtraction, projection onto components, and taking absolute values; therefore, it is continuous.

of ϵ to obtain a transition layer with an appropriate width to give useful results. We discretize the AC equation via convex splitting (Eyre 1998):

$$(1 + c\,dt)U^{n+1} - LU^{n+1}W = -dt\left(cU^n + k.*\,\mathrm{diag}(W) + T'(U^n) + \frac{1}{2m}kk^T Ue^{-W}\right),$$

where L is the combinatorial graph Laplacian matrix (see our previous discussions), T' denotes the derivative of T, and $c > 2/\epsilon$ (Luo and Bertozzi 2017). Using the constant c leads to an unconditionally stable scheme, which negates the stiffness caused by the $1/\epsilon$ scale.

It is necessary to solve a linear system of the form

$$(1 + c\,dt)U^{n+1} - LU^{n+1}W = F^n \tag{20}$$

many times. In a continuum setting, one can use a fast Fourier transform, but we do not know of a graph analog with comparable computational efficiency. Instead, we find the $2\hat{n}$ eigenvectors that correspond to the smallest eigenvalues[20] of L; and we employ the entire spectrum of W. Therefore, L is approximated by $V_L D_L V_L^T$, where D_L is a $2\hat{n} \times 2\hat{n}$ diagonal matrix of the smallest eigenvectors of L, sorted from smallest to largest, and V_L is the associated matrix of eigenvectors. Furthermore, let $W = V_W D_W V_W^T$ be the full spectral decomposition of W. The system (20) is then approximately equivalent to

$$(1 + c\,dt)V_L^T U^{n+1} V_W - D_L V_L^T U^{n+1} V_W D_W = V_L^T F^n V_W\,.$$

Letting $\hat{U}^n = V_L^T U^n V_W$ and $\hat{F}^n = V_L^T F^n V_W$, we write

$$(1 + c\,dt)\hat{U}^{n+1} - D_L \hat{U}^{n+1} D_W = \hat{F}^n, \tag{21}$$

which is easy to solve for $\hat{U}^{n+1}$. We convert $\hat{U}^{n+1}$ to a solution using $U^{n+1} = V_L \hat{U}^{N+1} V_W^T$. See Bertozzi and Flenner (2012) for a discussion of this method of recovering U^{n+1} from $\hat{U}^{n+1}$.

One final detail that we wish to note is that we want the evolution of U to be restricted to have a row sum of 1, so that we can interpret it in terms of probabilities. To do this, we use a modification of the projection algorithm from Chen and Ye (2011) at each time step.[21]

The MBO solver uses a very similar pseudospectral scheme, although it does not include convex splitting. Unlike in the AC scheme, we need to estimate two time steps

[20] The number $2\hat{n}$ is somewhat arbitrary; we choose it to exceed $\hat{n}$, but for computational convenience, we do not want it to be too large.

[21] The algorithm from Chen and Ye (2011) acts on a single row vector, and our modification is simply to process all rows at once by replacing operations on row-vector components with operations on matrix columns. The result is mathematically equivalent (up to round-off errors), but it is much faster because it vectorizes the operations.

automatically, instead of tuning them by hand.[22] The first is the inner-loop step (i.e., the time step that we use for computing the diffusion), which we determine using a restriction (which one can show is necessary for stability[23]) that the time step should not exceed twice the reciprocal of the largest eigenvalue of the linear operator that maps $U \to \frac{1}{m} k k^T U e^{-W}$. The time step between thresholdings of U is given by the reciprocal of the geometric mean of the largest and smallest eigenvalues of the operator that maps $U \to LUW$. The associated intuition is that linear diffusion should have enough time to evolve (to avoid getting stuck) but not enough time to evolve to steady state (because the steady state does not depend on the initial condition, so it carries no information about it). The reciprocal of the smallest eigenvalue gives an estimate of the time that it takes to reach steady state, and the reciprocal of the largest eigenvalue gives an estimate of the fastest evolution of the system. We choose the geometric mean between these values to produce a number between these two extremes.[24] Boyd et al. (2018) and van Gennip et al. (2014) proved bounds (although in a simpler setting) that support these time-step choices for MBO schemes.

References

Amini, A.A., Chen, A., Bickel, P.J., Levina, E.: Pseudo-likelihood methods for community detection in large sparse networks. Ann. Statist. **41**, 2097–2122 (2013)

Arenas, A., Fernández, A., Gómez, S.: Analysis of the structure of complex networks at different resolution levels. New J. Phys. **10**, 053039 (2008)

Ashcroft, N.W., Mermin, N.D.: Solid State Physics, 1st edn. Brooks Cole, Pacific Grove (1976)

Ayati, M., Erten, S., Chance, M.R., Koyuturk, M.: MOBAS: identification of disease-associated protein subnets using modularity-based scoring. EURASIP J. Bioinf. Sys. Bio. **1**, 1–14 (2015)

Bassett, D.S., Owens, E.T., Porter, M.A., Manning, M.L., Daniels, K.E.: Extraction of force-chain network architecture in granular materials using community detection. Soft Matter **11**, 2731–2744 (2015)

Bentley, J.L.: Multidimensional binary search trees used for associative searching. Commun. ACM **18**, 509–517 (1975)

Bertozzi, A.L., Flenner, A.: Diffuse interface models on graphs for classification of high dimensional data. Multiscale Model. Simul. **10**, 1090–1118 (2012)

Bertozzi, A.L., Flenner, A.: Diffuse interface models on graphs for classification of high dimensional data. SIAM Rev. **58**, 293–328 (2016)

Bertozzi, A.L., Luo, X., Stuart, A.M., Zygalakis, K.C.: Uncertainty quantification in graph-based classification of high dimensional data. SIAM/ASA J. Uncertain. Quantif. **6**, 568–595 (2018)

Betzel, R.F., Bassett, D.S.: Multi-scale brain networks. NeuroImage **160**, 73–83 (2017)

Boettinger, W.J., Warren, J.A., Beckermann, C., Karma, A.: Phase-field simulation of solidification. Ann. Rev. Mater. Res. **32**, 163–194 (2002)

Boyd, Z.M., Bae, E., Tai, X.-C., Bertozzi, A.L.: Simplified energy landscape for modularity using total variation. SIAM J. App. Math. **78**, 2439–2464 (2018)

Boykov, Y., Kolmogorov, V.: Computing geodesics and minimal surfaces via graph cuts. In: Proceedings of the Ninth IEEE International Conference on Computer Vision, vol. 2, ICCV '03, IEEE Computer Society, pp. 26–33. Washington, DC (2003)

[22] Tuning by hand is not only laborious, but it also is very problematic in cases involving recursion or when W changes, because the correct time step depends both on the (sub)graph that is being partitioned and on W. Consequently, there may be no time step that works for all subgraphs and choices of W even for a single data set.

[23] See, e.g., LeVeque (2007) for a description of the necessary techniques, which are standard in the numerical analysis of ordinary differential equations.

[24] From experimentation, we concluded that it is better to multiply this time step by 8 to avoid getting stuck too early. We chose the geometric mean because eigenvalues can have very different orders of magnitude.

Boykov, Y., Kolmogorov, V., Cremers, D., Delong, A.: An integral solution to surface evolution PDEs via geo-cuts. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7–13, 2006, Proceedings, Part III, pp. 409–422. Springer, Berlin (2006)

Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: Computer Vision and Pattern Recognition, vol. 2, pp. 60–65. (2005)

Candès, E.J., Romberg, J., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inform. Theory **52**, 489–509 (2006)

Cenna: Gr3gr.gif. Wikimedia Commons https://commons.wikimedia.org/wiki/File:Grgr3d_small.gif. (2012). Accessed 30 Mar 2019

Chen, Y., Ye, X.: Projection onto a simplex. arXiv:1101.6081 (2011)

Cleri, F., Phillpot, S.R., Wolf, D.: Atomistic simulations of intergranular fracture in symmetric-tilt grain boundaries. Interface Sci. **7**, 45–55 (1999)

Condon, A., Karp, R.M.: Algorithms for graph partitioning on the planted partition model. Random Struct. Algorithms **18**, 116–140 (2001)

Csermely, P., London, A., Wu, L.-Y., Uzzi, B.: Structure and dynamics of core/periphery networks. J. Complex Netw. **1**, 93–123 (2013)

Dal Maso, G.: An Introduction to Γ-Convergence. Birkhauser, Boston (1993)

Esedoglu, S., Otto, F.: Threshold dynamics for networks with arbitrary surface tensions. Commun. Pure Appl. Math. **68**, 808–864 (2015)

Eyre, D.J.: An unconditionally stable one-step scheme for gradient systems. Preprint, Available at https://www.math.utah.edu/~eyre/research/methods/stable.ps. Accessed 30 Mar 2019

Fienberg, S.E., Wasserman, S.S.: Categorical data analysis of single sociometric relations. Sociol. Meth. **12**, 156–192 (1981)

Fortunato, S., Barthélemy, M.: Resolution limit in community detection. Proc. Natl. Acad. Sci. USA **104**, 36–41 (2007)

Fortunato, S., Hric, D.: Community detection in networks: a user guide. Phys. Rep. **659**, 1–44 (2016)

Fosdick, B.K., Larremore, D.B., Nishimura, J., Ugander, J.: Configuring random graph models with fixed degree sequences. SIAM Rev. **60**, 315–355 (2018)

Frank, O., Harary, F.: Cluster inference by using transitivity indices in empirical graphs. J. Am. Stat. Soc. **77**, 835–840 (1982)

Frost, H.J., Thompson, C.V., Walton, D.T.: Simulation of thin film grain structures—I. Grain growth stagnation. Acta Metall. Mater. **38**, 1455–1462 (1990)

Garcia-Cardona, C., Merkurjev, E., Bertozzi, A.L., Percus, A.L., Flenner, A.: Multiclass segmentation using the Ginzburg–Landau functional and the MBO scheme. IEEE Trans. Pattern Anal. Mach. Intell. **36**, 1600–1614 (2014)

Gerhart, T., Sunu, J., Lieu, L., Merkurjev, E., Chang, J.-M., Gilles, J., Bertozzi, A.L.: Detection and tracking of gas plumes in LWIR hyperspectral video sequence data. SPIE. Int. Soc. Opt. Photon. **8743**, 87430J (2013)

Ghasemian, A., Hosseinmardi, H., Clauset, A.: Evaluating overfit and underfit in models of network community structure. arXiv:1802.10582. (2018)

Gilboa, G., Osher, S.: Nonlocal operators with applications to image processing. Multiscale Model. Simul. **7**, 1005–1028 (2008)

Goldstein, T., Osher, S.: The split Bregman method for L1-regularized problems. SIAM J. Imaging Sci. **2**, 323–343 (2009)

Good, B.H., de Montjoye, Y.-A., Clauset, A.: Performance of modularity maximization in practical contexts. Phys. Rev. E **81**, 046106 (2010)

Hegemann, R.A., Smith, L.M., Barbaro, A.B., Bertozzi, A.L., Reid, S.E., Tita, G.E.: Geographical influences of an emerging network of gang rivalries. Physica A **390**, 3894–3914 (2011)

Holland, P.W., Laskey, K.B., Leinhardt, S.: Stochastic blockmodels: first steps. Social Netw. **5**, 109–137 (1983)

Holme, P.: Modern temporal network theory: a colloquium. Eur. Phys. J. B **88**, 234 (2015)

Holme, P., Saramäki, J.: Temporal networks. Phys. Rep. **519**, 97–125 (2012)

Hric, D., Peixoto, T.P., Fortunato, S.: Network structure, metadata, and the prediction of missing nodes and annotations. Phys. Rev. X **6**, 031038 (2016)

Hu, H., Laurent, T., Porter, M.A., Bertozzi, A.L.: A method based on total variation for network modularity optimization using the MBO scheme. SIAM J. Appl. Math. **73**, 2224–2246 (2013)

Jacobs, M.: Algorithms for Multiphase Partitioning. University of Michigan, Ann Arbor (2017). PhD thesis
Jacobs, M., Merkurjev, E., Esedoglu, S.: Auction dynamics: a volume-constrained MBO scheme. J. Comput. Phys. **354**, 288–310 (2018)
Jeub, L.G.S., Balachandran, P., Porter, M.A., Mucha, P.J., Mahoney, M.W.: Think locally, act locally: detection of small, medium-sized, and large communities in large networks. Phys. Rev. E **91**, 012821 (2015)
Karrer, B., Newman, M.E.J.: Stochastic blockmodels and community structure in networks. Phys. Rev. E **83**, 016107 (2011)
Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. Bell Syst. Tech. J. **49**, 291–307 (1970)
Kim, M., Leskovec, J.: Inferring missing nodes and edges in networks. In: Chawla, N., Wang, W., (eds.) Proceedings of the 2011 SIAM International Conference on Data Mining, pp. 47–58 (2011)
Kinderlehrer, D., Livshits, I., Ta'asan, S.: A variational approach to modeling and simulation of grain growth. SIAM J. Sci. Comput. **28**, 1694–1715 (2006)
Kivelä, M., Arenas, A., Barthélemy, M., Gleeson, J.P., Moreno, Y., Porter, M.A.: Multilayer networks. J. Complex Netw. **2**, 203–271 (2014)
Kloumann, I.M., Ugander, J., Kleinberg, J.: Block models and personalized PageRank. Proc. Natl. Acad. Sci. USA **114**, 33–38 (2017)
Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Phys. Rev. E. **78**, 056117 (2008)
LeVeque, R.J.: Finite difference methods for differential equations. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2007). See also https://staff.washington.edu/rjl/fdmbook/
Luo, X., Bertozzi, A.L.: Convergence of the graph Allen–Cahn scheme. J. Stat. Phys. **167**, 934–958 (2017)
Manolakis, D., Siracusa, C., Shaw, G.: Adaptive matched subspace detectors for hyperspectral imaging applications. In: 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, pp. 3153–3156 (2001)
Mantegazza, C.: Lecture Notes on Mean Curvature Flow. Springer-Verlag, Berlin (2011)
Meng, Z., Merkurjev, E., Koniges, A., Bertozzi, A.L.: Hyperspectral image classification using graph clustering methods. Image Processing On Line **7**, 218–245 (2017)
Merkurjev, E., Sunu, J., Bertozzi, A.L.: Graph MBO method for multiclass segmentation of hyperspectral stand-off detection video. In: IEEE International Conference on Image Processing, pp 689–693 (2014)
Merkurjev, E., Bae, E., Bertozzi, A.L., Tai, X.-C.: Global binary optimization on graphs for data segmentation. J. Math. Imaging Vis. **52**, 414–435 (2015)
Merriman, B., Bence, J., Osher, S.: Diffusion generated motion by mean curvature. In: Proceedings of Computing Crystal Growers Workshop, pp. 73–83 (1992)
Modica, L.: The gradient theory of phase transitions and the minimal interface criterion. Arch. Ration. Mech. Anal. **98**, 123–142 (1987)
Moore, C.: The computer science and physics of community detection: landscapes, phase transitions, and hardness. arXiv:1702.00467 (2017). Also see the version in *Bulletin of the EATCS*, which is available at http://bulletin.eatcs.org/index.php/beatcs/article/view/480/471
Mullins, W.W.: Two-dimensional motion of idealized grain boundaries. J. Appl. Phys. **27**, 900–904 (1956)
Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E **74**, 036104 (2006)
Newman, M.E.J.: Equivalence between modularity optimization and maximum likelihood methods for community detection. Phys. Rev. E **94**, 052315 (2016)
Newman, M.E.J.: Networks, 2nd edn. Oxford University Press, Oxford (2018)
Newman, M.E.J., Clauset, A.: Structure and inference in annotated networks. Nat. Commun. **7**, 11863 (2016)
Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E **69**, 026113 (2004)
Newman, M.E.J., Reinert, G.: Estimating the number of communities in a network. Phys. Rev. Lett. **117**, 078301 (2016)
Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. J. Comput. Phys. **79**, 12–49 (1988)
Osting, B., Reeb, T.: Consistency of Dirichlet partitions. SIAM J. Math. Anal. **49**, 4251–4274 (2017)
Otter, N., Porter, M.A., Tillmann, U., Grindrod, P., Harrington, H.A.: A roadmap for the computation of persistent homology. EPJ Data Sci. **6**, 17 (2017)

Papadopoulos, L., Porter, M.A., Daniels, K.E., Bassett, D.S.: Network analysis of particles and grains. J. Complex Netw. **6**, 485–565 (2018)
Pastor-Satorras, R., Castellano, C., Van Mieghem, P., Vespignani, A.: Epidemic processes in complex networks. Rev. Mod. Phys. **87**, 925–979 (2015)
Peel, L., Larremore, D.B., Clauset, A.: The ground truth about metadata and community detection in networks. Sci. Adv. **3**, e1602548 (2017)
Peixoto, T.P.: Bayesian stochastic blockmodeling. arXiv:1705.10225. (2018). Chapter In: Doreian, P., Batagelj, V., Ferligoj, A. (eds.) Advances in Network Clustering and Blockmodeling, Wiley, New York City. [forthcoming]
Peixoto, T.P.: Hierarchical block structures and high-resolution model selection in large networks. Phys. Rev. X **4**, 011047 (2014)
Peixoto, T.P.: Inferring the mesoscale structure of layered, edge-valued, and time-varying networks. Phys. Rev. E **92**, 042807 (2015)
Peixoto, T.P.: Model selection and hypothesis testing for large-scale network models with overlapping groups. Phys. Rev. X **5**, 011033 (2015)
Porter, M.A., Mucha, P.J., Newman, M.E.J., Warmbrand, C.M.: A network analysis of committees in the U.S. House of Representatives. Proc. Natl. Acad. Sci. U.S.A. **102**, 7057–7062 (2005)
Porter, M.A., Onnela, J.-P., Mucha, P.J.: Communities in networks. Notices Am. Math. Soc. **56**, 1082–1097, 1164–1166 (2009)
Riolo, M.A., Cantwell, G.T., Reinert, G., Newman, M.E.J.: Efficient method for estimating the number of communities in a network. Phys. Rev. E **96**, 032310 (2017)
Rombach, P., Porter, M.A., Fowler, J.H., Mucha, P.J.: Core–periphery structure in networks (revisited). SIAM Rev. **59**, 619–646 (2017)
Rossi, R.A., Ahmed, N.K.: Role discovery in networks. IEEE Trans. Knowl. Data Eng. **27**, 1112–1131 (2015)
Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation noise removal algorithm. Physica D **60**, 259–268 (1992)
Smith, C.S.: Metal Interfaces. Grain shapes and other metallurgical applications of topology, pp. 65–113. American Society for Metals, Cleveland (1952)
Snijders, T.A.B., Nowicki, K.: Estimation and prediction for stochastic blockmodels for graphs with latent block structure. J. Classif. **14**, 75–100 (1997)
Traud, A.L., Kelsic, E.D., Mucha, P.J., Porter, M.A.: Comparing community structure to characteristics in online collegiate social networks. SIAM Rev. **53**, 526–543 (2011)
Traud, A.L., Mucha, P.J., Porter, M.A.: Social structure of Facebook networks. Physica A **391**, 4165–4180 (2012)
Trillos, N.G., Slepčev, D.: A variational approach to the consistency of spectral clustering. App. Comp. Harmonic Anal. **45**, 239–281 (2018)
Trillos, N.G., Slepčev, D., Von Brecht, J., Laurent, T., Bresson, X.: Consistency of Cheeger and ratio graph cuts. J. Mach. Learn. Res. **17**, 1–46 (2016)
Tudisco, F., Mercado, P., Hein, M.: Community detection in networks via nonlinear modularity eigenvectors. SIAM J. Appl. Math. **78**, 2393–2419 (2018)
van Gennip, Y., Bertozzi, A.L.: Γ-convergence of graph Ginzburg–Landau functionals. Adv. Differ. Equ. **17**, 1115–1180 (2012)
van Gennip, Y., Guillen, N., Osting, B., Bertozzi, A.L.: Mean curvature, threshold dynamics, and phase field theory on finite graphs. Milan J. Math. **82**, 3–65 (2014)
Vedaldi, A., Fulkerson, B.: VLFeat: an open and portable library of computer vision algorithms. Available at http://www.vlfeat.org. (2008). Accessed 30 Mar 2019
Veldt, N., Gleich, D.F., Wirth, A.: A correlation clustering framework for community detection. In: Proceedings of the 2018 World Wide Web Conference, WWW '18, Republic and Canton of Geneva, Switzerland, 2018, International World Wide Web Conferences Steering Committee, pp. 439–448
von Luxborg, U.: A tutorial on spectral clustering. Stat. Comput. **17**, 395–416 (2007)
Weaire, D., Kermode, J.P.: Computer simulation of a two-dimensional soap froth: I. Method and motivation. Phil. Mag. B **48**, 245–259 (1983)
Welk, M., Weickert, J., Gilboa, G.: A discrete theory and efficient algorithms for forward-and-backward diffusion filtering. J. Math. Imaging Vis. **60**, 1399–1426 (2018)
Zachary, W.W.: An information flow model for conflict and fission in small groups. J. Anthropol. Res. **33**, 452–473 (1977)

Zhu, W., Chayes, V., Tiard, A., Sanchez, S., Dahlberg, D., Bertozzi, A.L., Osher, S., Zosso, D., Kuang, D.: Unsupervised classification in hyperspectral imagery with nonlocal total variation and primal-dual hybrid gradient algorithm. IEEE Trans. Geosci. Remote Sens. **55**, 2786–2798 (2017)