

Efficient uncertainty minimization for fuzzy spectral clustering

Brian S. White* and David Shalloway†

Biophysics Program, Department of Molecular Biology and Genetics, Cornell University, Ithaca, New York 14853, USA

(Received 23 November 2008; revised manuscript received 8 July 2009; published 24 November 2009)

Spectral clustering uses the global information embedded in eigenvectors of an inter-item similarity matrix to correctly identify clusters of irregular shape, an ability lacking in commonly used approaches such as k -means and agglomerative clustering. However, traditional spectral clustering partitions items into hard clusters, and the ability to instead generate fuzzy item assignments would be advantageous for the growing class of domains in which cluster overlap and uncertainty are important. Korenblum and Shalloway [Phys. Rev. E **67**, 056704 (2003)] extended spectral clustering to fuzzy clustering by introducing the principle of uncertainty minimization. However, this posed a challenging nonconvex global optimization problem that they solved by a brute-force technique unlikely to scale to data sets having more than $O(10^2)$ items. Here we develop a method for solving the minimization problem, which can handle data sets at least two orders of magnitude larger. In doing so, we elucidate the underlying structure of uncertainty minimization using multiple geometric representations. This enables us to show how fuzzy spectral clustering using uncertainty minimization is related to and generalizes clustering motivated by perturbative analysis of almost-block-diagonal matrices. Uncertainty minimization can be applied to a wide variety of existing hard spectral clustering approaches, thus transforming them to fuzzy methods.

DOI: [10.1103/PhysRevE.80.056705](https://doi.org/10.1103/PhysRevE.80.056705)

PACS number(s): 02.70.Hm, 02.50.Fz, 89.75.Kd

I. INTRODUCTION

Coarse-graining data items i ($1 \leq i \leq N$) into clusters α ($1 \leq \alpha \leq m$) is important for large-scale data analysis [1–3]. For example, clustering genes according to their microarray expression profiles allows biologists to subsequently infer potential *cis*-regulatory elements from sequence commonalities within the clusters [4]. Clustering typically proceeds from a symmetric $N \times N$ similarity matrix S , where the non-negative off-diagonal element S_{ij} provides an inverse indicator of the “distance” d_{ij} between items i and j . The primary input (e.g., the alignment scores from sequence comparisons or edge weights of a graph) may directly define the S_{ij} . Alternatively, the data may consist of N_D properties for each item that can be embedded in a dataspace. For example, in microarray analysis each gene is an item, and its properties are its N_D expression levels under N_D different conditions. In that case, the d_{ij} are derived from the (not-necessarily Euclidean) distances between the items in the dataspace.

Spectral clustering methods ([5,6] for history and review) analyze the eigensystem of a transition (or Laplacian) matrix Γ , which is derived from S . Since the eigensystem depends globally on the entire data set, spectral methods have a perspective lacking in commonly used methods such as k -means and agglomerative clustering [2], which directly analyze the S_{ij} . Their dependence on pairwise similarities leads them to impose characteristic cluster shapes; e.g., k -means and complete-linkage clustering generate convex clusters while single-linkage clustering generates unbalanced and straggly clusters [2]. These shapes may not reflect the true geometries of the problem, such as the irregular boundaries of a subject within an image [7]. The ability of spectral methods to generate arbitrary cluster shapes lets them outperform k -means

across several benchmarks [8–10]. As we will see, they can also determine the optimal number of clusters automatically.

Γ typically satisfies [11,12]

$$\Gamma = \Gamma^S \cdot D_\pi^{-1}, \quad (1a)$$

$$\Gamma_{ij}^S = -S_{ij} \quad (i \neq j), \quad (1b)$$

$$\Gamma_{ii}^S = \sum_{j \neq i} S_{ji}, \quad (1c)$$

$$\mathbf{1} \cdot \Gamma = 0, \quad (1d)$$

where D_π is a diagonal normalizing matrix with non-negative elements satisfying $\text{Tr}(D_\pi) = 1$, $\mathbf{1}$ is the item-space vector having all components equal to one, and \cdot denotes the normalized item-space inner product:

$$\mathbf{x} \cdot \mathbf{y} \equiv N^{-1} \sum_{i=1}^N x_i y_i.$$

These conditions emerge when spectral clustering methods are used to approximate “min-cut” graph partitioning solutions [13,14] or when they are motivated by discrete-time [15–19] or continuous-time [10] dynamical models. [The first two motivations lead to analysis of the Markov matrix $T \equiv I - \Gamma$ (where I is the identity matrix), which satisfies $\mathbf{1} \cdot T = \mathbf{1}$ rather than Eq. (1d). But since the eigenvectors of T and Γ are identical and the eigenvalues are simply related, the same analysis applies with inconsequential changes.]

Equations (1) imply

$$\gamma_0 = 0, \quad (2a)$$

$$\psi_0^R \equiv N \boldsymbol{\pi}, \quad (2b)$$

$$\psi_0^L = \mathbf{1}, \quad (2c)$$

*bsw27@cornell.edu

†dis2@cornell.edu

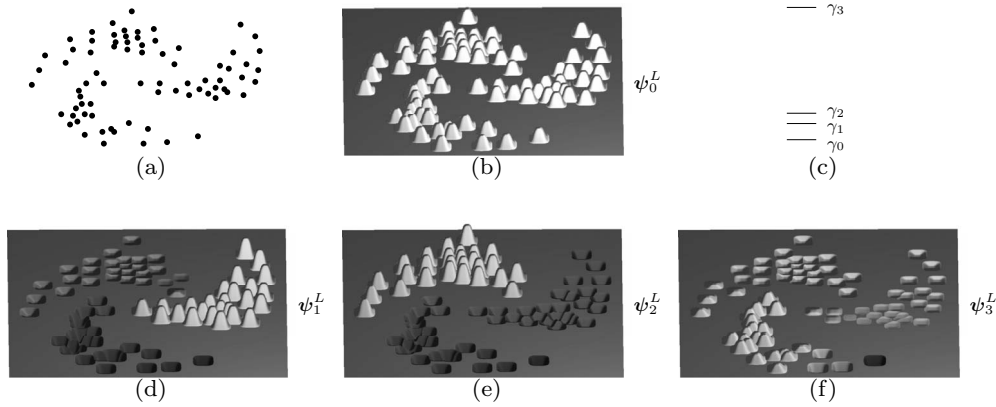


FIG. 1. The “spiral” clustering problem and its eigensystem. (a) The two-dimensional embedding of the spiral data set in dataspace. [(b), (d), (e), (f)] The (positive or negative) heights of the cones indicate the values of the clustering eigenvectors ψ_0^L , ψ_1^L , and ψ_2^L , and of the first nonclustering eigenvector ψ_3^L of the data set’s Γ . (c) The corresponding eigenvalues. Unless otherwise noted, figures are based on the Γ matrix defined by Korenblum and Shalloway [10] [see Eqs. (24)].

$$\psi_n^L = D_\pi^{-1} \cdot \psi_n^R, \tag{2d}$$

where ψ_n^L and ψ_n^R are the biorthogonal left and right eigenvectors of Γ , which we normalize such that $\psi_m^L \cdot \psi_n^R = \delta_{mn}$ and $\psi_n^L \cdot \psi_n^L = 1$, and π is the right equilibrium probability vector satisfying $\sum_i \pi_i = 1$. It follows that $(D_\pi^{-1})_{ii} = \pi_i^{-1}$. Eqs. (1) also imply that $\Gamma_{ij} \pi_j = \Gamma_{ji} \pi_i$ (i.e., that detailed balance holds), which ensures the reality and non-negativity of the eigenvalues [20].

Spectral methods begin by embedding each item i into the *low-frequency (or clustering) subspace* \mathbb{R}^m using as coordinates the m low-frequency vector components of $\tilde{\psi}^L(i) \equiv [\psi_0^L(i), \psi_1^L(i), \dots, \psi_{m-1}^L(i)]$ [21]. These are then used to identify m clusters [22]. Clustering (i.e., spatial coarse graining) is possible only if there is a gap in the distribution of the similarities S_{ij} [23].

The dynamical interpretation of spectral clustering provides a way to find a gap if it exists: each cluster is viewed as a metastable state of a diffusive relaxation process governed by Γ [24]

$$\frac{dp(t)}{dt} = -\Gamma \cdot p(t), \tag{3}$$

where $p(t)$ is a time-dependent probability vector over the discrete space of items [i.e., $p_i(t)$ is the probability of occupation of item i at time t], $-\Gamma_{ij}$ is the stochastic transition rate from item j to i , and Eqs. (1c) and (1d) ensure that probability is conserved. Because of the inverse relationship between eigenvector “wavelength” and eigenvalue, a spatial-scale gap in the distribution of the S_{ij} will appear as a time-scale *spectral gap*:

$$0 = \gamma_0 < \gamma_1 < \dots < \gamma_{m-1} \ll \gamma_m. \tag{4}$$

The gap between γ_{m-1} and γ_m indicates the existence of m clusters. When a spectral gap exists, the long-wavelength clustering eigenvectors $\psi_{n < m}^L$ will contain the information needed for clustering [25].

For example, Fig. 1 illustrates the $m=3$ “spiral” clustering

problem posed by 77 items embedded in a two-dimensional dataspace and the corresponding eigensystem of the Γ matrix of Ref. [10] [see Eqs. (24) below]. Panel (a) shows the spatial locations of the items, and it is subjectively evident that there are three interlocking clusters. Correspondingly, as predicted by Eq. (4), there is a gap between γ_2 and γ_3 [panel (c)]. The clustering eigenvectors ψ_1^L [panel (d)] and ψ_2^L [panel (e)] vary significantly only at the cluster boundaries and follow their distorted shapes. Thus, the shapes of the clusters defined using these eigenvectors will not be artificially restricted. In contrast, the nonclustering eigenvectors such as ψ_3^L [panel (f)] have large variations within clusters and thus are not used in the clustering analysis.

It remains to define the clustering from the clustering eigenvectors. Hard spectral clustering approaches do so simply by applying nonspectral methods such as k -means within the clustering subspace [9]. However, there are problems where hard partitioning is neither necessary nor ideal, for example, the separation of cell subpopulations by fluorescence-activated cell sorting (FACS) [26], automated biological database curation [27], complex network analysis [28], and gene expression analysis [29]. Such problems require *fuzzy clustering* that can represent uncertainty and overlapping clusters.

Nonspectral fuzzy clustering methods have already been applied to such problems [28,29], but spectral fuzzy methods could be advantageous because of their added ability to cope with irregular cluster boundaries (e.g., those within FACS dataspaces [26]). Moreover, fuzziness could provide further benefit in areas where hard spectral clustering has already been applied. For example, Paccanaro *et al.* [27] have used hard spectral clustering to faithfully reproduce many of the superfamily classifications from a subset of the structural classification of proteins (SCOP) database [30]; a fuzzy spectral approach would add the ability to assess the certainty of such classifications.

Formally, fuzzy clusterings are described by *assignment vectors* $w_\alpha \equiv [w_\alpha(1), w_\alpha(2), \dots, w_\alpha(N)]$, where $w_\alpha(i)$ is the probability that item i is a member of cluster α and therefore must satisfy the probabilistic constraints

$$w_\alpha(i) \geq 0 \quad (\forall \alpha, i), \tag{5a}$$

$$\sum_{\alpha} w_{\alpha}(i) = 1 \quad (\forall i). \quad (5b)$$

To define these in a spectral context, following Ref. [10] we use the low-frequency clustering eigenvectors as a linear basis for the w_{α} [31]:

$$w_{\alpha} = \sum_{n=0}^{m-1} M_{\alpha n} \psi_n^L \equiv \vec{M}_{\alpha} \circ \vec{\psi}^L, \quad (6)$$

where the $\vec{M}_{\alpha} \equiv [M_{\alpha 0}, M_{\alpha 1}, \dots, M_{\alpha(m-1)}]$ are m -vectors, $\vec{\psi}^L \equiv [\psi_0^L, \psi_1^L, \dots, \psi_{m-1}^L]$, and \circ denotes the inner product over the low-frequency subspace:

$$\vec{x} \circ \vec{y} = \sum_{n=0}^{m-1} x_n y_n.$$

Equation (6) transforms the clustering problem to that of finding the “best” \vec{M}_{α} subject to Eqs. (5). Korenblum and Shalloway [10] proposed that this was the one that minimized overlap between assignment vectors: since the w_{α} are non-negative and composed of only the long wavelength $\psi_{n < m}^L$, they will inevitably overlap each other and thus will give *uncertain* (i.e., fuzzy) item-to-cluster assignments. This uncertainty is minimized when the clusters’ self-overlap is maximized. The self-overlap of cluster α can be quantified by the fractional cluster certainty $\bar{Y}_{\alpha}(M)$ ($1 \leq \alpha \leq m$) [10],

$$\bar{Y}_{\alpha}(M) \equiv \frac{\langle w_{\alpha} | w_{\alpha} \rangle}{\langle \mathbf{1} | w_{\alpha} \rangle} \quad [N^{-1} \leq \bar{Y}_{\alpha}(M) \leq 1], \quad (7)$$

where M represents the components of all the \vec{M}_{α} and bracket notation denotes the equilibrium-weighted inner product [32]

$$\langle x | y \rangle \equiv x \cdot D_{\pi} \cdot y. \quad (8)$$

$\bar{Y}_{\alpha}(M) = 1$ when the cluster α is completely certain, i.e., $w_{\alpha}(i) = 0$ or 1; the total certainty is the product of the $\bar{Y}_{\alpha}(M)$ for all the clusters. Thus, the optimal M is determined by *uncertainty minimization* of the *overall uncertainty objective function*,

$$\Phi(M) \equiv - \sum_{\alpha} \log \bar{Y}_{\alpha}(M), \quad (9)$$

subject to the constraints of Eqs. (5). Korenblum and Shalloway showed that this procedure provided good fuzzy clusterings of a number of difficult problems. However, they solved the resulting challenging constrained nonconvex uncertainty minimization problem using a “brute-force” solver whose $O(m^2 N^{m+1})$ computational complexity limited its application to modest-sized problems ($N = 200$) and precluded application to the larger problems [e.g., $N \sim O(10^4)$] that emerge in areas such as gene microarray analysis [33].

A closely related approach was independently developed by Weber *et al.* [18]. They also used Eq. (6) but, instead of using uncertainty minimization, determined the M through an efficient but approximate method motivated by perturbative analysis of almost-block-diagonal matrices [34]. Their Perron cluster analysis (PCCA) defined the w_{α} as

“membership functions” that only approximate the probabilistic constraints of Eqs. (5). In PCCA the M are determined algorithmically rather than by objective function optimization, and clusterings for different values of m are accepted if the resultant approximation is regarded (by subjective criteria) to be adequate. While approximate, this method had the advantage of being computationally simpler than the initial uncertainty minimization algorithm of Korenblum and Shalloway [10].

Thus until now, practical exact fuzzy spectral data clustering has remained elusive. To resolve this problem, here we develop an efficient method for uncertainty minimization and show that it is generally applicable to any spectral clustering method satisfying Eqs. (1), including popular asymmetric approaches based on random walks over graphs [15–19]. Thus, we imbue a wide range of hard spectral clustering methods with the ability to compute fuzzy cluster assignments and, thereby, uncertainty and cluster overlap. In the process, we show that there are multiple geometric interpretations of the uncertainty minimization problem that can be used to illuminate its structure. Through these we relate uncertainty minimization to PCCA and extend the previously reported conditions under which the PCCA approximation is applicable.

II. COMPUTATIONAL THEORY

Minimization of $\Phi(M)$ subject to the constraints of Eqs. (5) poses a global nonlinear optimization problem in the m^2 degrees of freedom of M . To solve this it is convenient to re-express Eq. (9) explicitly in terms of the \vec{M}_{α} as

$$\Phi(M) \equiv - \sum_{\alpha} \log \bar{Y}_{\alpha}(M) = - \sum_{\alpha} \log \frac{\vec{M}_{\alpha} \circ \vec{M}_{\alpha}}{\vec{M}_{\alpha} \circ \hat{\epsilon}_0}, \quad (10)$$

where $\hat{\epsilon}_0$ is the m -vector $(1, 0, \dots, 0)$, and we have used $\langle w_{\alpha} | w_{\alpha} \rangle = \vec{M}_{\alpha} \circ \vec{M}_{\alpha}$ and $\langle \mathbf{1} | w_{\alpha} \rangle = \vec{M}_{\alpha} \circ \hat{\epsilon}_0$, which follow from Eqs. (2c), (2d), and (6) and the biorthogonality of the eigenvectors. Similarly, we re-express Eqs. (5) in terms of the \vec{M}_{α} :

$$w_{\alpha}(i) = \vec{M}_{\alpha} \circ \vec{\psi}^L(i) \geq 0 \quad (\forall \alpha, i), \quad (11a)$$

$$\sum_{\alpha} \vec{M}_{\alpha} = \hat{\epsilon}_0. \quad (11b)$$

Because $\Phi(M)$ is invariant under permutations of the indices associated with the clusters, its global minimum will have an $m!$ -fold permutation degeneracy.

We now describe two geometric representations that illuminate the problem (Sec. II A) and then show how to solve it in three steps: (1) precondition Γ to avoid numerical noise that can obfuscate spectral gaps when low-lying eigenvalues are nearly degenerate, to improve numerical efficiency, and to remove outliers (Appendix A), (2) find a zeroth-order solution (Sec. II B), and (3) iteratively refine using linear programming with a subset of the inequality constraints to determine the solution to the desired accuracy (Sec. II C). Since the procedure explicitly uses only the ψ_n^L , for notational convenience we subsequently denote them simply as the ψ_n .

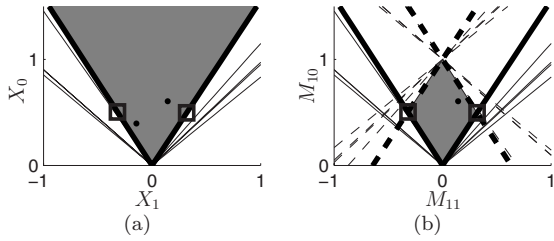


FIG. 2. Symmetric and asymmetric M representations of the $m=2$ “crescentric” problem [2,10]. (a) *Symmetric M representation*: the diagonal lines indicate the boundaries formed by the inequality constraints. The two bold lines forming the narrowest cone (shaded) define the feasible region in $\mathbb{R}^m=\mathbb{R}^2$. \vec{M}_1 and \vec{M}_2 are represented by dots. They are not independent since they are further constrained by the equality constraints of Eq. (11b). The global minimum of the uncertainty objective function $\Phi(M)$ corresponds to the dots being located at the positions indicated by small squares, and the invariance under particle exchange corresponds to the permutation degeneracy discussed in the text. (b) *Asymmetric M representation*: the solid lines indicate the boundaries of the homogeneous inequality constraints acting on the free particle $\vec{M}^{\text{free}}=\vec{M}_1$. The dashed lines indicate the boundaries of the inhomogeneous inequality constraints that derive from the slave particle \vec{M}_2 . Two of these (bold-dashed) lines cap the cone formed by the relevant homogeneous constraint boundaries (bold) to define a closed feasible polytope in $\mathbb{R}^{m(m-1)}=\mathbb{R}^2$. In this representation the single dot represents all $m(m-1)=2$ components of \vec{M}^{free} . $\Phi(M)$ is minimized at either of the two permutation-degenerate solutions (small squares).

A. Geometric representation of uncertainty minimization

1. Symmetric M representation

Each \vec{M}_α may be regarded as the coordinates of a particle α in \mathbb{R}^m with axes labeled $X_0, X_1, \dots, X_{(m-1)}$. Equation (11a) implies that the same N inequality constraints act on each particle; thus they restrict each one to the same half-space in \mathbb{R}^m that is bounded by a hypersurface passing through the origin and normal to $\vec{\psi}(i)$. The intersection of these half spaces determines the feasible region as a convex polyhedral cone in the upper half of \mathbb{R}^m . Only a subset of the inequality constraints will actually bound the feasible region since their satisfaction will automatically guarantee satisfaction of the other constraints. And, as proved in Appendix B 1, each particle lies on an edge of the polyhedral cone (i.e., is constrained by $m-1$ active inequality constraints) at every local minimizer of $\Phi(M)$.

An example of this *symmetric M representation* for an $m=2$ problem (based on the “crescentric” bivariate data set of Ref. [2]) is shown in Fig. 2(a). (It is only in the $m=2$ case that a simple graphical representation is possible; nonetheless it is useful for illustrating structural properties that also hold when $m>2$.) In this case, the feasible region is bounded by only two lines corresponding to $\vec{X} \circ \vec{\psi}(i_<)=0$ and $\vec{X} \circ \vec{\psi}(i_>)=0$, where $i_<$ and $i_>$ are the minimizer and maximizer of $\psi_1(i)$, respectively. The global minimum of Φ corresponds to the unique (up to the permutation degeneracy) situation where each particle lies on the feasible region boundary while the equality constraints of Eq. (11b) are simultaneously

satisfied. In Fig. 2(a), this is when the points are located at the two squares on the boundary. The two ways of associating the particles with the squares corresponds to the twofold permutation degeneracy of the solution.

2. Asymmetric M representation

The m particles in the symmetric M representation are not independent because of the equality constraints [Eq. (11b)]. We use these in the *asymmetric M representation* to explicitly eliminate the degrees of freedom of one *slave* particle that, without loss of generality, we take to be \vec{M}_m :

$$\vec{M}_m = \hat{\epsilon}_0 - \sum_{\alpha \neq m} \vec{M}_\alpha. \tag{12}$$

The homogeneous inequality constraints on the slave, $\vec{M}_m \circ \vec{\psi}(i) \geq 0 (\forall i)$, transform into inhomogeneous inequality constraints that couple the remaining $m-1$ *free* particles:

$$\sum_{\alpha \neq m} \vec{M}_\alpha \circ \vec{\psi}(i) \leq 1. \tag{13}$$

We consolidate the $m(m-1)$ degrees of freedom of the free particles into the supervector \vec{M}^{free} having components $(\vec{M}_1, \vec{M}_2, \dots, \vec{M}_{m-1})$ in $\mathbb{R}^{m(m-1)}$. Optimization then proceeds in $\mathbb{R}^{m(m-1)}$ with the \vec{M}^{free} restricted by $(m-1)N$ homogeneous inequality constraints from Eq. (11a) with $\alpha < m$ and N inhomogeneous inequality constraints from Eq. (13). The combination of homogeneous and inhomogeneous inequality constraints forms a closed convex polytope that bounds the feasible region. Each local minimum of $\Phi(M)$ (and thus, the global minimum) lies at a vertex of this polytope [10].

An example of the asymmetric M representation for $m=2$ is shown in Fig. 2(b). In this case there are four bounding constraints: two homogeneous inequality constraints having boundaries passing through the origin and two inhomogeneous inequality constraints (from the slave cluster) with boundaries intersecting at $\hat{\epsilon}_0$ [35]. Φ is infinite at the polytope vertices at the origin and $\hat{\epsilon}_0$. The two other vertices correspond to index-permutation-equivalent global minima.

The minimization problem can be visualized and easily solved in this manner only for $m=2$: as m increases the number of polytope vertices, and hence the number of local minima, grows rapidly, and the global minimization problem becomes difficult. Korenblum and Shalloway [10] solved this by an expensive random exploration of the vertices.

B. Cluster representatives and the approximate global solution

1. Representatives

We take a different approach: rather than trying to identify the minimizing vertex directly, we exploit the fact that the m^2 components of M can be determined by the m^2 low-frequency components of an appropriately chosen subset $\mathcal{R} = \{r_1, r_2, \dots, r_m\}$ of m items, which we call *representatives*. To make this explicit we write a matrix analog of Eq. (6) over \mathcal{R} as

$$W^{\mathcal{R}} = M \circ \Psi^{\mathcal{R}}, \quad (14)$$

where

$$W_{\alpha\beta}^{\mathcal{R}} \equiv w_{\alpha}(r_{\beta}) \quad (1 \leq \alpha, \beta \leq m), \quad (15a)$$

$$\Psi_{n\alpha}^{\mathcal{R}} \equiv \psi_n(r_{\alpha}) \quad \begin{cases} (1 \leq \alpha \leq m) \\ (0 \leq n < m) \end{cases}, \quad (15b)$$

and M is the matrix having the \vec{M}_{α} as its rows. According to Eq. (11b), M must satisfy

$$\sum_{\alpha} M_{\alpha n} = \delta_{n0}. \quad (16)$$

As shown in Appendix B 2, there always exists at least one subset \mathcal{R} such that $\Psi^{\mathcal{R}}$ is invertible. With such a subset we can solve Eq. (14) for M :

$$M = W^{\mathcal{R}} \bullet (\Psi^{\mathcal{R}})^{-1}, \quad (17)$$

where \bullet denotes the inner product over the cluster index α .

The usefulness of Eq. (17) may be questioned since *a priori* we do not know any $W^{\mathcal{R}}$ exactly. However, any data set amenable to clustering will have at least one item per cluster that will be strongly assigned in the clustering solution; we call such items *candidate representatives*. If we could select a set of representatives \mathcal{R}_c containing one candidate representative from each cluster, we could use our approximate foreknowledge of their assignment values at the solution, $W^{\mathcal{R}_c*}$, to approximate M at the solution, M^* , via Eq. (17).

For example, if item i_{α} were a candidate representative for cluster α , its assignment in the clustering solution would be [36]

$$w_{\beta}^*(i_{\alpha}) \approx \delta_{\alpha\beta}. \quad (18)$$

By choosing $r_{\alpha} = i_{\alpha}$ and making similar choices for the other clusters, we would get

$$W^{\mathcal{R}_c*} \approx I.$$

This zeroth-order estimate could be used to approximately solve Eq. (17) for M^* :

$$M^* \equiv W^{\mathcal{R}_c*} \bullet (\Psi^{\mathcal{R}_c})^{-1} \quad (19a)$$

$$\approx I \bullet (\Psi^{\mathcal{R}_c})^{-1} = (\Psi^{\mathcal{R}_c})^{-1} \equiv M^0. \quad (19b)$$

In agreement with Eq. (16), M^0 would satisfy [37]

$$\sum_{\alpha} M_{\alpha n}^0 = \delta_{n0}. \quad (20)$$

Knowing M^0 would allow us to define zeroth-order estimates w_{α}^0 for all the items via Eq. (6) with $\vec{M}_{\alpha} = \vec{M}_{\alpha}^0$, where the \vec{M}_{α}^0 are the rows of M^0 :

$$w_{\alpha}^0 = \vec{M}_{\alpha}^0 \circ \vec{\psi}. \quad (21)$$

However, the w_{α}^0 would not necessarily satisfy the inequality constraints of Eq. (5a). If they did, they would solve the

optimization problem (see Sec. II C 1). If they did not, they would provide a starting point for refining the solution as discussed in Sec. II C.

2. Finding \mathcal{R}_c

Equation (19b) implies that we only need to find the representatives to determine M^0 . This is trivial when $m=2$: the two active inequality constraints [identified by either pair of intersecting bold and bold-dashed lines in Fig. 2(b)] come from the extremal items r_1 and r_2 of ψ_1 , i.e., the minimizer and maximizer of $\psi_1(i)$. Thus, at the solution $w_1^*(r_2)=0$ and $w_2^*(r_1)=0$, and the equality constraints imply that $w_1^*(r_1)=1$ and $w_2^*(r_2)=1$: r_1 and r_2 not only generate the active constraints but are also the representatives, which in this case are perfectly assigned in the solution.

The situation is more complicated when $m>2$. The representatives: (1) may not be maxima and minima of the eigenvectors, (2) may not be the items associated with the active constraints, and (3) may not be perfectly assigned at the solution. Nonetheless, as discussed above, they will satisfy $w_{\alpha}(r_{\beta}) \approx \delta_{\alpha\beta}$, and we will use this property to identify them.

We show how this is done using the $m=3$ spiral problem as an example (Fig. 3). Its three low-frequency clustering eigenvectors are shown in panel (b), and the representatives that we would like to find are identified by circles, triangles, and squares. To find \mathcal{R}_c we imagine that we know M^* and the corresponding assignment vectors w_{α}^* so that we can map the items into \mathbb{R}^m at the points specified by the 3-vectors $\vec{w}^*(i) \equiv [w_1^*(i), w_2^*(i), w_3^*(i)]$ in panel (c) [38]. Because the $\vec{w}^*(i)$ satisfy the probabilistic equality constraints, these points lie in the two-dimensional plane that is normal to the vector (1,1,1) and at distance $1/\sqrt{3}$ from the origin. Moreover, because they satisfy the probabilistic inequality constraints, they lie within an equilateral triangle in this plane. (We use “within” to include points that lie on the boundary.) This provides barycentric coordinates [39] in which the three vertices of the triangle correspond to the cluster assignments (1,0,0), (0,1,0) and (0,0,1); we will call these the $\alpha=1, 2$, and 3 vertices, respectively. The three components of $\vec{w}^*(i)$ are given by the three distances of point i from the three sides of the triangle. Thus, if point i lies on the side of the triangle opposing vertex α , the inequality constraint $w_{\alpha}(i) \geq 0$ is active. We call this the \vec{w}^{Δ} representation [panel (c)]. Although it may not be evident in the figure, consistent with the even distribution of active inequality constraints between the clusters (Appendix B 1), each side of the triangle intersects exactly two items.

The candidate representatives are the items that are close to the three vertices, and we want to choose one from the vicinity of each vertex to compose \mathcal{R}_c . We can do this by choosing the three items that (when taken as vertices) define the triangle of largest area. It is easy to show that the triangular area defined by any subset \mathcal{R} of three items located at their solution positions is $|W^{\mathcal{R}}|/(2\sqrt{3})$. Thus, we can find a good \mathcal{R}_c by finding the subset \mathcal{R} that, via Eq. (14), maximizes $|W^{\mathcal{R}}|$.

Since we do not actually know M^* or the $\vec{w}^*(i)$, it is not obvious how to proceed. However, Eq. (14) implies that

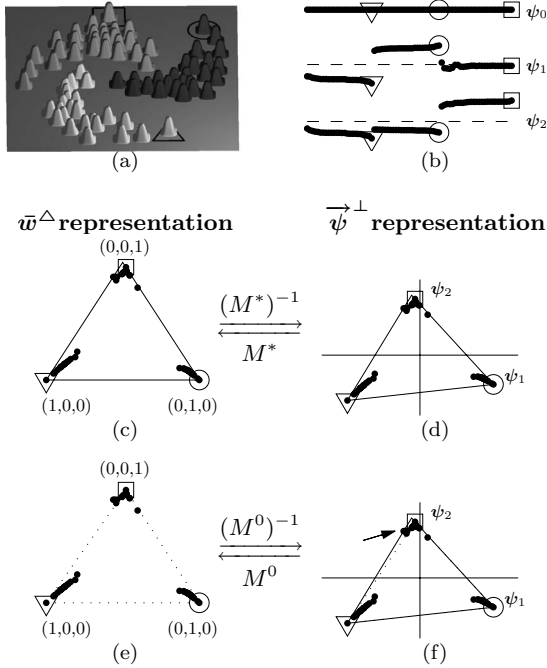


FIG. 3. \bar{w}^Δ and $\vec{\psi}^\perp$ representations of the spiral problem. The items are represented (a) in the dataspace as peaks with magnitudes determined by their maximal assignment, (b) in the clustering (low-frequency) eigenvector representation, (c) and (e) in the barycentric coordinates of the \bar{w}^Δ representation, or (d) and (f) in the $\vec{\psi}^\perp$ representation. Panels (c) and (d) correspond to the refined solution M^* , while (e) and (f) correspond to the zeroth-order solution M^0 . The solid and dotted triangles denote the M^* and M^0 feasible region boundaries. [The solid triangle is superimposed in panel (f) to show how the triangle expands slightly in the $\vec{\psi}^\perp$ representation during refinement. The arrow indicates the item that becomes an active constraint in M^* .] The left and right arrows connecting the representations are reminders that M determines the positions of the items in the \bar{w}^Δ representation and of the triangle vertices in the $\vec{\psi}^\perp$ representation. [Although it may not be evident in the figure, the points in panels (c) and (e) and the top vertex in panels (d) and (f) are at slightly different positions.] The different shades of gray in panel (a) denote the hard clustering obtained by quantizing the fuzzy clustering, while the height of a cone shows the strength of the probabilistic assignment of the item to the cluster. The ordering of items in panel (b) was chosen *post facto* to separate the clusters. The dashed lines in this panel are at $\psi_n(i)=0$. The representatives for clusters 1, 2, or 3 are enclosed within triangles, circles, or squares, respectively.

$$|W^{\mathcal{R}^*}| = |M^*| |\Psi^{\mathcal{R}}|, \quad (22)$$

so, since M^* is fixed (though unknown), selecting the \mathcal{R} that maximizes $|W^{\mathcal{R}^*}|$ is equivalent to selecting the \mathcal{R} that maximizes $|\Psi^{\mathcal{R}}|$. This is straightforward because $\Psi^{\mathcal{R}}$ does not depend on M . Formally, maximizing $|\Psi^{\mathcal{R}}|$ is a combinatoric problem that could be solved by comparing the determinants for all subsets \mathcal{R} . However, this would be exponentially expensive in N . Instead we use an efficient greedy algorithm that selects the representatives solely from the subset of candidate representatives. This may not exactly maximize the

determinant but will be adequate to determine an \mathcal{R}_c that gives, via Eq. (19b), an M^0 that can be used as a starting point for refinement.

We leave the details of the greedy algorithm to Appendix C, but it is useful to establish its geometric framework here, continuing to use the spiral problem as an example: we first plot each item in the two-dimensional $\vec{\psi}^\perp$ representation using the 2-vector $\vec{\psi}^\perp(i)=[\psi_1(i), \psi_2(i)]$ [panels (d) and (f)]. [No information is lost in the projection from the low-frequency subspace since $\psi_0(i)=1(\forall i)$.] These vectors are independent of M [40]; rather, in this representation M determines the position of the inequality constraint bounding triangle. As explained in Appendix B 4, the $\vec{\psi}^\perp$ coordinates of the three bounding triangle vertices are the columns of the bottom two rows of M^{-1} . When $M=M^*$ [panel (d)], the vertices may not coincide with any items, but all the items will lie within the bounding triangle. When $M=M^0$ [panel (f)], the vertices of the triangle coincide with the representatives, but some items may violate the inequality constraints and lie outside the triangle. (Four items in the upper left corner are outside the triangle in this example.) The greedy algorithm operates within the $\vec{\psi}^\perp$ representation to identify \mathcal{R}_c .

The approach generalizes easily to higher m : the $\bar{w}^*(i)$ are now m -vectors. The \bar{w}^Δ representation is in an $(m-1)$ -dimensional hyperplane normal to the vector $(1,1,\dots,1)$ in \mathbb{R}^m and provides barycentric coordinates for the $\bar{w}^*(i)$. \mathcal{R}_c is comprised of the subset of m items that, when located at their solution positions in the \bar{w}^Δ representation, are the vertices of the $(m-1)$ -simplex of largest hypervolume. This hypervolume, for any subset \mathcal{R} , is proportional to $|W^{\mathcal{R}^*}|$ so, via Eq. (22), we can transform the problem of selecting \mathcal{R}_c to that of finding the m items that maximize $|\Psi^{\mathcal{R}}|$. This problem is equivalent to maximizing the hypervolume of the $\vec{\psi}^\perp$ -representation simplex having vertices at $\{\vec{\psi}^\perp(i): i \in \mathcal{R}\}$. Once \mathcal{R}_c has been identified, it is used to determine M^0 via Eq. (19b), and M^0 is used to determine w_α^0 via Eq. (21).

C. Refinement

1. Case when M^0 is the exact solution

If the w_α^0 satisfy all the inequality constraints, they provide the unique solution to the uncertainty minimization problem. To prove this, consider the $\vec{\psi}^\perp$ representation of an $m=3$ problem where the inequality constraints are satisfied. As in the spiral problem, the representatives are at the vertices of the $\vec{\psi}^\perp$ triangle determined by M^0 , and as discussed above, transforming M^0 to M moves the sides of this triangle. Moving any side inwards would leave a representative outside the triangle, thus violating an inequality constraint. And, since all points are already within the triangle (i.e., all inequality constraints are satisfied), moving any side outwards would result in that side contacting less than two points, i.e., one of the clusters would have less than the required (Appendix B 1) $m-1=2$ active inequality constraints. Therefore, in this case $M^*=M^0$ must be the unique solution. As can be inferred from the analysis of Fig. 2, M^0 is always the unique solution for $m=2$ problems.

2. Linearizing $\Phi(M)$

If the w_α^0 violate any of the inequality constraints, M^0 is not a solution but can be used as the starting point for further refinement. Since it is expected to be near M^* , we can expand the objective function in its neighborhood to first order as

$$\begin{aligned} \Phi(M) &= - \sum_{\alpha} \log \frac{\vec{M}_{\alpha} \circ \vec{M}_{\alpha}}{\vec{M}_{\alpha} \circ \hat{\epsilon}_0} \\ &\approx \Phi(M^0) + \sum_{\alpha} (\vec{M}_{\alpha} - \vec{M}_{\alpha}^0) \circ \vec{\nabla}_{\alpha} \Phi(M) \Big|_{M=M^0}, \end{aligned} \quad (23)$$

where

$$\vec{\nabla}_{\alpha} \Phi(M) \equiv \frac{\delta \Phi(M)}{\delta \vec{M}_{\alpha}} = -2 \frac{\vec{M}_{\alpha}}{|\vec{M}_{\alpha}|^2} + \frac{\hat{\epsilon}_0}{\vec{M}_{\alpha} \circ \hat{\epsilon}_0}$$

is the gradient of $\Phi(M)$ with respect to \vec{M}_{α} . Local minimization using this linear approximation and the constraints of Eqs. (11) pose a linear programming (LP) problem, which can be solved by standard methods.

A simple approach would be to: (1) apply LP using Eq. (23) and all the constraints to find an improved, constraint-satisfying solution M^1 , (2) set $M^0 \leftarrow M^1$, and (3) repeat (1) and (2) until sufficient convergence is achieved. This amounts to constrained gradient-descent local minimization. However, we do not expect to encounter the slow convergence problems that sometimes plague gradient descent because all the LP solutions, as well as the true solution, are at vertices of the feasible polytope [41]. Therefore, even the first iteration will drive the solution to a vertex, and the solution will not change at the next iteration unless the vertices are very dense on the scale set by the curvature of $\Phi(M)$. Thus, rapid convergence is expected.

3. Reducing the number of constraints included in LP

However, the cost of standard LP solvers (e.g., simplex and interior point methods) grows rapidly [$O(N_c^{1.5})$] with the number of constraints N_c , which may be large [42]. While there are mN inequality constraints, only $m(m-1)$ of these are active at M^* . These alone need to be included in the LP problem to guarantee that all the inequality constraints will be satisfied. Since we will often be interested in problems where $m \sim O(10)$ and $N \sim O(10^4)$, it would accelerate the LP solver by multiple orders of magnitude if the number of constraints provided to it were reduced to $O(m^2)$.

We do not know the active constraints *a priori* but can find them rapidly by an iterative procedure that exploits the fact that (as discussed above) at M^* exactly $m-1$ points will lie on each of the m faces of the bounding simplex in the $\vec{\psi}^{\perp}$ representation. To motivate this procedure, consider the refinement of the spiral problem (Fig. 3). The left side of the (dotted) M^0 triangle [panel (f)] must move outwards to include the four points in the upper left region that are excluded from its interior; this motion must leave the side intersecting two points. Because the objective function $\Phi(M)$

constitutes an inward ‘‘pressure’’ on the triangle, M^* will correspond to the situation where the smallest expansion that can accomplish this is used. Consequently, the left side will pivot outwards about the lower left corner until it intersects the item identified by the arrow. Each side of the resulting M^* triangle [panel (d)] will intersect $m-1=2$ points, and these points will be near (but not identical with) the $m=3$ vertices of the M^0 triangle. These six intersections will identify the $m(m-1)=6$ active constraints.

This suggests that, for $m=3$ in general, the two points lying on a side of the M^* triangle will be near different vertices and, subject to this restriction, will be the points that are farthest outside the M^0 triangle. This observation easily generalizes to $m > 3$: Each of the m faces of the M^* simplex will contain $m-1$ item points, each near a different vertex. These $m(m-1)$ points are the most likely to lie outside the M^0 simplex. Thus, it is sensible to initially attempt a LP solution using only the inequality constraints corresponding to these $m(m-1)$ face-item point pairs. [If point i lies on the face opposing vertex α , this face-item pair corresponds to the active inequality constraint $w_{\alpha}(i)=0$.] However, this is only a heuristic argument, and inequality constraints may still be violated in the partially constrained LP solution. If so, we iterate while adding to an *included constraint list* \mathcal{C} (of face-item pairs) the violated constraints that are identified by the above criteria as most likely to be active. The procedure terminates when all the inequality constraints are satisfied. Termination is guaranteed because inequality constraints are only added to, and never removed from, the included constraint list. The procedure is formalized below.

4. Refinement algorithm

- (1) Initialize \mathcal{C} to the empty set.
- (2) Perform hard clustering based on the M^0 assignments: item i is assigned to the cluster (vertex) α that maximizes $w_{\alpha}^0(i)$. We call this subset of items \mathcal{S}_{α} .
- (3) Identify the item (designated i_{β}) from $\mathcal{S}_{\beta} (\beta \neq \alpha)$ that is farthest outside the face opposing vertex α . This identifies the $m-1$ constraints corresponding to the face-item pairs $(\alpha, i_{\beta}; \beta \neq \alpha)$. As shown in Appendix B 5, the ordering of the item points relative to the simplex faces is the same in the \vec{w}^{Δ} and $\vec{\psi}^{\perp}$ representations. Therefore, we determine the ordering in the \vec{w}^{Δ} -representation barycentric coordinates since this is simple: $w_{\alpha}(i)$ is the distance of an item point i from the α -opposing face (positive if inside, negative if outside the simplex). When executed for all m faces this procedure identifies $m(m-1)$ inequality constraints \mathcal{C}' .
- (4) $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}'$.
- (5) Apply the LP solver with the equality constraints, the inequality constraints in \mathcal{C} , and the linear objective function approximation of Eq. (23).
- (6) Check for satisfaction of all inequality constraints and for convergence according to $\max_{\alpha, i} |w_{\alpha}^1(i) - w_{\alpha}^0(i)| < \rho_{LP}$, where ρ_{LP} is a small number and $w_{\alpha}^1(i)$ and $w_{\alpha}^0(i)$ are the values determined by M^1 and M^0 , respectively. If both conditions are satisfied, terminate with $M^* = M^1$; if not, set $M^0 \leftarrow M^1$ and return to step 2.

When the algorithm is applied to the spiral problem, \mathcal{C} is set to the active constraints in a single step [43].

III. OVERALL COMPUTATIONAL ALGORITHM

Combining the steps described in Sec. II, the overall algorithm is as follows:

(1) Compute and precondition Γ as described in Appendix A.

(2) Compute 20 [44] low-frequency clustering eigenvalues and eigenvectors using the Lanczos method [45]. This is more efficient than computing the full eigensystem but will converge slowly if the eigenvalues are densely packed near zero (as they often are). To exclude this possibility we employ a shift-and-invert spectral transformation [46], which spreads out the small eigenvalues by transforming them into the large magnitude eigenvalues of a related spectral decomposition having the same eigenvectors.

(3) Following Ref. [10], determine m according to the lowest spectral gap satisfying $\gamma_m/\gamma_{m-1} > \rho_\gamma$ where ρ_γ is the *minimum gap parameter*. If there is no gap, the algorithm has determined that there are no clusters and terminates.

(4) Identify the representatives and compute the zeroth-order solution M^0 and w_α^0 using the procedure of Sec. II B.

(5) Determine if w_α^0 violates any inequality constraints. If so, iteratively refine M^0 to M^* using the procedure of Sec. II C and, via Eq. (6), compute the refined solution w_α^* . Otherwise, $w_\alpha^* = w_\alpha^0$.

(6) Following Ref. [10], test the solution against the minimum certainty conditions $\tilde{Y}_\alpha(M) > \rho_Y (\forall \alpha)$, where ρ_Y is the *minimum certainty parameter*. If it satisfies them, the solution is accepted. If not, the eigenspectrum can be tested for higher spectral gaps, and the algorithm proceeds with step 4. If desired, the fuzzy solution can be quantized to a hard clustering by assigning item each i to the cluster having the largest assignment value; these hard clusters may be recursively analyzed.

IV. RESULTS

We tested the efficiency of our method for uncertainty minimization by using it for fuzzy spectral clustering of a family of synthetic data sets containing up to $N=20\,000$ items. Further, we showed that it can be applied to both symmetric and asymmetric Γ matrices popular in the literature.

A. Implementation

The C++ implementation was compiled using GCC version 4.1.2 and G77 version 3.3.5 under $-O3$ optimization. It accesses low-level LAPACK [47] routines through LAPACK++ [48] version 2.5.2, interfaces to the ARPACK [49] Lanczos solver through the ARPACK++ C++ wrappers [50] and solves constrained linear programs using the GLPK simplex method [51] version 4.9. The scaling benchmarks of Sec. IV B were executed on a dedicated quad CPU 3.46 GHz PENTIUM 4, configured with 4 GB of RAM and 4 GB of swap space, and running a 64-bit version of SUSE LINUX. The numerical precision parameter was $\epsilon = 2.220\,45 \times 10^{-16}$. The minimum gap and minimum certainty parameters were set to $\rho_\gamma = 3$ and $\rho_Y = 0.68$ [10]. The LP convergence parameter was $\rho_{LP} = 0.001$.

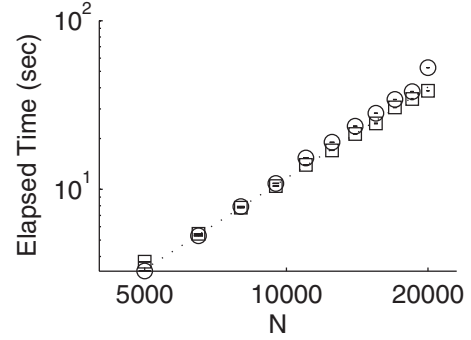


FIG. 4. Log-log plot of elapsed computational time versus N for synthetic benchmarks. N was varied from 5000 to 20 000 in steps of 1500. The results shown for $m=2$ (\circ) and 10 (\square) are averages over five runs and are representative of those for $2 < m < 10$. (The standard errors of the mean are too small to be discernible.) The dotted line is the least-squares linear fit and has slope 1.8.

B. Computational efficiency and scaling

To evaluate the efficiency and cost scaling of uncertainty minimization, we applied it to synthetic data sets containing from 2 to 10 clusters and from 5000 to 20 000 items arranged in a pyramid of blocks in a two-dimensional dataspace. For these tests we used the Laplacian Γ defined by Eqs. (1) and the definitions of S and D_π arising from the continuous dynamical interpretation of Ref. [10]:

$$S_{ij} = \frac{e^{-d_{ij}^2/2\langle d_0^2 \rangle}}{d_{ij}^2} \quad (i \neq j), \quad (24a)$$

$$(D_\pi)_{ii} = N^{-1}, \quad (24b)$$

where d_{ij} is the Euclidean distance between items i and j in the dataspace and d_0^2 is a characteristic distance of the problem:

$$\langle d_0^2 \rangle = N^{-1} \sum_{i=1}^N d_{i<}^2, \quad (25a)$$

$$d_{i<} \equiv \min_{j \neq i} d_{ij}. \quad (25b)$$

These problems required up to four invocations of the LP solver, with the number increasing with m , but not evidently with N . The log-log plot in Fig. 4 shows that execution time was proportional to $N^{1.8}$ with little dependence on m . Execution time was dominated by the calculation of S and by the eigensolver (each having roughly equal cost), with uncertainty minimization contributing $\leq 10\%$ of the total in all problems tested. The largest problem ($m=10$ and $N=20\,000$), which is of the scale of biological microarray gene expression data sets, only required about 30 s on a commodity processor.

C. General applicability

Uncertainty minimization is applicable to spectral clustering using any Γ defined by Eqs. (1), including unnormalized

and normalized forms that are popular in the literature. Of course, the success of any method will depend on the choices of S and D_π , which are highly problem specific, and we do not address this issue here. Our goal was to demonstrate the applicability of uncertainty minimization to this wide range of formulations. Thus, in addition to the tests described above using the Γ of Eqs. (24), we applied uncertainty minimization to the spiral problem using two other forms of Γ . The first one, an asymmetrically normalized Laplacian Γ ([52] for review) having S_{ij} a Gaussian function of d_{ij} , commonly arises when a Markovian [17–19,53] rather than a continuous [10] dynamical interpretation is used. It is specified by Eqs. (1) with

$$S_{ij} = e^{-d_{ij}^2/2\sigma^2}, \quad (26a)$$

$$(D_\pi)_{ii} = \frac{\sum_j S_{ji}}{\sum_{jk} S_{jk}}, \quad (26b)$$

where σ is chosen by empirical tuning [17–19,53] or heuristics [6,54]. We chose $\sigma^2 = \langle d_0^2 \rangle$. (This type of Γ , but with a non-Gaussian S , also frequently arises in image segmentation [7,15,16,55] where it is motivated by the “normalized cut” variant of the min-cut graph partitioning method [7].) We also tested the symmetric, unnormalized Laplacian form ([56] for review) of Γ specified by

$$S_{ij} = e^{-d_{ij}^2/2\sigma^2}, \quad (27a)$$

$$(D_\pi)_{ii} = N^{-1}. \quad (27b)$$

This form is popular in graph partitioning problems {e.g., very large scale integrated (VLSI) circuit partitioning [57–59], parallel matrix factorization [60], and computational load balancing [61,62]}, where it is used to approximate the solution to the “ratio cut” variant of the min-cut graph partitioning method [57]. When applied to graph partitioning the S_{ij} are simply edge weights but to apply it to the spiral data clustering problem the S_{ij} must be computed from the d_{ij} ; for this we again used the Gaussian form of Eq. (27a) because it is popular in dataspace clustering [6].

Figure 5 shows the results obtained by using uncertainty minimization for fuzzy spectral clustering of the spiral problem with the Γ matrices defined by Eqs. (24), (26), and (27). In each case the algorithm selected the same three representatives and the LP solver was invoked twice. While there were minor differences in the w_α along the cluster boundaries, the use of all three Γ gave essentially the same results. In contrast [panel (d)], the spiral problem confounded k -means with “extragrades” (which is an outlier-robust variant of k -means) [63]. As discussed in the introduction, this failure of k -means is not surprising, given the irregular, interlocking nature of the clusters.

V. DISCUSSION

To date, spectral clustering has been used primarily for hard partitioning. Prior studies [10,18] have suggested that

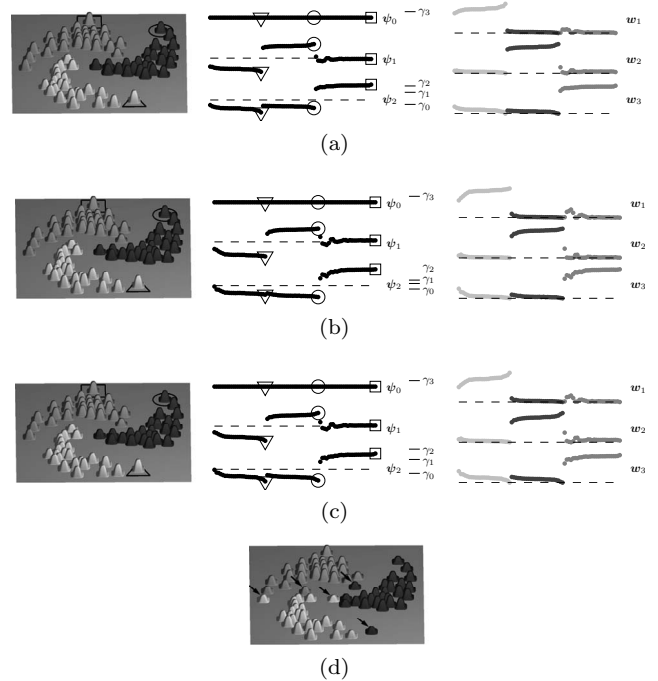


FIG. 5. Fuzzy spectral clustering by uncertainty minimization of the spiral problem using three different Γ matrices. The fuzzy clusterings and clustering eigenvectors, eigenvalues, and assignment vectors computed using the Γ matrices specified in (a) Eqs. (24), (b) Eqs. (26), and (c) Eqs. (27) are shown. Representatives are indicated by triangles, circles, and squares in the two left columns. (d) Application of fuzzy k -means with extragrades [63] to this problem; arrows identify misclassified items.

fuzzy spectral clustering could be accomplished by using the m low-frequency eigenvectors of Γ as a linear basis for expanding, via a transformation matrix M , the fuzzy cluster assignment vectors w_α , where $w_\alpha(i)$ is the probability that item i is assigned to cluster α . Korenblum and Shalloway [10] suggested that M^* , the optimal M , is best identified by uncertainty minimization, which minimizes the probabilistic overlap between clusters. Uncertainty minimization has the additional advantage of providing measures (the final values of the objective function and fractional cluster certainties) that quantify the quality of a clustering, which can be as important as the clusterings themselves. However, Korenblum and Shalloway did not provide an efficient means of solving this challenging nonconvex global minimization problem, which limited their approach to small data sets with $N \sim O(10^2)$ items. Alternatively, Weber *et al.* [18] suggested that M could be determined by perturbative approximation from almost-block-diagonal matrices, but this approach gives w_α that only approximately satisfy the probabilistic constraints of Eqs. (5). Thus, until now there has been no computationally practical, exact fuzzy spectral data clustering method.

To address this need we developed an efficient method for uncertainty minimization, which extends the number of items that can be clustered by at least two orders of magnitude: data sets with $N \sim O(10^4)$ can now be analyzed within ~ 30 seconds on a commodity processor. Using tests with synthetic data sets having up to 20 000 items and ten clusters

we showed that computational cost scaled $\sim O(N^{1.8})$ and was insensitive to the number of clusters. This implies that as many as $N \sim O(10^6)$ items can be clustered in modest time on a serial machine. The additional cost of uncertainty minimization was small compared to costs common to all spectral clustering methods (e.g., computing Γ from the d_{ij} and computing its low-frequency eigensystem).

In developing this approach we elucidated the underlying structure of the uncertainty minimization problem. This revealed fundamental relationships between four different geometric representations: the m -dimensional symmetric M representation, the $m(m-1)$ -dimensional asymmetric M representation, and the $(m-1)$ -dimensional \bar{w}^Δ and $\bar{\psi}^\perp$ representations. All are formally equivalent, but each has advantages: The symmetric M representation has the most direct connection to the minimization problem. The asymmetric M representation provides a closed feasible region; it is used to prove that all local minima are at polytope vertices and that the inequality constraints are evenly distributed between the clusters at these points. The \bar{w}^Δ representation provides barycentric coordinates and makes it evident that the m cluster representatives in \mathcal{R}_c are those items that determine the $(m-1)$ -simplex of largest hypervolume. The $\bar{\psi}^\perp$ representation motivates the greedy algorithm used to approximate \mathcal{R}_c , which in turn yields M^0 , the starting point for refinement to M^* .

The greedy algorithm we used is almost identical to the “inner simplex method” used in the Perron cluster cluster analysis method [18,64] for approximate fuzzy data clustering [65]. However, our motivation for the algorithm and consequently our understanding of its domain of validity are different. The inner simplex method was motivated by earlier studies [66,67] on perturbation theory of block-diagonal matrices [34]. These studies exploited two observations: (1) that the Γ of well-separated clusters can be brought into almost-block-diagonal form and (2) that the low-frequency eigenvectors of such a Γ are perturbed only in second order in the non-block-diagonal terms, and therefore, to this order, possess a “level structure” in which their components are concentrated near m different values. The inner simplex method aims at finding one item from each level set and thus, in principle, depends on their existence. In contrast, the analysis presented here makes no assumptions about level structure and only presumes that at least one item (i.e., the representative) can be well assigned to each cluster. An example where representatives exist, even though matrix perturbation theory is no longer applicable and the eigenvectors do not have a level structure, is illustrated in Fig. 6. Even in this case it is evident that there are three fuzzy clusters although many of the items will have weak assignments. Thus, the greedy algorithm is more generally applicable than previously stated.

In the two-cluster case the M^0 solution is always exact, but in the tested $m > 2$ problems, it always violated some of the inequality constraints required for a probabilistic interpretation of the w_α . These violations were removed by refinement. The corrections changed $w_\alpha(i)$ by < 0.05 ; so, except when high accuracy is needed, the most important role of the refinement may be to provide a rational method for

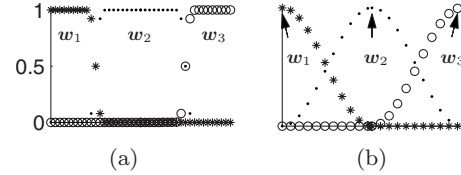


FIG. 6. Assignment vectors for a three-cluster problem for eigenvectors with or without a “level structure.” Eigenvectors arising from almost-block-diagonal Γ have a level structure leading to “almost-hard” assignment vectors such as those shown in panel (a). (Different symbols are used for the three assignment vectors.) When there is no level structure, the assignment vectors are much softer, as in panel (b). However, even in this case representatives (identified by arrows) exist.

ensuring that the w_α^0 satisfy the probabilistic constraints.

Deuffhard and Weber [68] used a metastability objective function for clustering protein conformations collected from molecular dynamics simulations that is closely related to the sum of the fractional cluster certainties $\bar{Y}_\alpha(M)$ defined in Eq. (7). Their objective function is the sum of terms

$$\bar{v}_\alpha(M;t) \equiv \frac{\langle \mathbf{w}_\alpha | e^{-\Gamma t} | \mathbf{w}_\alpha \rangle}{\langle \mathbf{1} | \mathbf{w}_\alpha \rangle},$$

where t denotes a time period which, in practice, is set to a multiple of the molecular dynamics integration time step [69,70]. $\bar{v}_\alpha(M;t)$ measures the fractional persistence of probability within subregion α of conformation space after stochastic evolution for time t and is identical to the $\bar{Y}_\alpha(M)$ except for the presence of the Markov matrix $e^{-\Gamma t}$, generated by a Γ derived from the molecular dynamics data. Thus, $\bar{Y}_\alpha(M)$ is the $t \rightarrow 0$ limit of $\bar{v}_\alpha(M;t)$. It is not clear if a t -dependent objective function is appropriate for clustering data that does not arise in a dynamic manner, though this may be worth considering.

Another potentially interesting objective function is the determinant of M . It is intriguing because of its simple geometric interpretation: we can show that maximizing $|M|$ is equivalent to maximizing the hypervolume of the $(m-1)$ -simplex formed in the \bar{w}^Δ representation by any subset of m items [71]. This property is attractive since we expect a good clustering to spread the items out in this barycentric representation as much as possible. However, $|M|$ does not have a simple information-theoretic interpretation as does $\Phi(M)$, defined in Eq. (10): $\exp[-\Phi(M)]$ is the product of the fractional cluster certainties, \bar{Y}_α , which are normalized to unity when the corresponding cluster is completely hard, but $|M|$ does not provide a normalized measure of cluster hardness. Moreover, while optimization using either $|M|$ or $\Phi(M)$ tends to minimize overlap, optimization of $|M|$ also tends to equalize the size of the clusters [71]. Although this is not necessarily desirable for data clustering, it may be of value in graph partitioning applications that seek to balance partition sizes [7,57].

Uncertainty minimization and the method for efficiently solving it presented here are applicable to the wide range of popular Γ matrices that satisfy Eqs. (1). To demonstrate this,

we applied uncertainty minimization to the spiral data set, a convenient two-dimensional example with visually discernible irregularly shaped clusters, using one asymmetric and two symmetric forms of Γ . The resulting fuzzy spectral clustering gave similar results with all three Γ matrices, while k -means did not provide a valid clustering. Of course, these particular forms may not be suitable for all data sets—as in hard spectral clustering, Γ must often be tailored to the problem. Our goal here was to demonstrate the ability of uncertainty minimization to efficiently fuzzify spectral clustering methods. It can now be applied to a wide variety of problem-specific domains, such as those noted in the introduction.

ACKNOWLEDGMENTS

The authors are grateful to Sally McKee for the use of computational resources and to Vince Weaver for help in administering them. Partial support was provided for B.S.W. by the Department of Energy and administered by The Krell Institute, Ames, IA.

APPENDIX A: Γ PRECONDITIONING

Numerical errors in computing the eigensystem increase with γ_{N-1}/γ_1 and, if this ratio is too large, can obscure differences between very small eigenvalues and obfuscate the spectral gap. This can occur if two items within a cluster are exceedingly close (and hence communicate very rapidly) or if clusters are nearly isolated (and hence communicate very slowly). The latter situation can also occur if the data contain outliers—items that are distant from the bulk of the items. We avoid these problems by preconditioning Γ and, at the same time, improve computational efficiency by sparsifying it (i.e., by setting very small transition rates exactly to zero). [This reduces memory requirements and improves cache performance and eigensolver efficiency so that uncertainty minimization may be practically applied to large problems. For example, for the largest of the scaling benchmarks considered in Section IV B. (i.e., $m=10$ and $N=20\,000$), the sparsified Γ matrix held less than 650 000 independent elements, representing a storage reduction in a factor of ~ 300 .] This involves three steps: (1) determine appropriate upper (d_{hi}) and lower (d_{lo}) bounds on the d_{ij} , (2) sparsify Γ using d_{hi} and check for any resultant graph disconnections, and (3) evaluate the remaining matrix elements and truncate the large-magnitude elements using d_{lo} , compute γ_1 , bound γ_{N-1} , and confirm that γ_{N-1}/γ_1 is properly constrained. If it is not, d_{lo} is increased so that it will be. (Increasing d_{lo} was not required for the examples in this paper, but this step is included as a precaution.)

To avoid excessive numerical error, we want to adjust Γ so that

$$\frac{\Delta_\gamma}{\gamma_1} \leq \alpha, \quad (\text{A1})$$

where Δ_γ is the expected computational error in the eigenvalues and α is the desired fractional precision, e.g., $\sim O(10^{-2})$. Typically [47,49]

$$\Delta_\gamma \leq \epsilon \gamma_{N-1},$$

where ϵ is machine precision. So Eq. (A1) will be satisfied if

$$\frac{\gamma_{N-1}}{\gamma_1} \leq \alpha/\epsilon. \quad (\text{A2})$$

We expect that γ_{N-1}/γ_1 will depend on $|\Gamma_{hi}|/|\Gamma_{lo}|$, the ratio of the largest to the smallest nonzero $|\Gamma_{i \neq j}|$, and one way to satisfy Eq. (A2) would be to limit this ratio. However, when clustering data, e.g., as in the examples of this paper, computing the $\Gamma_{i \neq j}$ from the d_{ij} constitutes a significant fraction of total cost because exponentiation is required [at least for forms of S in Eqs. (24), (26), and (27)], and this is wasted for the large fraction of the $\Gamma_{i \neq j}$ that are zeroed during preconditioning. Therefore, instead of directly limiting $|\Gamma_{hi}|/|\Gamma_{lo}|$, we gain the same result by limiting d_{hi}/d_{lo} , the ratio of the largest to the smallest d_{ij} . This allows us to sparsify before evaluating all but a few matrix elements. This indirect approach is not needed when applying uncertainty minimization to spectral clustering of graphs where S is specified *a priori* and, hence, all elements of Γ can be inexpensively computed.

1. Determining d_{hi} and d_{lo}

Although a rigorous *a priori* bound on γ_{N-1}/γ_1 depends on N as well as on $|\Gamma_{hi}|/|\Gamma_{lo}|$, we expect that in most cases the two ratios will be roughly of the same order of magnitude since $|\Gamma_{hi}|$ and $|\Gamma_{lo}|$ set the scales of the fastest and slowest dynamical processes in the system [72]. Thus, we can hope to satisfy Eq. (A2) by requiring that

$$\frac{|\Gamma_{hi}|}{|\Gamma_{lo}|} = \alpha/\epsilon \quad (\text{not used}). \quad (\text{A3})$$

However, when Γ is asymmetric [i.e., $(D_\pi)_{ii} \neq 1/N$ as in Eq. (26)], then even this requirement cannot be imposed until D_π is evaluated, and this would require costly evaluation of all the $\Gamma_{i \neq j}$ prior to sparsification. Thus, instead we apply Eq. (A3) to Γ^S :

$$\frac{|\Gamma_{hi}^S|}{|\Gamma_{lo}^S|} = \alpha/\epsilon. \quad (\text{A4})$$

We expect this to be adequate because in most cases multiplying by D_π^{-1} will result in $|\Gamma_{hi}|/|\Gamma_{lo}| < |\Gamma_{hi}^S|/|\Gamma_{lo}^S|$. (This indeed is the case for the examples we have considered.) However, exceptional sets of d_{ij} can be constructed where it will not, so this is not guaranteed. Nonetheless, we use Eq. (A4) because of its reduced cost and the guarantee that Eq. (A2) will ultimately be satisfied by the confirmation and possible iteration steps described in Appendix A 3.

To minimize the effect of preconditioning on the rest of the eigensystem, we multiplicatively center $|\Gamma_{hi}^S|$ and $|\Gamma_{lo}^S|$ around $|\Gamma_{mid}^S|$, a typical midrange rate. That is, we require

$$\frac{|\Gamma_{mid}^S|}{|\Gamma_{lo}^S|} = \frac{|\Gamma_{hi}^S|}{|\Gamma_{mid}^S|}. \quad (\text{A5})$$

We determine $|\Gamma_{mid}^S|$ by noting that $|\Gamma_{i>}^S|$, the magnitude of the largest $\Gamma_{i \neq j}^S$ in row i , is the largest transition rate connect-

ing i to other items. Thus, the median of the $|\Gamma_{i>}^S|$ is a reasonable choice for $|\Gamma_{\text{mid}}^S|$. Because $|\Gamma_{i\neq j}^S|$ depends monotonically on d_{ij} [e.g., see Eqs. (24), (26), and (27)], this is equivalent to $|\Gamma_{\text{mid}}^S| = |\Gamma^S(\text{med}\{d_{i<}\})|$, where $\text{med}\{d_{i<}\}$ is the median of the $\{d_{i<}\}$, the smallest off-diagonal elements in each row of the d_{ij} matrix. Thus, determining $|\Gamma_{\text{mid}}^S|$ requires computing only one element of Γ^S . Once this has been done, Eqs. (A4) and (A5) can be combined to give

$$\Gamma_{\text{lo}}^S = |\Gamma_{\text{mid}}^S| \sqrt{\epsilon/\alpha}, \quad (\text{A6a})$$

$$\Gamma_{\text{hi}}^S = |\Gamma_{\text{mid}}^S| \sqrt{\alpha/\epsilon}. \quad (\text{A6b})$$

We then numerically invert $\Gamma_{ij}^S(d_{ij})$ [e.g., using one of Eqs. (24), (26), and (27)] with $\Gamma_{ij}^S \rightarrow -|\Gamma_{\text{lo}}^S|$ and $\Gamma_{ij}^S \rightarrow -|\Gamma_{\text{hi}}^S|$ to determine d_{hi} and d_{lo} , respectively.

2. Sparsification and connected component analysis

Γ^S is sparsified by setting all off-diagonal elements having magnitudes less than Γ_{lo}^S to zero. That is,

$$\Gamma_{i\neq j}^S \rightarrow 0 \quad (\text{if } d_{ij} > d_{\text{hi}}).$$

To test if this disconnects the graph, we perform a standard connected component analysis [73]. This initially assigns items to individual sets and then iteratively merges sets whenever any of their respective members are connected. If distinct subsets (i.e., disconnected components) remain at the end, the algorithm creates hard assignment vectors identifying them. (This process may remove outliers.) Larger subsets may be analyzed as new clustering problems of their own.

3. Truncation and checking the eigenvalue range

Having sparsified the (typically large) fraction of insignificantly small off-diagonal elements, we now evaluate the remaining $\Gamma_{i\neq j}^S$ while truncating their maximum magnitudes using

$$\Gamma_{i\neq j}^S \rightarrow -|\Gamma_{\text{hi}}^S| \quad (\text{if } d_{ij} < d_{\text{lo}}),$$

and compute D_π and Γ . We can then compute γ_1 using a Lanczos solver (see Sec. III) and bound γ_{N-1} using the Gershgorin circle theorem [74] and Eqs. (1c) and (1d) to

$$\gamma_{N-1} \leq 2 \max |\Gamma_{ii}|. \quad (\text{A7})$$

If γ_1 and the Gershgorin bound on γ_{N-1} satisfy Eq. (A2), then preconditioning is complete. If not, d_{lo} , and hence $|\Gamma_{\text{hi}}^S|$, is adjusted so that it will be satisfied when the $|\Gamma_{i\neq j}^S|$ are truncated to the new bound and Γ is recomputed. Preconditioning is now complete.

APPENDIX B: VARIOUS PROOFS

1. Even distribution of active inequality constraints

We prove here that each cluster must be constrained by exactly $m-1$ inequality constraints at each local minimum of Φ in the feasible region. Consider a local minimum $\vec{M}^{\times \text{free}}$ in the asymmetric M representation discussed in Sec. II A. Kornblum and Shalloway [10] have already proved that this

must be at a vertex of the feasible polytope. The coordinates at the local minimum of the individual free particles, \vec{M}_α^\times ($1 \leq \alpha < m$), satisfy the inequality constraints of Eq. (11a), but their homogeneity means that they will also be satisfied for any $\xi_\alpha \vec{M}_\alpha^\times$ with $\xi_\alpha > 0$. Thus, the free particle inequality constraints acting alone leave the $m-1$ degrees of freedom ξ_α unspecified and are inadequate to force $\vec{M}^{\times \text{free}}$ to be at a vertex of the feasible polytope. Therefore, at least $m-1$ additional active constraints must come from the inhomogeneous inequality constraints associated with the slave particle [Eq. (13)]. However, the choice of the slave particle in Eq. (12) is arbitrary. Therefore, every particle must have at least $m-1$ active inequality constraints. But since only $m(m-1)$ inequality constraints are active at a vertex, each of the m particles must have exactly $m-1$ inequality constraints active. This proof extends to every vertex of the feasible polytope except for those vertices where at least one of the $\vec{M}_\alpha^\times = 0$ (since multiplying such an \vec{M}_α^\times by ξ_α has no effect). This proof does not preclude the possibility that a single item may be associated with multiple active constraints; i.e., it is possible that $w_\alpha(i)=0$ and $w_\beta(i)=0$ are both active constraints. (This is the case for the solution to the spiral problem [Figs. 3(c) and 3(d)] where $w_2(r_1)=0=w_3(r_1)$ and also $w_1(r_2)=0=w_3(r_2)$.)

2. Invertibility of $\Psi^{\mathcal{R}}$

We prove here that there is at least one subset of m items \mathcal{R} such that $\Psi^{\mathcal{R}}$ is invertible. We define the $m \times N$ matrix Ψ by $\Psi_{ni} \equiv \psi_n(i)$ ($0 \leq n < m; 1 \leq i \leq N$). Since its m rows (i.e., the low-frequency eigenvectors) are linearly independent, Ψ has rank m . Therefore, Ψ also has at least m linearly independent columns. If the items corresponding to these columns are selected to comprise \mathcal{R} , then the $m \times m$ matrix $\Psi^{\mathcal{R}}$ has full rank and is therefore invertible.

3. Invertibility of M

We prove here that each M corresponding to a local minimum of Φ within the feasible region is invertible. As proved in Appendix B 1, at any such minimum each cluster has $m-1$ active inequality constraints: $m-1$ items lie on each of the m faces of the bounding simplex in the \vec{w}^Δ representation. Consider a subset \mathcal{R} that contains one item from each face. It defines an $(m-1)$ -simplex (inscribed within or identical to the bounding simplex) with nonzero hypervolume. This hypervolume is proportional to $|W^{\mathcal{R}}|$, implying that $|W^{\mathcal{R}}| \neq 0$ and, with Eq. (17), implying that $|M| \neq 0$. Thus, M is invertible.

Actually, the proof holds for every M having all $\vec{M}_\alpha \neq 0$ that lies at a vertex of the feasible polytope in the asymmetric M representation since Appendix B 1 applies to all such M , not only those at local minima.

4. Bounding simplex in the $\vec{\psi}^\perp$ representation

Analogously to Eq. (14), we may write

$$W^{\text{vert}} = M \circ \Psi^{\text{vert}}, \quad (\text{B1})$$

where the columns of Ψ^{vert} are the coordinates of the bounding simplex vertices in the low-frequency eigenvector repre-

sensation and W^{vert} is the matrix whose rows are the coordinates of the vertices in the \bar{w}^Δ representation; i.e., $W^{\text{vert}}=I$. Inverting this gives $\Psi^{\text{vert}}=M^{-1}$. When $M=M^0$, Eqs. (19b) and (B1) imply that $\Psi^{\text{vert}}=\Psi^{\mathcal{R}_c}$, which is consistent with the zeroth-order placement of the representatives at the vertices. When $M=M^*$, the vertices may not correspond to item locations, but, as proved in Appendix B 3, M^* is invertible so $\Psi^{\text{vert}}=(M^*)^{-1}$. In both cases, the simplex vertex coordinates in the $\vec{\psi}^\perp$ representation are given by the columns of Ψ^{vert} with the first row omitted. (Just as for $\Psi^{\mathcal{R}}$, all elements in the first row of Ψ^{vert} are one for any invertible M , in particular, for M^0 and M^* [75].)

5. Same ordering of item points in the \bar{w}^Δ and $\vec{\psi}^\perp$ representations

To simplify the proof of identical ordering, we use the spiral problem illustrated in Fig. 3 as a specific example; the proof is easily generalized. We index the vertices in the \bar{w}^Δ representation as described in Sec. II B 2. For example, the top vertex in panel (c) is vertex 3, and we denote it as v_3 . We carry the same indexing over to the $\vec{\psi}^\perp$ representation.

Ordering the items according to their distances from the simplex faces is easy in the \bar{w}^Δ representation: because it provides barycentric coordinates, the distance of a point i from the side opposite vertex α is just $w_\alpha(i)$, with the sign negative if the point lies outside the simplex. The \bar{w}^Δ ordering can be related to the $\vec{\psi}^\perp$ ordering in a few steps. First, note that the distance of point i from the side opposite v_2 is linearly related to the area of the triangle having vertices at points i , v_1 , and v_3 , with sign depending on triangle orientation. This signed area is proportional to the ratio of determinants

$$A = \frac{|\bar{w}(i) \otimes \hat{e}_1 \otimes \hat{e}_3|}{|\hat{e}_2 \otimes \hat{e}_1 \otimes \hat{e}_3|},$$

where $\bar{w}(i) \otimes \hat{e}_1 \otimes \hat{e}_3$ is the 3×3 matrix formed by stacking the three row vectors and the denominator (which will always be ± 1) ensures the correct sign. Second, note that since

$$\bar{w}(i) \otimes \hat{e}_1 \otimes \hat{e}_3 = M \circ [\vec{\psi}(i) \otimes \vec{\psi}_{v_1} \otimes \vec{\psi}_{v_3}],$$

$$\hat{e}_2 \otimes \hat{e}_1 \otimes \hat{e}_3 = M \circ [\vec{\psi}_{v_2} \otimes \vec{\psi}_{v_1} \otimes \vec{\psi}_{v_3}],$$

where $\vec{\psi}_{v_k}$ is the m -vector having the coordinates of vertex v_k in the low-frequency eigenvector space,

$$A = \frac{|\vec{\psi}(i) \otimes \vec{\psi}_{v_1} \otimes \vec{\psi}_{v_3}|}{|\vec{\psi}_{v_2} \otimes \vec{\psi}_{v_1} \otimes \vec{\psi}_{v_3}|}. \quad (\text{B2})$$

Third, since all the m -vectors in Eq. (B2) have their zeroth component equal to one, A is proportional to the signed area of the triangle having vertices i , v_1 , and v_3 in the $\vec{\psi}^\perp$ representation. Fourth, this area is proportional to the distance of

point i from the side opposite to v_2 in the $\vec{\psi}^\perp$ representation. Combining all these proportionalities proves that the distance of point i from the side opposing a vertex in the \bar{w}^Δ representation is proportional to its distance in the $\vec{\psi}^\perp$ representation.

APPENDIX C: GREEDY ALGORITHM FOR SELECTING \mathcal{R}

The goal of the algorithm is to choose the subset of items \mathcal{R} that approximately defines the $(m-1)$ -simplex having maximum hypervolume V_{m-1} in the $\vec{\psi}^\perp$ representation. If the hypervolume V_{m-2} of one face of the simplex is already determined, V_{m-1} is proportional to the distance of the excluded vertex from that face. [For example, in the case of a 2-simplex (a triangle), this is the familiar area = 1/2 base \times height rule, where ‘‘base’’ is the length of the determined simplex face and ‘‘height’’ is the distance of the other point from that face.] This suggests a natural greedy algorithm: (a) initialize by finding the $(q-1=1)$ -simplex of greatest length, (b) extend the $(q-1)$ -simplex to a q -simplex by finding the item that is furthest from the hypersurface that embeds the $(q-1)$ -simplex, (c) $q \leftarrow q+1$ and return to step (b) until $q = m$.

Specifically,

1. Initialize

Select the two items i_1 and i_2 that maximize $\|\vec{\psi}^\perp(i_2) - \vec{\psi}^\perp(i_1)\|$,

$$\mathcal{R} = \{i_1, i_2\},$$

$$q = 2.$$

2. Repeat while $q < m$

(a) Select the item i_{q+1} that maximizes

$$d^\perp(i_{q+1}) = \|\mathcal{P}^q \ominus [\vec{\psi}^\perp(i_{q+1}) - \vec{\psi}^\perp(i_1)]\|,$$

where

$$\mathcal{P}_{nn'}^q = I_{nn'} - \sum_{q'=2}^q \frac{[\vec{\psi}^\perp(i_{q'}) - \vec{\psi}^\perp(i_1)]_n [\vec{\psi}^\perp(i_{q'}) - \vec{\psi}^\perp(i_1)]_{n'}}{\|\vec{\psi}^\perp(i_{q'}) - \vec{\psi}^\perp(i_1)\|^2},$$

(b) $\mathcal{R} \leftarrow \mathcal{R} \cup i_{q+1}$, and (c) $q \leftarrow q+1$.

Here \ominus denotes the inner product within the $(m-1)$ -dimensional $\vec{\psi}^\perp$ space and \mathcal{P}^q is the projection matrix in this space that removes the components of $[\vec{\psi}^\perp(i_{q+1}) - \vec{\psi}^\perp(i_1)]$ that lie within the subspace containing the $(q-1)$ -simplex. Therefore, $d^\perp(i_{q+1})$ is the distance of $\vec{\psi}^\perp(i_{q+1})$ from the subspace, and the q -simplex formed by adding $\vec{\psi}^\perp(i_{q+1})$ as a vertex is that of maximum hypervolume containing the previously computed $(q-1)$ -simplex as one of its faces.

- [1] A. K. Jain, M. N. Murty, and P. J. Flynn, *ACM Comput. Surv.* **31**, 264 (1999).
- [2] B. S. Everitt, S. Landau, and M. Leese, *Cluster Analysis* (Arnold, London, 2001).
- [3] R. Xu and D. Wunsch II, *IEEE Trans. Neural Netw.* **16**, 645 (2005).
- [4] R. J. Cho *et al.*, *Mol. Cell* **2**, 65 (1998).
- [5] D. A. Spielman and S.-H. Teng, *Proceedings of Annual Symposium on Foundations of Computer Science* (IEEE Computer Society, Washington, D.C., 1996), pp. 96–105.
- [6] U. von Luxburg, *Stat. Comput.* **17**, 395 (2007).
- [7] J. Shi and J. Malik, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (IEEE Computer Society, Washington, D.C., 1997), pp. 731–737.
- [8] S. D. Kamvar, D. Klein, and C. D. Manning, *Proceedings of International Joint Conference on Artificial Intelligence* (Morgan Kaufmann, San Mateo, 2003), pp. 561–566.
- [9] A. Y. Ng, M. I. Jordan, and Y. Weiss, *Proceedings of the 14th Neural Information Processing Systems Conference* (MIT Press, Cambridge, 2002), pp. 849–856.
- [10] D. Korenblum and D. Shalloway, *Phys. Rev. E* **67**, 056704 (2003).
- [11] Some authors analyze $I-\Gamma$, but this is an insignificant difference. Less frequently [9], the “normalized Laplacian” $\Gamma = D_\pi^{-1/2} \cdot S \cdot D_\pi^{-1/2}$ is analyzed, which does not satisfy Eqs. (1c) and (1d). Such Γ do not have a dynamical interpretation so we do not consider them.
- [12] Equation (1d) is a direct consequence of Eq. (1c).
- [13] W. E. Donath and A. J. Hoffman, *IBM J. Res. Dev.* **17**, 420 (1973).
- [14] M. Fiedler, *Czech. Math. J.* **23**, 298 (1973).
- [15] M. Meilä and J. Shi, *Proceedings of Neural Information Processing Systems Conference* (MIT Press, Cambridge, 2000), pp. 873–879.
- [16] M. Meilä and J. Shi, *Proceedings of International Workshop on Artificial Intelligence and Statistics* (Morgan Kaufman, San Francisco, 2001).
- [17] M. Belkin and P. Niyogi, *Neural Comput.* **15**, 1373 (2003).
- [18] M. Weber, W. Rungtaritoyin, and A. Schliep, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Report No. 04–39, 2004 (unpublished).
- [19] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis, *Appl. Comput. Harmon. Anal.* **21**, 113 (2006).
- [20] Although most spectral clustering Γ satisfy Eqs. (1), Pentney and Meila [76] use a Γ that does not and thereby gives rise to complex eigenvalues. We do not consider this uncommon situation.
- [21] Alternatively, the ψ^R could be used, but the contrast between Eqs. (2b) and (2c) makes it more convenient to use the ψ^L .
- [22] An exception is Ref. [59], which uses more low-frequency eigenvectors than clusters.
- [23] In some cases (e.g., integrated circuit design) spectral methods have been used to partition items into a fixed number of subsets regardless of the existence of such a gap. We distinguish such forced partitioning from clustering and do not consider it here.
- [24] Equation (3) is the continuous master equation form used by Korenblum and Shalloway [10]. Other authors consider Markov processes, which are discretized versions of this equation. Our analysis applies with little change to the Markov case as well.
- [25] In the dynamical interpretation of Eq. (3), the temporal decay of the low-frequency eigenvectors redistributes probability between clusters, while the decay of the higher-frequency eigenvectors redistributes probability within clusters. Thus, the low-frequency eigenvectors have the information needed to describe the metastable states of the system, which can be shown to correspond to the clusters [10]; see also D. Shalloway, *J. Chem. Phys.* **105**, 9986 (1996) for an extended discussion of such coarse graining in the context of a continuous (conformation) space.
- [26] D. Jeffries, I. Zaidi, B. de Jong, M. J. Holland, and D. J. C. Miles, *Cytometry, Part A* **73A**, 857 (2008).
- [27] A. Paccanaro, J. A. Casbon, and M. A. S. Saqi, *Nucleic Acids Res.* **34**, 1571 (2006).
- [28] J. Reichardt and S. Bornholdt, *Phys. Rev. Lett.* **93**, 218701 (2004).
- [29] A. P. Gasch and M. B. Eisen, *Genome Biol.* **3**, 1 (2002).
- [30] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, *J. Mol. Biol.* **247**, 536 (1995).
- [31] Reference [10] actually uses the $D_\pi^{-1} \cdot \psi_n^R$ as a basis, but by Eq. (2d) this is equivalent to using the ψ_n^L .
- [32] Reference [10] only explicitly considers the case when $(D_\pi)_{ii} = 1/N$. Here we introduce the straightforward generalization that follows from their Eq. (10). Its inclusion in Eq. (8) ensures that items with little contribution to the equilibrium probability make little contribution to the uncertainty.
- [33] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, *Proc. Natl. Acad. Sci. U.S.A.* **95**, 14863 (1998).
- [34] G. W. Stewart, in *Mathematical Computer Performance and Reliability*, edited by G. Iazeolla, P. J. Courtois, and A. Hordijk (Elsevier/North Holland, New York, Amsterdam, 1984), pp. 287–302.
- [35] For $m > 2$, the slave inequality constraint boundaries do not intersect at a point but in the subspace where Eq. (13) is an equality for all items. This condition can be expressed as $\sum_{k=1}^{m-1} Z_{ik} M_k^{\text{free}} = 1 (\forall i)$, where $M_k^{\text{free}} \equiv M_{\alpha n}$ with $\alpha \equiv 1 + [(k-1)/m]$, $n \equiv (k-1) \bmod m$, and $Z_{ik} \equiv \psi_{k'}(i)$ with $k' \equiv (k-1) \bmod m$. (M_k^{free} is the flattened form of the supervector \vec{M}^{free} .) Z_{ik} has m independent columns, i.e., it contains $m-1$ copies of each of the m low frequency ψ_n . Therefore, its rank is m and it only imposes m restrictions on the $m(m-1)$ variables. Thus, the intersection of the inhomogeneous constraints occurs in a subspace of dimension $m(m-1) - m = m(m-2)$.
- [36] Equation (18) breaks the permutation symmetry by assigning each representative to a specific cluster index.
- [37] To prove this, we postmultiply Eq. (19b) by $\Psi^{\mathcal{R}c}$ and sum to get $1 = \sum_\alpha J_{\alpha\beta} = \sum_{\alpha n} M_{\alpha n}^0 \Psi_{n\beta}^{\mathcal{R}c} (\forall \beta)$. Since $\Psi_{0\beta}^{\mathcal{R}c} \equiv \psi_0(r_\beta) = 1 (\forall \beta)$, this is solved by $\sum_\alpha M_{\alpha n}^0 = \delta_{n0}$. Moreover, since $\Psi^{\mathcal{R}c}$ is invertible, this solution is unique, thereby proving Eq. (20).
- [38] Although they are constructed from the same components, do not confuse the m N -vectors w_α^* , which are vectors over the space of items, with the N m -vectors $\bar{w}^*(i)$, which are vectors over the space of clusters. (These vectors can be viewed as the rows and columns, respectively, of an $m \times N$ matrix W^* .)
- [39] H. S. M. Coxeter, *Introduction to Geometry*, 2nd ed. (Wiley and Sons, New York, 1969).
- [40] When Γ is symmetric, the covariance matrix, $\sum_{mn} \psi_m \cdot \psi_n = \delta_{mn}$, is diagonal, so that the item points in the ψ^L representation have principal components $\psi_n (0 < n < m)$ and center of mass at the origin.

- [41] We ignore the rare case where LP has a space of degenerate solutions corresponding to a face of the feasible polytope.
- [42] Interior point methods have cost $O(m^3 N_c^{1.5} L / \log N_c)$, where $L \geq O(\log N_c)$ [K. M. Anstreicher, *SIAM J. Optim.* **9**, 803 (1999)]; Simplex methods require $O[\min(N_c^2, m^4)]$ pivot steps [I. Adler, R. M. Karp, and R. Shamir, *J. Complex.* **3**, 372 (1987)]. The cost of each pivot step is $O(N_c^2)$ as that is the size of the matrix that is updated. Thus, the number of operations for simplex methods, like that of interior point methods, grows at least as fast as $N_c^{1.5}$.
- [43] In the spiral problem the representatives at vertices 1 and 2 each contribute two active constraints to the solution M^* . Because the left side pivots out during refinement, the representative at vertex 3 only contributes one of the two constraints that are active at M^0 ; the second constraint associated with this vertex comes from the item identified by the arrow in Fig. 3(f). Specifically, the constraints active at M^* are $w_\alpha(i)=0$, where $(\alpha, i)=(2, r_1), (3, r_1), (1, r_2), (3, r_2), (1, r_3)$, and $(2, x_3)$, where x_3 is the index of the item identified by the arrow. In this problem, these are all identified in a single step of refinement.
- [44] The number of eigenvectors computed is arbitrary and is chosen to limit computational cost on the assumption that the number of potential clusters to be identified at a single level of recursion is <20 . A larger value could be used if needed.
- [45] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. (John Hopkins U. Press, Baltimore, Md., 1996), Chap. 9.
- [46] R. Lehoucq and D. Sorensen, in *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, edited by Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst (SIAM, Philadelphia, 2000). Since we are interested in the eigenvalues near zero, we choose the shift of the shift-and-invert Lanczos method to be $\sqrt{\epsilon}$, where ϵ is machine precision, to maximize the spread without introducing significant numerical error.
- [47] E. Anderson, Z. Bai, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide* (SIAM, Philadelphia, PA, 1999).
- [48] C. Stimming, LAPACK++ <http://lapackpp.sourceforge.net> (2008).
- [49] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods* (SIAM, Philadelphia, 1998).
- [50] F. A. M. Gomes and D. C. Sorensen, Rice University, Report No. TR97729, 1997 (unpublished).
- [51] A. Makhorin, GNU Linear Programming Kit: Reference Manual <http://www.gnu.org/software/glpk> (2006).
- [52] F. Chung, *Spectral Graph Theory* (American Mathematical Society, Providence, RI, 1997).
- [53] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis, *Proceedings of Neural Information Processing Systems Conference* (MIT Press, Cambridge, 2006), pp. 955–962.
- [54] L. Zelnik-Manor and P. Perona, *Proceedings of Neural Information Processing Systems Conference* (MIT Press, Cambridge, 2005), pp. 1601–1608.
- [55] J. Malik, S. Belongie, T. Leung, and J. Shi, *Computer Vision* **43**, 7 (2001).
- [56] B. Mohar, in *Graph Theory, Combinatorics, and Applications*, edited by Y. Alavi, G. Chartrand, O. R. Oellermann, and A. J. Schwenk (Wiley, 1991), Vol. 2, pp. 871–898.
- [57] L. Hagen and A. Kahng, *IEEE Trans. Comput.-Aided Des.* **11**, 1074 (1992).
- [58] P. K. Chan, M. D. F. Schlag, and J. Y. Zien, *ACM IEEE Design Automation Conference* (ACM Press, New York, 1993), pp. 749–754.
- [59] C. J. Alpert, A. B. Kahng, and S.-Z. Yao, *Discrete Appl. Math.* **90**, 3 (1999).
- [60] A. Pothén, H. D. Simon, and K.-P. Liou, *SIAM J. Matrix Anal. Appl.* **11**, 430 (1990).
- [61] S. T. Barnard and H. D. Simon, *Concurrency* (Chichester, Engl.) **6**, 101 (1994).
- [62] B. Hendrickson and R. Leland, *SIAM J. Sci. Comput. (USA)* **16**, 452 (1995).
- [63] J. J. De Gruijter and A. B. McBratney, in *Classification and Related Methods of Data Analysis*, edited by H. H. Bock (Elsevier, New York, 1988), pp. 97–104. When applied to the spiral problem [Fig. 5(d)] k -means was configured using the default configuration distributed with the software: $k=m=3$, a fuzzy exponent $\phi=2$, maximum number of iterations=300, convergence criterion= 10^{-6} , d_{ij} computed as Mahalanobis distances, scatter parameter=0.2, and the number of attempts to choose an optimal solution=10.
- [64] M. Weber, Konrad-Zuse-Zentrum für Informationstechnik Berlin Report No. 03–04 2003 (unpublished).
- [65] Reference [18] uses a slightly different variant in which the greedy algorithm is initialized by selecting the item that is farthest from the origin rather than the two items that are farthest from each other. This effectively maximizes the hypervolume of the m -simplex in the $\vec{\psi}$ space, which includes the origin as one of the $m+1$ vertices, as opposed to maximizing the hypervolume of the $m-1$ -simplex in the $\vec{\psi}^\perp$ representation. Given that $\psi_0=1$, all $m-1$ -simplices in the $\vec{\psi}^\perp$ representation lie in a plane equidistant from the origin. Hence, exact maximization in either space is equivalent. However, since the greedy algorithms may only approximately maximize the volumes, the two variants may give slightly different results. But such differences become irrelevant when uncertainty minimization is used because the greedy solution only provides the starting point for subsequent refinement.
- [66] P. Deuffhard, W. Huisinga, A. Fischer, and C. Schütte, *Linear Algebr. Appl.* **315**, 39 (2000).
- [67] C. Schütte and W. Huisinga, *Handbook of Numerical Analysis X* (Elsevier, Amsterdam, 2003), p. 699.
- [68] P. Deuffhard and M. Weber, *Linear Algebr. Appl.* **398**, 161 (2005).
- [69] S. Kube and M. Weber, *J. Chem. Phys.* **126**, 024103 (2007).
- [70] F. Noé, I. Horenko, C. Schütte, and J. Smith, *J. Chem. Phys.* **126**, 155102 (2007).
- [71] The hypervolume in the \vec{w}^Δ representation of the $(m-1)$ -simplex formed by any subset \mathcal{R} of m items (not necessarily representatives) is proportional to $|W^{\mathcal{R}}|$, which, according to Eq. (14), is $|W^{\mathcal{R}}|=|M||\Psi^{\mathcal{R}}|$. Thus, the hypervolume for every subset is proportional to $|M|$ although the constant of proportionality, $|\Psi^{\mathcal{R}}|$, is different in each case. Therefore, maximizing $|M|$ maximizes the hypervolume of each $(m-1)$ -simplex. Moreover, as discussed in Appendix B 4, $\Psi^{\text{vert}}=M^{-1}$, and since $|\Psi^{\text{vert}}|$ is proportional to the area of the inequality constraint boundary $(m-1)$ -simplex in the $\vec{\psi}^\perp$ representation, maximizing $|M|$ minimizes the hypervolume of

this simplex. To see that maximizing $|M|$ tends to equalize the cluster sizes, note that $|M| = |M \circ M^T|^{1/2} = |\Omega|^{1/2}$, where $\Omega_{\alpha\beta} \equiv \langle \mathbf{w}_\alpha | \mathbf{w}_\beta \rangle$. The off-diagonal elements of Ω are the equilibrium probability-weighted cluster overlaps and will be small compared to the diagonal elements. Therefore the determinant is dominated by the product of the diagonal elements, $\Pi_\alpha \langle \mathbf{w}_\alpha | \mathbf{w}_\alpha \rangle = \Pi_\alpha \langle (\mathbf{1} - \sum_{\beta \neq \alpha} \mathbf{w}_\beta) | \mathbf{w}_\alpha \rangle \approx \Pi_\alpha \langle \mathbf{1} | \mathbf{w}_\alpha \rangle$. This will be maximized when all the $\langle \mathbf{1} | \mathbf{w}_\alpha \rangle$ (i.e., the weighted fractions of items assigned to each cluster) are equal.

[72] Of course, γ_{N-1} and γ_1 will be affected if extremal-rate processes occur in series or parallel with other processes of similar extremal magnitude. Thus, these are only rough guidelines,

but they are sufficient for preconditioning, subject to the subsequent verification described in Appendix A 3.

- [73] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms* (MIT Press, Cambridge, MA, 2001), Chap. 22.3.
- [74] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. (John Hopkins U. Press, Baltimore, Md., 1996).
- [75] To prove this, multiply both sides of Eq. (16) by $M_{n\beta}^{-1}$ and sum over $n=0, \dots, m-1$. This gives $1 = M_{0\beta}^{-1}$.
- [76] W. Pentney and M. Meila, *Proceedings of the National Conference on Artificial Intelligence*, 2005 (AAAI, Menlo Park, CA, 2005), pp. 845–850.