Appl. Comput. Harmon. Anal. 57 (2022) 58–100

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha

A multiscale environment for learning by diffusion

James M. Murphy, Sam L. Polk *

Department of Mathematics, Tufts University, United States of America

ARTICLE INFO

Article history:
Received 31 January 2021
Received in revised form 7 August 2021
Accepted 9 November 2021
Available online 17 November 2021
Communicated by Amit Singer

Dataset link:
https://github.com/sampolk/MultiscaleDiffusionClustering

Keywords:
Clustering
Diffusion geometry
Hierarchical clustering
Machine learning
Spectral graph theory

ABSTRACT

Clustering algorithms partition a dataset into groups of similar points. The clustering problem is very general, and different partitions of the same dataset could be considered correct and useful. To fully understand such data, it must be considered at a variety of scales, ranging from coarse to fine. We introduce the Multiscale Environment for Learning by Diffusion (MELD) data model, which is a family of clusterings parameterized by nonlinear diffusion on the dataset. We show that the MELD data model precisely captures latent multiscale structure in data and facilitates its analysis. To efficiently learn the multiscale structure observed in many real datasets, we introduce the Multiscale Learning by Unsupervised Nonlinear Diffusion (M-LUND) clustering algorithm, which is derived from a diffusion process at a range of temporal scales. We provide theoretical guarantees for the algorithm's performance and establish its computational efficiency. Finally, we show that the M-LUND clustering algorithm detects the latent structure in a range of synthetic and real datasets.

1. Introduction

Unsupervised machine learning algorithms detect structure in data given no known class labels [23]. Among the many branches of unsupervised learning, clustering is perhaps the most developed and widely used. A clustering algorithm partitions a dataset into groups. In a good partition, data points from the same group are "similar" to one another, while data points from distinct groups are "dissimilar" from one another. The specific notion of similarity used varies widely [40,52,56,59,64,69]. Often, cluster analysis is one of the first tasks performed by a user interested in learning more about an unexplored dataset.

Given no further information, the clustering problem is quite general. One could easily imagine cases in which there are multiple correct separations in a single dataset, and the most useful clustering often depends on the specifics of how it will be applied in practice. A coarse separation of a dataset may be desired in one problem setting, while another problem setting may call for finer separation within the data. Thus, it may make sense to consider a dataset at various scales and analyze all of the many possible "correct" partitions. The property of data having multiple scales of relevant structure is readily observable in

[*] Corresponding author. Present address: 503 Boston Avenue, Medford, MA, USA.
E-mail addresses: JM.Murphy@Tufts.edu (J.M. Murphy), Samuel.Polk@Tufts.edu (S.L. Polk).

https://doi.org/10.1016/j.acha.2021.11.004
1063-5203/© 2021 Elsevier Inc. All rights reserved.

many empirical datasets; for example, in social networks (e.g., geographical community structures), protein-protein interaction networks (e.g., scales of chemical secondary structures), and gene interaction networks (e.g., co-expressed gene clusters) [1,19,61]. Thus, to understand the structure of a dataset in its entirety, it is necessary to understand how it is structured at a multitude of scales.

Recent decades have brought significant advances in the development of clustering algorithms meant to detect multiple scales of separation [6,39,40,54]. Typically, these approaches have relied upon a data model allowing for latent hierarchical structure in a dataset to provide performance guarantees on clustering algorithms. However, in many data models allowing for multiscale cluster structure, it is difficult to understand how separation at one scale relates to separation at another [40,54]. Diffusion geometry on graphs has been proposed to efficiently capture latent low-dimensional structure in high dimensional data, where the time scale of the diffusion process corresponds to a scale of separation—short time scales reflect fine, local structures in the data, while large time scales reflect coarse, global structures in the data [14,15,41,50]. It is of interest to understand the precise nature of this time scaling and build clustering algorithms that allow for all time scales of interest to be considered simultaneously.

1.1. Major contributions

This article makes two significant contributions. The first is the *Multiscale Environment for Learning by Diffusion* (*MELD*) data model. The MELD data model is a family of clusterings, parameterized by a diffusion time parameter. For each of these clusterings, we show that diffusion distances (a data-dependent distance metric) between clusters are bounded away from the diffusion distances within clusters during an interval determined by the geometric properties of the underlying data and that clustering. We show that clusterings with coherent and well-separated clusters are more stable in the diffusion process and emphasized within the MELD data model. Finally, we show that when the clusterings in the MELD data model exhibit hierarchical structure, the number of latent clusters is monotonically non-increasing as a function of the diffusion time parameter.

The second major contribution is the *Multiscale Learning by Unsupervised Nonlinear Diffusion* (*M-LUND*) clustering algorithm. This algorithm is a multiscale generalization of the Learning by Unsupervised Nonlinear Diffusion (LUND) algorithm, which leverages diffusion distances' attractive theoretical properties to efficiently and accurately cluster high-dimensional data [41]. The M-LUND algorithm extracts all clusterings in the MELD data model using the LUND algorithm. It then chooses the clustering that minimizes the variation of information (VI) between nontrivial extracted clusterings [43]. In this way, it is able to not only suggest a few salient clusterings but also output the one that best represents all the others from an information-theoretic perspective. In addition to theoretical guarantees, we show the strong empirical performance of M-LUND on synthetic datasets associated with poor performance of many popular clustering algorithms [41,49], as well as a range of real data [20,27].

1.2. Notation and outline

Abbreviations are provided below.[2] Notation used throughout this article appears in Table A.4 in Appendix A. In Section 2, we review preliminaries and introduce pertinent background on how graph diffusion

[2] DGM: Diffusion Geometry Model.
DPC: Density Peak Clustering.
GDM: Geometric Data Model.
HSBM: Hierarchical Stochastic Blockmodel.
HSC: Hierarchical Spectral Clustering.
HSI: Hyperspectral Image
KDE: Kernel Density Estimate.
LUND: Learning by Unsupervised Nonlinear Diffusion.
M-GDM: Multiscale Geometric Data Model.
M-LUND: Multiscale Learning by Unsupervised Nonlinear Diffusion.
MELD: Multiscale Environment for Learning by Diffusion.
MMS: Multiscale Markov Stability.
NMI: Normalized Mutual Information.
SBM: Stochastic Blockmodel.
SC: Spectral Clustering.
SLC: Single-Linkage Clustering.
SL-LUND: Single-Linkage Learning by Unsupervised Nonlinear Diffusion.
VI: Variation of Information.

is well-suited to the clustering problem. In Section 3, we introduce the MELD data model, which we will show efficiently captures multiscale cluster structure within a dataset. In Section 4, we present and analyze the M-LUND algorithm, which leverages the theory of Section 3 to detect the most representative clustering among the many possible latent clusterings of a dataset. In Section 5, we provide numerical corroboration of the theory developed in Sections 3 and 4 on synthetic data and also present comparisons of the M-LUND algorithm against related clustering schemes on eleven real-world benchmark datasets and one real-world hyperspectral image (HSI) in Section 5. In Section 6, we conclude and discuss future research.

2. Background

2.1. Background on unsupervised clustering

Clustering algorithms partition a dataset $X = \{x_i\}_{i=1}^n \subset \mathbb{R}^D$ into K subsets $X_1, \dots, X_K$. The partition $\{X_k\}_{k=1}^K$ is called a *clustering* of X, while each X_k is called a *cluster*. Clustering algorithms are typically *unsupervised*, meaning that no expert annotations or labels are used in the partitioning of X. Thus, the number of clusters K is often (though not always [22,38,41]) a hyperparameter in clustering algorithms. Typically, we want a clustering to satisfy both a separation condition—that if $k \neq k'$, most points in X_k are "far" from those in $X_{k'}$—and a coherence condition—that most points in each X_k are "close." For a detailed overview of important classical clustering algorithms, see Appendix B.

2.2. Background on spectral graph theory and its applications to clustering

Spectral graph theory is widely used in clustering [6,38,39,41,52,59,64]. Typically, spectral methods construct a local connectivity graph that stores information about the pairwise similarity between data points [52,59]. The spectral decomposition of the graph Laplacian can then be used to locate highly connected regions within the graph [59]. Because spectral methods rely on nonlinear transformations derived from graph structure, they are highly effective at clustering datasets containing nonlinear or elongated structures [52,64]. This is in contrast to K-Means and density peak clustering (DPC) [56], which may fail on datasets containing these structures [41].

2.2.1. Spectral graph theory

In spectral graph theory, the points in X may be represented as nodes in a graph. Let the edge weight between two nodes x_i and x_j be $\mathbf{W}_{ij}$. Typically, $\mathbf{W}_{ij}$ is computed using a symmetric, radial, and rapidly decaying similarity measure such as $\mathbf{W}_{ij} = \exp\big(-\|x_i - x_j\|_2^2/\sigma^2\big)$ for some choice of scaling parameter $\sigma > 0$ that reflects the interaction radius between points [59]. If σ is large, then long-range interactions between points are considered, while if σ is small, only short-range interactions are emphasized.

One can construct a Markov transition matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$ associated to $\mathbf{W} \in \mathbb{R}^{n\times n}$ with an appropriate normalization [14,52,64]. Let the degree matrix $\mathbf{D}$ be the diagonal matrix with $\mathbf{D}_{ii} = \sum_{j=1}^n \mathbf{W}_{ij}$. We call $\mathbf{D}_{ii}$ the *degree* of the point $x_i \in X$. Let $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$. This matrix stores transition probabilities for a Markov diffusion process on the dataset, where P_{ij} reflects the probability of transitioning from x_i to x_j.[3] We assume that the Markov chain described by $\mathbf{P}$ is reversible, irreducible (i.e., the graph is connected), and aperiodic. Hence, $\mathbf{P}$ has a unique stationary distribution π satisfying $\pi\mathbf{P} = \pi$ [35]. The eigendecomposition of $\mathbf{P}$ is strongly associated with connectivity in X, making it useful for clustering. Let $\{\psi_i\}_{i=1}^n$ be the right eigenfunctions of $\mathbf{P}$ with corresponding eigenvalues $\{\lambda_i\}_{i=1}^n$. We will order eigenvalues according to $|\lambda_i|$ in non-increasing order; so when we say the "first k eigenfunctions," we refer to the k eigenfunctions

[3] We remark that, with an abuse of notation, P_{ij} denotes the entries of $\mathbf{P}$, while $\mathbf{P}_{ij}$ shall denote block submatrices of $\mathbf{P}$.

$\psi_i(x)$ corresponding to $|\lambda_i|$ closest to 1. In general, the multiplicity of the unity eigenvalue is the number of connected components in the graph [64], which is 1 by our assumption that $\mathbf{P}$ is irreducible.

Each eigenfunction $\psi_i(x)$ of $\mathbf{P}$ is also an eigenfunction of the random walk graph Laplacian $\mathbf{L}_{\text{rw}} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{W}$ corresponding to the eigenvalue $1 - \lambda_i$. The graph Laplacian is a discrete approximation of the Laplacian operator, so the eigenvectors of $\mathbf{L}_{\text{rw}}$ (and therefore $\mathbf{P}$) are discrete approximations of the continuous eigenfunctions of the Laplace operator [46,62]. Each eigenfunction $\psi_i(x)$ has a frequency related to the corresponding eigenvalue λ_i. Hence, we will say that an eigenfunction $\psi_i(x)$ of $\mathbf{P}$ is *low-frequency* if λ_i is close to 1 and *high-frequency* if $\lambda_i \ll 1$. In particular, the K lowest-frequency eigenfunctions of the graph Laplacian of X tend to concentrate on the K components of the graph that are most highly connected. This property has been used to cluster data with nonlinear structure [32,38,52,59,64].

2.2.2. Spectral clustering

Many classical clustering algorithms perform well when applied to certain classes of well-behaved data but fail on datasets with nonlinear structure [32,41,52,64]. Applying the eigenmap $\Phi(x) = (\psi_1(x), \psi_2(x), \dots, \psi_K(x))$ for $K \leq n$ as a preprocessing step before the application of K-Means often produces better separation in a new data-dependent feature space independent of nonlinear structure in X [52,59]. This is, in its essence, the spectral clustering (SC) algorithm. Typically (but not always [6]), the number of clusters K is assumed a priori and the first K eigenvectors of $\mathbf{P}$ are extracted to compute $\Phi(X)$. A simple clustering algorithm like K-Means is then applied to $\Phi(X)$ rather than X, usually after a normalization step [52]. Since SC was first introduced [52,59], its theoretical properties have been investigated [2,57–59]. It was shown that, when $K = 2$, SC produces an approximate solution to the normalized graph cut problem [59]. However, there are some classes of data for which SC has been observed to fail; for example, datasets with structure varying in scale and/or density [49].

2.3. Background on diffusion geometry

The matrix $\mathbf{P}$ is the transition matrix for a Markov diffusion process on a graph generated from the dataset X. Diffusion distances capture the structure encoded in $\mathbf{P}$ as a data-dependent distance metric between points [14,15,50].

Definition 2.1. Let $\mathbf{P}$ be an irreducible, aperiodic Markov transition matrix on $X \subset \mathbb{R}^D$ with stationary distribution π. For points $x_i, x_j \in X$ and $t \geq 0$, let $p_t(x_i, x_j) = (P^t)_{ij}$. The diffusion distance at time t between x_i and x_j is defined to be $D_t(x_i, x_j) = \|p_t(x_i, :) - p_t(x_j, :)\|_{\ell^2(1/\pi)} = \sqrt{\sum_{u \in X} [p_t(x_i, u) - p_t(x_j, u)]^2 \frac{1}{\pi(u)}}$.

Importantly, diffusion distances are data-dependent, enabling the detection of nonlinear structure in data [14,15,50]. Moreover, diffusion distances have a natural connection with the clustering problem. The diffusion distance at time t can be identified as the Euclidean distance between rows of $\mathbf{P}^t$, weighted according to $1/\pi$. If each cluster in a clustering of X is highly-connected, irreducible, and well-separated from other clusters, then $p_t(x_i, :)$ will be nearly equal to $p_t(x_j, :)$ for any pair of points x_i and x_j in the same cluster X_k, implying a low diffusion distance between points within the same cluster. Conversely, if x_i and x_j are in distinct clusters, $p_t(x_i, :)$ is expected to be very different from $p_t(x_j, :)$. This will be formalized in Section 2.5.

Definition 2.2. Let $\{(\psi_i, \lambda_i)\}_{i=1}^n$ be the right eigenvector-eigenvalue pairs of an irreducible, aperiodic transition matrix $\mathbf{P}$, sorted according to $|\lambda_i|$ in non-increasing order. The *diffusion map* at time $t \geq 0$ is defined to be $\Psi_t(x) = (\psi_1(x), \lambda_2^t \psi_2(x), \dots, \lambda_n^t \psi_n(x))$.

Diffusion maps and diffusion distances are related as $D_t(x, y) = \|\Psi_t(x) - \Psi_t(y)\|_2$ [14]. In particular, diffusion distances can be identified as Euclidean distances in a new data-dependent feature space consisting

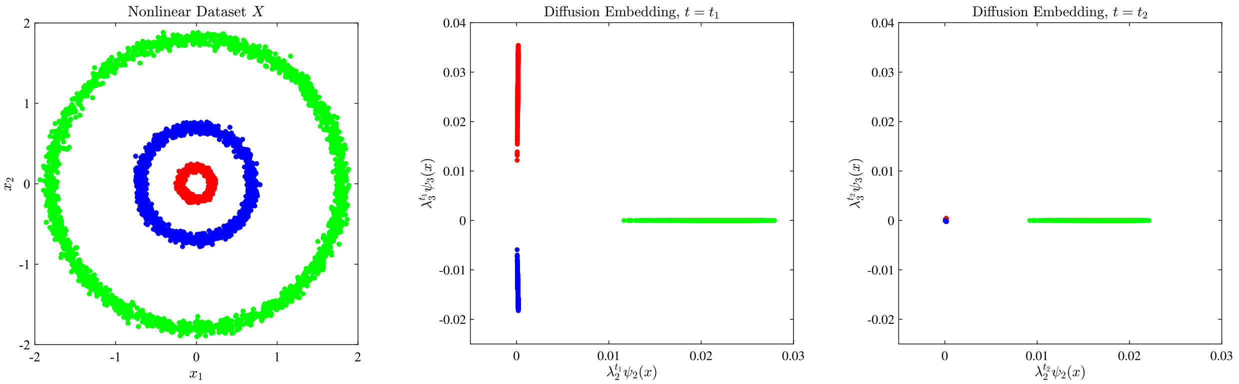


Fig. 1. Diffusion embedding of a nonlinear dataset. We plot the second and third diffusion map coordinates, which are the first coordinates of $\Psi_t(x)$ that depend on t and the data; $\lambda_1 = 1$ and ψ_1 is constant by construction. When t is small ($t = t_1$), the diffusion map sends each ring in the dataset X to a different cluster. Each ring is well-separated in the new data-dependent feature space. When t becomes large ($t = t_2$), the diffusion map sends the inner two rings to a one-point mass. This corresponds to a different scale of separation by diffusion distances. Thus, the diffusion map exhibits multiscale structure as a function of t.

of the coordinates of the diffusion map. The diffusion map can be identified as a natural extension of the eigenmap $\Phi(x)$ defined in Section 2.2.2. In $\Phi(x)$, each of the first K eigenfunctions is weighted equally in the new feature space [51,52]. Conversely, in $\Psi_t(x)$, the i^{th} eigenfunction is weighted according to λ_i^t. Thus, as t increases, the coordinates of $\Psi_t(x_i)$ corresponding to higher-frequency eigenfunctions become vanishingly small. Because lower-frequency eigenfunctions of $\mathbf{P}$ tend to concentrate on highly-connected regions in the data, this fact may facilitate the detection of different scales of structure in the data for different values of the time parameter t [15], as observed in Fig. 1.

2.4. Background on nearly reducible Markov chains

Suppose that X admits a latent clustering $\{X_k\}_{k=1}^K$. Write $\mathbf{P}$, possibly after permuting the indices of data points, as

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \dots & \mathbf{P}_{1K} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \dots & \mathbf{P}_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{K1} & \mathbf{P}_{K2} & \dots & \mathbf{P}_{KK} \end{bmatrix}, \tag{1}$$

where the block $\mathbf{P}_{kk'}$ reflects the probability of transitioning from points $x \in X_k$ to points $y \in X_{k'}$. Thus, if the mass of $\mathbf{P}$ is centered on its block diagonal, diffusion is unlikely to exit any given cluster in the latent clustering of X. Moreover, if these blocks are in some sense irreducible, diffusion will explore a cluster quickly and diffuse within it for a long period of time. The stochastic complement, defined below, provides some formalism for this intuition.

Definition 2.3. Let $\mathbf{P}$ be an irreducible, aperiodic Markov transition matrix on X, partitioned as in (1). Let $\mathbf{P}_k$ be the principal block submatrix generated by deleting the k^{th} row and column of blocks from (1). Similarly, define the matrices $\mathbf{P}_{*k} = [\mathbf{P}_{1,k}\ \ \mathbf{P}_{2,k} \dots \mathbf{P}_{k-1,k}\ \ \mathbf{P}_{k+1,k} \dots \mathbf{P}_{n,k}]^\top$ and $\mathbf{P}_{k*} = [\mathbf{P}_{k,1}\ \ \mathbf{P}_{k,2} \dots \mathbf{P}_{k,k-1}\ \ \mathbf{P}_{k,k+1} \dots \mathbf{P}_{k,n}]$. The *stochastic complement* of the submatrix $\mathbf{P}_{kk}$ of $\mathbf{P}$ is defined to be $\mathbf{S}_{kk} = \mathbf{P}_{kk} + \mathbf{P}_{k*}(\mathbf{I} - \mathbf{P}_k)^{-1}\mathbf{P}_{*k}$. The *stochastic complement* of $\mathbf{P}$ with respect to the clustering $\{X_k\}_{k=1}^K$ is defined to be the completely reducible, row-stochastic, block-diagonal matrix consisting of the stochastic complements of the diagonal blocks of $\mathbf{P}$:

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{11} & 0 & \dots & 0 \\ 0 & \mathbf{S}_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{S}_{KK} \end{bmatrix}.$$

The stochastic complement $\mathbf{S}_{kk}$ consists of two terms: $\mathbf{P}_{kk}$ and $\mathbf{P}_{k*}(\mathbf{I}-\mathbf{P}_k)^{-1}\mathbf{P}_{*k}$. The term $\mathbf{P}_{kk}$ captures the probability of directly transitioning between points in X_k, while the term $\mathbf{P}_{k*}(\mathbf{I}-\mathbf{P}_k)^{-1}\mathbf{P}_{*k}$ captures the probability of transitioning into X_k indirectly after first moving through the points in other clusters. Indeed, $(\mathbf{I}-\mathbf{P}_k)^{-1}$ can be expanded as $\sum_{t=0}^{\infty}\mathbf{P}_k^t$. Thus, the stochastic complement of $\mathbf{P}_{kk}$ encodes the probability of transitioning within X_k after a path of arbitrary length from inside or outside of X_k. The stochastic complement can be viewed as an approximation of the transition matrix $\mathbf{P}$ that contains information about a latent clustering $\{X_k\}_{k=1}^K$ of X. The following Theorem illustrates when this approximation is successful [45]. Recall $\|\mathbf{A}\|_\infty = \max_{1\le i\le n}\sum_{j=1}^n |\mathbf{A}_{ij}|$.

Theorem 2.1. *[45] Let $\mathbf{P}$ be an irreducible, aperiodic Markov transition matrix on X, partitioned as in (1). Let $\mathbf{S}$ be the stochastic complement of $\mathbf{P}$ with respect to the clustering $\{X_k\}_{k=1}^K$. Suppose each $\mathbf{S}_{kk}$ is primitive (i.e., non-negative, irreducible, and aperiodic) so that the eigenvalues of $\mathbf{S}$ are $1 = \lambda_1 = \dots = \lambda_K > |\lambda_{K+1}| > \dots > |\lambda_n| \ge 0$. Suppose that $\mathbf{Z}$ diagonalizes $\mathbf{S}$, let $\delta = \|\mathbf{P}-\mathbf{S}\|_\infty$, and let $\kappa = \|\mathbf{Z}\|_\infty\|\mathbf{Z}^{-1}\|_\infty$. Finally, let $\mathbf{S}^\infty = \lim_{t\to\infty}\mathbf{S}^t$. Then for any $t \ge 0$, $\|\mathbf{P}^t - \mathbf{S}^\infty\|_\infty \le \delta t + \kappa|\lambda_{K+1}|^t$. Moreover, if for $\epsilon > 0$, $t \in \left[\frac{\log(2\kappa/\epsilon)}{\log(1/|\lambda_{K+1}|)}, \frac{\epsilon}{2\delta}\right]$, then $\|\mathbf{P}^t - \mathbf{S}^\infty\|_\infty < \epsilon$.*

We will henceforth refer to the interval referenced in Theorem 2.1 as $\mathcal{I}_\epsilon = \left[\frac{\log(2\kappa/\epsilon)}{\log(1/|\lambda_{K+1}|)}, \frac{\epsilon}{2\delta}\right]$. The interval $\mathcal{I}_\epsilon$ is dependent not only on ϵ, but also on data-driven quantities derived from the transition matrix $\mathbf{P}$ and its stochastic complement $\mathbf{S}$: λ_{K+1}, δ, and κ. These parameters will be of importance in Section 3, where we develop a theory of multiscale clustering based on graph diffusion. If we assume—as in Theorem 2.1—that the stochastic complement $\mathbf{S}$ of $\mathbf{P}$ is block primitive and diagonalizable [41,45], then these parameters may be interpreted as follows:

- λ_{K+1}: Clearly, $\mathbf{S}$ is primitive if and only if each $\mathbf{S}_{kk}$ is primitive, so $|\lambda_{K+1}| = \max_{1\le k\le K}|\lambda_2(\mathbf{S}_{kk})|$. The second eigenvalue of an irreducible, row-stochastic matrix like $\mathbf{S}_{kk}$ is related to the conductance of the subgraph X_k of X [35,60]. Indeed, if all clusters are highly connected, $|\lambda_{K+1}|$ will be small [35,45,60]. Conversely, $|\lambda_{K+1}|$ will be near 1 if any cluster is only loosely connected.
- δ: Note that $\delta = \|\mathbf{P}-\mathbf{S}\|_\infty = 2\max_{1\le k\le K}\|\mathbf{P}_{k*}\|_\infty$ [45]. Thus, the parameter δ can be interpreted as the maximum probability across all points in X of transitioning from one cluster to another in a single time step. If transitions between any pair of clusters are likely, then δ will be large. Conversely, if transitions between all pairs of clusters are unlikely, then δ will be small. In this sense, δ measures the separation between clusters in X [41]. Since δ is the maximal probability of transitioning between clusters, it is somewhat pessimistic in datasets in which outliers from one cluster overlap with outliers from another [41]. In such datasets, δ will be large, but the probability of transitioning between points in cluster cores is still small.
- κ: By definition, κ tells us how difficult it is to diagonalize the stochastic complement $\mathbf{S}$ of $\mathbf{P}$. Suppose that the latent clustering of X is the one consisting of n singletons. Clearly, the stochastic complement would be the identity matrix, so at this extreme $\kappa = 1$. If X is sampled from a common manifold, then $\kappa = O(1)$ with respect to n [41]. If each cluster is sampled from a different common manifold, a similar result is expected to hold. However, the parameter κ is admittedly not well-studied, and research on it is still ongoing.

2.5. Background on diffusion distances in clustering

Define the (worst-case) within-cluster and between-cluster diffusion distance at time t with respect to the clustering $\{X_k\}_{k=1}^K$ by $D_t^{\text{in}} = \max\limits_{1\leq k\leq K} \max\limits_{x,y\in X_k} D_t(x,y)$ and $D_t^{\text{btw}} = \min\limits_{1\leq k<k'\leq K} \min\limits_{x\in X_k, y\in X_{k'}} D_t(x,y)$ respectively [41]. We desire a clustering of X which will yield D_t^{in} small (a coherence condition) and D_t^{btw} large (a separation condition). In this section, we review a result bounding diffusion distances within and between clusters in terms of the underlying statistical and geometric properties of $\mathbf{P}$ [41]. The following piece of machinery will prove useful in this analysis:

Definition 2.4. Let X, $\mathbf{P}$, and $\mathbf{S}^\infty$ be as in Theorem 2.1, let $p_t(x_i,x_j) = (P^t)_{ij}$, and let $s^\infty(x_i,x_j) = (S^\infty)_{ij}$. Define

$$\gamma(t) = \max_{x\in X}\left(1 - \frac{1}{2}\sum_{u\in X}\left|\frac{|p_t(x,u) - s^\infty(x,u)|}{\|p_t(x,:) - s^\infty(x,:)\|_2} - \frac{1}{\sqrt{n}}\right|^2\right)^{-1}.$$

For any vector $u \in \mathbb{R}^n$, we can write $\|u\|_2 = \frac{c_u}{\sqrt{n}}\|u\|_1$, where $c_u = \left(1 - \frac{1}{2}\sum_{i=1}^n \left|\frac{|u_i|}{\|u\|_2} - \frac{1}{\sqrt{n}}\right|^2\right)^{-1}$ [9]. Thus, $\gamma(t)$ can be identified as the maximum c_u, where the vectors u are chosen from the rows of $\mathbf{P}^t - \mathbf{S}^\infty$. In this sense, $\gamma(t)$ indicates how much the ℓ^1- and ℓ^2-norms of the rows of $\mathbf{P}^t - \mathbf{S}^\infty$ differ. Diffusion distances are written using the ℓ^2-norm, which gives the spectral decomposition. However, diffusion distances are arguably more natural in an ℓ^1-norm framework: the setting of Theorem 2.1 [17,41,45]. The function $\gamma(t)$ bridges this disconnect and enables bounding diffusion distances using results that are written in the ℓ^1-norm (e.g., Theorem 2.1) [41,45].

Theorem 2.2. *[41] Let $\{X_k\}_{k=1}^K$ be a partition of $X = \{x_i\}_{i=1}^n$. Let $\mathbf{P}$, δ, κ, and λ_{K+1} be as in Theorem 2.1, and define $s^\infty(x_i,x_j) = (S^\infty)_{ij}$. Then for any $t \geq 0$,*

$$D_t^{\text{in}} \leq \frac{2\gamma(t)}{\sqrt{n}}\left(\delta t + \kappa|\lambda_{K+1}|^t\right), \qquad D_t^{\text{btw}} \geq 2\min_{w\in X}\|s^\infty(w,:)\|_{\ell^2(1/\pi)} - \frac{2\gamma(t)}{\sqrt{n}}\left(\delta t + \kappa|\lambda_{K+1}|^t\right).$$

Moreover, if, for $\epsilon > 0$, $t \in \mathcal{I}_\epsilon$, then

$$D_t^{\text{in}} \leq \frac{2\gamma(t)}{\sqrt{n}}\epsilon, \quad D_t^{\text{btw}} \geq 2\min_{w\in X}\|s^\infty(w,:)\|_{\ell^2(1/\pi)} - \frac{2\gamma(t)}{\sqrt{n}}\epsilon.$$

For $\epsilon > 0$, if $t \in \mathcal{I}_\epsilon$, all clusters are of equal size n/K, and s^∞ is uniform on each X_k, then Theorem 2.2 implies that $D_t^{\text{in}}/D_t^{\text{btw}} = O(\epsilon)$ [41]. For ϵ small, this indicates that the maximum within-cluster diffusion distance at time t will be much less than the minimum between-cluster diffusion distance at time t. Notably, there is tension between the assumption that $t \in \mathcal{I}_\epsilon$ and the conclusion that diffusion distances at time t induce a good separation among the clusters of the clustering $\{X_k\}_{k=1}^K$ [41]. If ϵ is large, the assumption that $t \in \mathcal{I}_\epsilon$ may be lax, but the conclusion of Theorem 2.2 may be weak or even trivial. Conversely, if ϵ is small, Theorem 2.2 implies that $D_t^{\text{in}}/D_t^{\text{btw}}$ will also be small, but this strong result comes at the expense of narrowing the interval of time during which it can be attained. For fixed κ, δ, and λ_{K+1}, $\mathcal{I}_\epsilon$ shrinks to the empty set as $\epsilon \to 0^+$ [41]. In the idealized case in which there are no between-cluster transitions (so that $\delta = 0$) and each cluster is a point mass (so that $|\lambda_{K+1}| = 0$), then $\mathcal{I}_\epsilon = [0,\infty)$ [41].

2.6. The LUND clustering algorithm

The LUND algorithm (Algorithm 1) was introduced to leverage diffusion distances to cluster data [41]. This clustering algorithm locates high-density points that are far in diffusion distance from other high-density points and labels them as cluster modes. Non-modal points are then paired with a labeled point

Algorithm 1: Learning by Unsupervised Nonlinear Diffusion (LUND).

```
Input: $X$ (dataset), $\sigma$ (diffusion scale parameter), $\sigma_0$ (KDE bandwidth), $t$ (diffusion time parameter)
Output: $\mathcal{C}$ (clustering), $K$ (no. clusters)
Construct transition matrix $\mathbf{P}$ with a Gaussian kernel and diffusion scale parameter $\sigma$;
Compute the KDE $p(x)$ with KDE bandwidth $\sigma_0$ for each $x \in X$;
Compute $\rho_t(x)$ according to Definition 2.5 for each $x \in X$;
Store $\mathcal{D}_t(x) = p(x)\rho_t(x)$ for each $x \in X$;
Sort $X$ in non-increasing order according to $\mathcal{D}_t(x)$. Denote this sorting $\{x_{m_k}\}_{k=1}^n$;
Solve $K = \text{argmax}\Big\{\frac{\mathcal{D}_t(x_{m_k})}{\mathcal{D}_t(x_{m_{k+1}})}\Big\}_{k=1}^{n-1}$ and label each cluster mode, $x_{m_k}$ $(k = 1, \dots, K)$ by $\mathcal{C}(x_{m_k}) = k$;
Sort $X$ in non-increasing order according to $p(x)$. Denote this sorting $\{x_{\ell_k}\}_{k=1}^n$;
for $k = 1 : n$ do
    if $\mathcal{C}(x_{\ell_k}) = 0$ then
        $x^* = \text{argmin}_{y\in X}\{D_t(x_{\ell_k}, y) \mid p(y) \geq p(x_{\ell_k}),\ y \text{ is labeled}\}$;
        $\mathcal{C}(x_{\ell_k}) = \mathcal{C}(x^*)$;
    end
end
```

iteratively. More precisely, the LUND algorithm captures density using a kernel density estimate (KDE) $p(x) = \frac{1}{Z}\sum_{y\in NN(x,N)} \exp\big(-\|x - y\|_2^2/\sigma_0^2\big)$, where σ_0 is a KDE bandwidth, $NN(x, N)$ is the set of N ℓ^2-nearest neighbors of x, and Z is a normalization constant such that $p(x)$ sums to one [41]. To capture diffusion geometry, we introduce a different function:

Definition 2.5. Let X and $\mathbf{P}$ be as in Theorem 2.1, and let $p(x)$ be a KDE of X. Define

$$\rho_t(x) = \begin{cases} \min_{y\in X}\{D_t(x,y) \mid p(y) \geq p(x)\} & x \neq \text{argmax}_{y\in X} p(y), \\ \max_{y\in X} D_t(x,y) & x = \text{argmax}_{y\in X} p(y). \end{cases}$$

Thus, $\rho_t(x)$ assigns x the diffusion distance at time t between x and its D_t-nearest neighbor of higher density. The LUND algorithm then analyzes $\mathcal{D}_t(x) = p(x)\rho_t(x)$. The maximizers of $\mathcal{D}_t(x)$ tend to be high in empirical density and far in diffusion distance from other high-density points, making them suitable choices as cluster modes. The function $\mathcal{D}_t(x)$ can also be used to estimate the number of latent clusters in X [41]. While K-Means and SC can estimate the number of latent clusters K via the scree plot [10] and eigengap [37], respectively, these estimates of K have been shown to fail on data classes in which the LUND estimate succeeds (e.g., datasets with nonlinear structure for the scree plot and datasets with multimodal bottleneck structure for the eigengap [38,41]). It has been shown that, under plausible assumptions on cluster structure and density, the estimate provided by the LUND algorithm on the number of clusters in the dataset is accurate, even for datasets with these problematic structures [41]. We review theoretical guarantees on the performance of the LUND algorithm in Section 4.3.1.

The LUND algorithm relies on the diffusion time parameter t when calculating the diffusion distance between points. As discussed in Section 2.3, this parameter tends to affect the scale of a clustering separable by diffusion distances [14,50]. Thus, as t varies, the clustering that the LUND algorithm estimates will change. To improve the LUND algorithm, it is necessary to understand how its cluster assignments change as a function of t. A better theoretical understanding of how the diffusion process changes may enable the elimination of the dependence on t as well as a deeper understanding of which time scale yields the most representative clustering of X.

3. A multiscale environment for learning by diffusion

Theorem 2.2 states that diffusion distances induce strong separation on a latent clustering during an interval in the diffusion process. However, it is limited in that it only considers a fixed scale and a single latent clustering. Many datasets exhibit multiscale structure with many partitions that could be considered “correct” and useful [1,15,19,61]. In this section, we will generalize Theorem 2.2 by allowing multiple latent clusterings, varying in scale, to exist within the same dataset [41]. We will then introduce the MELD data model, which parameterizes the clusterings of X by t.

Suppose there are M latent clusterings of X, denoted $\{X_k^{(\ell)}\}_{k=1}^{K_\ell}$ for $\ell \in \{1, \dots, M\}$. We will not require that these clusterings are hierarchical but consider that special case in Section 3.2. For $1 \leq \ell \leq M$, define the submatrices $\mathbf{P}_{kk'}^{(\ell)}$ of the transition matrix $\mathbf{P}$, possibly after permuting the indices of data points, implicitly by

$$\mathbf{P}^{(\ell)} = \begin{bmatrix} \mathbf{P}_{11}^{(\ell)} & \mathbf{P}_{12}^{(\ell)} & \dots & \mathbf{P}_{1K_\ell}^{(\ell)} \\ \mathbf{P}_{21}^{(\ell)} & \mathbf{P}_{22}^{(\ell)} & \dots & \mathbf{P}_{2K_\ell}^{(\ell)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{K_\ell 1}^{(\ell)} & \mathbf{P}_{K_\ell 2}^{(\ell)} & \dots & \mathbf{P}_{K_\ell K_\ell}^{(\ell)} \end{bmatrix}, \tag{2}$$

where the block $\mathbf{P}_{kk'}^{(\ell)}$ reflects the probability of transitioning from points $x \in X_k^{(\ell)}$ to points $y \in X_{k'}^{(\ell)}$. In particular, the block matrix $\mathbf{P}_{kk}^{(\ell)}$ reflects the probability of remaining in the cluster $X_k^{(\ell)}$ in the ℓ^{th} latent clustering of X, while $\mathbf{P}_{kk'}^{(\ell)}$ reflects the probability of transitioning from the cluster $X_k^{(\ell)}$ to $X_{k'}^{(\ell)}$ in the ℓ^{th} latent clustering of X.

The stochastic complement of $\mathbf{P}$ depends on the clustering assumed a priori. Thus, for each of the latent clusterings of X, a different stochastic complement can be extracted. We will refer to the stochastic complement of the submatrix $\mathbf{P}_{kk}^{(\ell)}$ as $\mathbf{S}_{kk}^{(\ell)}$ and the stochastic complement of $\mathbf{P}$ with respect to the ℓ^{th} clustering of X as $\mathbf{S}^{(\ell)}$. Let $\mathbf{S}_\infty^{(\ell)} = \lim_{t\to\infty}[\mathbf{S}^{(\ell)}]^t$. Similar to the case in which there was only one latent clustering, the stochastic complement $\mathbf{S}_{kk}^{(\ell)}$ may be interpreted as capturing the probability of transitioning into the cluster $X_k^{(\ell)}$, either directly from inside of $X_k^{(\ell)}$ or indirectly after a path of arbitrary length starting outside of $X_k^{(\ell)}$ [41,45]. As before, we require $\mathbf{S}_{kk}^{(\ell)}$ to be primitive and diagonalizable for each $k \in \{1, \dots, K_\ell\}$ and $\ell \in \{1, \dots, M\}$. Denote the invertible $n \times n$ matrix that diagonalizes $\mathbf{S}^{(\ell)}$ as $\mathbf{Z}^{(\ell)}$ [41,45].

The interval $\mathcal{I}_\epsilon$ is regulated by three constants—λ_{K+1}, δ, and κ—each derived from the stochastic complement of $\mathbf{P}$ corresponding to a clustering. Therefore, the interval $\mathcal{I}_\epsilon$ will change as a function of the scale of clustering.

Definition 3.1. Let $\mathbf{P}$ be an aperiodic, irreducible Markov transition matrix on X, partitioned as in (2). Let $\mathbf{S}^{(\ell)}$ be the stochastic complement of $\mathbf{P}$ with respect to the ℓ^{th} clustering of X. Define $\lambda_{K_\ell+1}^{(\ell)} = \lambda_{K_\ell+1}[\mathbf{S}^{(\ell)}]$, $\delta^{(\ell)} = \|\mathbf{P} - \mathbf{S}^{(\ell)}\|_\infty$ and $\kappa^{(\ell)} = \|\mathbf{Z}^{(\ell)}\|_\infty \|[\mathbf{Z}^{(\ell)}]^{-1}\|_\infty$. For $\epsilon > 0$, define the interval $\mathcal{I}_\epsilon^{(\ell)} = \left[\frac{\log(2\kappa^{(\ell)}/\epsilon)}{\log(1/|\lambda_{K_\ell+1}^{(\ell)}|)}, \frac{\epsilon}{2\delta^{(\ell)}}\right]$.

We will refer to the maximum within-cluster and minimum between-cluster diffusion distance at time t for the clustering $\{X_k^{(\ell)}\}_{k=1}^{K_\ell}$ as $D_t^{\text{in}}(\ell)$ and $D_t^{\text{btw}}(\ell)$ respectively. We argued in Section 2.3 that the dependence of diffusion distances on the diffusion time parameter affects what scales of structure can be uncovered by diffusion distances [14,50]. Thus, for any fixed t, the ratio $D_t^{\text{in}}(\ell)/D_t^{\text{btw}}(\ell)$ may be large for some ℓ and small for others. In the following definition, the notion of strong separation by diffusion distances at a given time scale is formalized.

Definition 3.2. Let $\epsilon > 0$. A clustering $\{X_k^{(\ell)}\}_{k=1}^{K_\ell}$ is *ϵ-separable by diffusion distances at time t* if $\frac{D_t^{\text{in}}(\ell)}{D_t^{\text{btw}}(\ell)} \leq \frac{\epsilon}{1/\sqrt{n}-\epsilon}$.

The criterion for ϵ-separation by diffusion distances is related to the notion of a perfect clustering. A clustering $\{X_k\}_{k=1}^K$ is said to be *perfect* under the metric $m(:,:)$ if there is an $r > 0$ for which the maximum within-cluster distance is at most r and the minimum between-cluster distance is at least $4r$, where distance is measured using m [42,65]. More precisely, if $0 < \epsilon \leq \frac{1}{5\sqrt{n}}$, a clustering that is ϵ-separable by diffusion distances at time t is also perfect under the metric D_t. In datasets with a perfect partition, cluster structure may be detected with K-Means using the metric D_t (if r is unknown) or by thresholding a minimum spanning tree (if r is known) [42,65].

Let $\gamma^{(\ell)}(t)$ be the multiscale extension of $\gamma(t)$, where $s^{\infty}(x_i, x_j)$ is replaced by $s_{\infty}^{(\ell)}(x_i, x_j) = (S_{\infty}^{(\ell)})_{ij}$ in Definition 2.4. This function measures how much the ℓ^1-norm of rows in $\mathbf{P}^t - \mathbf{S}_{\infty}^{(\ell)}$ differs from the ℓ^2-norm of rows in $\mathbf{P}^t - \mathbf{S}_{\infty}^{(\ell)}$. As was the case for $\gamma(t)$, $1 \leq \gamma^{(\ell)}(t) \leq \sqrt{n}$ for any t and $\ell \in \{1, \dots, M\}$. Using the established notation, we are able to provide the following Corollary, which serves as a multiscale extension of Theorem 2.2.

Corollary 3.1. *Let* $\mathbf{P}$ *be an aperiodic, irreducible Markov transition matrix on a dataset* X*, partitioned as in (2). Let* $\ell \in \{1, \dots, M\}$ *be a fixed clustering scale, and let* $\mathbf{S}^{(\ell)}$ *be the stochastic complement of* $\mathbf{P}$ *with respect to the clustering* $\{X_k^{(\ell)}\}_{k=1}^{K_\ell}$*. Let* $\delta^{(\ell)}$*,* $\kappa^{(\ell)}$*, and* $\lambda_{K_\ell+1}^{(\ell)}$ *be the geometric constants introduced in Definition 3.1, and let* $s_{\infty}^{(\ell)}(x_i, x_j) = (S_{\infty}^{(\ell)})_{ij}$*.*

(a) For any $t \geq 0$*,*

$$D_t^{\text{in}}(\ell) \leq \frac{2\gamma^{(\ell)}(t)}{\sqrt{n}}\left(\delta^{(\ell)} t + \kappa^{(\ell)} |\lambda_{K_\ell+1}^{(\ell)}|^t\right);$$

$$D_t^{\text{btw}}(\ell) \geq \min_{w \in X} \|s_{\infty}^{(\ell)}(w,:)\|_{\ell^2(1/\pi)} - \frac{2\gamma^{(\ell)}(t)}{\sqrt{n}}\left(\delta^{(\ell)} t + \kappa^{(\ell)} |\lambda_{K_\ell+1}^{(\ell)}|^t\right).$$

Moreover, if, for $\epsilon > 0$*,* $t \in \mathcal{I}_{\epsilon}^{(\ell)}$*, then*

$$D_t^{\text{in}}(\ell) \leq \frac{2\gamma^{(\ell)}(t)}{\sqrt{n}}\epsilon; \qquad D_t^{\text{btw}}(\ell) \geq 2\min_{w \in X} \|s_{\infty}^{(\ell)}(w,:)\|_{\ell^2(1/\pi)} - \frac{2\gamma^{(\ell)}(t)}{\sqrt{n}}\epsilon.$$

(b) If $\epsilon < 1/\sqrt{n}$*, then the clustering* $\{X_k^{(\ell)}\}_{k=1}^{K_\ell}$ *is* ϵ*-separable by diffusion distances at times* $t \in \mathcal{I}_{\epsilon}^{(\ell)}$*.*

Proof. Theorem 2.2 gives (a) immediately [41]. To obtain (b), note first that $\min_{w\in X} \|s_{\infty}^{(\ell)}(w,:)\|_{\ell^2(1/\pi)}$ can be bounded from below: $\frac{1}{\sqrt{n}} = \frac{1}{\sqrt{n}} \min_{w\in X} \|s_{\infty}^{(\ell)}(w,:)\|_1 \leq \min_{w\in X} \|s_{\infty}^{(\ell)}(w,:)\|_2 \leq \min_{w\in X} \|s_{\infty}^{(\ell)}(w,:)\|_{\ell^2(1/\pi)}$. The assumption $\epsilon < \frac{1}{\sqrt{n}}$ implies $\epsilon < \frac{1}{\sqrt{n}} \leq \frac{1}{\gamma^{(\ell)}(t)} \leq \frac{\sqrt{n}}{\gamma^{(\ell)}(t)} \min_{w\in X} \|s_{\infty}^{(\ell)}(w,:)\|_{\ell^2(1/\pi)}$. Rearranging yields $2\min_{w\in X} \|s_{\infty}^{(\ell)}(w,:)\|_{\ell^2(1/\pi)} > \frac{2\gamma^{(\ell)}(t)}{\sqrt{n}}\epsilon$, so the lower bound on $D_t^{\text{btw}}(\ell)$ given in (a) is positive for $t \in \mathcal{I}_{\epsilon}^{(\ell)}$. Thus,

$$\begin{aligned}
\frac{D_t^{\text{in}}(\ell)}{D_t^{\text{btw}}(\ell)} &\leq \frac{2\gamma^{(\ell)}(t)\epsilon/\sqrt{n}}{2\min_{w\in X} \|s_{\infty}^{(\ell)}(w,:)\|_{\ell^2(1/\pi)} - 2\gamma^{(\ell)}(t)\epsilon/\sqrt{n}} \\
&= \frac{\gamma^{(\ell)}(t)\epsilon}{\sqrt{n}\min_{w\in X} \|s_{\infty}^{(\ell)}(w,:)\|_{\ell^2(1/\pi)} - \gamma^{(\ell)}(t)\epsilon} \\
&\leq \frac{\epsilon}{\min_{w\in X} \|s_{\infty}^{(\ell)}(w,:)\|_{\ell^2(1/\pi)} - \epsilon} \\
&\leq \frac{\epsilon}{1/\sqrt{n} - \epsilon},
\end{aligned}$$

where $\gamma^{(\ell)}(t) \leq \sqrt{n}$ was used to obtain the second to last inequality and $\min_{w\in X} \|s_{\infty}^{(\ell)}(w,:)\|_{\ell^2(1/\pi)} \geq 1/\sqrt{n}$ was used to obtain the last inequality. □

The proof of Corollary 3.1 suggests that the notion of ϵ-separation by diffusion distances is somewhat pessimistic, as it relies on worst-case assumptions on the behavior of the parameters $\gamma^{(\ell)}(t)$ and $\min_{w\in X}\|s_\infty^{(\ell)}(w,:)\|_{\ell^2(1/\pi)}$. In practice, if $t\in\mathcal{I}_\epsilon^{(\ell)}$, the rows of $\mathbf{P}^t-\mathbf{S}_\infty^{(\ell)}$ tend to be nearly uniform, so $\gamma^{(\ell)}(t)=O(1)$ with respect to n [41]. Similarly, $\min_{w\in X}\|s_\infty^{(\ell)}(w,:)\|_{\ell^2(1/\pi)}\geq 1/\sqrt{n}$ is a worst-case lower bound. If the rows of $\mathbf{P}^t-\mathbf{S}_\infty^{(\ell)}$ are completely uniform, then $\gamma^{(\ell)}(t)=1$ [41]. Assuming this is the case for each $t\in\mathcal{I}_\epsilon^{(\ell)}$, where $\epsilon\in(0,1)$, the lower bound of $D_t^{\text{btw}}(\ell)$ in (a) of Corollary 3.1 is positive. Hence,

$$\frac{D_t^{\text{in}}(\ell)}{D_t^{\text{btw}}(\ell)}\leq\frac{\gamma^{(\ell)}(t)\epsilon}{\sqrt{n}\min_{w\in X}\|s_\infty^{(\ell)}(w,:)\|_{\ell^2(1/\pi)}-\gamma^{(\ell)}(t)\epsilon}=\frac{\epsilon}{\sqrt{n}\min_{w\in X}\|s_\infty^{(\ell)}(w,:)\|_{\ell^2(1/\pi)}-\epsilon}\leq\frac{\epsilon}{1-\epsilon},$$

where we have used $\min_{w\in X}\|s_\infty^{(\ell)}(w,:)\|_{\ell^2(1/\pi)}\geq 1/\sqrt{n}$ to obtain the last inequality. This is clearly a tighter bound than the one required for ϵ-separation and is notably independent of n. For $\epsilon\ll 1$, this new inequality implies that diffusion distances at times $t\in\mathcal{I}_\epsilon^{(\ell)}$ will induce excellent separation on the clusters in the ℓ^{th} clustering.

By Corollary 3.1, there are M intervals in the diffusion process, during each of which a different clustering is ϵ-separable by diffusion distances. If two distinct intervals $\mathcal{I}_\epsilon^{(\ell)}$ and $\mathcal{I}_\epsilon^{(\ell')}$ ever overlapped for some fixed $\epsilon\in\left(0,\frac{1}{\sqrt{n}}\right)$, time steps t would exist during which multiple clusterings are ϵ-separable by diffusion distances at the same time. We therefore make the simplifying assumption that, if $\ell\neq\ell'$ and ϵ are fixed, $\mathcal{I}_\epsilon^{(\ell)}\bigcap\mathcal{I}_\epsilon^{(\ell')}=\varnothing$ so that the intervals $\mathcal{I}_\epsilon^{(\ell)}$ do not intersect. Thus, at times $t\in\mathcal{I}_\epsilon^{(\ell)}$, $\{X_k^{(\ell)}\}_{k=1}^{K_\ell}$ is the unique clustering that is ϵ-separable by diffusion distances as a result of Corollary 3.1. This is the basis of the MELD data model, wherein the unique latent partitions of X are parameterized by the diffusion time parameter.

Definition 3.3. Let X be a dataset with M distinct latent clusterings $\{X_k^{(\ell)}\}_{k=1}^{K_\ell}$ for $1\leq\ell\leq M$. Fix $\epsilon\in\left(0,\frac{1}{\sqrt{n}}\right)$, and assume that the intervals $\mathcal{I}_\epsilon^{(\ell)}$ are nonintersecting. For each $t\geq 0$, if $t\in\mathcal{I}_\epsilon^{(\ell)}$ for some $\ell\in\{1,\dots,M\}$, then we define $\mathcal{C}_t=\{X_k^{(\ell)}\}_{k=1}^{K_\ell}$ to be the clustering that is ϵ-separable by diffusion distances at time t as a result of Corollary 3.1. Define the *Multiscale Environment for Learning by Diffusion (MELD)* data model for this choice of ϵ to be $MELD_\epsilon(X)=\left\{\mathcal{C}_t\;\middle|\;t\in\mathcal{I}_\epsilon^{(\ell)}\textit{ for some }\ell\in\{1,\dots,M\}\right\}$.

The MELD data model is similar in spirit to the *diffusion geometry model* (*DGM*), which also assumes the existence of M latent clusterings of X: $\{\mathcal{C}^{(\ell)}\}_{\ell=1}^M$ [17]. In the DGM, the stochastic complement of $\mathbf{P}$ with respect to each of the M clusterings is extracted, along with the corresponding geometric constants: $\{(\lambda_{K_\ell+1}^{(\ell)},\delta^{(\ell)},\kappa^{(\ell)})\}_{\ell=1}^M$. The DGM was used to aggregate information about the many latent scales of structure in the dataset into a single distance metric [17]. On the other hand, the MELD data model provides a fine-scale view of how latent cluster structure changes as a function of the diffusion process. Corollary 3.1 guarantees that a clustering $\mathcal{C}_t\in\text{MELD}_\epsilon(X)$ is ϵ-separable by diffusion distances at time t. In this sense, a clustering $\mathcal{C}_t$ in the MELD data model can be interpreted as the latent clustering of X at time t in the diffusion process.

Notably, the union of the intervals $\mathcal{I}_\epsilon^{(\ell)}$ may not be $[0,\infty)$. If there are time steps t not contained in any interval $\mathcal{I}_\epsilon^{(\ell)}$, then diffusion distances at those time steps may not induce an ϵ-separation on any clustering whatsoever. The time steps between any two intervals $\mathcal{I}_\epsilon^{(\ell)}$ and $\mathcal{I}_\epsilon^{(\ell')}$ can be thought of as transition regions between two latent clusterings. Transition regions are, in datasets with very well-defined multiscale cluster structure, short intervals during which $\mathbf{P}$ is rapidly mixing and transitioning between states. During these transition regions, there is not necessarily a “true” latent clustering of X. Therefore, the MELD data model naturally only captures the time steps at which diffusion distances yield strong separation of clusters in a clustering of X. We will refer to the time steps during which a latent clustering of X is ϵ-separable by diffusion distances as $A_\epsilon=\bigcup_{\ell=1}^M\mathcal{I}_\epsilon^{(\ell)}$.

3.1. Stability in the MELD data model

A cluster can be viewed as a region of the graph on which diffusion is unlikely to exit [44]. The duration the random walk is "trapped" on the cluster, aggregated across clusters, can be interpreted as a clustering's stability.

Definition 3.4. Fix $\epsilon \in \left(0, \frac{1}{\sqrt{n}}\right)$ and let $MELD_\epsilon(X)$ be as in Definition 3.3. Let $\mathcal{C}_t$ and $\mathcal{C}_s$ be clusterings in $MELD_\epsilon(X)$ with $t \in \mathcal{I}_\epsilon^{(\ell)}$ and $s \in \mathcal{I}_\epsilon^{(\ell')}$. We say that $\mathcal{C}_t$ is more ϵ-*stable* than $\mathcal{C}_s$ if

$$\log\left[\frac{\epsilon}{2\delta^{(\ell)}} - \frac{\log(2\kappa^{(\ell)}/\epsilon)}{\log(1/|\lambda_{K_\ell+1}^{(\ell)}|)}\right] \geq \log\left[\frac{\epsilon}{2\delta^{(\ell')}} - \frac{\log(2\kappa^{(\ell')}/\epsilon)}{\log(1/|\lambda_{K_{\ell'}+1}^{(\ell')}|)}\right].$$

Thus, a clustering $\mathcal{C}_t$ is considered more ϵ-stable than $\mathcal{C}_s$ if the interval of time during which $\mathcal{C}_t$ is ϵ-separable by diffusion distances is longer on a logarithmic scale than the interval of time during which $\mathcal{C}_s$ is ϵ-separable by diffusion distances. We examine stability on a logarithmic scale because of the exponential dependence of diffusion distances on the spectrum of $\mathbf{P}$. Transitions between clusterings of X typically occur after a component of $\Psi_t(x)$ is sent to zero. Because each component of $\Psi_t(x)$ converges exponentially to zero, the length of intervals $\mathcal{I}_\epsilon^{(\ell)}$ later in the diffusion process tends to be exponentially longer than that of intervals early in the diffusion process. Thus, if we choose to examine stability on a linear scale (rather than logarithmic), a clustering that is ϵ-separable by diffusion distances later in the diffusion process will tend to be more ϵ-stable than a clustering that is ϵ-separable by diffusion distances early in the diffusion process. Taking logarithms allows for a more fair comparison between clusterings that are ϵ-separable at different stages in the diffusion process. The connection between the stability and geometry of a clustering is explored in Proposition 3.1.

Proposition 3.1. *Fix* $\epsilon \in \left(0, \frac{1}{\sqrt{n}}\right)$, *and let* $\mathcal{C}_t, \mathcal{C}_s \in \text{MELD}_\epsilon(X)$ *for* $t \in \mathcal{I}_\epsilon^{(\ell)}$ *and* $s \in \mathcal{I}_\epsilon^{(\ell')}$. *If* $|\lambda_{K_\ell+1}^{(\ell)}| \leq |\lambda_{K_{\ell'}+1}^{(\ell')}|$, $\delta^{(\ell)} \leq \delta^{(\ell')}$, *and* $\kappa^{(\ell)} = \kappa^{(\ell')}$, *then* $\mathcal{C}_t$ *is more* ϵ*-stable than* $\mathcal{C}_s$.

Proof. If $\delta^{(\ell)} \leq \delta^{(\ell')}$, then $\frac{\epsilon}{2\delta^{(\ell)}} \geq \frac{\epsilon}{2\delta^{(\ell')}}$. Similarly, if $|\lambda_{K_{\ell'}+1}^{(\ell')}| \geq |\lambda_{K_\ell+1}^{(\ell)}|$, then $\frac{1}{\log(1/|\lambda_{K_{\ell'}+1}^{(\ell')}|)} \geq \frac{1}{\log(1/|\lambda_{K_\ell+1}^{(\ell)}|)}$. By our assumption that $\kappa^{(\ell)} = \kappa^{(\ell')}$, this implies that $\frac{\log(2\kappa^{(\ell')}/\epsilon)}{\log(1/|\lambda_{K_{\ell'}+1}^{(\ell')}|)} \geq \frac{\log(2\kappa^{(\ell)}/\epsilon)}{\log(1/|\lambda_{K_\ell+1}^{(\ell)}|)}$. Thus, $\frac{\epsilon}{2\delta^{(\ell)}} - \frac{\log(2\kappa^{(\ell)}/\epsilon)}{\log(1/|\lambda_{K_\ell+1}^{(\ell)}|)} \geq \frac{\epsilon}{2\delta^{(\ell')}} - \frac{\log(2\kappa^{(\ell')}/\epsilon)}{\log(1/|\lambda_{K_{\ell'}+1}^{(\ell')}|)}$. Taking logarithms on both sides yields the result. □

Proposition 3.1 implies that if the clusters in a clustering $\mathcal{C}_t \in \text{MELD}_\epsilon(X)$ with $t \in \mathcal{I}_\epsilon^{(\ell)}$ are better separated (so that $\delta^{(\ell)}$ is small) and more coherent (so that $|\lambda_{K_\ell+1}^{(\ell)}|$ is small) than the clusters in a different clustering $\mathcal{C}_s \in \text{MELD}_\epsilon(X)$, then $\mathcal{C}_t$ will be more ϵ-stable and appear more frequently than $\mathcal{C}_s$ in the MELD data model. This implies that clusterings with well-separated, coherent clusters are emphasized within the MELD data model.

3.2. Applications of the MELD data model to hierarchical clustering

In this section, we investigate the relationship between the clusterings in the MELD data model and the diffusion time parameter in the case that the MELD data model exhibits hierarchical structure.

Definition 3.5. Let $\mathcal{C} = \{X_k\}_{k=1}^K$ and $\mathcal{C}' = \{X_k'\}_{k=1}^{K'}$ be clusterings of X. The clustering $\mathcal{C}$ is a *refinement* of the clustering $\mathcal{C}'$ if $K \geq K'$ and if, for every cluster $X_k' \in \mathcal{C}'$, $X_k' = \bigcup_{j=1}^m X_{k_j}$ for some subsequence $\{k_j\}_{j=1}^m$

of $\{1, 2 \ldots, K\}$. The family of clusterings $\mathscr{C} = \{\mathcal{C}_\alpha\}_{\alpha \in A}$ *exhibits hierarchical structure* if, for each pair $\mathcal{C}_\alpha$ and $\mathcal{C}_\beta$ in $\mathscr{C}$, either $\mathcal{C}_\alpha$ is a refinement of $\mathcal{C}_\beta$ or $\mathcal{C}_\beta$ is a refinement of $\mathcal{C}_\alpha$.

Thus, the MELD data model exhibits hierarchical structure if, for each pair of coarse and fine clusterings in the model, any cluster in the coarse clustering can be expressed as the union of clusters from the fine clustering. In general, the MELD data model does not assume hierarchical structure because not all multiscale cluster structure is hierarchical. Indeed, one of the advantages of the MELD data model is its ability to capture non-hierarchical multiscale structure in data. Nevertheless, the assumption that the MELD data model exhibits hierarchical structure does provide us with the ability to provide concrete analysis about the structure of the MELD data model.

Lemma 3.1. *Fix* $\epsilon \in \left(0, \frac{1}{\sqrt{n}}\right)$, *and let* $\mathcal{C}_t, \mathcal{C}_s \in \text{MELD}_\epsilon(X)$, *where* $t \in \mathcal{I}_\epsilon^{(\ell)}$ *and* $s \in \mathcal{I}_\epsilon^{(\ell')}$. *If* $\mathcal{C}_t$ *is a refinement of* $\mathcal{C}_s$, *then* $\delta^{(\ell)} \leq \delta^{(\ell')}$.

Proof. Let $\mathbf{P}_{k*}^{(\ell)} = [\mathbf{P}_{k,1}^{(\ell)}\ \mathbf{P}_{k,2}^{(\ell)} \ldots \mathbf{P}_{k,k-1}^{(\ell)}\ \mathbf{P}_{k,k+1}^{(\ell)} \ldots \mathbf{P}_{k,n}^{(\ell)}]$ for each $k \in \{1, \ldots, K_\ell\}$ and $\ell \in \{1, \ldots, M\}$. By assumption, the clusters in $\{X_k^{(\ell)}\}_{k=1}^{K_\ell}$ can be merged to form the any of the clusters in $\{X_k^{(\ell')}\}_{k=1}^{K_{\ell'}}$. So, for all block rows k, any $\mathbf{P}_{k*}^{(\ell)}$ is a submatrix of some $\mathbf{P}_{j*}^{(\ell')}$. Therefore, for each $k \in \{1, \ldots, K_\ell\}$, $\|\mathbf{P}_{k*}^{(\ell)}\|_\infty \leq \|\mathbf{P}_{j*}^{(\ell')}\|_\infty$ for some $j \in \{1 \ldots, K_{\ell'}\}$. Thus, $\delta^{(\ell')} = 2\max_{1 \leq k \leq K_{\ell'}} \|\mathbf{P}_{k*}^{(\ell')}\|_\infty \leq 2\max_{1 \leq k \leq K_\ell} \|\mathbf{P}_{k*}^{(\ell)}\|_\infty = \delta^{(\ell)}$. □

As diffusion progresses, diffusion distances separate ever coarser structure within the dataset. Because the i^{th} component of the diffusion map $\Psi_t(x)$ is weighted by λ_i^t, each eigenfunction's contribution to diffusion distances will decay exponentially with t. As low-frequency eigenfunctions are annihilated, different mesoscopic equilibria will arise, during which diffusion distances induce different clusterings on X. Moreover, because fewer low-frequency eigenfunctions contribute to diffusion distances as t increases, it is reasonable to expect the latent structure separated by diffusion distances to go from fine to coarse in scale.

Proposition 3.2. *Fix* $\epsilon \in \left(0, \frac{1}{\sqrt{n}}\right)$ *and let* K_t *denote the number of clusters in the clustering* $\mathcal{C}_t \in \text{MELD}_\epsilon(X)$. *If* $\text{MELD}_\epsilon(X)$ *exhibits hierarchical structure, then* K_t *is monotonically non-increasing during* A_ϵ.

Proof. Let $\mathcal{C}_t, \mathcal{C}_s \in \text{MELD}_\epsilon(X)$ be any two distinct clusterings of X. We will show that if $K_t > K_s$, then $t < s$. Because $\mathcal{C}_t, \mathcal{C}_s \in \text{MELD}_\epsilon(X)$, there are intervals $\mathcal{I}_\epsilon^{(\ell)}$ and $\mathcal{I}_\epsilon^{(\ell')}$ such that $t \in \mathcal{I}_\epsilon^{(\ell)}$ and $s \in \mathcal{I}_\epsilon^{(\ell')}$. Since $\text{MELD}_\epsilon(X)$ exhibits hierarchical structure and $K_t > K_s$, $\mathcal{C}_t$ is a refinement of $\mathcal{C}_s$. By Lemma 3.1, $\delta^{(\ell')} \geq \delta^{(\ell)}$, which reduces to $\frac{\epsilon}{2\delta^{(\ell)}} \leq \frac{\epsilon}{2\delta^{(\ell')}}$. By the assumption that $\mathcal{I}_\epsilon^{(\ell)} \cap \mathcal{I}_\epsilon^{(\ell')} = \varnothing$, this implies that $\mathcal{I}_\epsilon^{(\ell)}$ is earlier in the diffusion process than $\mathcal{I}_\epsilon^{(\ell)}$. In particular, $t < s$, as desired. Thus, for any pair of clusterings $\mathcal{C}_t, \mathcal{C}_s \in \text{MELD}_\epsilon(X)$, if $K_s < K_t$, then $\mathcal{C}_t$ is ϵ-separable by diffusion distances earlier in the diffusion process than $\mathcal{C}_s$. We conclude that K_t is monotonically non-increasing as a function of the diffusion time parameter t during A_ϵ, as desired. □

3.3. Comparison to related models

In order to understand MELD in greater detail, we compare to two related data models.

3.3.1. Geometric data model

The geometric data model (GDM) models X by assuming points are sampled from a probability measure $\mu = \sum_{k=1}^K w_k \mu_k$, where each μ_k is itself a probability measure on X and $\sum_{k=1}^K w_k = 1$. Each μ_k is assumed to be supported on some subset of $\mathbb{R}^D$, which is allowed to be nonlinear, nonconvex, and multimodal. Typically, separation and coherence conditions are imposed on $\{\mu_k\}_{k=1}^K$ (e.g., if $k \neq k'$, the support of μ_k

does not overlap too much with that of $\mu_{k'}$), but connections are strong between data points sampled from each μ_k. The GDM is non-parametric and assumes very little about the distributions $\{\mu_k\}_{k=1}^K$. Moreover, the assumption that each cluster is sampled from a different distribution leads to a simple interpretation of the clustering problem: to recover the correct index of the distribution μ_k from which each $x_i \in X$ was sampled given solely the information provided by the dataset. However, the GDM requires there to be but one latent clustering to be learned, even though many datasets exhibit multiscale structure in practice. In contrast, the MELD data model allows for many different scales of cluster analysis, in some sense generalizing the GDM. Moreover, the GDM assumes a latent distribution on the data itself. While this generality offers significant theoretical advantages, it is also not always clear which clustering algorithm can best recover the correct distribution μ_k from which each data point was sampled [3,4,12,58,63,67,68].

We note that the assumptions of the GDM can be modified to allow for multiscale structure. Indeed, suppose that for scales $\ell \in \{1, 2, \dots M\}$ that $\mu = \sum_{k=1}^{K_\ell} w_k^{(\ell)} \mu_k^{(\ell)}$ for measures $\{\mu_k^{(\ell)}\}_{k=1}^{K_\ell}$ where, naturally, $\sum_{k=1}^{K_\ell} w_k^{(\ell)} = 1$. Hierarchical structure can be accounted for by requiring, for each $1 \leq \ell < M-1$ and each $k \in \{1, \dots, K_{\ell+1}\}$,

$$\mu_k^{(\ell+1)} = \sum_{k' \in I_k^{(\ell+1)}} \left(\frac{w_{k'}^{(\ell)}}{\sum_{k' \in I_k^{(\ell+1)}} w_{k'}^{(\ell)}} \right) \mu_{k'}^{(\ell)}, \quad w_k^{(\ell+1)} = \sum_{k' \in I_k^{(\ell+1)}} w_{k'}^{(\ell)}$$

for some index set $I_k^{(\ell+1)} \subset \{1, 2, \dots, K_\ell\}$, where $\{I_k^{(\ell+1)}\}_{k=1}^{K_{\ell+1}}$ is a partition of $\{1, 2, \dots, K_\ell\}$. The map $(k, \ell) \mapsto I_k^{(\ell+1)}$ may be understood as mapping the *parent* k at scale $\ell+1$ to the *children* $I_k^{(\ell+1)}$ at scale ℓ. We will call this data model the *multiscale geometric data model* (*M-GDM*). The M-GDM gives an avenue for understanding a continuous analogue to the MELD data model, where the concentration of each $\mu_k^{(\ell)}$ is related to the cluster coherence parameter $\lambda_{K_\ell+1}^{(\ell)}$ and the separation between the support of pairs of $\mu_k^{(\ell)}$ and $\mu_{k'}^{(\ell)}$ is related to the cluster separation parameter $\delta^{(\ell)}$. More precisely, for a positive, symmetric, rapidly decaying kernel function $\mathcal{K} : \mathbb{R}^D \times \mathbb{R}^D$ (e.g., a Gaussian kernel), continuum notions of coherence and separation may be defined [58] as

$$\Delta^{(\ell)} = \max_{k,k'=1,\dots,K_\ell, k\neq k'} \frac{\int_{\mathbb{R}^D} \int_{\mathbb{R}^D} \mathcal{K}(x,y) d\mu_k^{(\ell)}(x) d\mu_{k'}^{(\ell)}(y)}{\int_{\mathbb{R}^D} \int_{\mathbb{R}^D} \mathcal{K}(x,y) d\mu(x) d\mu_{k'}^{(\ell)}(y)},$$

$$\Lambda^{(\ell)} = \min_{k=1,\dots,K_\ell} \inf_{S \subset \mathbb{R}^D} \frac{|\mathcal{X}|_{\mathcal{K},k,\ell} \int_S \int_{\mathbb{R}^D \setminus S} \mathcal{K}(x,y) d\mu_k^{(\ell)}(x) d\mu_k^{(\ell)}(y)}{|S|_{\mathcal{K},k,\ell} \, |\mathbb{R}^D \setminus S|_{\mathcal{K},k,\ell}},$$

where $|S|_{\mathcal{K},k,\ell} = \int_S \int_S \mathcal{K}(x,y) d\mu_k^{(\ell)}(x) d\mu_k^{(\ell)}(y)$ is a kernelized notion of volume associated to a subset $S \subset \mathbb{R}^D$ with respect to the measure $\mu_k^{(\ell)}$ and the measures are all supported in the compact set $\mathcal{X} \subset \mathbb{R}^D$. Then $\Delta^{(\ell)}$ is comparable to $\delta^{(\ell)}$ and $\Lambda^{(\ell)}$ is like a continuum notion of *conductance* minimized across each $\mu_k^{(\ell)}$, which by Cheeger's inequality is comparable to $\lambda_{K_\ell+1}^{(\ell)}$ [11,13].

Developing a continuum limit theory for MELD that allows for performance guarantees in terms of $\{\Delta^{(\ell)}\}_{\ell=1}^M$ and $\{\Lambda^{(\ell)}\}_{\ell=1}^M$ or similar quantities would separate the statistical aspects (dependence on the sample of size n) from the underlying geometric structure of the measure μ. We conjecture this can be done by relating the decay in the kernel $\mathcal{K}$ (e.g. the scaling parameter in the case of Gaussian kernels) to the time parameter t in an associated diffusion operator [14], which would allow precise analogues of the intervals of Definition 3.1, but in the continuum setting. The Γ-convergence framework [18,25] would then allow for high-probability results—depending on n and structural properties of the $\mu_k^{(\ell)}$ such as their intrinsic dimensionality—showing the comparability of performance in the discrete and continuum settings. This is a topic of ongoing research.

3.3.2. The stochastic blockmodel

Another class of data models that remains widely used in clustering models the points in X as nodes in a random network. The edges between nodes typically are sampled independently according to some probability distribution. The stochastic blockmodel (SBM) [29] is a random network model that assumes K latent clusters $X_1, \ldots, X_K$ exist in the graph and that there is a $K \times K$ matrix $\mathbf{Q}$ storing between-cluster edge probabilities. More precisely, an edge will exist between $x \in X_i$ and $y \in X_j$ with probability $\mathbf{Q}_{ij}$, independently of other edges. The SBM is a useful tool for proving performance guarantees on clustering algorithms because of its statistical construction [57].

By its definition, however, the SBM assumes a single scale of latent structure. The hierarchical stochastic blockmodel (HSBM) is a multiscale extension of the SBM [40]. The HSBM is similar to the SBM in that points are modeled as the nodes of a graph, the edges between which are sampled according to a probability distribution. However, unlike the SBM, the HSBM allows for multiple scales of separation to exist within the same graph. The benefits of the HSBM result from its statistical framework, which facilitates the analysis of hierarchical clustering algorithms. However, it is difficult to gain a geometric interpretation of the communities in HSBMs, because edges are generated independently.

4. Multiscale learning by unsupervised nonlinear diffusion

An advantage of the MELD data model is that it encapsulates a range of scales of separation within a single dataset. The possible separations range from coarse to fine, and we have shown that there is a natural relationship between the scale of latent cluster structure and the time parameter in a diffusion process. An important implication is that there are many "correct" clusterings of the same dataset. It is then natural to ask: which among the many latent clusterings of a dataset contains the most information about its underlying structure? In this section, we introduce the M-LUND algorithm: a multiscale extension of the LUND algorithm. The M-LUND algorithm chooses the partition of X that best represents all latent multiscale structure. In particular, it finds the barycenter among all nontrivial clusterings of X learned by the LUND algorithm, where distance is measured using *variation of information* (*VI*) [43].

4.1. Background on the variation of information between clusterings

Let $\mathcal{C} = \{X_k\}_{k=1}^{K}$ and $\mathcal{C}' = \{X_k'\}_{k=1}^{K'}$ be two clusterings of X with cluster sizes $|X_k| = n_k$ for $1 \leq k \leq K$ and $|X_k'| = m_k$ for $1 \leq k \leq K'$ respectively. A data point sampled from a uniform distribution over X has probability n_k/n of being a point from X_k. Hence, the clustering $\mathcal{C}$ can be associated with a discrete random variable taking K values. A similar discrete random variable taking K' values can be constructed for the clustering $\mathcal{C}'$ [43].

The uncertainty associated with a random variable can be quantified by its *entropy*. The entropy of the clustering $\mathcal{C}$ is identified as the entropy of the random variable associated to $\mathcal{C}$: $H(\mathcal{C}) = -\sum_{i=1}^{K} \frac{n_i}{n} \log\left(\frac{n_i}{n}\right)$ [43]. The entropy of a clustering will be zero whenever there is no uncertainty whatsoever about which cluster each point belongs to (i.e., the single-cluster clustering). Conversely, the entropy of $\mathcal{C}$ is maximal when it consists of n singleton clusters.

The random variables associated with $\mathcal{C}$ and $\mathcal{C}'$ also have a joint distribution: $\mathbb{P}(x \in X_i \bigcap X_j') = n_{ij}/n$, where $n_{ij} = |X_i \bigcap X_j'|$. Define the *mutual information* between the clusterings $\mathcal{C}$ and $\mathcal{C}'$ by the mutual information between the random variables associated with them: $I(\mathcal{C}, \mathcal{C}') = -\sum_{i=1}^{K} \sum_{j=1}^{K'} \frac{n_{ij}}{n} \log\left(\frac{n_{ij}/n}{(n_i/n)(m_j/n)}\right)$ [43]. Mutual information quantifies the information gained about one random variable by observing another. In the context of clustering, $I(\mathcal{C}, \mathcal{C}')$ quantifies the information gained about the clustering $\mathcal{C}$ of X from the observation of a different clustering $\mathcal{C}'$ of X [43].

The VI between $\mathcal{C}$ and $\mathcal{C}'$ can be defined in terms of the entropy of and mutual information between the clusterings $\mathcal{C}$ and $\mathcal{C}'$. The VI comparison scheme has the advantageous property of being a distance metric measuring how much information is maintained across two clusterings of the same dataset [43].

Definition 4.1. The *VI* between two clusterings $\mathcal{C}$ and $\mathcal{C}'$ of X is defined to be $VI(\mathcal{C}, \mathcal{C}') = H(\mathcal{C}) + H(\mathcal{C}') - 2I(\mathcal{C}, \mathcal{C}')$.

4.2. The M-LUND clustering algorithm

In Section 3, we noted that a clustering $\mathcal{C}_t \in \text{MELD}_\epsilon(X)$ can be interpreted as the latent clustering of X at time t. Under assumptions on cluster density and diffusion at time t, the LUND algorithm with input t is guaranteed to recover the latent clustering $\mathcal{C}_t$ [41]. Thus, the MELD data model and the LUND algorithm are closely linked, and the LUND algorithm can be interpreted as an algorithm to find the MELD clustering at a fixed time step. In this section, we leverage this relationship for a multiscale extension of the LUND algorithm based on the MELD data model.

We begin by considering how the LUND algorithm's cluster assignments behave at very large time. We note that, when diffusion is close to stationarity, these clusterings become independent of t. Indeed, for $x, y \in X$,

$$\begin{aligned} D_t(x,y)^2 &= \lambda_2^{2t}(\psi_2(x) - \psi_2(y))^2 + \sum_{k=3}^{n} \lambda_k^{2t}(\psi_k(x) - \psi_k(y))^2 \\ &= \lambda_2^{2t}\left[(\psi_2(x) - \psi_2(y))^2 + \sum_{k=3}^{n} \left(\frac{\lambda_k}{\lambda_2}\right)^{2t} (\psi_k(x) - \psi_k(y))^2\right]. \end{aligned}$$

If there is a gap between $|\lambda_2|$ and $|\lambda_3|$, then $\left|\frac{\lambda_k}{\lambda_2}\right| < 1$ for each $k \geq 3$. Therefore, while all eigenfunctions' contributions to diffusion distances converge to zero as $t \to \infty$, they do not do so at the same rate. When diffusion is near stationarity, higher-frequency eigenfunctions' contributions to diffusion distances are nearly zero relative to the contribution of the second eigenfunction. This implies that the clustering generated by the second eigenfunction of $\mathbf{P}$ will persist until diffusion distances are numerically zero. The persistence of that clustering, however, does not reflect its stability, as the diffusion process will have effectively arrived at stationarity.

To avoid artificially increasing the stability of the clustering generated by the second eigenfunction of $\mathbf{P}$, we choose to terminate cluster analysis at a maximum time step. We will cluster X using the LUND algorithm for all t in the set $\{0, 1, \beta, \dots, \beta^T\}$, where $\beta > 1$ is an exponential sampling rate and $T = \left\lceil \log_\beta \left[\log_{|\lambda_2|}\left(\frac{\tau \pi_{\min}}{2}\right)\right]\right\rceil$ is a maximum time index depending on the quantity $\pi_{\min} = \min_{u\in X} \pi(u)$ and a stationarity threshold $\tau \in (0,1)$. Intuitively, smaller β corresponds to a finer sampling and more precision when recovering latent multiscale structure in X. However, each additional time sample would correspond to another run of the LUND algorithm, so β should be tuned according to the size of the dataset and available computational resources. The quantity τ is a threshold for how close to stationarity diffusion should be to end cluster analysis and is typically small ($\tau \ll 10^{-2}$) in practice. The quantity T will be justified further in our theoretical guarantees in Section 4.3.3.

We remark that there is possibly be a more data-driven method of choosing time steps at which the LUND algorithm should be evaluated. For example, one reasonable modification is to generate an adaptive sampling of the diffusion process based on the spectrum of $\mathbf{P}$. In this scheme, the LUND algorithm would be implemented using the time steps t at which the lowest-frequency eigenfunctions are annihilated in the diffusion map. This modification is, in many cases, likely to produce the same clusterings of the dataset and may result in a reduction in the computational complexity by a constant multiple. Nevertheless, it is not

Algorithm 2: Multiscale Learning by Unsupervised Nonlinear Diffusion (M-LUND).

```
Input: $X$ (dataset), $\sigma$ (diffusion scale), $\sigma_0$ (KDE bandwidth), $\beta$ (sampling rate), $\tau$ (stationarity threshold)
Output: $\{\mathcal{C}_{t_i} | t_i = 0, 1, \beta, \dots, \beta^T\}$ (multiscale clusterings), $\mathcal{C}_{t^*}$ (optimal clustering), $K_{t^*}$ (optimal no. clusters)
Construct the transition matrix $\mathbf{P}$ and its stationary distribution $\pi$ with a Gaussian kernel and diffusion scale $\sigma$;
Calculate $T = \lceil \log_\beta \left[\log_{|\lambda_2(\mathbf{P})|} \left(\frac{\tau \min(\pi)}{2}\right)\right]\rceil$;
for $t_i \in \{0, 1, \beta, \beta^2, \dots, \beta^T\}$ do
    $[\mathcal{C}_{t_i}, K_{t_i}] = \text{LUND}(X, \sigma_0, \sigma, t_i)$;
end
$J = \{t_i \,|1 < K_{t_i} < \frac{n}{2}\}$ ;
for $t_i \in J$ do
    $\text{VI}^{(\text{tot})}(\mathcal{C}_{t_i}) = \sum_{s \in J} \text{VI}(\mathcal{C}_{t_i}, \mathcal{C}_s)$;
end
$\mathcal{C}_{t^*} = \text{argmin}\{\text{VI}^{(\text{tot})}(\mathcal{C}_{t_i}) | t_i \in J\}$;
$K_{t^*}$ = number of unique clusters in $\mathcal{C}_{t^*}$;
```

clear that this modification reduces the computational complexity of clustering extraction asymptotically. In Section 4.4, we will argue that the value T is $O(1)$ with respect to sample size n. Thus, any reduction in the number of times that the LUND algorithm is implemented in a multiscale extension must be a reduction by a constant multiple with respect to n.

To find the optimal clustering of X, we solve $\mathcal{C}_{t^*} = \text{argmin}\big\{\text{VI}^{(\text{tot})}(\mathcal{C}_t)\big| t \in J\big\}$, where the *total VI* of the clustering $\mathcal{C}_t$ is defined to be $\text{VI}^{(\text{tot})}(\mathcal{C}_t) = \sum_{s \in J} \text{VI}(\mathcal{C}_t, \mathcal{C}_s)$ and $J = \big\{t = \beta^j \;|j \in \{-\infty, 0, 1, \dots, T\}, K_t \in \;[2, \frac{n}{2})\big\}$. We restrict our analysis to clusterings sampled during J because it is possible that some clusterings extracted by the LUND algorithm are not meaningful; for example if the LUND algorithm is evaluated during a transition region. We will refer to a clustering $\mathcal{C}_t$ as *nontrivial* if $K_t \in [2, \frac{n}{2})$ and *trivial* otherwise. Thus, J corresponds to the time steps during which the LUND algorithm extracts a nontrivial clustering. We choose a lower bound of $K_t = 2$ because the single-cluster clustering yields no meaningful information about the dataset. We choose an upper bound of $K_t = \frac{n}{2}$ to avoid singleton clusters. Thus, the clustering $\mathcal{C}_{t^*}$ is the partition of X that best represents the nontrivial multiscale structure detected by the LUND algorithm across the diffusion process. The M-LUND algorithm is provided in Algorithm 2.

The clustering $\mathcal{C}_{t^*}$ is the barycenter of all nontrivial clusterings of X learned by the M-LUND algorithm, where distances between clusterings are measured using VI. In this sense, $\mathcal{C}_{t^*}$ incorporates information from clusterings of all scales into a single representative partition. Importantly, however, VI minimization is not a simple average; clusterings may appear multiple times if they are extracted by the M-LUND algorithm at multiple time steps in the diffusion process. To see this, we must introduce new notation. Suppose M unique nontrivial clusterings are learned by the M-LUND algorithm, and let $J_\ell = \{s \in J \mid \mathcal{C}_s = \mathcal{C}_\ell\}$ be the set of time steps during which a clustering $\mathcal{C}_\ell$ is extracted by the M-LUND algorithm for $\ell \in \{1, 2, \dots, M\}$. Using this new notation, the total VI of a clustering $\mathcal{C}_t$ can be rewritten $VI^{(\text{tot})}(\mathcal{C}_t) = \sum_{\ell=1}^{M} |J_\ell| VI(\mathcal{C}_t, \mathcal{C}_\ell)$. We remark that $|J_\ell|$ can be interpreted as an approximation of the size of $\mathcal{I}_\epsilon^{(\ell)}$ on a logarithmic scale. Indeed, if $\mathcal{C}_\ell$ is extracted at only times within $\mathcal{I}_\epsilon^{(\ell)}$, as $\beta \to 1^+$, it can be shown that $|J_\ell| \to \log\left[\frac{\epsilon}{2\delta^{(\ell)}} - \frac{\log(2\kappa^{(\ell)}/\epsilon)}{\log(1/|\lambda_{K_\ell+1}|^{(\ell)})}\right]$: the log-length of $\mathcal{I}_\epsilon^{(\ell)}$. In particular, stable clusterings are emphasized by the M-LUND minimization scheme. Stability in the diffusion process is highly related to desirable properties in a clustering. For example, Proposition 3.1 implies that clusterings that consist of well-separated and coherent clusters are, all else equal, ϵ-separable by diffusion distances for a longer interval of time on a logarithmic scale. Therefore, by emphasizing clusterings' stability in its optimization, the M-LUND algorithm weights representative clusterings with coherent and well-separated clusters higher.

In our performance guarantees (Section 4.3), we show that, under assumptions on entropy and mutual information of the clusterings in the MELD data model, the $VI^{(\text{tot})}$-minimizer will be in $\text{MELD}_\epsilon(X)$. In this sense, the M-LUND algorithm chooses the stability-weighted VI barycenter of the MELD data model.

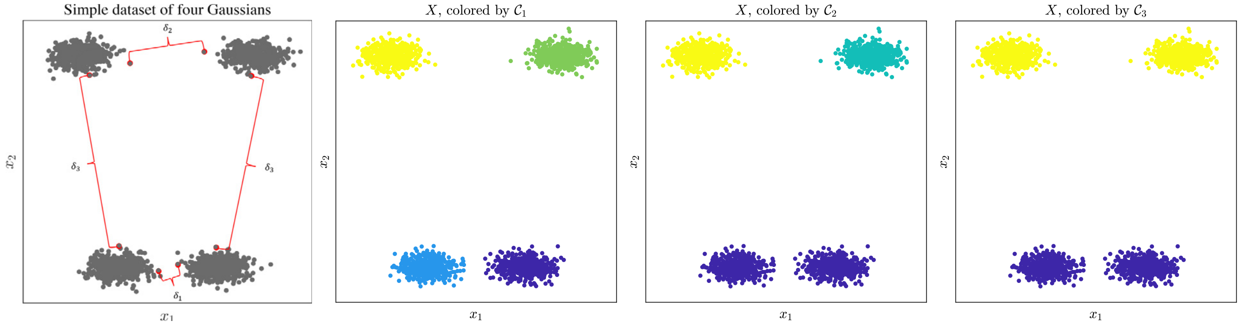


Fig. 2. A mixture of Gaussians example considered in this section. In the leftmost panel, the distances δ_1, δ_2, and δ_3 are indicated on the dataset, with between-cluster Euclidean distance minimizer indicated in red. The clusterings $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$ are visualized in the right three panels. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

4.2.1. Relationship of the LUND and M-LUND algorithms

In this section, we highlight some key differences between the LUND and M-LUND clustering algorithms. The most significant of these differences, perhaps, is the difference between the standard clustering problem (where only one clustering at a fixed scale is learned) and the multiscale clustering problem (where many scales of latent clusterings are learned). In the LUND algorithm, the user must input a time parameter t. As we have discussed in Section 3, the value of t corresponds to the scale of cluster structure that can be learned using the LUND algorithm. Because the choice of the scale of t is not unsupervised, the LUND algorithm itself cannot be completely unsupervised. In contrast, the M-LUND algorithm eliminates the dependence on t by varying it across all relevant scales. After extracting all scales of latent cluster structure, the M-LUND algorithm then offers a way to incorporate all scales of latent cluster structure into a single representative clustering of the dataset. In this way, the M-LUND algorithm is able to cluster the dataset in a truly unsupervised fashion.

In addition, we remark that the dependence of the LUND algorithm on the time parameter t leaves it prone to user error. For two different datasets, the same value of t may not correspond to the same scale of clustering. This is because the intervals on which diffusion distances separate a given clustering are inherently dependent on the geometry of the dataset and that specific clustering. It is thus difficult to give guidance to the user about how to use the LUND algorithm to cluster at a preset scale on a general dataset. In contrast, the M-LUND algorithm varies the time parameter t across all scales of interest and finds the clustering of the dataset that is most representative of all underlying cluster structure. In this sense, the issues faced by the LUND algorithm in its reliance on the user inputting t are mitigated entirely by the proposed M-LUND algorithm. By eliminating the dependence on t by clustering a dataset at all relevant time scales, the M-LUND algorithm is also easier to use by a user unfamiliar with graph-based clustering algorithms.

4.2.2. The role of diffusion stability in the output of the M-LUND algorithm

Consider four well-separated clusters in $\mathbb{R}^D$ of equal size $n/4$, arranged on the vertices of a trapezoid (see Fig. 2). Mathematically, we let $X = \bigcup_{k=1}^4 X_k$, where X_k consists of points from the k^{th} cluster. We define $\delta_1 = \min_{x\in X_1, y\in X_2} \|x - y\|_2$, $\delta_2 = \min_{x\in X_3, y\in X_4} \|x - y\|_2$, and assume that $\delta_3 = \min_{x\in X_1, y\in X_3} \|x - y\|_2 = \min_{x\in X_2, y\in X_4} \|x - y\|_2$ so that the clusters X_1 and X_3 are as well-separated as the clusters X_2 and X_4. Supposing $0 \ll \delta_1 < \delta_2 < \delta_3$, consider three distinct nontrivial clusterings: $\mathcal{C}_1 = \{X_1, X_2, X_3, X_4\}$, $\mathcal{C}_2 = \{X_1 \bigcup X_2, X_3, X_4\}$, and $\mathcal{C}_3 = \{X_1 \bigcup X_2, X_3 \bigcup X_4\}$.

The separation parameters δ_k are related to the stability of the clusterings $\mathcal{C}_\ell$. If the δ_k are large and nearly equal, then transitions between any pair of clusters will tend to be unlikely. In this case, $\mathcal{C}_1$ will be more stable in the diffusion process than $\mathcal{C}_2$ and $\mathcal{C}_3$. Conversely, if δ_1 and δ_2 are very small compared to δ_3, then $\mathcal{C}_3$ will be more stable in the diffusion process than $\mathcal{C}_1$ and $\mathcal{C}_2$. The M-LUND algorithm learns multiscale cluster structure by evaluating the LUND algorithm at an exponential sampling of the diffusion process. Thus, more stable clusterings will be extracted more frequently and weighted higher in the M-LUND minimization problem. We can explicitly derive when the M-LUND algorithm will choose one clustering over another as a function of the stability of those clusterings to graph diffusion.

Proposition 4.1. *For $\ell = 1, 2, 3$, assume that a fraction $p_\ell \in [0, 1]$ of the nontrivial clusterings extracted by the M-LUND algorithm are $\mathcal{C}_\ell$ and that $p_1 + p_2 + p_3 = 1$ so that $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$ are the only nontrivial clusterings extracted by the M-LUND algorithm. Then,*

1. *$\mathcal{C}_1$ is chosen by the M-LUND algorithm if and only if $p_1 \geq p_2 + p_3$.*
2. *$\mathcal{C}_2$ is chosen by the M-LUND algorithm if and only if $p_2 \geq |p_1 - p_3|$.*
3. *$\mathcal{C}_3$ is chosen by the M-LUND algorithm if and only if $p_3 \geq p_1 + p_2$.*

Proof. By the stated assumptions, $p_\ell = m_\ell/(m_1 + m_2 + m_3)$, where m_ℓ is the number of times that $\mathcal{C}_\ell$ is extracted by the M-LUND algorithm. Since no other nontrivial clustering is extracted by the M-LUND algorithm, $VI^{(tot)}(\mathcal{C}_\ell) = \sum_{k=1}^{3} m_\ell VI(\mathcal{C}_\ell, \mathcal{C}_k)$. By Definition 4.1, $VI(\mathcal{C}_1, \mathcal{C}_2) = 0.5\log(2)$, $VI(\mathcal{C}_1, \mathcal{C}_3) = \log(2)$, and $VI(\mathcal{C}_2, \mathcal{C}_3) = 0.5\log(2)$. Since $VI(\mathcal{C}_\ell, \mathcal{C}_\ell) = 0$ for all ℓ, total VI is calculated to be

$$\begin{aligned} VI^{(\text{tot})}(\mathcal{C}_1) &= m_2 VI(\mathcal{C}_1, \mathcal{C}_2) + m_3 VI(\mathcal{C}_1, \mathcal{C}_3) = 0.5\log(2)[m_2 + 2m_3] \\ VI^{(\text{tot})}(\mathcal{C}_2) &= m_1 VI(\mathcal{C}_1, \mathcal{C}_2) + m_3 VI(\mathcal{C}_2, \mathcal{C}_3) = 0.5\log(2)[m_1 + m_3] \\ VI^{(\text{tot})}(\mathcal{C}_3) &= m_1 VI(\mathcal{C}_1, \mathcal{C}_3) + m_2 VI(\mathcal{C}_2, \mathcal{C}_3) = 0.5\log(2)[2m_1 + m_2]. \end{aligned}$$

Algebra comparing $VI^{(\text{tot})}(\mathcal{C}_\ell)$ across $\ell \in \{1, 2, 3\}$ yields the result. □

Proposition 4.1 suggests that stability in the diffusion process is critical in the M-LUND algorithm's optimization scheme. If $p_\ell = p_{\ell'}$ for all $1 \leq \ell, \ell' \leq 3$, Proposition 4.1 implies that the M-LUND algorithm will output the intermediate clustering $\mathcal{C}_2$. Conversely, if $\mathcal{C}_1$ is extracted $m > 2$ times more frequently than $\mathcal{C}_2$ and $\mathcal{C}_3$ so that $p_1 > mp_\ell$ for $\ell = 2, 3$, then $\mathcal{C}_1$ will be the minimizer of total VI. Thus, even though $\mathcal{C}_2$ is an intermediate clustering closest in VI to both $\mathcal{C}_1$ and $\mathcal{C}_3$, the M-LUND algorithm chooses $\mathcal{C}_1$ because of its relatively higher stability in the diffusion process. We remark that, in the limiting case that $\beta \to 1^+$, the set $\{p_1, p_2, p_3\}$ tends to converge to a distribution that is a function of the stability of the clusterings $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$. More precisely, assuming that each $\mathcal{C}_\ell$ can only be learned during the interval $\mathcal{I}_\epsilon^{(\ell)}$, then in the limit of $\beta \to 1^+$, p_ℓ tends towards the log-length of interval $\mathcal{I}_\epsilon^{(\ell)}$, divided by the sum of the log-lengths of $\mathcal{I}_\epsilon^{(1)}$, $\mathcal{I}_\epsilon^{(2)}$, and $\mathcal{I}_\epsilon^{(3)}$. Thus, Proposition 4.1 is robust to the choice of β when the value of β is taken sufficiently small.

4.3. Performance guarantees for unsupervised clustering

In this section, we provide performance guarantees on the M-LUND algorithm. We begin by reviewing guarantees on the performance of the LUND algorithm in Section 4.3.1 and extend these to performance guarantees of the M- LUND algorithm in Section 4.3.2. In Section 4.3.3, we provide theoretical justification for the termination of the first for-loop in the M-LUND algorithm at time β^T by showing that, for any $t > \beta^T$ and pair of points $x, y \in X$, $D_t(x, y) < \tau$.

4.3.1. Performance guarantees on the LUND clustering algorithm

In this section, we review previously-introduced guarantees on the performance of the LUND algorithm at recovering latent cluster structure at a fixed time step [41]. We will assume that there is a latent clustering $\mathcal{C}_t = \{X_k^{(t)}\}_{k=1}^{K_t}$ of X at time $t \geq 0$ and refer to the maximum within-cluster and minimum between-cluster diffusion distance at time t for the clustering $\mathcal{C}_t$ as $D_t^{\text{in}}(\mathcal{C}_t)$ and $D_t^{\text{btw}}(\mathcal{C}_t)$ respectively. We aim to show that, under plausible assumptions on density and diffusion at time t, the LUND algorithm with input t recovers the latent clustering $\mathcal{C}_t$ [41].

Definition 4.2. For a latent clustering $\mathcal{C}_t = \{X_k^{(t)}\}_{k=1}^{K_t}$ of X at time $t \geq 0$, define the set of *cluster density maxima* at time t by $\mathcal{M}_t = \Big\{p(x)\Big| \exists k \in \{1, \dots, K_t\} : x = \text{argmax}_{x \in X_k^{(t)}} p(x)\Big\}$.

The LUND algorithm estimates the modes of clusters in the latent clustering at time $t \geq 0$ as the maximizers of $\mathcal{D}_t(x)$. The following theorem guarantees that the cluster modes learned by the LUND algorithm are the highest-density points within clusters in the latent clustering at time t [41].

Theorem 4.1. *[41] For a latent clustering $\mathcal{C}_t = \{X_k^{(t)}\}_{k=1}^{K_t}$ of X at time $t \geq 0$, denote the K_t maximizers of $\mathcal{D}_t(x)$ as $\{x_i^{(t)*}\}_{k=1}^{K_t}$. If $\frac{D_t^{\text{in}}(\mathcal{C}_t)}{D_t^{\text{btw}}(\mathcal{C}_t)} < \frac{\min(\mathcal{M}_t)}{\max(\mathcal{M}_t)}$, there is a permutation $(k_1, \dots, k_{K_t})$ of $(1, \dots, K_t)$ such that $x_i^{(t)*}$ maximizes empirical density among points in the cluster $X_{k_i}^{(t)}$.*

Thus, the cluster modes estimated by the LUND algorithm are cluster-wise empirical density maximizers. The LUND algorithm estimates the number of clusters at time t using the ratio of the sorted values taken by $\mathcal{D}_t(x)$.

Corollary 4.1. *[41] For a latent clustering $\mathcal{C}_t = \{X_k^{(t)}\}_{k=1}^{K_t}$ of X at time $t \geq 0$, let $\{x_{m_i}^{(t)}\}_{i=1}^{n}$ be the points in X sorted in non-increasing order by $\mathcal{D}_t(x)$. Then, for $j < K_t$, $\frac{\mathcal{D}_t(x_{m_j}^{(t)})}{\mathcal{D}_t(x_{m_{j+1}}^{(t)})} \leq \frac{\max(\mathcal{M}_t)}{\min(\mathcal{M}_t)} \frac{\max_{1 \leq k \leq K} \rho_t(x_{m_k}^{(t)})}{\min_{1 \leq k \leq K} \rho_t(x_{m_k}^{(t)})}$. Conversely, $\frac{\mathcal{D}_t(x_{m_{K_t}}^{(t)})}{\mathcal{D}_t(x_{m_{K_t+1}}^{(t)})} \geq \frac{\min(\mathcal{M}_t)}{\max(\mathcal{M}_t)} \frac{D_t^{\text{btw}}(\mathcal{C}_t)}{D_t^{\text{in}}(\mathcal{C}_t)}$.*

By Corollary 4.1, under reasonable assumptions on the data and the latent clustering, $\mathcal{D}_t(x_{m_k}^{(t)})/\mathcal{D}_t(x_{m_{k+1}}^{(t)})$ will be small for the first K_t values and large thereafter, yielding an accurate estimation of the number of clusters at time t. In Corollary 4.2, these assumptions imply that the LUND algorithm perfectly recovers the latent clustering at time t [41].

Corollary 4.2. *For a latent clustering $\mathcal{C}_t = \{X_k^{(t)}\}_{k=1}^{K_t}$ of X at time $t \geq 0$, let $\{x_{m_i}^{(t)}\}_{i=1}^{n}$ be the points in X sorted in non-increasing order by $\mathcal{D}_t(x)$. The LUND algorithm with input t will recover the latent clustering $\mathcal{C}_t$ if*

$$\frac{D_t^{\text{in}}}{D_t^{\text{btw}}} < \min\left\{ \frac{\min_{1 \leq i \leq K_t} \rho_t(x_{m_i}^{(t)})}{\max_{1 \leq i \leq K_t} \rho_t(x_{m_i}^{(t)})} \left(\frac{\min(\mathcal{M}_t)}{\max(\mathcal{M}_t)}\right)^2, \frac{\min_{y \in X} p(y)}{\max(\mathcal{M}_t)} \frac{\min_{1 \leq k \leq K_t} \min_{x \neq y \in X_k^{(t)}} D_t(x,y)}{D_t^{\text{in}}(\mathcal{C}_t)} \right\}.$$

Proof. First, we prove that the LUND algorithm correctly recovers the number of clusters in $\mathcal{C}_t$, denoted K_t. For $j < K_t$,

$$\frac{\mathcal{D}_t(x_{m_j}^{(t)})}{\mathcal{D}_t(x_{m_{j+1}}^{(t)})} \leq \frac{\max(\mathcal{M}_t)}{\min(\mathcal{M}_t)} \frac{\max_{1 \leq k \leq K_t} \rho_t(x_{m_k}^{(t)})}{\min_{1 \leq k \leq K_t} \rho_t(x_{m_k}^{(t)})} < \frac{\min(\mathcal{M}_t)}{\max(\mathcal{M}_t)} \frac{D_t^{\text{btw}}(\mathcal{C}_t)}{D_t^{\text{in}}(\mathcal{C}_t)} \leq \frac{\mathcal{D}_t(x_{m_{K_t}}^{(t)})}{\mathcal{D}_t(x_{m_{K_t+1}}^{(t)})},$$

where Corollary 4.1 was used to gain the first and last inequalities and $\frac{D_t^{\text{in}}(\mathcal{C}_t)}{D_t^{\text{btw}}(\mathcal{C}_t)} \frac{\max_{1 \leq i \leq K_t} \rho_t(x_{m_i}^{(t)})}{\min_{1 \leq i \leq K_t} \rho_t(x_{m_i}^{(t)})} < \left(\frac{\min(\mathcal{M}_t)}{\max(\mathcal{M}_t)}\right)^2$ was used to gain the second. Next, let $j > K_t$. Because $D_t^{\text{in}} \leq D_t^{\text{btw}}$ and $p(x_{m_j}^{(t)}) \leq \min(\mathcal{M}_t)$ by Theorem 4.1, we clearly have that $\frac{\rho_t(x_{m_j}^{(t)})}{\rho_t(x_{m_{j+1}}^{(t)})} \leq \frac{D_t^{\text{in}}(\mathcal{C}_t)}{\min_{1 \leq k \leq K_t} \min_{x,y \in X_k^{(t)}} D_t(x,y)}$. Thus,

$$\frac{\mathcal{D}_t(x^{(t)}_{m_j})}{\mathcal{D}_t(x^{(t)}_{m_{j+1}})} \leq \frac{\min(\mathcal{M}_t)}{\min_{y\in X} p(y)} \frac{D^{\text{in}}_t(\mathcal{C}_t)}{\min_{1\leq k\leq K_t} \min_{x,y\in X^{(t)}_k} D_t(x,y)} < \frac{\min(\mathcal{M}_t)}{\max(\mathcal{M}_t)} \frac{D^{\text{btw}}_t(\mathcal{C}_t)}{D^{\text{in}}_t(\mathcal{C}_t)} \leq \frac{\mathcal{D}_t(x^{(t)}_{m_{K_t}})}{\mathcal{D}_t(x^{(t)}_{m_{K_t+1}})},$$

where $\frac{D^{\text{in}}_t(\mathcal{C}_t)}{\min_{1\leq k\leq K_t}\min_{x\neq y\in X^{(t)}_k} D_t(x,y)} < \frac{\min_{y\in X} p(y)}{\max(\mathcal{M}_t)}\frac{D^{\text{btw}}_t(\mathcal{C}_t)}{D^{\text{in}}_t(\mathcal{C}_t)}$ was used to gain the second inequality, and Corollary 4.1 was used to gain the last. So, the LUND algorithm correctly estimates $K_t = \text{argmax}_{1\leq k\leq n-1}\mathcal{D}_t(x^{(t)}_{m_k})/\mathcal{D}_t((x^{(t)}_{m_{k+1}})$ and labels cluster modes $\mathcal{C}(x^{(t)}_{m_k}) = k$ $(k = 1,\dots,K_t)$. Lastly, we show that non-modal labels are assigned correctly. Let $x \in X^{(t)}_k$ be any unlabeled, non-modal point. Because $D^{\text{in}}_t(\mathcal{C}_t) \leq D^{\text{btw}}_t(\mathcal{C}_t)$, $x^* = \text{argmin}_{y\in X}\{D_t(x,y)|p(y) \geq p(x), y \text{ is labeled}\}$ must be a point in $X^{(t)}_k$. So, $\mathcal{C}(x) = \mathcal{C}(x^*) = k$. By induction, all non-modal points are labeled correctly. □

Corollary 4.2 relies on a technical assumption that is sufficient (though not necessary) for successful recovery of $\mathcal{C}_t$ by LUND. The first assumption, that $\frac{D^{\text{in}}_t(\mathcal{C}_t)}{D^{\text{btw}}_t(\mathcal{C}_t)} < \frac{\min_{1\leq i\leq K_t}\rho_t(x^{(t)}_{m_i})}{\max_{1\leq i\leq K_t}\rho_t(x^{(t)}_{m_i})}\left(\frac{\min(\mathcal{M}_t)}{\max(\mathcal{M}_t)}\right)^2$, holds if $p(x)$ yields comparable values for cluster modes and between-cluster mode diffusion distances are roughly constant. For such datasets, $\min_{1\leq i\leq K_t}\rho_t(x^{(t)}_{m_i})/\max_{1\leq i\leq K_t}\rho_t(x^{(t)}_{m_i})$ will be insignificant, and $\min(\mathcal{M}_t)/\max(\mathcal{M}_t)$ will be close to 1. The second assumption, that $\frac{D^{\text{in}}_t(\mathcal{C}_t)}{D^{\text{btw}}_t(\mathcal{C}_t)} < \frac{\min_{y\in X} p(y)}{\max(\mathcal{M}_t)}\frac{\min_{1\leq k\leq K_t}\min_{x\neq y\in X^{(t)}_k} D_t(x,y)}{D^{in}_t(\mathcal{C}_t)}$, holds for datasets in which $p(x)$ has low variance and for which $\Psi_t(x)$ sends each cluster approximately to a point mass (e.g., in Fig. 1). For such datasets, within-cluster diffusion distances are nearly constant, so $D^{\text{in}}_t(\mathcal{C}_t)/D^{\text{btw}}_t(\mathcal{C}_t)$ is small compared to $\min_{1\leq k\leq K_t}\min_{x\neq y\in X^{(t)}_k} D_t(x,y)/D^{\text{in}}_t(\mathcal{C}_t)$.

4.3.2. Performance guarantees on the M-LUND clustering algorithm

We now provide performance guarantees for the M-LUND algorithm, all of which rely on the following setup:

Definition 4.3. We refer to the following as *the usual setup*: let $\epsilon \in \left(0, \frac{1}{\sqrt{n}}\right)$, $\beta > 1$, and $\tau \in (0,1)$. For each $t \in \left\{0, 1, \beta, \dots, \beta^T\right\} \bigcap A_\epsilon$, let $\mathcal{C}_t \in \text{MELD}_\epsilon(X)$ be the latent clustering of X at time t, and let $\{x^{(t)}_{m_i}\}_{i=1}^n$ be the points in X sorted in non-increasing order by $\mathcal{D}_t(x)$. Assume $\min\limits_{1\leq \ell\leq M}\delta^{(\ell)} > \frac{\epsilon}{2}\log_{\frac{\tau\pi_{\min}}{2}}(|\lambda_2|)$ and that, for each $t \in \left\{0, 1, \beta, \dots, \beta^T\right\} \bigcap A_\epsilon$,

$$\frac{\epsilon}{1/\sqrt{n}-\epsilon} < \min\left\{\frac{\min_{1\leq i\leq K_t}\rho_t(x^{(t)}_{m_i})}{\max_{1\leq i\leq K_t}\rho_t(x^{(t)}_{m_i})}\left(\frac{\min(\mathcal{M}_t)}{\max(\mathcal{M}_t)}\right)^2, \frac{\min_{y\in X} p(y)}{\max(\mathcal{M}_t)}\frac{\min_{1\leq k\leq K_t}\min_{x\neq y\in X^{(t)}_k} D_t(x,y)}{D^{in}_t(\mathcal{C}_t)}\right\}. \tag{3}$$

There are two main assumptions in the usual setup. The first, that $\min\limits_{1\leq \ell\leq M}\delta^{(\ell)} > \frac{\epsilon}{2}\log_{\frac{\tau\pi_{\min}}{2}}(|\lambda_2|)$, requires that the separation between clusters not be too strong. To gain some intuition for this condition, consider the idealized case in which, for some clustering scale ℓ, $\delta^{(\ell)} = 0$ so that the clusters $X^{(\ell)}_k$ are perfectly separated. Then, the upper limit of $\mathcal{I}^{(\ell)}_\epsilon$ is infinite; to accurately estimate $\mathcal{I}^{(\ell)}_\epsilon$, the M-LUND algorithm would need to sample an infinite number of time steps. Thus, separation must not be so strong that diffusion spreads within clusters of a single clustering ad infinitum.

Lemma 4.1. *Let $\epsilon > 0$, $\beta > 1$, and $\tau \in (0,1)$. If $\min\limits_{1\leq \ell\leq M}\delta^{(\ell)} > \frac{\epsilon}{2}\log_{\frac{\tau\pi_{\min}}{2}}(|\lambda_2|)$, then $A_\epsilon \subset [0, \beta^T]$.*

Proof. If $\min\limits_{1\leq \ell\leq M}\delta^{(\ell)} > \frac{\epsilon}{2}\log_{\frac{\tau\pi_{\min}}{2}}(|\lambda_2|)$, then $\max\limits_{1\leq \ell\leq M}\frac{\epsilon}{2\delta^{(\ell)}} < \log_{|\lambda_2|}\left(\frac{\tau\pi_{\min}}{2}\right) \leq \beta^T$. Since $A_\epsilon \subset \left[0, \max\limits_{1\leq \ell\leq M}\frac{\epsilon}{2\delta^{(\ell)}}\right]$, it follows that $A_\epsilon \subset [0, \beta^T]$. □

The M-LUND algorithm extracts the latent clusterings of X by implementing the LUND algorithm at different choices of t. However, because cluster analysis is terminated at time $t=\beta^T$, the cluster extraction stage of the M-LUND algorithm may end before the end of the last interval $\mathcal{I}_\epsilon^{(\ell)}$. In this case, important information about the latent structure of X will be lost, and the performance of the M-LUND algorithm will correspondingly worsen. Lemma 4.1 guarantees that M-LUND samples all relevant time scales in the diffusion process when extracting cluster structure. We remark that, if there exists a $\delta^{(\ell)}$ near zero, the ℓ^{th} MELD clustering would be easy to find by conventional means; e.g., running K-Means on the rows of $\mathbf{P}^t$ for t very large. Moreover, the technical assumption of Lemma 4.1 is lax (e.g. if $\lambda_2 = 1-10^{-5}$, $\tau = 10^{-5}$, $\pi_{\min} = 10^{-2}$, and $\epsilon = 10^{-2}$, it holds if $\min_{1\leq \ell\leq M}\delta^{(\ell)} > 10^{-8}$).

The second major assumption, that (3) holds for each t sampled from A_ϵ, links the MELD data model and the M-LUND clustering algorithm. Indeed, when this condition holds, it implies that diffusion distances at any sampled $t\in A_\epsilon$ will induce sufficiently strong separation on the clusterings $\mathcal{C}_t \in \text{MELD}_\epsilon(X)$ that these clusterings can be learned by the M-LUND algorithm. In this sense, the M-LUND clustering algorithm is guaranteed to recover the MELD data model. The condition (3) is easier to satisfy when the variance of p is low and diffusion maps send clusters of MELD clusterings to coherent, well-separated clusters. For example, if density is uniform and $\Psi_t(x)$ maps each cluster in $\mathcal{C}_t$ to a point mass for $t\in\{0,1,\beta,\dots,\beta^T\}\bigcap A_\epsilon$, the right hand side of (3) will be 1. In this idealized case, (3) will be satisfied by any $\epsilon\in\left(0,\frac{1}{2\sqrt{n}}\right)$. Conversely, (3) is more difficult to satisfy when the variance of p is high, or if diffusion distances do not separate cluster structure well. Proposition 4.2 summarizes the recovery of the MELD data model under the usual setup.

Proposition 4.2. *Under the usual setup, the M-LUND algorithm extracts a superset of an exponential sampling of* $\text{MELD}_\epsilon(X)$.

Proof. By Lemma 4.1, $A_\epsilon \subset [0,\beta^T]$. For each $t\in\left\{0,1,\beta,\dots,\beta^T\right\}\bigcap A_\epsilon$, $\frac{D_t^{\text{in}}(\mathcal{C}_t)}{D_t^{\text{btw}}(\mathcal{C}_t)} \leq \frac{\epsilon}{1/\sqrt{n}-\epsilon}$ by Corollary 3.1, so the assumptions of Corollary 4.2 are satisfied. Hence, the LUND algorithm perfectly recovers $\mathcal{C}_t\in\text{MELD}_\epsilon(X)$. This yields a superset of an exponential sampling of $\text{MELD}_\epsilon(X)$. □

In the M-LUND algorithm, the ϵ-stability of a clustering is approximated by exponentially sampling the interval $[0,\beta^T]$. In particular, if a clustering $\{X_k^{(\ell)}\}_{k=1}^{K_\ell}$ is more ϵ-stable, the interval $\mathcal{I}_\epsilon^{(\ell)}$ will be sampled more frequently. The M-LUND algorithm may obtain a fine-scale perspective of the ϵ-stability of the clusterings in $\text{MELD}_\epsilon(X)$ by decreasing the exponential sampling rate β. However, this requires implementations of the LUND algorithm at more time steps, increasing computational complexity. On the other hand, if β is too large, the M-LUND algorithm may not sample a MELD clustering of X. It is important to understand what choices of β are suitable for the M-LUND algorithm.

Proposition 4.3. *Let* $\epsilon\in\left(0,\frac{1}{\sqrt{n}}\right)$ *and* $\beta\in\left(1,\frac{\epsilon}{2\delta^{(\ell^*)}}\middle/\frac{\log(2\kappa^{(\ell^*)}/\epsilon)}{\log(1/|\lambda_{K_{\ell^*}+1}^{(\ell^*)}|)}\right]$, *where* $\ell^* = \underset{1\leq\ell\leq M}{\text{argmin}}\left[\log_\beta\left(\frac{\epsilon}{2\delta^{(\ell)}}\right) - \log_\beta\left(\frac{\log(2\kappa^{(\ell)}/\epsilon)}{\log(1/|\lambda_{K_\ell+1}^{(\ell)}|)}\right)\right]$. *Then under the usual setup, the M-LUND algorithm extracts each of the clusterings in* $MELD_\epsilon(X)$ *at least once.*

Proof. By Lemma 4.1, $\mathcal{I}_\epsilon^{(\ell)}\subset[0,\beta^T]$ for each $\ell\in\{1,\dots,M\}$. It therefore suffices to show that there exists a sample $\beta^{k_\ell}\in\mathcal{I}_\epsilon^{(\ell)}$ ($k_\ell\in\{0,\dots,T\}$) for every scale $\ell\in\{1,\dots,M\}$. If $\beta\in\left(1,\frac{\epsilon}{2\delta^{(\ell^*)}}\middle/\frac{\log(2\kappa^{(\ell^*)}/\epsilon)}{\log(1/|\lambda_{K_{\ell^*}+1}^{(\ell^*)}|)}\right]$, then for each $\ell\in\{1,\dots,M\}$,

$$1\leq \log_\beta\left(\frac{\epsilon}{2\delta^{(\ell^*)}}\right) - \log_\beta\left(\frac{\log(2\kappa^{(\ell^*)}/\epsilon)}{\log(1/|\lambda_{K_{\ell^*}+1}^{(\ell^*)}|)}\right) \leq \log_\beta\left(\frac{\epsilon}{2\delta^{(\ell)}}\right) - \log_\beta\left(\frac{\log(2\kappa^{(\ell)}/\epsilon)}{\log(1/|\lambda_{K_\ell+1}^{(\ell)}|)}\right).$$

Thus, for each $\ell \in \{1, \dots, M\}$, there is a $k_\ell \in \{0, \dots, T\}$ such that $\log_\beta \left(\frac{\log(2\kappa^{(\ell)}/\epsilon)}{\log(1/|\lambda^{(\ell)}_{K_\ell+1}|)} \right) \leq k_\ell \leq \log_\beta \left(\frac{\epsilon}{2\delta^{(\ell)}} \right)$, implying $\beta^{k_\ell} \in \mathcal{I}^{(\ell)}_\epsilon$. □

Proposition 4.3 illustrates that there is a tension between finding a β that will sample all intervals $\mathcal{I}^{(\ell)}_\epsilon$ and satisfying (3) for all time steps $t \in A_\epsilon \bigcap \{0, 1, \beta, \dots, T\}$. If ϵ is large, then there will be a wide range of exponential sampling rates β that can be used to sample all intervals $\mathcal{I}^{(\ell)}_\epsilon$. However, if ϵ is too large, ϵ-separation by diffusion distances might not guarantee strong enough separation of clusters to satisfy (3) at all sampled time steps. On the other hand, if ϵ is small, then (3) is easier to satisfy because of strong ϵ-separation by diffusion distances. However, because the intervals $\mathcal{I}^{(\ell)}_\epsilon$ shrink as ϵ becomes smaller, β must be decreased to guarantee that the M-LUND algorithm samples each interval $\mathcal{I}^{(\ell)}_\epsilon$. Proposition 4.3 also illustrates how ϵ-stability affects the range of suitable choices of β. For fixed $\epsilon \in \left(0, \frac{1}{\sqrt{n}}\right)$, if each interval $\mathcal{I}^{(\ell)}_\epsilon$ is large on a logarithmic scale, β can be chosen to be large and the M-LUND algorithm will still recover all clusterings in $\text{MELD}_\epsilon(X)$. On the other hand, if one of the clusterings $\{X^{(\ell)}_k\}_{k=1}^{K_\ell}$ in $\text{MELD}_\epsilon(X)$ is unstable so that $\mathcal{I}^{(\ell)}_\epsilon$ is small on a logarithmic scale, β must be decreased to guarantee that the M-LUND algorithm samples $\mathcal{I}^{(\ell)}_\epsilon$.

Because the intervals $\mathcal{I}^{(\ell)}_\epsilon$ are not known a priori, the entire time domain $[0, \beta^T]$ must be sampled to learn the clusterings in the MELD data model. Thus, it is possible that the minimizer of total VI is not within $\text{MELD}_\epsilon(X)$ and is instead a clustering obtained during a transition region, i.e. intervals of time during which the transition matrix is rapidly mixing, and there is no “true” latent clustering. Because no latent clustering exists during transition regions, the VI between a clustering sampled during a transition region and a MELD clustering is expected to be high. Proposition 4.4 provides a lax technical assumption that guarantees a MELD clustering is the minimizer of total VI.

Proposition 4.4. *Assume the usual setup, and let $B_\epsilon = [0, \beta^T] \setminus A_\epsilon$ be the transition regions between clusterings in $MELD_\epsilon(X)$. If there is a $t \in J \bigcap A_\epsilon$ such that for any $r \in J \bigcap B_\epsilon$, $\frac{1}{|J|}\sum_{s \in J} \left[I(\mathcal{C}_r, \mathcal{C}_s) - I(\mathcal{C}_t, \mathcal{C}_s) \right] < \frac{1}{2}\left[H(\mathcal{C}_r) - H(\mathcal{C}_t) \right]$, then the M-LUND algorithm outputs a clustering from* $\text{MELD}_\epsilon(X)$ *as the minimizer of total VI.*

Proof. By Lemma 4.1, B_ϵ is well-defined. Moreover, by Corollary 4.2, at each time step $t \in \{0, 1, \beta, \dots, \beta^T\} \bigcap A_\epsilon$, the LUND algorithm extracts the latent clustering at time t: $\mathcal{C}_t \in \text{MELD}_\epsilon(X)$. It suffices to show that the total VI will be lower for a clustering sampled during A_ϵ than for any sampled during B_ϵ. By the stated assumption, there is a $t \in J \bigcap A_\epsilon$ such that for any $r \in J \bigcap B_\epsilon$,

$$
\begin{aligned}
& \frac{1}{|J|}\sum_{s \in J} \left[I(\mathcal{C}_r, \mathcal{C}_s) - I(\mathcal{C}_t, \mathcal{C}_s) \right] < \frac{1}{2}\left[H(\mathcal{C}_r) - H(\mathcal{C}_t) \right] \\
\iff \quad & |J| H(\mathcal{C}_t) - 2\sum_{s \in J} I(\mathcal{C}_t, \mathcal{C}_s) < |J| H(\mathcal{C}_r) - 2\sum_{s \in J} I(\mathcal{C}_r, \mathcal{C}_s) \\
\iff \quad & \sum_{s \in J} \Big[H(\mathcal{C}_t) + H(\mathcal{C}_s) - 2I(\mathcal{C}_t, \mathcal{C}_s) \Big] < \sum_{s \in J} \Big[H(\mathcal{C}_r) + H(\mathcal{C}_s) - 2I(\mathcal{C}_r, \mathcal{C}_s) \Big] \\
\iff \quad & \sum_{s \in J} VI(\mathcal{C}_t, \mathcal{C}_s) < \sum_{s \in J} VI(\mathcal{C}_r, \mathcal{C}_s).
\end{aligned}
$$

Since the total VI of $\mathcal{C}_t$ is less than that of $\mathcal{C}_r$ where $r \in B_\epsilon$, the minimizer of total VI must be sampled during A_ϵ. □

Proposition 4.4 relies on a technical assumption on the entropy of and mutual information between nontrivial clusterings extracted by the LUND algorithm. The quantity $\frac{1}{|J|}\sum_{s \in J} \left[I(\mathcal{C}_r, \mathcal{C}_s) - I(\mathcal{C}_t, \mathcal{C}_s) \right]$ is the average difference in mutual information encoded in $\mathcal{C}_r$ and $\mathcal{C}_t$, where the average is across all nontrivial extracted clusterings $\mathcal{C}_s$. The assumption of Proposition 4.4 is easier to satisfy if this quantity is small; i.e., if a clustering in $\text{MELD}_\epsilon(X)$ stores more information about the latent structure in X than the clusterings sampled during transition regions. On the other hand, the quantity $\frac{1}{2}\left[H(\mathcal{C}_r) - H(\mathcal{C}_t)\right]$ is half the difference in entropy between $\mathcal{C}_r$ and $\mathcal{C}_t$. The assumption of Proposition 4.4 is easier to satisfy if this quantity is large. The entropy of a clustering is maximal if it consists of n singleton clusters, so the constraint on the entropy of $\mathcal{C}_t$ can be viewed as regularization: downweighting complicated clusterings that may not actually correspond to meaningful structure. Thus, a simple partition that shares high levels of mutual information with the other nontrivial extracted clusterings of X tends to satisfy the assumption of Proposition 4.4.

4.3.3. Diffusion near equilibrium

In this section, we will justify the termination of the first for-loop of the M-LUND algorithm at time $t = \beta^T$. If $|\lambda_2| > |\lambda_3|$, then for any $\eta > 0$, there exists t such that $\max_{x,y\in X} |D_t(x,y) - |\lambda_2|^t|\psi_2(x) - \psi_2(y)|| \leq \eta$. This leads the LUND algorithm to continue to label the clustering generated by the second eigenfunction of $\mathbf{P}$ until diffusion distances are numerically zero. However, the persistence of that clustering may not reflect its stability, as the diffusion process will have effectively arrived at its stationary distribution. For this reason, cluster analysis is terminated once diffusion is sufficiently close to stationarity in the M-LUND scheme. The following quantity will prove useful in measuring how close the diffusion process is to its stationary distribution:

Definition 4.4. Let $\mathbf{P}$ be a reversible, irreducible, and aperiodic transition matrix of a Markov chain on state space X with stationary distribution π. The *relative pointwise distance* of $\mathbf{P}^t$ to π at time t is $\Delta(t) = \max_{1\leq i,j\leq n} |(P^t)_{ij} - \pi_j|/\pi_j$.

It is known that $\Delta(t) \leq |\lambda_2(\mathbf{P})|^t/\pi_{\min}$ [30,60]. This yields a uniform bound on D_t.

Proposition 4.5. *For any $\tau \in (0,1)$ and $x, y \in X$, if $t > \log_{|\lambda_2|}\left(\frac{\tau\pi_{\min}}{2}\right)$, then $D_t(x,y) < \tau$.*

Proof. Let $\epsilon > 0$ and $x, y \in X$ be given. By the definition of diffusion distances,

$$\begin{aligned}
D_t(x,y) &= \|p_t(x,:) - p_t(y,:)\|_{\ell^2(1/\pi)} \\
&\leq \|p_t(x,:) - \pi\|_{\ell^2(1/\pi)} + \|p_t(y,:) - \pi\|_{\ell^2(1/\pi)} \\
&\leq 2\sqrt{\sum_{u\in X} \max_{z\in X} \frac{|p_t(z,u) - \pi(u)|^2}{\pi(u)^2}\pi(u)} \\
&\leq 2\Delta(t)\sqrt{\sum_{u\in X}\pi(u)} \\
&= 2\Delta(t) \\
&\leq 2|\lambda_2|^t/\pi_{\min}.
\end{aligned}$$

Thus, if $t > \log_{|\lambda_2|}\left(\frac{\tau\pi_{\min}}{2}\right)$, then $D_t(x,y) < \tau$. □

The value $\log_{|\lambda_2|}\left(\frac{\tau\pi_{\min}}{2}\right)$ is determined by quantities pertaining to the original graph and how diffusion spreads on it. In a graph with coherent components that have few edges between each other, $|\lambda_2|$ is close to 0. Hence, coherent cluster structure in the dataset indicates that a longer time horizon is needed. Similarly, a smaller τ indicates that more time is needed before the threshold for stationarity is met. One can interpret the dependence of T on $\pi_{\min}$ as capturing the fact that more time is needed for diffusion to reach points of lower degree.

4.4. Computational complexity

We will now analyze the computational complexity of the M-LUND algorithm, which is essentially linear when nearest neighbor searches are performed using the *cover tree* indexing structure [7,41]. Often, high-dimensional datasets $X \subset \mathbb{R}^D$ lie on or near intrinsically low-dimensional sets (e.g. subspaces or manifolds). The *doubling dimension* of X quantifies this notion of latent low-dimensionality [7]. Let $c > 0$ be the minimum value such that any ball $B(p,r) = \{q \in X \mid \|p-q\|_2 \leq r\}$ can be covered by c balls of half the radius. The doubling dimension of X is defined to be $d = \log_2 c$. Note that a uniform sample on a d-dimensional manifold has doubling dimension d. If $X \subset \mathbb{R}^D$ has doubling dimension d, the calculation of all N nearest neighbors for X using cover trees has a computational complexity of $O(NDC^d n\log(n))$ [7] with C a constant independent of n, N, d, D.

Theorem 4.2. *Let d be the doubling dimension of X. Suppose $\mathbf{P}$ is built using a KNN graph with $O(\log(n))$ nearest neighbors, cover trees are used for nearest neighbor searches, and $O(1)$ eigenfunctions are used to compute diffusion distances. If $T = \left\lceil \log_\beta \left[\log_{|\lambda_2|}\left(\frac{\tau\pi_{\min}}{2}\right)\right]\right\rceil$, the complexity of the M-LUND algorithm is $O(TDC^d n\log^2(n) + T^2 n\log(n))$ with C a constant independent of n, N, d, D.*

Proof. Under the stated assumptions, a single run of the LUND algorithm has complexity $O(DC^d\log(n)^2 n)$ [7,41]. Thus, the first for-loop in the M-LUND algorithm has complexity $O(TDC^d n\log^2(n))$. Computing $VI(\mathcal{C}_t, \mathcal{C}_s)$ costs $O(n\log(n))$ operations [43]. Since $|J| \leq T+2$, the complexity of the second for-loop in the M-LUND algorithm is $O(T^2 n\log(n))$. Combining these two results, the overall complexity is $O(TDC^d n\log(n)^2 + T^2 n\log(n))$. □

Note that if β is replaced with $\beta^{1/m}$, the complexity of the first for-loop increases by a factor of m, and the complexity of the second increases by a factor of m^2. This is because a finer sampling frequency is used; i.e., the LUND algorithm must be evaluated more frequently. Similarly, τ indicates how close to stationarity $\mathbf{P}^t$ is required to be before terminating cluster analysis. So, if τ is decreased, the M-LUND algorithm's complexity will increase. More precisely, if τ is replaced by τ^m, the value of T will increase slightly to $\left\lceil \log_\beta \left[\log_{|\lambda_2|}\left(\frac{\pi_{\min}}{2}\right) + m\log_{|\lambda_2|}\tau\right]\right\rceil$.

We expect that $T = O(1)$ with respect to n because T reflects the length of the interval for which diffusion distances remain bounded away from τ. If $T = O(1)$ with respect to n, the following simplification holds:

Corollary 4.3. *Under the assumptions of Theorem 4.2, if $T = O(1)$ with respect to n, M-LUND has complexity $O(DC^d\log(n)^2 n)$.*

Thus, no further complexity (with respect to n) is added to a single implementation of the LUND algorithm in its multiscale extension. Importantly, because the $T+2$ implementations of the LUND algorithm are independent of each other, the dominant for-loop in which the LUND clusterings are computed is embarrassingly parallelizable, as is computing total VI in the second dominant for-loop.

4.5. Comparisons with other multiscale clustering algorithms

In this section, we compare the M-LUND algorithm to related hierarchical and multiscale clustering schemes.

4.5.1. Comparison with dendrogram-based hierarchical clustering algorithms

In Appendix B.2, we describe classical hierarchical clustering algorithms, which extract a family of partitions of the dataset using a linkage function and express them as a dendrogram. Most linkage functions use

Algorithm 3: Hierarchical Spectral Clustering (HSC) [6].

```
Input: $X$ (dataset), $\sigma$ (diffusion scale), $T_{\max}$ (maximum time step)
Output: $M$ partitions of $X$ with stability measure and eigengap $\{(\mathcal{C}^{(\ell)}, \alpha_\ell, \beta_\ell)\}_{\ell=1}^M$
Construct transition matrix $\mathbf{P}$ with a Gaussian kernel and diffusion scale $\sigma$;
Calculate the eigenvalues of $\mathbf{P}$: $\{\lambda_i\}_{k=1}^n$;
For $t \in \{1, 2, \dots, T_{\max}\}$, compute $\Delta_t = \max_{1\leq k\leq n-1} |\lambda_k^t - \lambda_{k+1}^t|$ and $K_t = \text{argmax}_{1\leq k\leq n-1}|\lambda_k^t - \lambda_{k+1}^t|$;
Find the $M$ local maxima of $\Delta_t$ and denote them $\Delta_{t_1}, \Delta_{t_2} \dots, \Delta_{t_M}$. Set $t_0 = 0$.;
for $\ell = 1 : M$ do
    $\mathcal{C}^{(\ell)} = \text{SC}[X, \sigma, K_{t_\ell}]$;
    Store $(\mathcal{C}^{(\ell)}, \alpha_\ell, \beta_\ell)$, where $\alpha_\ell = (t_\ell - t_{\ell-1})/T_{\max}$ and $\beta_\ell = \Delta_{t_\ell}$;
end
```

Euclidean distances to compare clusters, causing poor performance on outliers. Conversely, because LUND relies on the function $\mathcal{D}_t(x) = p(x)\rho_t(x)$ to compute modes, it downweights low-density points that are high in diffusion distance from their D_t-nearest neighbor. Thus, the M-LUND algorithm is able to capture latent multiscale structure while remaining robust to outliers. Moreover, linkage-based clustering algorithms are typically greedy, optimizing for the best split at each iteration. This makes the output of these algorithms prone to small perturbations in the data. Conversely, all scales of clusterings extracted by the M-LUND algorithm arise from the same graph, making it more robust to minor variations in the data.

4.5.2. Comparison with hierarchical spectral clustering

SC has the disadvantage of requiring a priori knowledge of the number of clusters K. However, K may be estimated from $\mathbf{P}$ using the number of eigenvalues close to 1 [64]. A similar fact is true of $\mathbf{P}^t$: the number of eigenvalues with $|\lambda_i|^t$ near 1 may be descriptive of the number of latent clusters at time t [6]. Hierarchical Spectral Clustering (HSC) leverages this property of $\mathbf{P}$ in a multiscale adaptation of classical SC [6]. Define $\mathcal{S}_t = \{\lambda_k^t - \lambda_{k+1}^t\}_{k=1}^{n-1}$. If each of the K clusters in a latent clustering of X is a complete graph of equal size and if the effective rank of $\mathbf{P}^t$ is K, then the first $K-1$ entries of $\mathcal{S}_t$ will be small because $\lambda_1^t \approx \lambda_2^t \approx \dots \approx \lambda_K^t \approx 1$. Similarly, the last $n - K - 1$ entries of $\mathcal{S}_t$ will be small because $\lambda_{K+1}^t \approx \lambda_{K+2}^t \approx \dots, \lambda_n^t \approx 0$. Thus, $\mathcal{S}_t$ is expected to be a sequence of nearly-zero numbers in all but the K^{th} entry. The quantity $\lambda_K^t - \lambda_{K+1}^t$ is called the *eigengap at time t* [6,64]. As t varies, $K_t = \text{argmax}_k(\lambda_k^t - \lambda_{k+1}^t)$ varies as well, so t can be interpreted as a scaling parameter [6]. HSC finds the K_t corresponding to local maxima of $\Delta_t = \max\limits_{1\leq k\leq n-1}\{\lambda_k^t - \lambda_{k+1}^t\}$ and uses these as inputs for the SC algorithm (Section 2.2.2). HSC is provided in Algorithm 3.

There are similarities between the M-LUND algorithm and HSC. For example, both algorithms rely on a Markov diffusion process to extract multiscale cluster structure. However, there are some key differences between them, the most important being that HSC does not directly incorporate density into predictions. SC exhibits fundamental limitations on datasets with clusters that are not of uniform density and scale [49]; these limitations persist in the multiscale implementation of SC. In contrast, the M-LUND algorithm has performance guarantees for recovering the correct clusterings on datasets of varying scale and density.

Another difference between the M-LUND algorithm and HSC is the latter algorithm's reliance on the eigengap to estimate the number of clusters at time t. While the eigengap is effective at uncovering the effective rank of $\mathbf{P}^t$ in some idealized cases, it may fail when Euclidean distances are used to extract $\mathbf{P}$ from a dataset that does not consist of well-separated spherical clusters [2,38]. Conversely, there is strong empirical and theoretical evidence to support the use of the function $\mathcal{D}_t(x)$ to measure the number of latent clusters at time t [41].

4.5.3. Comparison with multiscale Markov stability clustering

Another diffusion-based approach to multiscale community detection uses the *Markov stability* of a clustering at time t as a quality measure for multiscale clustering [33,34,39]. Markov stability is derived from the autocovariance matrix $\mathbf{B}(t) = \Pi\mathbf{P}^t - \pi^\top\pi$, where Π is the diagonal matrix with $\Pi_{ii} = \pi_i$ [33,34]. The quantity $\mathbf{B}(t)$ reflects the probability of a random walk beginning in a cluster X_k and being in that cluster after t steps,

Algorithm 4: Single-Linkage Learning by Unsupervised Nonlinear Diffusion (SL-LUND).

```
Input: $X$ (dataset), $\sigma$ (diffusion scale), $\sigma_0$ (KDE bandwidth), $\beta$ (sampling rate), $\tau$ (stationarity threshold)
Output: $\{\mathcal{C}^{(\ell)}\}_{\ell=1}^{K_1}$ (multiscale clusterings)
Construct the transition matrix $\mathbf{P}$ and its stationary distribution $\pi$ with a Gaussian kernel and diffusion scale $\sigma$;
Calculate $T = \lceil \log_\beta [\log_{|\lambda_2(\mathbf{P})|}(\frac{\tau \min(\pi)}{2})]\rceil$;
for $t_i \in \{0, 1, \beta, \beta^2, \dots, \beta^T\}$ do
    $[\{X_k^{(1)}\}_{k=1}^{K_1}, K_1] = \text{LUND}(X, \sigma_0, \sigma, t_i)$;
    if $2 \leq K_1 < n/2$ then
        Set $t = t_i$ and $\mathcal{C}^{(1)} = \{X_k^{(1)}\}_{k=1}^{K_1}$;
        break
    end
end
for $\ell = 2, 3, \dots, K_1,$ do
    Solve $[X_{k_1}^{(\ell-1)}, X_{k_2}^{(\ell-1)}] = \text{argmin}_{1\leq k<k'\leq K_\ell-1}\mathcal{L}_{\text{SL-LUND}}(X_k^{(\ell-1)}, X_{k'}^{(\ell-1)})$;
    Merge $X_{k_1}^{(\ell-1)}$ and $X_{k_2}^{(\ell-1)}$ in $\mathcal{C}^{(\ell-1)}$ to obtain $\mathcal{C}^{(\ell)}$;
end
```

minus the probability that two independent random walks end in X_k, evaluated at stationarity [39]. Thus, $\mathbf{B}(t)$ is expected to be large for stable clusterings and nearly zero for intermediate, less stable clusterings. Nevertheless, to our knowledge, this relationship is more an intuition than a theoretical result. The Markov stability of a clustering $\mathcal{C} = \{X_k\}_{k=1}^K$ of X is defined to be $r(t, \mathcal{C}) = \sum_{k=1}^K \sum_{x_i,x_j\in X_k} \mathbf{B}(t)_{ij}$ [33,34,39]. The Markov stability $r(t, \mathcal{C})$ will be large if a t-step random walk is likely to terminate in the cluster in which it began, and it is likely to be small if a between-cluster transition is likely [39].

The Multiscale Markov Stability (MMS) clustering algorithm optimizes $r(t, \mathcal{C})$ across partitions $\mathcal{C}$ using a modified Louvain algorithm: $\mathcal{C}_t = \text{argmax}\big\{r(t, Z) \;\big|\; Z \text{ is a partition of } X\big\}$ [8,39]. This optimization is performed across an exponential sampling of the diffusion process to learn multiscale structure. For each pair of times s and t, $VI(\mathcal{C}_s, \mathcal{C}_t)$ is calculated [39,43]. If $\mathcal{C}_t$ is stable in the diffusion process, it is likely to be close in VI to other extracted clusterings [39]. The authors therefore look for large diagonal blocks with small values in the $VI(\mathcal{C}_s, \mathcal{C}_t)$ matrix.

The M-LUND clustering algorithm is similar in some crucial ways to the MMS algorithm [39]. Both algorithms rely on a Markov diffusion process to indicate scale [39,41]. Similarly, both algorithms use the VI between clusterings to determine which partition is most representative of the dataset on a whole [39,43]. The main difference between the M-LUND and MMS algorithms is how clusterings are derived and the theoretical guarantees both clustering algorithms provide. The M-LUND algorithm uses the LUND algorithm to extract K_t and $\mathcal{C}_t$ at times $t \geq 0$. In Corollary 4.2, we showed that under reasonable assumptions on cluster density and diffusion at time t, the LUND algorithm will perfectly recover K_t and $\mathcal{C}_t$. While Markov stability-based clustering has been shown to perform well on many benchmark datasets, it does not enjoy similar theoretical backing.

4.5.4. Comparison with single-linkage learning by unsupervised nonlinear diffusion

In this section, we compare the M-LUND algorithm against the single-linkage learning by unsupervised nonlinear diffusion (SL-LUND) algorithm (Algorithm 4), which is a simpler multiscale extension of the LUND algorithm based in part on single-linkage clustering (SLC). In the first for-loop of the SL-LUND algorithm, an initial LUND clustering is obtained. Once that clustering, denoted $\mathcal{C}^{(1)}$, is found, the SL-LUND algorithm treats this clustering as the finest-scale clustering in a dendrogram. To generate intermediate, coarser-scale clusterings, the SL-LUND algorithm iteratively merges clusters based on a diffusion distance-based linkage function: $\mathcal{L}_{\text{SL-LUND}}(X_1, X_2) := \min_{x_1\in X_1, x_2\in X_2} D_t(x_1, x_2)$, which is identical to the one used in SLC (see Appendix B.2), except that diffusion distances at time t—the time step at which $C^{(1)}$ was extracted by the LUND algorithm—are used, rather than Euclidean distances.

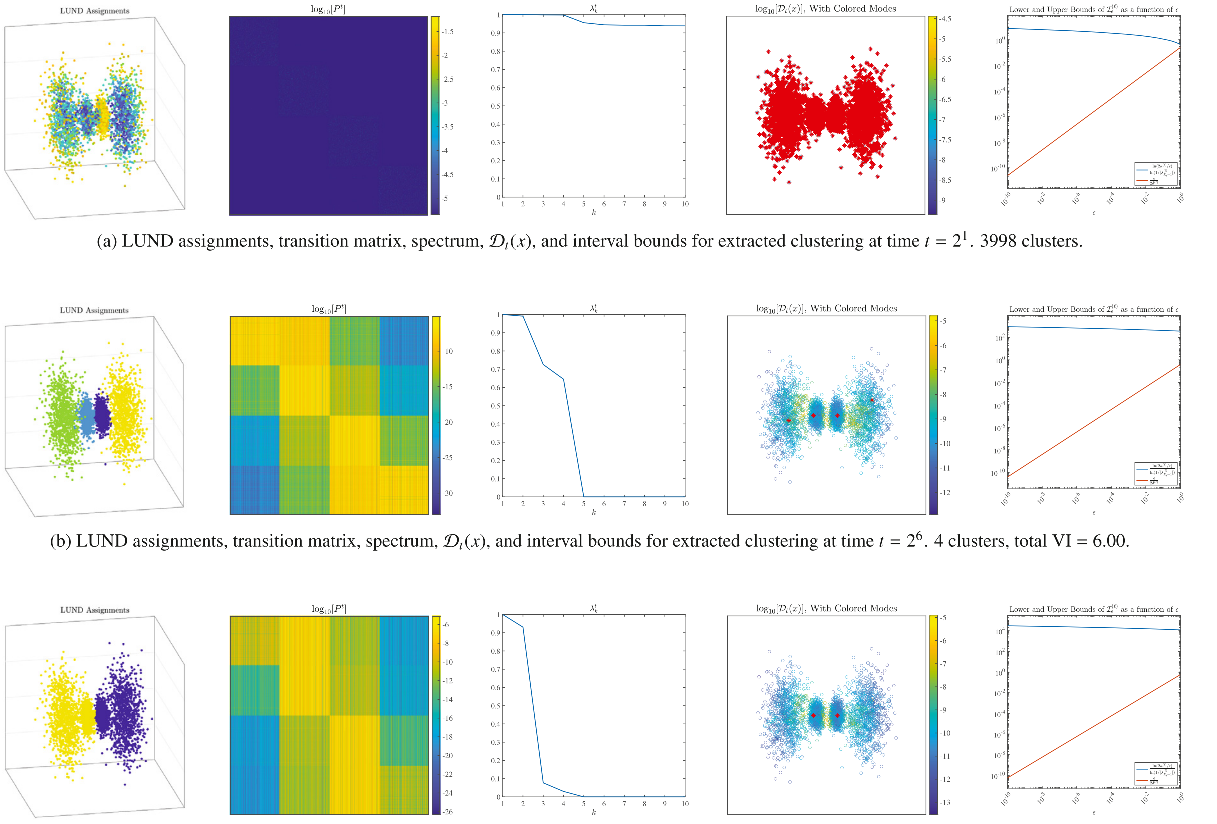
In the best-case scenario—when a nontrivial clustering with few latent clusters is extracted early in the diffusion process—the SL-LUND algorithm has a computational complexity that is a factor of T less than that of the M-LUND algorithm. However, the worst-case complexity of the SL-LUND algorithm is actually greater than or equal to that of the M-LUND algorithm. In the worst-case scenario, only trivial clusterings are extracted until time $t = \beta^T$: a time at which a clustering with $K_1 = \lfloor n/2 - 1 \rfloor$ is extracted. In this case, the computational complexity of the first for-loop is identical to the cluster extraction stage of the M-LUND algorithm. Moreover, assuming cover trees are used to compute $\mathcal{L}_{\text{SL-LUND}}$, the computational complexity of a single iteration of the second for-loop in the SL-LUND algorithm is $O(mC^d n \log(n))$, where m is the number of eigenfunctions used to compute diffusion distances and d is the doubling dimension of X. In this worst-case scenario, $K_1 = O(n)$, so the second for-loop has complexity $O(mC^d n^2 \log(n))$, which is a factor of n greater than the worst-case computational complexity of the M-LUND algorithm.

Finally, the method by which clusters are merged in the SL-LUND algorithm is necessarily limited. As we have discussed in Section 3, for any fixed t, diffusion distances at time t can generally only separate one clustering. In the SL-LUND algorithm, however, we fix the time parameter t. In particular, all subsequent steps of merging clusters rely on diffusion distances at a fixed value of t. Thus, the SL-LUND algorithm in some sense is attempting to learn multiscale cluster structure from a snapshot of the graph at a fixed scale. In contrast, the M-LUND algorithm relies on the graph directly to build multiscale clusterings of the dataset. By relying upon diffusion at many time steps to enable multiscale clustering detection, we incorporate all scales of hierarchical clustering into the proposed M-LUND clustering algorithm.

5. Numerical experiments

In this section, we illustrate the performance of the M-LUND algorithm. We compute a number of statistics on its performance on four synthetic datasets (overlapping Gaussians, concentric rings, data with bottlenecks, and Gaussians on the unit sphere S^1), eleven real, benchmark datasets, and the Salinas A HSI. Discussion on how the synthetic datasets and Salinas A HSI exhibit multiscale structure is provided in Appendix C. For synthetic datasets, we implemented the M-LUND algorithm on 100 samples of the latent distribution and provide detailed analysis of its performance on a representative sample. Weight matrices were calculated using a Gaussian kernel with diffusion scale $\sigma > 0$. KDEs were computed (as described as in Section 2.6) with KDE bandwidth σ_0 and N ℓ^2-nearest neighbors. Diffusion maps were truncated to only include the first 10 eigenfunctions. Because the transition matrices analyzed in this section were approximately low rank, this resulted in a much-reduced computational complexity for the M-LUND algorithm while retaining high levels of accuracy in diffusion distance computations. For all numerical experiments, we used a stationarity threshold of $\tau = 10^{-5}$ and sampling rate $\beta = 2$, but we used different choices of N, σ, and σ_0 for different datasets.

We computed the stochastic complements of $\mathbf{P}$ with respect to the clusterings extracted using the LUND algorithm in order to measure the geometric constants $\lambda_{K_\ell+1}^{(\ell)}$, $\delta^{(\ell)}$, and $\kappa^{(\ell)}$ and the intervals $\mathcal{I}_\epsilon^{(\ell)}$. In the fifth column of the figures in this section, the lower and upper limits of the interval $\mathcal{I}_\epsilon^{(\ell)}$ are plotted as a function of ϵ. The blue curve corresponds to the lower limit of $\mathcal{I}_\epsilon^{(\ell)}$, while the red curve corresponds to its upper limit. If, for a fixed $\epsilon > 0$, the value taken by the blue curve is greater than the value taken by the red, then $\mathcal{I}_\epsilon^{(\ell)} = \varnothing$ and the clustering is not guaranteed to be ϵ-separable by diffusion distances at any time in the diffusion process. On the other hand, if the red curve takes a greater value than the blue for some $\epsilon \in \left(0, \frac{1}{\sqrt{n}}\right)$, a range of time exists during which the clustering can be ϵ-separable by diffusion distances. Hence, from the clusterings of X extracted by the M-LUND algorithm, we recovered the MELD data model of X. We remark that the condition $t \in \mathcal{I}_\epsilon^{(\ell)}$ for $\epsilon \in \left(0, \frac{1}{\sqrt{n}}\right)$ is sufficient for the ℓ^{th} MELD clustering to be ϵ-separable by diffusion distances at time t but not necessary. Thus, even if $\mathcal{I}_\epsilon^{(\ell)}$ is empty, it is possible for the clustering $\{X_k^{(\ell)}\}_{k=1}^{K_\ell}$ to be ϵ-separable by diffusion distances.



(a) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^1$. 3998 clusters.

(b) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^6$. 4 clusters, total VI = 6.00.

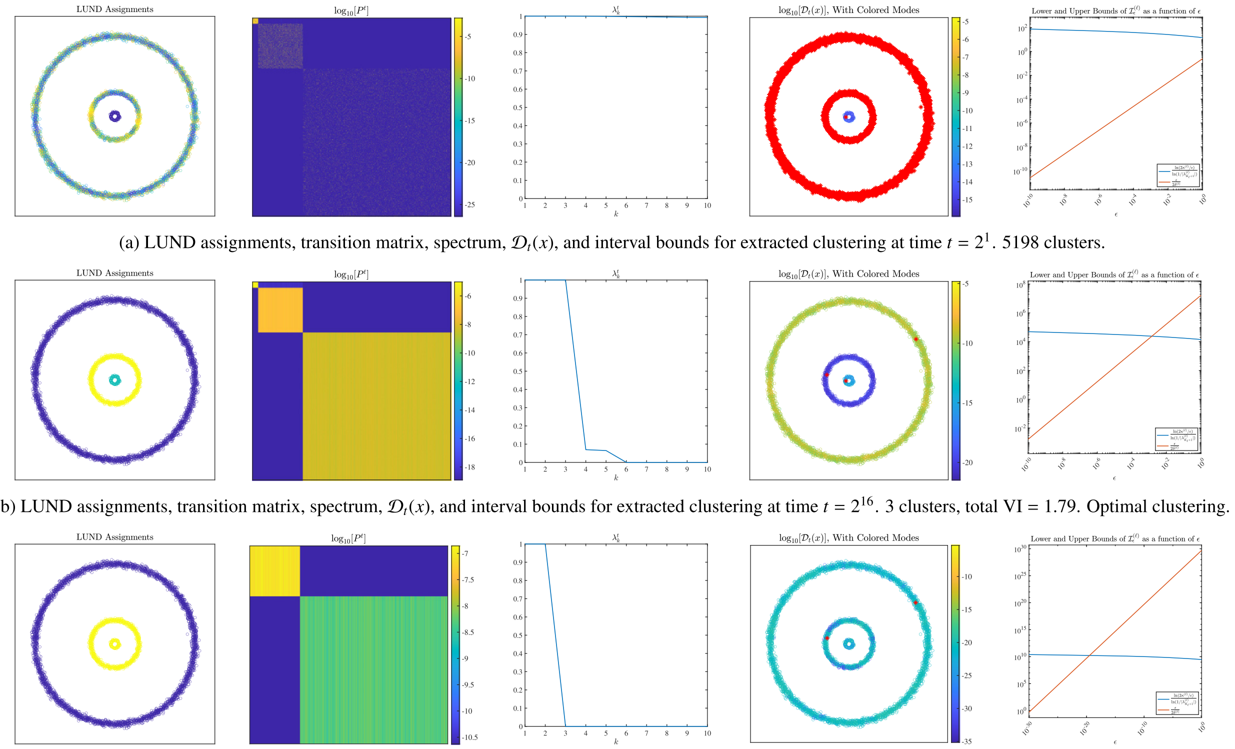
(c) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^{10}$. 2 clusters, total VI = 4.00. Optimal clustering.

Fig. 3. Diffusion on four three-dimensional Gaussians of variable density ($n = 4000$). Red points indicate cluster modes. To generate plots of $\mathcal{D}_t(x)$, we project the points in X onto two dimensions. The number of estimated clusters monotonically decreases with t. Because of poor separation between Gaussians, $\text{MELD}_\epsilon(X)$ is empty for all choices of ϵ.

5.1. Synthetic Gaussians in $\mathbb{R}^3$

In this section, we analyze a dataset sampled from four overlapping Gaussians in $\mathbb{R}^3$. The outer Gaussians have larger radii than the inner Gaussians, which are higher density. We implemented the M-LUND algorithm using a KNN graph with edges weighted with a Gaussian kernel. The parameters we used were $N = 25$ nearest neighbors, diffusion scale $\sigma = 3.10$ and KDE bandwidth $\sigma_0 = 1.45$. In Fig. 3, we show how the labels assigned by the LUND algorithm change as a function of the diffusion time parameter. Early in the diffusion process, higher-frequency eigenfunctions still contribute to diffusion distance computations and the LUND algorithm assigns a trivial singleton clustering (Fig. 3a, $t \in [0, 2^5]$). Later in the diffusion process, only the first four eigenfunctions contribute significantly to diffusion distances, and each of the four Gaussians is assigned to its own cluster (Fig. 3b, $t \in [2^6, 2^9]$). Finally, each large-radius Gaussian is merged with the nearest small-radius Gaussian (Fig. 3c, $t \in [2^{10}, 2^{16}]$). The M-LUND algorithm assigns a total VI of 6.00 to the 4-cluster clustering and a total VI of 4.00 to the 2-cluster clustering, which is more stable and thus the total VI minimizer.

Fig. 3 makes clear that the MELD data model is highly sensitive to cluster overlap. If outliers of one cluster are close to outliers of another, then δ will be large [41]. Because of the significant overlap between the four Gaussians in this dataset, the interval $\mathcal{I}_\epsilon^{(\ell)}$ is empty for both clusterings of X across all choices of $\epsilon > 0$. The numerical experiments given in this section therefore imply that the reliance of $\text{MELD}_\epsilon(X)$ on the separation parameter δ is somewhat pessimistic, and the M-LUND algorithm is able to detect latent structure at a variety of scales even when ϵ-separation is not necessarily achieved.



(a) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^1$. 5198 clusters.

(b) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^{16}$. 3 clusters, total VI = 1.79. Optimal clustering.

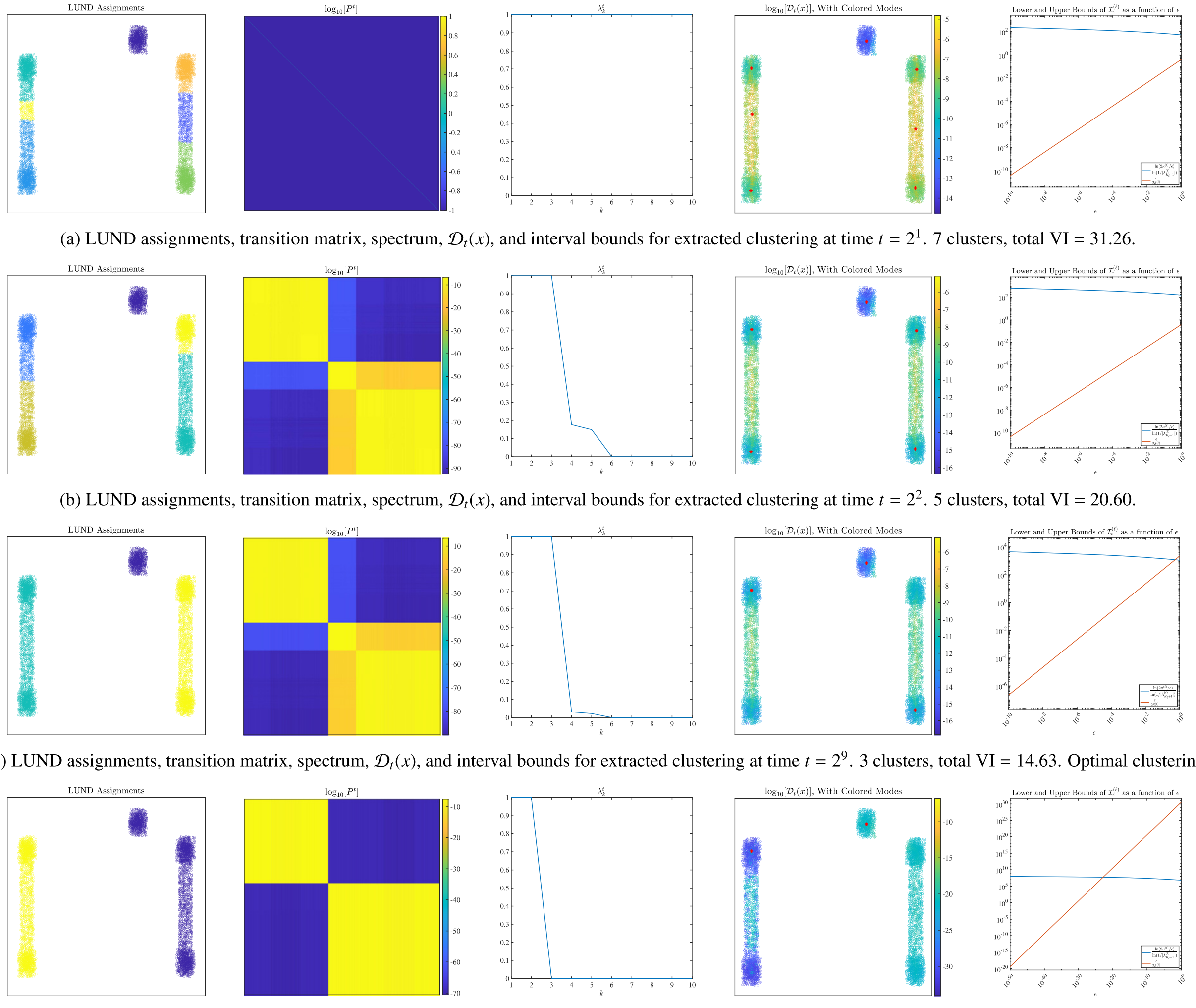
(c) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^{33}$. 2 clusters, total VI = 3.80.

Fig. 4. Diffusion on three nested circles in $\mathbb{R}^2$ ($n = 5380$). Red points indicate cluster modes. Nonlinear structure is detected. The number of estimated clusters monotonically decrease with t. Clusterings are observed to transition as a function of when a given component of the diffusion map is annihilated. Notably, the intervals $\mathcal{I}_\epsilon^{(\ell)}$ do not intersect for any choice of $\epsilon > 0$.

5.2. *Synthetic nonlinear data in* $\mathbb{R}^2$

In this section, we analyze a dataset sampled from three nested rings of uniform density. We implemented the M-LUND algorithm using a complete graph. Edges were weighted using a Gaussian kernel with diffusion scale $\sigma = 0.21$. The parameters we used for the KDE were $N = 200$ nearest neighbors and a KDE bandwidth of $\sigma_0 = 3.00$. The distance between the middle and inner rings is smaller than the distance between the outer and middle rings. In Fig. 4, we show how the labels assigned by the LUND algorithm change as a function of time. Many classical clustering algorithms (e.g., K-Means, K-medoids, DPC [23,56]) may not perform well on data with nonlinear structure, but the LUND algorithm returns reasonable clusterings for much of the diffusion process. For t small (Fig. 4a, $t \in [0, 2^{15}]$), diffusion has not passed a critical point at which enough higher-frequency eigenfunctions have been annihilated that diffusion distances can accurately separate cluster structure in the outer rings. Notably, because of poor separation between clusters, the interval $\mathcal{I}_\epsilon^{(\ell)} = \varnothing$ for any choice of $\epsilon > 0$. In particular, this clustering is not included in $\text{MELD}_\epsilon(X)$ for any $\epsilon > 0$.

When t is sufficiently large, only the first four eigenfunctions contribute significantly to diffusion distances, and the LUND algorithm assigns each ring to its own cluster (Fig. 4b, $t \in [2^{16}, 2^{32}]$). Later in the diffusion process, the third and fourth coordinates of the diffusion map decay to zero as well, and the middle and inner ring clusters merge (Fig. 4c, $t \in [2^{33}, 2^{35}]$). Notice that, in Figs. 4b-4c, $\mathcal{D}_t(x)$ returns relatively small values on all $x \in X$ except cluster modes. On modal points, the value taken by $\mathcal{D}_t(x)$ is several orders of magnitude larger than that which is taken on the surrounding dataset, implying that the LUND estimate for K_t is highly robust for these clusterings. Total VI was minimized for the 3-cluster clustering, which was assigned a total VI of 1.79. Conversely, the 2-cluster clustering was assigned a total VI of 3.80. For



(a) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^1$. 7 clusters, total VI = 31.26.

(b) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^2$. 5 clusters, total VI = 20.60.

(c) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^9$. 3 clusters, total VI = 14.63. Optimal clustering.

(d) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^{24}$. 2 clusters, total VI = 23.77.

Fig. 5. Diffusion on data with bottlenecks in $\mathbb{R}^2$ ($n = 6550$). Red points indicate cluster modes. The number of estimated clusters monotonically decreases with t, and clusterings are observed to transition as components of the diffusion map are annihilated. The clusterings in Figs. 5c-5d are ϵ-separable by diffusion distances for choices of $\epsilon > 0$. The intervals $\mathcal{I}_\epsilon^{(\ell)}$ do not intersect for any choice of $\epsilon > 0$.

ϵ sufficiently large, there are non-intersecting intervals of time $\mathcal{I}_\epsilon^{(\ell)}$ during which each of these clusterings is ϵ-separable by diffusion distances. In this sense, the MELD data model recovered from this nonlinear dataset consists of the 3-cluster and 2-cluster clusterings (Figs. 4b-4c).

5.3. Synthetic bottleneck data in $\mathbb{R}^2$

In this section, we analyze a dataset with bottlenecks. Each bottleneck consists of two Gaussians of the same radius, connected by data sampled from a uniform distribution. Density is the same for all Gaussians but is higher than the density of data sampled from uniform distributions. We have added an additional Gaussian that is slightly closer to the right bottleneck. We implemented the M-LUND algorithm using a complete graph with a Gaussian kernel and diffusion scale $\sigma = 0.86$. The parameters we used for the KDE were $N = 200$ nearest neighbors and a KDE bandwidth of $\sigma_0 = 0.50$. We visualize the performance of the M-LUND algorithm in Fig. 5. For t small, many higher-frequency eigenfunctions contribute to diffusion distance computations, and the LUND algorithm estimates seven latent clusters with significant overlap (Fig. 5a, $t \in [0, 2^1]$). As t becomes larger, fewer higher-frequency eigenfunctions contribute to diffusion distances, and X is partitioned into five clusters (Fig. 5b, $t \in [2^2, 2^8]$). Both of these clusterings have $\mathcal{I}_\epsilon^{(\ell)}$ empty due to poor separation between clusters.

Once higher-frequency components of the diffusion map decay to zero, the LUND algorithm detects a 3-cluster clustering in which each bottleneck is assigned to a cluster and the separated Gaussian is assigned to a cluster (Fig. 5c, $t \in [2^9, 2^{23}]$). After the third diffusion map coordinate is annihilated, the LUND algorithm groups the separated Gaussian with the right bottleneck (Fig. 5d, $t \in [2^{24}, 2^{25}]$). The minimizer of total VI is the 3-cluster clustering, which is assigned a total VI of 14.63. The clusterings in Figs. 5c-5d are well-separated within the original graph, and we observe that there are choices $\epsilon \in \left(0, \frac{1}{\sqrt{n}}\right)$ for which the intervals $\mathcal{I}_\epsilon^{(\ell)}$ are nonempty. In this sense, the unique clusterings in $\text{MELD}_\epsilon(X)$, for ϵ sufficiently large, consist of the 2-cluster and 3-cluster clusterings of X. As was the case for the nonlinear dataset discussed in Section 5.2, the intervals $\mathcal{I}_\epsilon^{(\ell)}$ do not intersect for any choice of $\epsilon \in \left(0, \frac{1}{\sqrt{n}}\right)$.

5.4. Synthetic manifold data

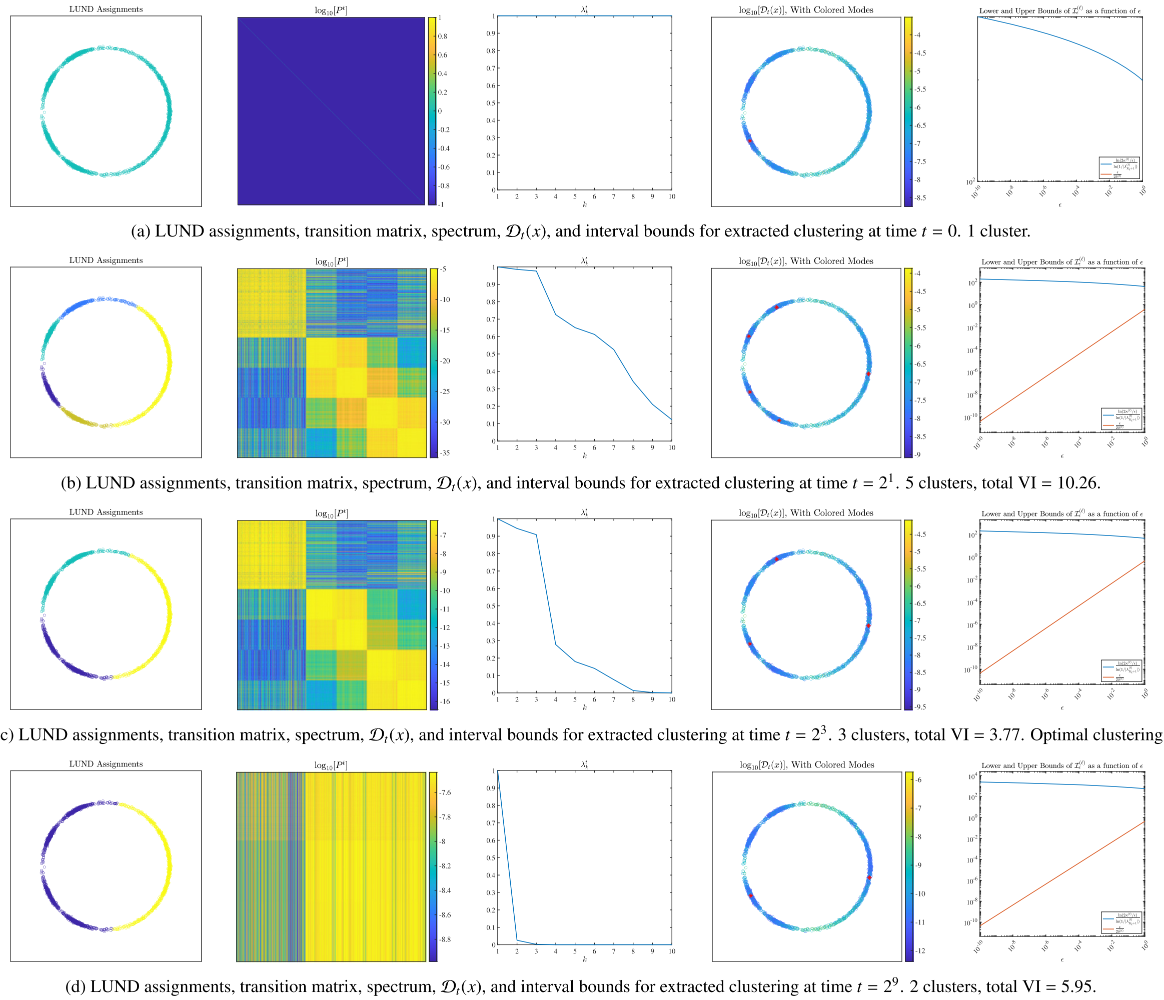
In this section, we analyze results on a dataset that consists of points sampled from a manifold. More precisely, we sampled the angular coordinate of points on the manifold S^1 from a mixture of five Gaussian distributions on $[0, 2\pi)$. One Gaussian has mean 0 rad and standard deviation 0.64 rad. The other four Gaussians each have standard deviation 0.11 rad and means $\frac{21}{32}\pi$ rad, $\frac{27}{32}\pi$ rad, $\frac{37}{32}\pi$ rad, and $\frac{43}{32}\pi$ rad respectively. The M-LUND algorithm was implemented using a complete graph, and edges between points were weighted using a Gaussian kernel with diffusion scale $\sigma = 0.358$. We used a KDE with $N = 20$ nearest neighbors and $\sigma_0 = 0.006$.

In Fig. 6, we show how labels assigned by the LUND algorithm vary as a function of the time parameter t. Early in the diffusion process, the M-LUND algorithm assigns a trivial $K = 1$ clustering of X (Fig. 6a, $t \in [0, 1]$). Once higher-frequency eigenfunctions have been annihilated in the diffusion map, the LUND algorithm assigns a $K = 5$ clustering, successfully recovering the latent distribution from which each data point was sampled (Fig. 6b, $t \in [2^1, 2^2]$). After some progression in the diffusion process, the neighboring Gaussians in quadrants 2 and 3 of S^1 are merged, yielding a $K = 3$ clustering (Fig. 6c, $t \in [2^3, 2^8]$). Once just the second eigenfunction contributes to diffusion distance computations, the LUND algorithm merges the clusters in quadrants 2 and 3 in a final $K = 2$ clustering (Fig. 6c, $t \in [2^9, 2^{12}]$). Due to poor separation between Gaussian distributions on the manifold, $\mathcal{I}_\epsilon^{(\ell)}$ is empty for all clusterings. The minimizer of total VI is the highly-stable $K = 3$ clustering, which was assigned the a total VI of 3.77.

5.5. Benchmark real data

In this section, we present analysis of multiscale clustering algorithms on eleven publicly-available, real-world datasets that are frequently used as benchmarks for clustering. These datasets and their ground truth labels (denoted $\mathcal{C}_G$) were obtained from the University of California, Irvine's Machine Learning Repository [20]. This choice of eleven real datasets was proposed by [39], wherein MMS clustering was compared against conventional clustering schemes. Attributes of these datasets and the parameters used to generate $\mathbf{P}$ and $p(x)$ are summarized in Table 1.

The *normalized mutual information* (*NMI*) between an estimated clustering and the ground truth labels is used as the performance measure of the clusterings in the dataset in this section. NMI, which is defined by $NMI(\mathcal{C}, \mathcal{C}') = \sqrt{\frac{I(\mathcal{C},\mathcal{C}')^2}{H(\mathcal{C})H(\mathcal{C}')}}$, is a measure of similarity between two clusterings, ranging $[0, 1]$. NMI is closely related to VI, and it can be shown that $NMI(\mathcal{C}, \mathcal{C}') = 1$ if and only if $VI(\mathcal{C}, \mathcal{C}') = 0$ (i.e., $\mathcal{C} = \mathcal{C}'$). Thus, if $NMI(\mathcal{C}, \mathcal{C}_G)$ is near 1, the clustering $\mathcal{C}$ is very close in VI to the ground truth labels. Conversely, $NMI(\mathcal{C}, \mathcal{C}') = 0$ if and only if the random variables associated with the clusterings $\mathcal{C}$ and $\mathcal{C}'$ are independent; i.e., observing $\mathcal{C}$ yields no new information about the clustering $\mathcal{C}'$. Thus, if $NMI(\mathcal{C}, \mathcal{C}_G)$ is near 0, there is only a weak relationship between $\mathcal{C}$ and the ground truth labels, and $\text{VI}(\mathcal{C}, \mathcal{C}_G)$ will be large.



(a) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 0$. 1 cluster.

(b) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^1$. 5 clusters, total VI = 10.26.

(c) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^3$. 3 clusters, total VI = 3.77. Optimal clustering.

(d) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^9$. 2 clusters, total VI = 5.95.

Fig. 6. Diffusion on data sampled from Gaussians on the manifold S^1 in $\mathbb{R}^2$ ($n = 2100$). Red points indicate cluster modes. The number of estimated clusters monotonically decreases with t, and clusterings are observed to transition as components of the diffusion map are annihilated. Because of poor separation between clusters, $\text{MELD}_\epsilon(X)$ is empty for all choices of ϵ.

We first compare M-LUND against related algorithms (MMS, HSC, SLC, K-Means, and LUND) [23,39,41,52]) at a fixed scale by setting the number of clusters K to be the number of ground truth classes in $\mathcal{C}_G$. Diffusion-based algorithms (M-LUND, MMS, HSC, LUND) were implemented using the same KNN graph with edges weighted with a Gaussian kernel. In our implementation of the LUND algorithm, the time parameter t is set to be the first value at which nontrivial cluster structure is extracted. Graph parameters are summarized in Table 1, and the results of this analysis are provided in Table 2. We compared performances using the two-sided paired-sample t-test [55], which tests the null hypothesis that the difference in performance between the M-LUND algorithm and its competitors is distributed normally with mean zero. Under this null hypothesis, the test statistic (denoted t_S) follows a Student's t-distribution with $n_D - 1$ degrees of freedom, where n_D reflects the number of datasets on which these algorithms were evaluated. We rejected the null hypothesis when comparing the M-LUND algorithm against each of MMS, HSC, SLC, K-Means, and LUND at the $\alpha = 0.05$ significance level. Thus, the M-LUND algorithm produces clusterings that are significantly closer to ground truth labels than those produced by the other four algorithms.

We next compare the M-LUND algorithm against related algorithms (MMS, HSC, SLC, and SL-LUND [23,39,52]) in a multiscale setting. As before, diffusion-based algorithms were implemented using the same graph as well as the same exponential time sampling of the diffusion process. To evaluate

Table 1
Summary of the eleven benchmark datasets analyzed. All datasets were obtained from the University of California, Irvine's Machine Learning Repository [20]. Graph and KDE parameters for this section's analysis are stored in the rightmost column. Here, N denotes the number of nearest neighbors, while σ and σ_0 are the diffusion scale and KDE bandwidth respectively.

Dataset	Number of samples, n	Ambient data dimensionality, D	Number of ground truth classes, K	Parameters		
				N	σ	σ_0
Breast Tissue	106	9	6	5	16 140	1230
Control Chart	600	60	6	200	58.97	45.05
Glass	214	9	6	5	1.07	0.41
Image Seg.	2310	19	7	5	748	15.50
Iris	150	4	3	50	1.34	0.457
Parkinsons	195	22	2	5	111	9.96
Seeds	210	7	3	100	0.91	1.09
Vertebral	310	6	3	5	18.15	12.39
WBCD	569	30	2	20	234	283
Wine	178	13	3	50	78.57	117.56
Yeast	1484	8	10	10	33.66	0.78

an algorithm's performance, we compare the optimal clustering it outputs to the ground truth labels. For M-LUND, MMS, and HSC, we select the clustering that minimizes total VI as the optimal outputted clustering. SLC does not rely on a diffusion process to generate its clusterings, so we select $\mathcal{C}^{(\ell^*)} = \text{argmin}_{2\leq \ell<n/2} L^{\text{in}}_{SLC}(\mathcal{C}^{(\ell)})/L^{\text{btw}}_{SLC}(\mathcal{C}^{(\ell)})$, where $L^{\text{in}}_{SLC}(\mathcal{C}^{(\ell)}) = \max_{1\leq k\leq \ell}\max_{x,y\in X_k^{(\ell)}} \|x-y\|_2$ is the maximum within-cluster Euclidean distance for the ℓ-cluster clustering in the dendogram, denoted $\mathcal{C}^{(\ell)}$, and $L^{\text{btw}}_{SLC}(\mathcal{C}^{(\ell)}) = \min_{1\leq k<k'\leq \ell}\mathcal{L}_{SLC}(X_k^{(\ell)}, X_{k'}^{(\ell)})$ is the minimum between-cluster value taken by the SLC linkage function. Similarly, SL-LUND does not directly rely on varying a diffusion time parameter to generate multiscale clusterings, so we select $\mathcal{C}^{(\ell^*)} = \text{argmin}_{2\leq \ell\leq K_1} L^{\text{in}}_{SL-LUND}(\mathcal{C}^{(\ell)})/L^{\text{btw}}_{SL-LUND}(\mathcal{C}^{(\ell)})$, where $L^{\text{in}}_{SL-LUND}(\mathcal{C}^{(\ell)}) = \max_{1\leq k\leq \ell}\max_{x,y\in X_k^{(\ell)}} D_t(x,y)$ is the maximum within-cluster diffusion distance for the ℓ^{th} SL-LUND clustering in the dendogram, denoted $\mathcal{C}^{(\ell)}$, and $L^{\text{btw}}_{SL-LUND}(\mathcal{C}^{(\ell)}) = \min_{1\leq k<k'\leq \ell}\mathcal{L}_{SL-LUND}(X_k^{(\ell)}, X_{k'}^{(\ell)})$ is the minimum between-cluster value taken by the SL-LUND linkage function.

Table 3 indicates that the M-LUND algorithm generates clusterings that are, on average, closer to the ground truth labels than those assigned by the algorithms it is compared against. Indeed, the performance achieved by the M-LUND algorithm is greater than or equal to that of its competitors across all datasets. We remark that, on six datasets, the SL-LUND algorithm achieves equal performance to the M-LUND algorithm, reflecting that the first nontrivial clustering extracted by the LUND algorithm is the total VI minimizer for these datasets. Regardless, the SL-LUND algorithm's use of a single time step to analyze

Table 2
Comparison of clustering algorithms' performance on eleven benchmark datasets when K is fixed to be the number of clusters in $\mathcal{C}_G$. MMS clustering did not learn a K-cluster clustering from the Control Chart and Parkinsons datasets, so we did not include performances on these datasets in averages or statistical tests. We used the NMI between the outputted K-cluster clustering and ground truth labels to measure performance on a dataset. Thus, a high value reflects more similarity to the ground truth labels in the dataset. On average, the K-cluster M-LUND clustering is significantly closer to the ground truth labels than the K-cluster clusterings produced by the algorithms we compare against: MMS ($p = 0.03$, $t_S = 2.66$), HSC ($p = 0.001$, $t_S = 4.94$), SLC ($p = 7.45 \times 10^{-5}$, $t_S = 7.45$), K-Means ($p = 0.001$, $t_S = 4.75$), and LUND ($p = 0.03$, $t_S = 2.55$).

Dataset	M-LUND	MMS	HSC	SLC	K-means	LUND
Breast Tissue	**0.415**	0.402	0.355	0.122	0.294	0.164
Control Chart	**0.806**	—	0.713	0.695	0.749	**0.806**
Glass	**0.427**	0.400	0.254	0.0724	0.378	**0.427**
Image Seg.	**0.644**	0.638	0.474	0.366	0.507	**0.644**
Iris	**0.901**	0.743	0.766	0.717	0.742	0.735
Parkinsons	**0.039**	—	0.001	0.005	0.001	0.001
Seeds	**0.739**	0.732	0.667	0.066	0.710	**0.739**
Vertebral	**0.574**	0.550	0.515	0.009	0.403	0.532
WBCD	**0.498**	0.403	0.473	0.005	0.465	0.326
Wine	**0.450**	0.435	0.405	0.062	0.423	**0.450**
Yeast	**0.351**	0.284	0.276	0.066	0.244	0.253
Average	**0.555**	0.510	0.465	0.165	0.463	0.475

Table 3
Comparison of clustering algorithms' performance on eleven benchmark datasets. We used the NMI between the optimal outputted clustering and the ground truth labels to measure an algorithm's performance on a dataset. Thus, a high value reflects more similarity to the ground truth labels in the dataset. The highest NMI for each dataset is marked in bold. On average, the optimal M-LUND clustering is significantly closer to the ground truth labels than the optimal clusterings produced by the algorithms we compare against: MMS ($p = 0.02$, $t_S = 2.75$), HSC ($p = 0.004$, $t_S = 3.68$), SLC ($p = 2.47 \times 10^{-4}$, $t_S = 5.54$), and SL-LUND ($p = 0.05$, $t_S = 2.20$).

Dataset	M-LUND	MMS	HSC	SLC	SL-LUND
Breast Tissue	**0.480**	0.378	0.00	0.016	**0.480**
Control Chart	**0.760**	**0.760**	0.712	0.571	**0.760**
Glass	**0.467**	0.365	0.034	0.034	0.463
Image Seg.	**0.630**	0.029	0.480	0.009	**0.630**
Iris	**0.734**	**0.734**	**0.734**	**0.734**	**0.734**
Parkinsons	**0.193**	0.113	0.001	0.013	**0.193**
Seeds	**0.739**	0.551	0.635	0.397	0.578
Vertebral	**0.623**	0.465	0.515	0.004	0.501
WBCD	**0.443**	0.357	0.374	0.005	0.326
Wine	**0.448**	0.375	0.379	0.314	**0.448**
Yeast	**0.301**	0.195	0.035	0.035	0.035
Average	**0.529**	0.393	0.354	0.194	0.468

multiscale cluster structure forces it to perform, on average, worse on the eleven datasets we considered. As before, we compared performances using the two-sided, paired-sample t-test [55], testing the null hypothesis that the difference in performance between the M-LUND algorithm and its competitors is distributed normally with mean zero. We again rejected this null hypothesis when comparing the M-LUND algorithm against each of MMS, HSC, and SLC at the $\alpha = 0.05$ significance level. Thus, in both the fixed-scale and multiscale settings, the M-LUND algorithm produces clusterings that are significantly closer to the ground truth labels compared to those generated by related algorithms.

5.6. Salinas A hyperspectral image

HSIs are images of a scene, typically generated by airborne sensors or satellites in orbit, that encode information about a hundred or more bands of electromagnetic activity. While HSIs are very high-dimensional and encode rich information about a scene, they often exhibit intrinsically low-dimensional structure [47,48,71]. In this section, we show that the M-LUND algorithm detects latent multiscale structure in the Salinas A HSI [27]. The Salinas scene was generated using the Airborne Visible/Infrared Imaging Spectrometer Sensor over Salinas Valley, California, United States in 1998. We examined the Salinas A subset of the Salinas scene, which is 83×86 pixels with $D = 224$ spectral bands per pixel. To differentiate two pixels exhibiting the same exact value in each spectral band, we added Gaussian noise with variance $= 10^{-4}$ to the Salinas A HSI as a preprocessing step [48]. In Fig. 7, we visualize the ground truth labels for the Salinas A image and the spectra of a random subset of the image. Black pixels reflect bare soil, while other colors reflect different crop types. Notably, the spectra of broccoli greens pixels, which are the dark blue crop type, is highly distinct from the spectra of all other ground truth classes.

In Fig. 8, we show how the pixel labels assigned by the LUND algorithm change as a function of the diffusion time parameter. The parameters we used were $N = 75$ nearest neighbors, diffusion scale $\sigma = 1.90$ and KDE bandwidth $\sigma_0 = 4.25 \times 10^{-3}$. Early in the diffusion process, broccoli greens pixels are grouped together, while all other pixels are assigned singleton clusters (Fig. 8a, $t \in [0, 2^{11}]$). Once higher-frequency coordinates of the diffusion map have been annihilated, the LUND algorithm detects 5 clusters (Fig. 8b, $t \in [2^{12}, 2^{13}]$), reflecting fine-scale structure in the HSI. The clusters in this clustering correspond to broccoli greens, corn-senesced greens grouped with 5-week maturity romaine lettuce crops, 6-week maturity romaine lettuce crops, 7-week maturity romaine lettuce crops, and 8-week maturity romaine lettuce crops. Later in the diffusion process, mature romaine lettuce (7-8 week maturity) clusters merge in a 4-cluster clustering with total VI $= 5.47$ (Fig. 8c, $t = 2^{14}$). Finally, all romaine lettuce clusters merge, leaving only the broccoli

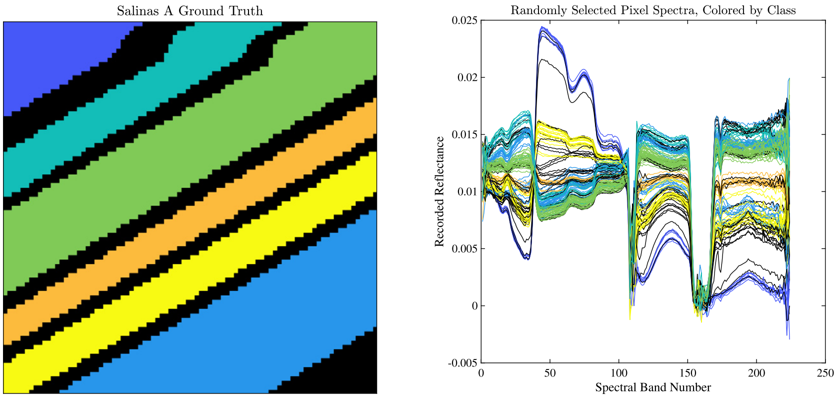
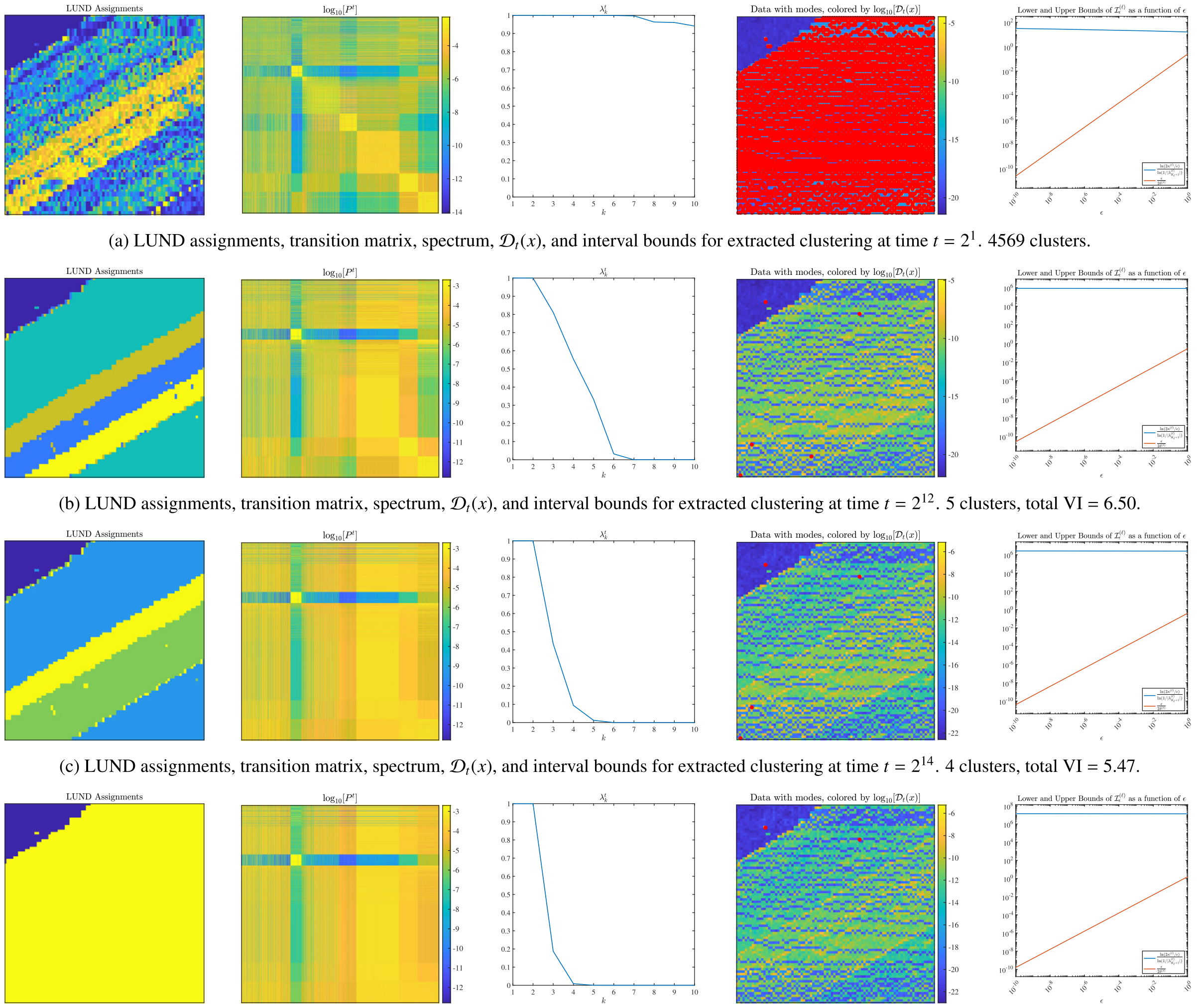


Fig. 7. Visualization of Salinas A scene. In the left panel, the ground truth labels for the pixels are provided. Black indicates bare soil and other colors indicate crop type. Specifically, dark blue indicates broccoli greens, teal indicates corn senesced greens, green indicates 5-week romaine, orange indicates 6-week romaine, yellow indicates 7-week romaine, and light blue indicates 8-week romaine. In the right panel, the spectra of a random subset of the Salinas A HSI are provided. Each pixel is colored by its ground truth label.



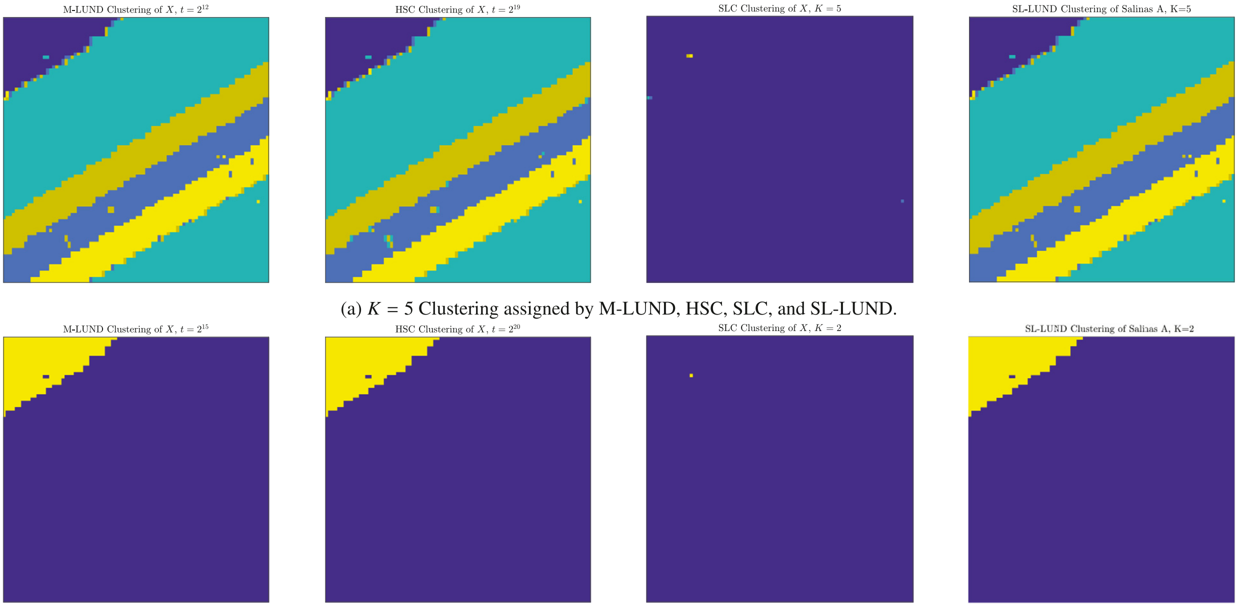
(a) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^1$. 4569 clusters.

(b) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^{12}$. 5 clusters, total VI = 6.50.

(c) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^{14}$. 4 clusters, total VI = 5.47.

(d) LUND assignments, transition matrix, spectrum, $\mathcal{D}_t(x)$, and interval bounds for extracted clustering at time $t = 2^{16}$. 2 clusters, total VI = 2.86. Optimal clustering.

Fig. 8. Diffusion process on the Salinas A HSI ($n = 7138$) [27]. Red points indicate cluster modes. Multiscale structure is detected. The number of estimated clusters monotonically decreases with t. Data indices in $\mathbf{P}$ are ordered by their ground truth class to illustrate multiscale structure.



(a) $K = 5$ Clustering assigned by M-LUND, HSC, SLC, and SL-LUND.

(b) $K = 2$ Clustering assigned by the M-LUND algorithm, HSC, SLC, and SL-LUND.

Fig. 9. Comparison of the $K = 5$ and $K = 2$ clusterings extracted by the M-LUND algorithm from the Salinas A HSI [27] against the $K = 5$ and $K = 2$ clusterings assigned by HSC and SLC [6,23,26].

greens crops separated (Fig. 8d, $t \in [2^{15}, 2^{18}]$). This highly stable, 2-cluster clustering is the total VI minimizer for the HSI (total VI = 2.86). Due to poor separation between clusters, no clustering of the HSI is guaranteed to be ϵ-separable by diffusion distances at any interval in the diffusion process.

As in Section 5.5, we compare the performance of the M-LUND clustering algorithm against MMS clustering, HSC, SLC, and SL-LUND. The same KNN graph with Gaussian kernel and same exponential sampling of the diffusion process were used for diffusion-based algorithms (M-LUND, MMS, and HSC, SL-LUND). All algorithms except for MMS clustering produce 5-cluster and 2-cluster clusterings of the Salinas A image (Fig. 9). The clusterings produced by SLC do not meaningfully correspond to ground truth labels. In contrast, the M-LUND algorithm, SL-LUND, and HSC extract clusterings that can be related to the ground truth labels in Fig. 7. The $K = 5$ clusterings generated by the three algorithms are similar, but romaine lettuce clusters estimated by the M-LUND (and hence SL-LUND) algorithm are slightly more coherent than those estimated by HSC (Fig. 9a). M-LUND, and HSC, and SL-LUND produce identical $K = 2$ clusterings, wherein broccoli greens pixels are separated from all else in the scene (Fig. 9b). We observe a different trend in the outputs of MMS clustering (Fig. 10). Indeed, clusterings rapidly transition early in the diffusion process from $K = 12$ to $K = 6$. This different trend may be due to MMS clustering not explicitly relying on the spectral decomposition of $\mathbf{P}$. In contrast, the M-LUND algorithm, HSC, and SL-LUND directly rely on the eigenfunctions of $\mathbf{P}$ to learn multiscale structure from X. Regardless, it is clear from the results of this section that a nonlinear diffusion-based clustering scheme is able to extract latent multiscale structure from the Salinas A HSI.

6. Conclusions and future work

We have shown that Markov chains derived from a data-generated graph facilitate the detection of clusterings at many scales, and the specific scale at which one wishes to cluster is tightly linked to the diffusion time parameter. With this in mind, we introduced the Multiscale Environment for Learning by Diffusion (MELD) data model: a family of latent clusterings of the dataset, parameterized by the diffusion time parameter. We have shown that each clustering in the MELD data model can be separated by diffusion

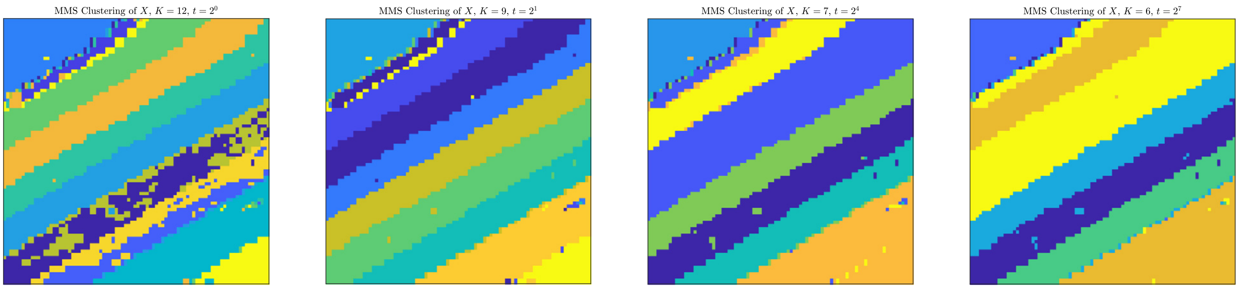


Fig. 10. Nontrivial clusterings of the Salinas A HSI [27] extracted by MMS clustering [39]. Fine-scale multiscale structure is extracted from the HSI, but MMS clusterings did not return any clusterings coarser than the $K = 6$ clustering in the rightmost panel.

distances during an interval of time that depends on the geometry of the dataset and that clustering. We showed that clusterings that consist of well-separated, coherent clusters are more stable in the diffusion process and will occur more frequently in the MELD data model.

We introduced the Multiscale Learning by Unsupervised Nonlinear Diffusion (M-LUND) clustering algorithm. The M-LUND algorithm is a multiscale extension of the LUND algorithm, which was introduced to leverage the attractive theoretical properties of diffusion distances [14,15,41,50]. The M-LUND algorithm learns multiscale cluster structure from data by varying a time parameter in the LUND algorithm across an exponential sampling of the diffusion process. It was proved that under reasonable assumptions on density and cluster structure, the M-LUND algorithm is guaranteed to extract an exponential sampling of the MELD data model from the dataset and choose a clustering from it as the minimizer of total VI. Our theoretical results were corroborated on synthetic and real data experiments.

The reliance of the MELD data model on ϵ results in a tension between the choice of an exponential sampling rate β that will sample the intervals during which MELD clusterings are ϵ-separable by diffusion distances and the guarantee that diffusion distances produce strong enough separation for the M-LUND algorithm to recover the MELD data model. For ϵ small, diffusion distances are guaranteed to provide strong separation on MELD clusterings during MELD intervals, but the range of β that is guaranteed to sample these intervals is small. On the other hand, for ϵ large, there is a wide range of β that are suitable for sampling these intervals, but in this case, ϵ-separation by diffusion distances may not be strong enough to guarantee that the M-LUND algorithm will recover the MELD data model.

A limitation of the MELD data model is its reliance on the geometric constant $\delta^{(\ell)}$: the maximum probability, across all points in X, of transitioning between clusters in a single time step. In a dataset in which outliers in one cluster overlap with outliers in another, $\delta^{(\ell)}$ can be quite pessimistic. For such a dataset, $\delta^{(\ell)}$ will be large, but diffusion is not likely to spread between cluster cores [41]. Indeed, in Section 5.1, we showed that the LUND algorithm performed well on overlapping Gaussians even though the separation parameter $\delta^{(\ell)}$ was large across the extracted clusterings. This suggests that the reliance of the MELD data model on the geometric constant $\delta^{(\ell)}$ forces it to exclude latent partitions of X that lack sufficiently strong separation within the original graph. Thus, the MELD data model may be improved by future work to allow for weaker separation between clusters.

To capture all scales of latent structure in a dataset, the M-LUND algorithm implements the LUND algorithm across an exponential sampling of the diffusion process. However, our numerical experiments suggest that latent clusterings are only extracted during a subset of the diffusion process. If this subset could be more precisely estimated before cluster analysis, the LUND algorithm could be implemented at those time steps alone, resulting in a decrease in complexity for the M-LUND algorithm. We hope to study this problem in future work as well.

All results in the present manuscript are for finite samples, and there is a dependence on n in many results. It is natural to consider continuum formulations of the data model and associated clustering algorithms, which may necessitate new models for multiscale mixtures of manifold data; see Section 3.3.1 for a more technical discussion.

Declaration of competing interest

The authors declare no competing interests.

Availability of data and code

Code to replicate results is available at https://github.com/sampolk/MultiscaleDiffusionClustering.

Acknowledgments

This research is partially supported by the US National Science Foundation grants NSF-DMS 1912737, NSF-DMS 1924513, and NSF-CCF 1934553. We thank Z. Liu and M. Barahona (Imperial College London) for their code. We thank the two anonymous referees for their constructive comments.

Appendix A. Notation

In Table A.4 we provide a table for easy referencing of the notation used throughout this article.

Appendix B. Background on classical clustering algorithms

Here, we review pertinent classical hierarchical clustering algorithms: K-Means, SLC, and DPC.

B.1. K-means clustering

The K-Means algorithm is a classical clustering algorithm that remains widely used. This algorithm estimates clusters $\{X_k\}_{k=1}^K$ by optimizing the distance of points to within-cluster means: $\mathcal{C} = \text{argmin}_{\{X_k\}_{k=1}^K} \sum_{k=1}^K \sum_{x\in X_k} \|x-\bar{x}_k\|_2^2$, where $\bar{x}_k$ denotes the mean data point of a cluster X_k. Many variants of K-Means exist [5,36,53,66]. However, it is easy to see that K-Means is sensitive to outliers because of its use of Euclidean distances. One extension of K-Means that is less sensitive to outliers is the K-medoids clustering algorithm, which replaces the cluster mean with a cluster medoid [53]. Nevertheless, K-Means and its variants exhibit poor performance on data that do not resemble well-separated, near-spherical clusters of the same size [52].

B.2. Dendrogram-based hierarchical clustering

Dendrogram-based clustering algorithms extract a family of partitions from a dataset X, varying from fine to coarse in scale, that can be expressed in a *dendrogram*: a diagram representing a tree of clusterings [23,26]. More formally, a dendrogram represents a family of n clusterings $\big\{\{X_k^{(\ell)}\}_{k=1}^{\ell}\big\}_{\ell=1}^{n}$ such that $\{X_k^{(n)}\}_{k=1}^n$ is the clustering consisting of n singletons and $\{X_k^{(1)}\}_{k=1}^1$ is the clustering consisting of a single cluster. Agglomerative hierarchical clustering algorithms initialize at $\ell = n$ and create intermediate clusterings $\{X_k^{(\ell)}\}_{k=1}^{\ell}$ by merging two clusters in $\{X_k^{(\ell+1)}\}_{k=1}^{\ell+1}$ found to minimize a linkage function [23,26]. On the other hand, divisive hierarchical clustering algorithms initialize at $\ell = 1$ and create intermediate clusterings $\{X_k^{(\ell+1)}\}_{k=1}^{\ell+1}$ by splitting a cluster at each scale. One of the more popular hierarchical clustering algorithms is the SLC algorithm, which builds a hierarchy of partitions by iteratively merging clusters using $\mathcal{L}_{\text{SLC}}(X_1, X_2) = \min_{x_1\in X_1, x_2\in X_2} \|x_1 - x_2\|_2$ as its linkage function [23,26]. Despite its widespread use

Table A.4
Notation used in the article listed in order of its first appearance. We refer to the multiscale analogues of certain notations after a back-slash.

Notation	Interpretation
$X = \{x_i\}_{i=1}^n \subset \mathbb{R}^D$	Data points to cluster
$X_k \ / \ X_k^{(\ell)}$	The k^{th} cluster of the clustering $\{X_k\}_{k=1}^K \ / \ \{X_k^{(\ell)}\}_{k=1}^{K_\ell}$
$\mathcal{C}$	Estimated cluster assignments
$\mathbf{W}$	Weight matrix
σ	Diffusion scale parameter
$\mathbf{D}$	Degree matrix
$\mathbf{P}$	Markov transition matrix
π	Stationary distribution of $\mathbf{P}$
$\{(\psi_i, \lambda_i)\}_{i=1}^n$	Right eigenvectors and eigenvalues of $\mathbf{P}$, sorted according to $\lvert\lambda_i\rvert$ in non-increasing order
$\Phi(x)$	Laplacian eigenmap, evaluated at $x \in X$
$D_t(x, y)$	Diffusion distance between points x and y at diffusion time step t
$\Psi_t(x)$	Diffusion map at time t, evaluated at $x \in X$
$\mathbf{S} \ / \ \mathbf{S}^{(\ell)}$	Stochastic complement of $\mathbf{P}$ with respect to the clustering $\{X_k\}_{k=1}^K \ / \ \{X_k^{(\ell)}\}_{k=1}^{K_\ell}$
$\mathbf{S}^\infty \ / \ \mathbf{S}_\infty^{(\ell)}$	$\lim_{t\to\infty} \mathbf{S}^t \ / \ \lim_{t\to\infty} [\mathbf{S}^{(\ell)}]^t$
$\mathbf{Z} \ / \ \mathbf{Z}^{(\ell)}$	Invertible matrix diagonalizing $\mathbf{S} \ / \ \mathbf{S}^{(\ell)}$
$\lambda_{K+1} \ / \ \lambda_{K_\ell+1}^{(\ell)}$	First non-unity eigenvalue of $\mathbf{S} \ / \ \mathbf{S}^{(\ell)}$
$\delta \ / \ \delta^{(\ell)}$	$\lVert\mathbf{P} - \mathbf{S}\rVert_\infty \ / \ \lVert\mathbf{P} - \mathbf{S}^{(\ell)}\rVert_\infty$
$\kappa \ / \ \kappa^{(\ell)}$	Infinity-norm condition number of diagonalizing $\mathbf{S} \ / \ \mathbf{S}^{(\ell)}$
$\mathcal{I}_\epsilon \ / \ \mathcal{I}_\epsilon^{(\ell)}$	Interval of time during which $\{X_k\}_{k=1}^K \ / \ \{X_k^{(\ell)}\}_{k=1}^{K_\ell}$ is ϵ-separable by diffusion distances
$D_t^{\text{in}} \ / \ D_t^{\text{in}}(\ell)$	Maximum within-cluster diffusion distance for the clustering $\{X_k\}_{k=1}^K \ / \ \{X_k^{(\ell)}\}_{k=1}^{K_\ell}$
$D_t^{\text{btw}} \ / \ D_t^{\text{btw}}(\ell)$	Minimum between-cluster diffusion distance for the clustering $\{X_k\}_{k=1}^K \ / \ \{X_k^{(\ell)}\}_{k=1}^{K_\ell}$
$\gamma(t) \ / \ \gamma^{(\ell)}(t)$	Measure of how the ℓ^1- and ℓ^2-norm differ across rows of $\mathbf{P}^t - \mathbf{S}^\infty \ / \ \mathbf{P}^t - \mathbf{S}_\infty^{(\ell)}$
$p(x)$	Kernel density estimate, evaluated at $x \in X$
σ_0	KDE bandwidth
$NN(x, N)$	Set of N ℓ^2-nearest neighbors of x in the dataset X
$\rho_t(x)$	Diffusion distance between x and that data point's D_t-nearest neighbor of higher density
$\mathcal{D}_t(x)$	$p(x)\rho_t(x)$
M	Number of distinct latent clusterings of X
$\mathcal{C}_t$	Latent clustering at time t
$\text{MELD}_\epsilon(X)$	The MELD data model of X for $\epsilon \in \left(0, \frac{1}{\sqrt{n}}\right)$
A_ϵ	$\bigcup_{\ell=1}^M \mathcal{I}_\epsilon^{(\ell)}$
$H(\mathcal{C})$	Entropy of the clustering $\mathcal{C}$
$I(\mathcal{C}, \mathcal{C}')$	Mutual information between clusterings $\mathcal{C}$ and $\mathcal{C}'$
$VI(\mathcal{C}, \mathcal{C}')$	Variation of information between clusterings $\mathcal{C}$ and $\mathcal{C}'$
β	Exponential sampling rate
T	$\left\lceil \log_\beta \left(\log_{\lvert\lambda_2\rvert} \left(\frac{\tau \pi_{\min}}{2} \right) \right) \right\rceil$
τ	Stationarity threshold
J	Time samples during which nontrivial clusterings are extracted by the LUND algorithm
$\text{VI}^{(\text{tot})}(\mathcal{C}_t)$	$\sum_{s\in J} \text{VI}(\mathcal{C}_t, \mathcal{C}_s)$
$\mathcal{M}_t$	Cluster-wise empirical density maximizers for the latent clustering at time t
$\{x_{m_k}^{(t)}\}_{k=1}^n$	The points in X, sorted according to $\mathcal{D}_t(x)$
B_ϵ	Transition regions between clusterings in $\text{MELD}_\epsilon(X)$
$\Delta(t)$	Relative pointwise distance of $\mathbf{P}^t$ to its stationary distribution π
d	Doubling dimension of the dataset
$NMI(\mathcal{C}, \mathcal{C}')$	Normalized mutual information between clusterings $\mathcal{C}$ and $\mathcal{C}'$

in practice, SLC has been shown to be statistically inconsistent if the dimension of the dataset is greater than 1 [28]. Moreover, its linkage function's reliance on Euclidean distances makes it sensitive to small perturbations and outliers.

B.3. Density peak clustering

Density-based clustering algorithms learn regions of high and low empirical density to cluster a dataset [16,21,24,31,70]. DPC is a widely-utilized example of a mode-based clustering algorithm [56]. DPC labels high-density points that are far in Euclidean space from other high-density points as *modes* of clusters in the dataset. Non-modal points are then paired with a labeled nearest neighbor iteratively. Due to its use of Euclidean distances, DPC often fails on data with nonlinear structure [41].

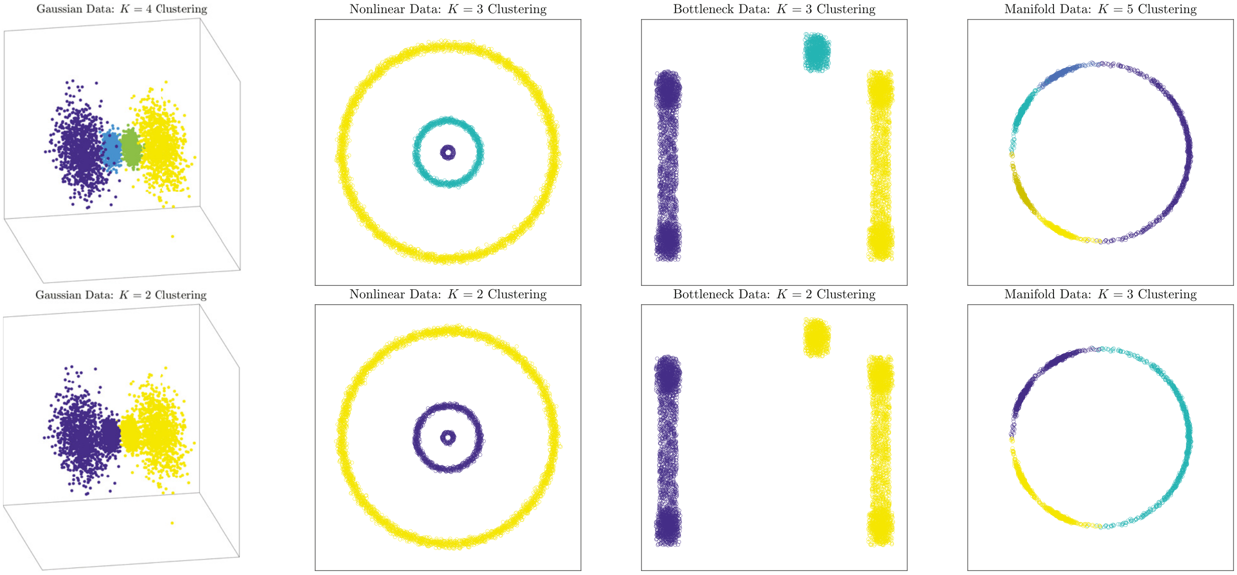
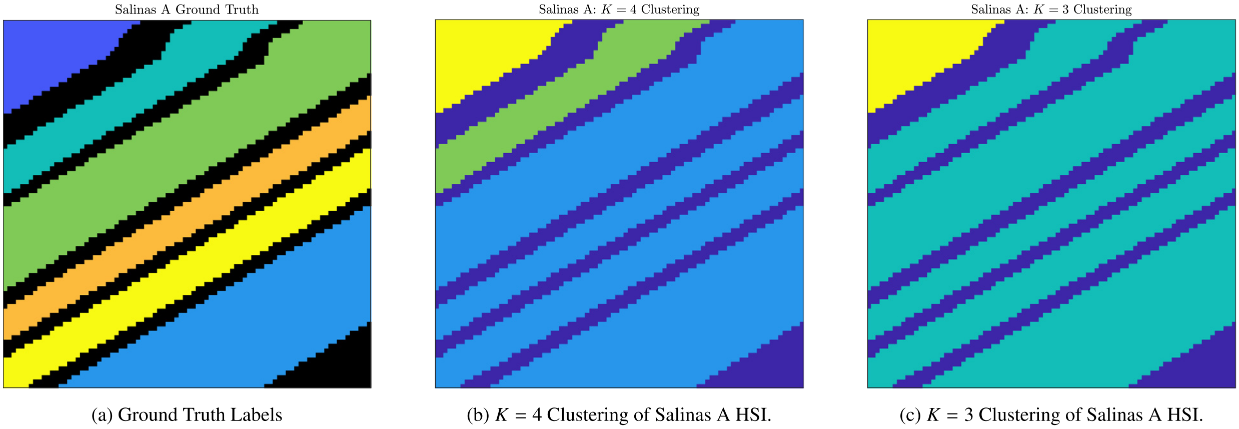


Fig. C.11. Multiscale ground truth labels for synthetic datasets analyzed in Sections 5.1-5.4. We remark that there is a $K = 2$ clustering missing for the Manifold data, where the first and fourth quadrants of S^1 are separated from the second and third; see Fig. 6d.

Appendix C. Multiscale labels of synthetic and hyperspectral data

In this appendix, visualizations are provided for the multiscale labels of the datasets analyzed in the numerical experiments presented in Section 5. The multiscale labels for the synthetic datasets analyzed in Sections 5.1-5.4 are given in Fig. C.11. In the first row of Fig. C.11, each point is colored according to the distribution from which it was been sampled, yielding fine-scale ground truth labels. In the second row of Fig. C.11, data points from nearby clusters are grouped together, yielding a coarser scale of ground truth labels. The multiscale labels for the Salinas A HSI analyzed in Section 5.6.are provided in Fig. C.12. The ground truth labels for the HSI are given in Fig. C.12a; similar ground truth classes were combined to obtain the coarser ground truth clusterings visualized in Figs. C.12b-C.12c. Specifically, in Fig. C.12b, broccoli greens (visualized in yellow) are separated from corn-senesced greens (visualized in green) and romaine lettuce of all maturity (visualized in light blue). On the other hand, in Fig. C.12c, the corn-senesced greens class is combined with the romaine lettuce classes (visualized in teal) but broccoli greens remain separate.



(a) Ground Truth Labels

(b) $K = 4$ Clustering of Salinas A HSI.

(c) $K = 3$ Clustering of Salinas A HSI.

Fig. C.12. Multiscale labels for the Salinas A HSI [27] analyzed in Section 5.6.

References

[1] Y.Y. Ahn, J.P. Bagrow, S. Lehmann, Link communities reveal multiscale complexity in networks, Nature 466 (2010) 761–764.

[2] E. Arias-Castro, Clustering based on pairwise distances when the data is of mixed dimensions, IEEE Trans. Inf. Theory 57 (2011) 1692–1706.

[3] E. Arias-Castro, G. Chen, G. Lerman, et al., Spectral clustering based on local linear approximations, Electron. J. Stat. 5 (2011) 1537–1587.

[4] E. Arias-Castro, G. Lerman, T. Zhang, Spectral clustering based on local PCA, J. Mach. Learn. Res. 18 (2017) 253–309.

[5] D. Arthur, S. Vassilvitskii, K-means++: the advantages of careful seeding, Technical Report, Stanford, 2006.

[6] A. Azran, Z. Ghahramani, Spectral methods for automatic multiscale data clustering, in: Proceedings CVPR, IEEE, 2006, pp. 190–197.

[7] A. Beygelzimer, S. Kakade, J. Langford, Cover trees for nearest neighbor, in: Proc. Int. Conf. Mach. Learn., 2006, pp. 97–104.

[8] V.D. Blondel, J.L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. Theory Exp. 2008 (2008) P10008.

[9] S. Botelho-Andrade, P.G. Casazza, D. Cheng, T. Tran, The exact constant for the $\ell_1 - \ell_2$ norm inequality, Math. Inequal. Appl. 22 (2019) 59–64.

[10] R.B. Cattell, The scree test for the number of factors, Multivar. Behav. Res. 1 (1966) 245–276.

[11] J. Cheeger, A lower bound for the smallest eigenvalue of the Laplacian, in: Problems in Analysis (Papers Dedicated To Salomon Bochner), Princeton Univ. Press, Princeton, NJ, 1970, pp. 195–200.

[12] G. Chen, S. Atev, G. Lerman, Kernel spectral curvature clustering (KSCC), in: Int. Conf. Comput. Vis., ICCV Workshops, IEEE, 2009, pp. 765–772.

[13] T. Chu, G. Miller, N. Walkington, A. Wang, Weighted Cheeger-Buser inequalities, with applications to cutting probability densities-as easy as 1, 2, 3, preprint, arXiv:2004.09589, 2020.

[14] R.R. Coifman, S. Lafon, Diffusion maps, Appl. Comput. Harmon. Anal. 21 (2006) 5–30.

[15] R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, S.W. Zucker, Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps, Proc. Natl. Acad. Sci. USA 102 (2005) 7426–7431.

[16] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 603–619.

[17] L. Cowen, K. Devkota, X. Hu, J.M. Murphy, K. Wu, Diffusion state distances: multitemporal analysis, fast algorithms, and applications to biological networks, SIAM J. Math. Data Sci. 3 (2021) 142–170.

[18] G. Dal Maso, An Introduction to Γ-Convergence, vol. 8, Springer Science & Business Media, 2012.

[19] A. Delmotte, E.W. Tate, S.N. Yaliraki, M. Barahona, Protein multi-scale organization through graph partitioning and robustness analysis: application to the myosin–myosin light chain interaction, Phys. Biol. 8 (2011) 055010.

[20] D. Dua, C. Graff, UCI machine learning repository, http://archive.ics.uci.edu/ml, 2017.

[21] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: KDD, 1996, pp. 226–231.

[22] X. Fan, Y. Yue, P. Sarkar, Y.X.R. Wang, A unified framework for tuning hyperparameters in clustering problems, preprint, arXiv:1910.08018, 2019.

[23] J. Friedman, T. Hastie, R. Tibshirani, The Elements of Statistical Learning, vol. 1, Springer Series in Statistics, Springer, New York, 2001.

[24] K. Fukunaga, L. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, IEEE Trans. Inf. Theory 21 (1975) 32–40.

[25] N. Garcia Trillos, D. Slepčev, A variational approach to the consistency of spectral clustering, Appl. Comput. Harmon. Anal. 45 (2018) 239–281.

[26] J.C. Gower, G.J. Ross, Minimum spanning trees and single linkage cluster analysis, J. R. Stat. Soc., Ser. C, Appl. Stat. 18 (1969) 54–64.

[27] J.A. Gualtieri, S.R. Chettri, R.F. Cromp, L. Johnson, Support vector machine classifiers as applied to AVIRIS data, in: JPL Airborne Geosci., 1999, pp. 217–227.

[28] J.A. Hartigan, Consistency of single linkage for high-density clusters, J. Am. Stat. Assoc. 76 (1981) 388–394.

[29] P.W. Holland, K.B. Laskey, S. Leinhardt, Stochastic blockmodels: first steps, Soc. Netw. 5 (1983) 109–137.

[30] M. Jerrum, A. Sinclair, Approximating the permanent, SIAM J. Comput. 18 (1989) 1149–1178.

[31] K. Jisu, Y.C. Chen, S. Balakrishnan, A. Rinaldo, L. Wasserman, Statistical inference for cluster trees, in: Adv. Neur. In., 2016, pp. 1839–1847.

[32] S. Lafon, A.B. Lee, Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 1393–1403.

[33] R. Lambiotte, J.C. Delvenne, M. Barahona, Random walks, Markov processes and the multiscale modular organization of complex networks, IEEE Trans. Netw. Sci. Eng. 1 (2014) 76–90.

[34] R. Lambiotte, J.C. Delvenne, M. Barahona, Dynamics and modular structure in networks, IEEE Trans. Netw. Sci. Eng. 1 (2015) 76–90.

[35] D.A. Levin, Y. Peres, Markov Chains and Mixing Times, AMS, 2017.

[36] A. Likas, N. Vlassis, J.J. Verbeek, The global K-means clustering algorithm, Pattern Recognit. 36 (2003) 451–461.

[37] A. Little, A. Byrd, A multiscale spectral method for learning number of clusters, in: Proc. Int. Conf. Mach. Learn., IEEE, 2015, pp. 457–460.

[38] A. Little, M. Maggioni, J.M. Murphy, Path-based spectral clustering: guarantees, robustness to outliers, and fast algorithms, J. Mach. Learn. Res. 21 (2020) 1–66.

[39] Z. Liu, M. Barahona, Graph-based data clustering via multiscale community detection, Appl. Netw. Sci. 5 (2020) 3.

[40] V. Lyzinski, M. Tang, A. Athreya, Y. Park, C.E. Priebe, Community detection and classification in hierarchical stochastic blockmodels, IEEE Trans. Netw. Sci. Eng. 4 (2016) 13–26.

[41] M. Maggioni, J.M. Murphy, Learning by unsupervised nonlinear diffusion, J. Mach. Learn. Res. 20 (2019) 1–56.

[42] F. McSherry, Spectral partitioning of random graphs, in: Ann. IEEE Symp. Found., IEEE, 2001, pp. 529–537.

[43] M. Meilă, Comparing clusterings–an information based distance, J. Multivar. Anal. 98 (2007) 873–895.

[44] M. Meilă, J. Shi, Learning segmentation by random walks, in: Adv. Neur. In., 2001, pp. 873–879.

[45] C.D. Meyer, Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems, SIAM Rev. 31 (1989) 240–272.

[46] B. Mohar, Y. Alavi, G. Chartrand, O. Oellermann, The Laplacian spectrum of graphs, in: Graph Theory, Combinatorics, and Applications, vol. 2, 1991, pp. 871–898.

[47] J.M. Murphy, M. Maggioni, Unsupervised clustering and active learning of hyperspectral images with nonlinear diffusion, IEEE Trans. Geosci. Remote Sens. 57 (2019) 1829–1845.

[48] J.M. Murphy, M. Maggioni, Spectral-spatial diffusion geometry for hyperspectral image clustering, IEEE Geosci. Remote Sens. Lett. 17 (2020) 1243–1247.

[49] B. Nadler, M. Galun, Fundamental limitations of spectral clustering, in: Adv. Neur. In., 2007, pp. 1017–1024.

[50] B. Nadler, S. Lafon, R.R. Coifman, I.G. Kevrekidis, Diffusion maps, spectral clustering and reaction coordinates of dynamical systems, Appl. Comput. Harmon. Anal. 21 (2006) 113–127.

[51] B. Nadler, S. Lafon, I. Kevrekidis, R.R. Coifman, Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators, in: Adv. Neur. In., 2006, pp. 955–962.

[52] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: Adv. Neur. In., 2002, pp. 849–856.

[53] H.S. Park, C.H. Jun, A simple and fast algorithm for K-medoids clustering, Expert Syst. Appl. 36 (2009) 3336–3341.

[54] T.P. Peixoto, Hierarchical block structures and high-resolution model selection in large networks, Phys. Rev. X 4 (2014) 011047.

[55] J.A. Rice, Mathematical Statistics and Data Analysis, Nelson Education, 2006.

[56] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (2014) 1492–1496.

[57] K. Rohe, S. Chatterjee, B. Yu, Spectral clustering and the high-dimensional stochastic blockmodel, Ann. Stat. 39 (2011) 1878–1915.

[58] G. Schiebinger, M.J. Wainwright, B. Yu, The geometry of kernelized spectral clustering, Ann. Stat. 43 (2015) 819–846.

[59] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 888–905.

[60] A. Sinclair, M. Jerrum, Approximate counting, uniform generation and rapidly mixing Markov chains, Inf. Comput. 82 (1989) 93–133.

[61] W.M. Song, B. Zhang, Multiscale embedded gene co-expression network analysis, PLoS Comput. Biol. 11 (2015) e1004574.

[62] A.D. Szlam, M. Maggioni, R.R. Coifman, J.C. Bremer Jr, Diffusion-driven multiscale analysis on manifolds and graphs: top-down and bottom-up constructions, in: Wavelets XI, SPIE, 2005, pp. 445–455.

[63] N.G. Trillos, F. Hoffmann, B. Hosseini, Geometric structure of graph Laplacian embeddings, preprint, arXiv:1901.10651, 2019.

[64] U. Von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (2007) 395–416.

[65] V. Vu, A simple SVD algorithm for finding hidden partitions, Comb. Probab. Comput. 27 (2018) 124–140.

[66] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, Constrained K-means clustering with background knowledge, in: Proc. Int. Conf. Mach. Learn., 2001, pp. 577–584.

[67] X. Wang, K. Slavakis, G. Lerman, Riemannian multi-manifold modeling, preprint, arXiv:1410.0095, 2014.

[68] X. Wang, K. Slavakis, G. Lerman, Multi-manifold modeling in non-Euclidean spaces, in: Artificial Intelligence and Statistics, in: PMLR, 2015, pp. 1023–1032.

[69] R. Xu, D.C. Wunsch, Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (2005) 645–678.

[70] X. Xu, M. Ester, H.P. Kriegel, J. Sander, A distribution-based clustering algorithm for mining in large spatial databases, in: Proc. Int. Conf. Data, IEEE, 1998, pp. 324–331.

[71] S. Zhang, J.M. Murphy, Hyperspectral image clustering with spatially-regularized ultrametrics, Remote Sens. 13 (2021) 955.