GEOMETRIC AND TOPOLOGICAL METHODS IN DATA ANALYSIS

CASEY JAO AND QIAO ZHOU

This is an on-going set of learning notes for geometric and topological techniques in data analysis. Some key words include manifold learning and topological data analysis.

Contents

1. Clustering

There is abundance of literature on various clustering techniques for data. This section begins by describing two (related) recently proposed methods for clustering data coming from intersecting surfaces, and concludes with some remarks on topological clustering.

1.1. Spectral clustering basics.

These notes are from the tutorial by Luxburg [8].

Many approaches to clustering data $X \subset \mathbb{R}^N$ seek to build a weighted graph $G = (X, W)$, where each weight w_{ij} is nonnegative and denotes some measure of "closeness" between vertices x_i and x_j, and apply standard graph clustering algorithms. The general problem of graph clustering is to find a partition of G into subsets $C_1, \ldots, C_k$ such that w_{ij} is "small" whenever vertices x_i, x_j belong to different clusters and w_{ij} is "large" whenever x_i, x_j belong to the same cluster.

Spectral clustering is a popular method based on the graph Laplacian. Its implementation involves standard linear algebra and can be motivated from several points of view.

There are several related definitions of a graph Laplacian, each of which has an associated spectral clustering algorithm. Let D be the diagonal matrix of degrees $d_1, \ldots, d_n$, where $d_i = \sum_j w_{ij}$.

The (unnormalized) *graph Laplacian* is

$$L = D - W.$$

Then L is positive semidefinite since

$$\langle Lf, f \rangle = \sum_{i,j} w_{ij} f_i^2 - \sum_{i,j} w_{ij} f_i f_j = \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2,$$

From this it is not hard to deduce that

Lemma 1. *The number of connected components of the graph is* $\dim \ker L$. *Moreover,* $\ker L$ *is spanned by the indicator functions of the components.*

Suppose now that G has k connected components and we want to partition the vertices $\{x_1, \ldots, x_n\}$ into k clusters. In this case the clusters should be precisely the connected components.

The basic unnormalized spectral clustering algorithm, which dates back to the 1970's, works as follows:

(1) Compute eigenvectors $x_1, \ldots, x_k \in \mathbb{R}^n$ corresponding to the k smallest eigenvalues of L, and form the $n \times k$ matrix $Y = [x_1, \ldots, x_k]$.
(2) Apply k-means clustering to the rows of Y. Call the rows $y_1, \ldots, y_n$ and clusters $C_1, \ldots, C_k$.
(3) Assign x_i to cluster j if $y_i \in C_j$.

k-means clustering refers to the following problem: given points $X \subset \mathbb{R}^N$ and a positive integer k, find a partition of X into k clusters $S_1, \ldots, S_k$ which minimizes

$$\sum_{i=1}^{k} \sum_{\mathbf{y} \in S_i} \|\mathbf{y} - \mu_i\|^2,$$

where $\mu_i = \frac{1}{|S_i|} \sum_{\mathbf{y} \in S_i} \mathbf{y}_i$. An iterative algorithm is common used.

More recent variants have been developed based on normalized graph Laplacians:

$$L_{sym} := D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2} \quad \text{(Ng, Jordan, Weiss 2002)}$$

$$L_{rw} := D^{-1} L = I - D^{-1} W \quad \text{(Shi and Malik 2000)}.$$

Remark 2. Spectral clustering works exactly when the graph has k connected components and we look for k clusters. It is also stable under small perturbations. However, its main attraction is that it boils down to a standard linear algebra problem.

Each of the following manifold clustering methods ultimately applies spectral clustering to the graph (X, W) with judicious choice of weights.

1.2. **Local PCA.** "Spectral clustering based on local PCA" [Arias-Castro, Lerman, Zhang] [1].

Clustering for surfaces in Euclidean space. This is basically a simplified version of previous algorithms proposed by Goldberg/Kushnir but with proofs.

1.2.1. *Problem.* Given data points $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^D$ which are assumed to concentrate near smooth d-dimensional surfaces $S_1, \ldots, S_K \subset \mathbb{R}^D$, assign each point to one of K clusters $C_1, \ldots, C_K$ such that each cluster consists of all the points from one surface.

1.2.2. *Algorithms.* Parameters: neighborhood radius $r > 0$, spatial scale $\varepsilon > 0$, covariance/projection scale $\eta > 0$.

Idea: build a similarity graph on the data points so that two points $\mathbf{x}_i$ and $\mathbf{x}_j$ are "close" if both of the following hold:

- $\mathbf{x}_i$ and $\mathbf{x}_j$ are close in the Euclidean sense: $\|\mathbf{x}_i - \mathbf{x}_j\| < \varepsilon$ for some distance threshold ε.
- The estimated tangent spaces of the surfaces at $\mathbf{x}_i$ and $\mathbf{x}_j$ look nearly the same.

Tangent spaces are compared as follows. Choose a parameter $0 < r \ll \varepsilon$. For each $\mathbf{x}_i$, form the sample covariance matrix $\mathbf{C}_i$ for the points in the ball $B_r(\mathbf{x}_i)$ of radius r with center $\mathbf{x}_i$. Assuming that the samples are roughly uniform, one expects $\|\mathbf{C}_i\| \approx r^2$. The discrepancy between tangent spaces at two points $\mathbf{x}_i$, $\mathbf{x}_j$ is measured by comparing directly the covariance matrices, or by comparing the orthogonal projections $\mathbf{Q}_i$ onto the space spanned by the eigenvectors corresponding to the largest eigenvalues of $\mathbf{C}_i$. The latter measure is less sensitive to the scaling of the data (e.g.

points on a line near a boundary will have smaller covariance matrices but identical projections as interior points).

The comparison of tangent spaces becomes relevant near an intersection of surfaces. Assuming that the surfaces bend much more slowly than the angle of intersection, two points $\mathbf{x}_i$ and $\mathbf{x}_j$ in a neighborhood of an intersection should come from the same surface iff the tangent spaces at those points have small relative angle (as measured by comparing the covariance matrices or the associated orthogonal projections).

The authors propose three related algorithms and analyze the first two:

Algorithm 2 (analyzed):

- For each $\mathbf{x}_i$, discard each $\mathbf{x}_j \in B_\varepsilon(\mathbf{x}_i)$ such that $\|\mathbf{C}_i - \mathbf{C}_j\| \geq \eta r^2$. The aim of this step is to temporarily remove points near intersections that could mess up the following
- Define the affinity matrix
$$W_{ij} = 1_{\{\|\mathbf{x}_i - \mathbf{x}_j\| < \varepsilon\}} 1_{\{\|\mathbf{C}_i - \mathbf{C}_j\| < \eta r^2\}},$$
and determine the connected components of the resulting graph.
- Group each of the removed points with the closest connected component.

Algorithm 3 (analyzed):

- Let $\mathbf{Q}_i$ be the orthogonal projection onto the span of the eigenvectors whose eigenvalues exceed $\sqrt{\eta}\|\mathbf{C}_i\|$, and determine the connected components of the graph with affinity matrix
$$W_{ij} = 1_{\{\|\mathbf{x}_i - \mathbf{x}_j\| < \varepsilon\}} 1_{\{\|\mathbf{Q}_i - \mathbf{Q}_j\| < \eta\}},$$

Algorithm 4 (the one actually implemented but not analyzed):

- Preliminary reduction: choose an r-sparse subset $Y = \{\mathbf{y}_i\}$; idea is that without sampling at two different scales (different values of r), the covariance matrices within a fixed ball yield little additional information.
- For each $\mathbf{y}_i$, let $\mathbf{C}_i$ be the covariance matrix for the points in $B_r(\mathbf{y}_i)$, and $\mathbf{Q}_i$ be the orthogonal projection onto the first d eigenvectors of $\mathbf{C}_i$.
- Construct the similarity graph on Y with affinity matrix
$$W_{ij} = \exp\Big(-\frac{\|\mathbf{y}_i - \mathbf{y}_j\|}{\varepsilon^2}\Big) \exp\Big(-\frac{\|\mathbf{Q}_i - \mathbf{Q}_j\|}{\eta^2}\Big),$$
and apply spectral clustering to obtain clusters $C_1, \ldots, C_k$
- Group each $\mathbf{x}_i$ with the cluster with the nearest mean.

Remark 3. Using soft cutoffs (thus making everything weakly connected) makes the algorithm more robust. With hard cutoffs, points falling just outside the prescribed thresholds would form their connected components.

Remark 4. How should the parameters be chosen? The authors set the spatial scale ε and projection scale ηas follows:

$$\varepsilon = \max_i \min_{j \neq i} \|\mathbf{y}_i - \mathbf{y}_j\|,$$
$$\eta = \text{median}\{\|\mathbf{Q}_i - \mathbf{Q}_j\| : (i,j) : \|\mathbf{y}_i - \mathbf{y}_j\|\}$$

The parameter r is chosen by hand in each case and should be smaller than ε. It should be small enough so that the surfaces look nearly flat in a ball of radius r, but also large enough so that there are enough points in a ball of radius r to compute tangent space approximations. Further, the distance parameter ε should be small enough so that each surface looks relatively flat in a ball of radius ε. *Automatic tuning of parameters still remains a challenging problem.*

Remark 5. These algorithms focus mainly on correctly resolving intersections. Two nearly parallel surfaces S_1 and S_2 that are within distance ε of each other are not distinguished since their estimated tangent spaces will be very similar. The "geodesic angle filtering" criterion of Lerman et. al. would address this issue.

1.2.3. *Theoretical results.* Let S_1, S_2 be compact C^2 d-submanifolds of $\mathbb{R}^D$ with *reach* $\geq r$. This means that for each x in the r-tubular neighborhood of S_i, there is a unique closest point of S_i to x. The reach provides an upper bound on the curvature of S_i.

Assume iid samples $X = \{x_1, \ldots, x_n\}$ have the form $x_i = s_i + z_i$, where s_i is sampled uniformly with respect to the surface measure of $S_1 \cup S_2$ and z_i is uniform over the ball of radius $\tau > 0$(the noise level).

Theorem 6. *Assume S_1 and S_2 intersect at positive angle such that $S_1 \cap S_2$ has positive reach. If the parameters $\tau, r, \eta, \varepsilon$ are chosen such that*

$$\eta/1/C,\ \varepsilon \leq \eta/C,\ r \leq \varepsilon/C,\ \tau \leq r\eta/C,$$

for a sufficiently large $C > 0$ depending on the geometry of $S_1, S_2, S_1 \cap C_2$, then

- *Algorithm 2 returns at exactly two groups which correctly clusters the data points outside distance Cr from the intersection.*
- *Algorithm 3 returns at least three groups such that data points from different surfaces are not clustered together unless they are within distance Cr from the intersection.*

1.3. **MGM-clustering.** "Riemannian multi-manifold modeling" [Wang, Slavkis, Lerman] [9]

1.3.1. *Problem.* Let M be a Riemannian manifold and $S_1, S_2 \subset M$ be two (possibly intersecting) submanifolds of equal dimensions. The submanifolds are not observed in advance. Given points $X = \{x_1, \ldots, x_n\} \subset M$ which are known *a priori* to concentrate near each submanifold, determine to which submanifold each x_i belongs.

1.3.2. *Theoretical Algorithm (TGCT).* Given points $X = \{x_1, \ldots, x_n\}$, a number $K \geq 2$ of clusters, a neighborhood radius $r > 0$, a projection threshold η, a distance threshold σ_d, and an angle threshold σ_a.

For each x_i:

(1) Compute the points $x_i^j \in J(x_i, r) := B(x_i, r) \cap X$ in geodesic normal coordinates centered at x_i.
(2) Use PCA on the points $J(x_i, r)$ (in geodesic coordinates) to compute the estimated tangent subspace $T^E_{x_i}S$ of the underlying submanifold S. More specifically, $T^E_{x_i}S$ is spanned by the eigenvectors of the sample covariance matrix $\mathbf{C}_{x_i}$ of x_i^j whose eigenvalues exceed $\eta\|\mathrm{C}_{x_i}\|$. The number of such eigenvectors is an estimate of the dimension of S. Note however that S might not pass through x_i, but if x_i is close to S then S should still look basically like a (slightly translated) subspace
(3) Assuming that S is a *geodesic* submanifold, for each point $x_j \in X$, x_j should belong to S iff the geodesic from x_i to x_j intersects $T^E_{x_i}S$ at a small angle θ_{ij}, called the *empirical geodesic angle*.
(4) Form a similarity graph on the points X with weights W_{ij} of the form
$$W_{ij} = 1_{d_g(x_i,x_j)<\sigma_d} 1_{\dim(T^E_{x_i}S)=\dim(T^E_{x_j}S)} 1_{(\theta_{ij}+\theta_{ji})<\sigma_a}.$$
(5) Run spectral clustering on the similarity graph.

Remark 7. Since $\dim S_1 = \dim S_2$, the only purpose of comparing dimensions is to ensure that points near an intersection of two manifolds S_1 and S_2 form their own cluster. The "practical"

algorithm omits this comparison, which is okay so long as only a small fraction of points lie near intersections.

Remark 8. The similarity matrix encodes the same underlying notions of "closeness" as in the local PCA paper (Algorithm 2). The empirical geodesic angle θ_{ij} plays an analogous role as $\|\mathbf{C}_i - \mathbf{C}_j\|$ or $\|\mathbf{Q}_i - \mathbf{Q}_j\|$; local covariance matrices can't be directly compared on a manifold if the points lie in different charts.

Remark 9. A "practical" algorithm is also formulated and implemented with technical tweaks. Hard cutoffs are replaced by soft ones. Also, the distance component $1_{d_g(x_i,x_j)<\sigma_d}$ is replaced by weights S_{ij} coming from a sparse coding problem.

Remark 10. If $x \in S_1$, the geodesic submanifold hypothesis implies that for any other $y \in S_1$ the shortest length geodesic from x to y lies in S_1 (so the geodesic angle θ_{xy} is zero in this case). If y is close to S_1, then θ_{xy} should still be small regardless of whether S_1 is geodesic. Their "practical" algorithm appears to work for examples where the hypothesis is violated.

Remark 11. This algorithm relies on being able to compute the logarithm map efficiently (i.e. the inverse of the exponential map $\exp_x : T_xM \to M$). Grassmannians, symmetric positive-definite matrices, spheres were considered in the article.

1.3.3. *Theoretical results.*

Theorem 12. *Let $S_1, S_2 \subset M$ be smooth compact d-submanifolds. Assume data $X = \{x_1, \ldots, x_N\}$ are sampled from the uniform distribution on the tubular neighborhood of each S_i with radius $\tau > 0$ (the noise level). For suitable choices of the parameters $r, \sigma_d, \sigma_a, \eta$ depending on τ and the geometry of the submanifolds and their intersection, then with probability $1 - O(Ne^{-CNr^{d+2}})$ the TGCT algorithm correctly classifies a subset $X^* \subset X$ of fraction at least $1 - C(r+\tau)^{d-\dim(S_1 \cap S_2)}$.*

Remark 13 (Section 5.8 in the paper)**.** Most of the proof considers the noiseless case $\tau = 0$, so that $X \subset S_1 \cup S_2$, and incorporates small levels of noise $\tau \leq cr$, $c \ll 1$, at the very end by simple perturbative considerations. Robustness under higher levels of noise was investigated in their numerical experiments and the authors think their theoretical results could be improved.

1.4. **Topological clustering.**

1.4.1. *Morse Theory.* Morse Theory is mostly about understanding the topology of a manifold M through the level sets of certain differentiable functions $f : M \to \mathbb{R}$. A smooth real-valued function on a manifold M is a Morse function if it all of its critical points have nondegenerate Hessian matrices (a Hessian matrix at a critical point is the matrix of second derivatives at that point). A basic result of Morse theory says that almost all functions are Morse functions. Technically, the Morse functions form an open, dense subset of all smooth functions $M \to \mathbb{R}$ in the C^2 topology. This is sometimes expressed as "a typical function is Morse" or "a generic function is Morse". The index of a non-degenerate critical point b of f is the dimension of the largest subspace of the tangent space to M at b on which the Hessian is negative definite. This corresponds to the intuitive notion that the index is the number of directions in which f decreases.

Given a Morse function f, one important question is about the changes in the topology of $M^a = f^{-1}(-\infty, a]$ as a varies.

Theorem 14. *Suppose f is a smooth real-valued function on M, $a < b$, $f^{-1}([a,b])$ is compact, and there are no critical values between a and b. Then M^a is diffeomorphic to M^b, and M^b deformation retracts onto M^a.*

It is also of interest to know how the topology of M^a changes when a passes a critical point. The following theorem answers that question.

Theorem 15. *Suppose f is a smooth real-valued function on M and p is a non-degenerate critical point of f of index γ, and that $f(p) = q$. Suppose $f^{-1}([q-\epsilon, q+\epsilon])$ is compact and contains no critical points besides p. Then $M^{q+\epsilon}$ is homotopy equivalent to $M^{q-\epsilon}$ with a γ-cell attached.*

1.4.2. *Density Clustering and Mode Clustering.* This part of our notes is a re-explanation of section 2 of [10].

Let $X = \{x_1, \ldots, x_n\}$ be a random sample from a probability distribution P supported on some subset $\mathcal{X} \subset \mathbb{R}^d$.

Assume P has a density $p(x)$. *Density clusters* are regions in $\mathcal{X}$ where $p(x)$ is large. More specifically, the density clusters at level t, denoted by $\mathcal{C}_t$, are the connected components of the super-level sets

$$L_t := \{x : p(x) > t\}.$$

The set $\mathcal{C} := \bigcup_{t \geq 0} \mathcal{C}_t$ has a tree structure induced by set inclusion. This is very similar to the perspective of looking at level sets of Morse functions.

In practice, it is common to use a kernel density estimator $\hat{p}_h(x)$ which involves the kernel K(e.g. Gaussian kernel) and a "bandwidth" h. To find the clusters, we choose another tuning parameter and make a graph with the nodes being $X_1, ..., X_n$.

Since the density clusters defined above have a tree structure, we define the density tree to be the two-dimensional tree that keeps track of the merging of clusters as t decreases. This is completely analogous to the changes in the topology of the level sets as we pass critical values of a Morse function.

For another related application of Morse theory, let's consider a different clustering method called mode clustering. In this case, the clusters are the attracting sets for the local maxima of the probability density function, with the flow given by the gradient flow. This gives a decomposition of the underlying space $\mathbb{R}^d$, and therefore divides our data set into different clusters according to the modes (local maxima/critical points).

2. Data Simplification and De-noising

2.1. **Dimension Reduction.** The general goal of dimension reduction is to map data $X \subset \mathbb{R}^D$ from a high-dimensional space to a low-dimensional space $\mathbb{R}^d$, $d \ll D$ in a manner that "preserves structure". Principal components analysis (PCA) is the method of choice if X concentrates near a linear subspace. However, describing data with nonlinear structure is a more delicate problem.

We briefly review some classical methods on nonlinear dimension reduction, including MDS, Isomap and Laplacian eigenmaps. There are other important methods that are not explained here, like LLE [6], t-SNE [5] , etc. For instance, LLE first finds weights w_{ij} such that the square of the distance between any data point x_i and the linear combination of its neighboring data points $\Sigma_j w_{ij} x_j$ is minimized, and then finds lower dimensional images y_i such that $||y_i - \Sigma_j w_{ij} y_j||^2$ is minimized.

On the other hand, a different method called t-SNE uses a gradient descent algorithm to minimize the "relative entropy" between the original data set X and its lower dimensional image Y. It is quite different from the aforementioned local methods and appears to preserve topological features of the data better. One interesting open question is: *is it possible to derive low-dimensional embedding methods that explicitly preserve topological features of the data?* [10]

2.1.1. *MDS and Isomap.* Multidimensional scaling (MDS) and Isomap [7] both try to find a map $f : X \to \mathbb{R}^d$ that preserves all pairwise distances to the best extent possible. Suppose $d_X(\cdot, \cdot)$ is a

metric on X. The mapping $Y = f(X)$ is chosen to minimize

$$\sum_{i,j} \big| d_X(x_i, x_j)^2 - |y_i - y_j|^2) \big|^2.$$

Classical MDS takes global Euclidean distances $d_X(x_i, x_j) = |x_i - x_j|$ for all i, j. However, this does not faithfully represent data coming from a nonlinear space M, such as the "Swiss roll" dataset, since points close together in the ambient Euclidean space could be far apart in geodesic distance d_G on the manifold.

The main innovation of Isomap is to replace the global distances on X with approximate geodesic distances. First, d_X is used to discover a "neighbor" relation on X, for instance, based on a threshold criterion $x_i \sim x_j$ if $d_X(x_i, x_j) < \varepsilon$ for some $\varepsilon > 0$, or by a k-nearest-neighbors criterion. Whatever the method chosen, this relation defines a weighted graph $G = (X, W)$ with $w_{ij} = d_X(x_i, x_j)$ if $x_i \sim x_j$. Finally, the shortest-path distance d_G on the graph is used in place of d_X as the input to the above minimization procedure.

2.1.2. *Laplacian eigenmaps.* The method of Laplacian eigenmaps emphasizes preserving *locality* – nearby points in the data space should get mapped to nearby points in the lower dimensional space.

Given a data set $X \subset \mathbb{R}^N$, construct a similarity graph $G = (X, W)$, where x_i and x_j are joined by an edge according to one of the following rules

- n-nearest-neighbors: $w_{ij} = 1$ if x_i is among the n nearest neighbors of x_j *or* x_j is among the n nearest neighbors of x_i. Distances are measured in the ambient space $\mathbb{R}^N$.
- $|x_i - x_j| < \varepsilon$ for some threshold ε.

Also choose weights w_{ij} according to one of the following schemes:

- $w_{ij} = 1$ iff x_i and x_j are joined by an edge.
- Heat kernel: for some parameter t, set $w_{ij} := \exp(-\frac{|x_i - x_j|}{t})$ if x_i and x_j are connected.

Let $L = D - W$ be the unnormalized Laplacian matrix of G. Then for any map $f = (f^1, \ldots, f^d) : X \to \mathbb{R}^d$, writing $y_k^i = f^i(x_k)$, the quantity

$$\frac{1}{2} \sum_{ij} w_{ij} |y^i - y^j|^2 = \sum_{i=1}^{X} |\langle L y_i, y_i \rangle| = \operatorname{tr}(Y^T L Y)$$

measures how well the neighborhood property is preserved. Of course the constant map minimizes this quantity, but that would yield no information about X. Instead, we want to insist also that the vectors y_i be linearly independent and orthogonal to the constant vector $\mathbf{1}$. More precisely, one determines the desired mapping $Y = f(X)$ by solving the constrained minimization problem

$$\min_{Y^T D Y = I} \operatorname{tr}(Y^T L Y).$$

The solution is given precisely by the eigenvectors y_i corresponding to the smallest eigenvalues λ_k of the normalized Laplacian $D^{-1}L$; equivalently, $D^{\frac{1}{2}} y_k$ are the orthonormal eigenvectors for the symmetrized Laplacian $D^{-1/2} L D^{-1/2}$.

When X comes from a compact manifold (M, g) and the similarity graph G is defined with the heat kernel weights, the graph Laplacian L can be regarded as a discrete analog of the Laplace-Beltrami operator $-\Delta_g$ of the underlying manifold and the eigenvectors y_i as approximations to the lowest-energy eigenfunctions of $-\Delta_g$, which solve a continuous analog of the preceding embedding problem. See Belkin and Niyogi (2003) for further details [2].

2.2. **Submanifold Estimation.** This part of our notes is a re-explanation of some parts of section 3 of [10].

Let $X = \{x_1, \ldots, x_n\}$ be a random sample from a probability distribution P supported on some subset $\mathcal{X} \subset \mathbb{R}^d$.

Sometimes the distribution P is supported on a submanifold $S \subset \mathbb{R}^d$ with $r = \dim(S) < d$.

A natural estimate for S is to take the union of ε-balls of each point:

$$\hat{S} := \bigcup_{i=1}^{n} B(x_i, \varepsilon).$$

It has been shown that under suitable assumptions on S and P, the estimator $\hat{S}$ can be made to converge in *Hausdorff distance* $H(\cdot, \cdot)$ to S; more precisely one has the bound

$$P(H(\hat{S}, S) > \varepsilon) \leq Cr^{-d}e^{-cn\varepsilon^d}.$$

It is unlikely that a sample will fall precisely on a submanifold S. A more realistic model is to allow a noise term from a distribution such as a Gaussian.

Most manifold learning methods assume that the distribution P is supported on some manifolds S. This is a very strong and unrealistic assumption. A weaker assumption is that there may exist some low dimensional sets where the density p has a relatively high local concentration. One way to make this more precise is through the idea of density ridges. Let $p_h(x)$ be a (estimated) probability density function. Without loss of generality, we may assume it to be a Morse function. Each critical point y of $p_h(x)$ a unique signature, which is given by the number of positive/negative eigenvalues of the Hessian at y. The tangent space at y is a direct sum of the subspace E_+ generated by the positive eigenvectors of the Hessian and the subspace E_- generated by the negative eigenvectors of the Hessian. Then a density ridge a lower-dimensional submanifold given by the exponential of the tangent space E_+ at a critical point y. Note that the modes/local maxima used in mode clustering are special cases of density ridges.

In addition, there is also a theory for estimating "stratified spaces". This applies to the case when P is supported on a union of intersecting manifolds. This is much less well developed than standard manifold estimation. Local PCA work is referenced here.

2.3. **Persistent Homology.** There are various constructions and algorithms for Persistent Homology. While simplicial, celluar and Cech complexes work well for computational purposes, conceptually we find it most natural to start from Morse theory and Morse homology. We first present the introduction to Morse theory and Morse homology from Wikipedia, then define Morse homology, followed by a quick explanation of some of its applications. These are based on "Persistent Homology-a Survey" by Herbert Edelsbrunner and John Harer [3], as well as "Topological Persistence and Simplification" by Herbert Edelsbrunner, David Letscher, and Afra Zomorodian [4].

2.3.1. *Morse Homology and Construction of Persistence Homology.* Given any (compact) smooth manifold M, let f be a Morse function and g a Riemannian metric on the manifold. The pair (f, g) gives us a gradient vector field. We say that (f, g) is Morse-Smale if the stable and unstable manifolds (attracting and repelling sets) associated to all of the critical points of f intersect each other transversely.

For any such (f, g), it can be shown that the difference in index between any two critical points is equal to the dimension of the moduli space of gradient flows between those points. Thus there is a one-dimensional moduli space of flows between a critical point of index i and one of index $i - 1$. Each flow can be reparametrized by a one-dimensional translation in the domain. After modding out by these reparametrizations, the quotient space is zero-dimensional — that is, a collection of oriented points representing unparametrized flow lines.

A chain complex $C_*(M,(f,g))$ may then be defined as follows. The set of chains is the $\mathbb{Z}$(or $\mathbb{Z}^2$ for data analysis)-module generated by the critical points. The differential d of the complex sends a critical point p of index i to a sum of index-$(i-1)$ critical points, with coefficients corresponding to the (signed) number of unparametrized flow lines from p to those index-$(i-1)$ critical points. The fact that the number of such flow lines is finite follows from the compactness of the moduli space.

The fact that this defines a complex (that is, that $d^2=0$) follows from an understanding of how the moduli spaces of gradient flows compactify. Namely, in d^2p the coefficient of an index-$(i-2)$ critical point q is the (signed) number of broken flows consisting of an index-1 flow from p to some critical point r of index $i-1$ and another index-1 flow from r to q. These broken flows exactly constitute the boundary of the moduli space of index-2 flows: The limit of any sequence of unbroken index-2 flows can be shown to be of this form, and all such broken flows arise as limits of unbroken index-2 flows. Unparametrized index-2 flows come in one-dimensional families, which compactify to compact one-manifolds. The fact that the boundary of a compact one-manifold is always zero proves that $d^2p=0$.

We denote the kernel of ∂_k to by Z_k, and the image of ∂_k by B_k.

The $k-$th homology group of a chain complex (C_*,d) is the quotient $H_k(Z_k/B_k)$. In the setting of Morse Homology this is basically the quotient of the group of critical points of index k by the group of critical points of index k which are the images of the gradient flow from some critical points of index $k+1$.

Intuitively, persistence homology is a test of how long homology classes live. In the context of Morse Homology, let $t_1<t_2...<t_m$ be the critical values of f and consider an interleaved sequence with $s_{i-1}<t_i<s_i$ for $1\leq i\leq m$. To capture the homology that exists at the beginning and the end we set $s_{-1}=t_0=-\infty$ and $t_{m+1}=s_{m+1}=\infty$. For each $-1\leq i\leq j\leq m+1$ we have the inclusion $M^{s_i}\subseteq M^{s_j}$ and the induced homomorphism between the corresponding homology groups $f_p^{i,j}:H_p(M^{s_i})\to H_p(M^{s_j})$. On the level of critical points, this map coincides with gradient flow. We call the image of $f_p^{i,j}$ a persistent homology group because it consists of classes born before s_i that are still alive at s_j. More explicitly, this is the quotient group $Z_p(M^{s_i})/B_p(M^{s_j})\cap Z_p(M^{s_i})$. Here kernels of differentials persist as move into bigger sublevel sets that include more critical points, but images of differentials could change as cycles that were not boundaries could become boundaries as we get more higher dimensional cells. Intuitively, we are trying to keep track of the stage at which a given homology cycle becomes a boundary in this filtration indexed by critical points.

We could generalize the notion of persistence homology to a more generalized class of functions called tame functions. We call a function f from a topological space X to $\mathbb{R}$ tame if the homology groups of every $X^a=f^{-1}((-\infty,a])$ have finite ranks and there are only finitely many values t across which the homology groups are not isomorphic.

2.3.2. *Relevance for Data Analysis.* Now we have a finite collection S of data in a metric space, and data clustering is an important problem. One thing we could do is to consider the topological space which is a finite union of open balls $X_\epsilon=\bigcup_{p\in S,\epsilon\in\mathbb{R}}B_\epsilon$. The topology of X_ϵ changes as ϵ varies. Therefore we could apply generalized Morse theory to the distance function on the space $X=\bigcup_{\epsilon\in\mathbb{R}}X_\epsilon$.

For computational purposes, we could define compatible Delaunay triangulations of all X_ϵ and get a filtration of simplicial complexes (just as in the case of Morse homology). Then we set up a big matrix which keeps track of the boundaries of faces at each stage of this filtration of simplicial complexes. Then computing homology or persistent homology would be about computing ranks and kernels of matrices.

Persistence homology has a good stability properties, meaning that the bottle neck distances between the persistence diagrams of two functions are bounded above by the differences between the two functions. This stability property enables us to apply persistence homology to some interesting

problems. For instance, because of stability, we can study persistence homology for time series data. When we have one continuous family of functions $f_t : M \to \mathbb{R}$, the data in persistence diagram also move in continuous curves.

More generally, we can replace the distance function by any probability density function $p(x)$, and study its persistence homology.

Now recall the mode clustering method for density clustering. In the language of persistent homology, each mode has a lifetime. Persistent homology precisely captures the creation and annihilation of clusters in the density tree construction.

2.4. **Cech Cohomology and Mapper.** First, consider a topological space M with an open cover $\mathcal{U} = \cup_{i \in I} U_i$, where I is a finite indexing set. The nerve $N(\mathcal{U})$ is defined to be the simplicial complex whose vertex set is the indexing set I, and where a family $\{i_0, ..., i_k\}$ spans a $k-$simplex in $N(\mathcal{U})$ if and only if $U_{i_0} \cap ... \cap U_{i_k} \neq \emptyset$. The nerve gives a combinatorial and computational model of M. In particular, when the open cover $\mathcal{U}$ is a good cover, the space M and the nerve $N(\mathcal{U})$ have the same cohomology. This is the idea behind the definition of Cech complex and Cech cohomology in terms of intersections of open sets in an open cover. For any continuous map $f : X \to Z$, the preimage of an open cover of Z is an open cover of X.

The algorithm Mapper [**?**] adopts the topological ideas above to discrete data sets, and aims to reduce complex data sets to combinatorial objects while keeping some of the essential topological features. More specifically, let X be a point cloud data set with n data points. Let $f : X \to Z$ be a function from X to a parameter space Z. Some common examples of Z include $\mathbb{R}, S^1, \mathbb{R}^3$ and other higher dimensional spaces. This function is called a filter. We assume that it is possible to compute inter-point distances between the points in X.

To implement Mapper, the first step is to choose an open cover $\mathcal{V}$ of the parameter space Z. When $Z = \mathbb{R}$, the range of the filter f is usually divided into a set of smaller intervals which overlap. This in turn gives a natural partition of X into intersecting subsets of data points. To quantify the topological notion of connected components in subsets of data points, clustering algorithms that take the inter-point distance matrix as an input are used.

One clustering algorithm implemented by the authors works as follows. First, we run the single-linkage clustering algorithm on the n data points. At the beginning, each individual data point is considered as a cluster. At each stage of the algorithm, two clusters that are closest are merged. At the end, it returns a vector $C \in \mathbb{R}^{n-1}$ whose coordinates are the distances between the two clusters that merge at each stage of the algorithm. Then the number of clusters in each partition of X is determined based on the idea that intra-cluster distances should be much smaller than inter-cluster distances. If we plot the coordinates of C in a histogram, we expect to see jumps. More explicitly, if we divide the histogram into k intervals for some positive integer k, we expect to see a set of interval(s) corresponding to the small coordinates of C, a set of empty interval(s), followed by a set of interval(s) corresponding to the large coordinates of C. By allowing all edges of length shorter than the length at which we observe the empty interval in the histogram, we recover a clustering of the data. Increasing k will increase the number of clusters we observe, and decreasing k will reduce it. While this method worked well for many data sets, it still has various limitations. This part of the procedure is open to exploration and change in the future.

One interesting feature of Mapper is that by varying the coarseness of the open cover of Z, the coarseness of the corresponding open cover of X changes and we get natural maps between the resulting simplicial complexes. This is called multi-resolution structure, and help us removing noises by singling out topological features that do not persist as the covers change.

In the topological story, when the open cover $\mathcal{U}$ of the space M is not a good cover, its nerve may not recover the Betti numbers of M. Therefore, sometimes Mapper is implemented for parameter

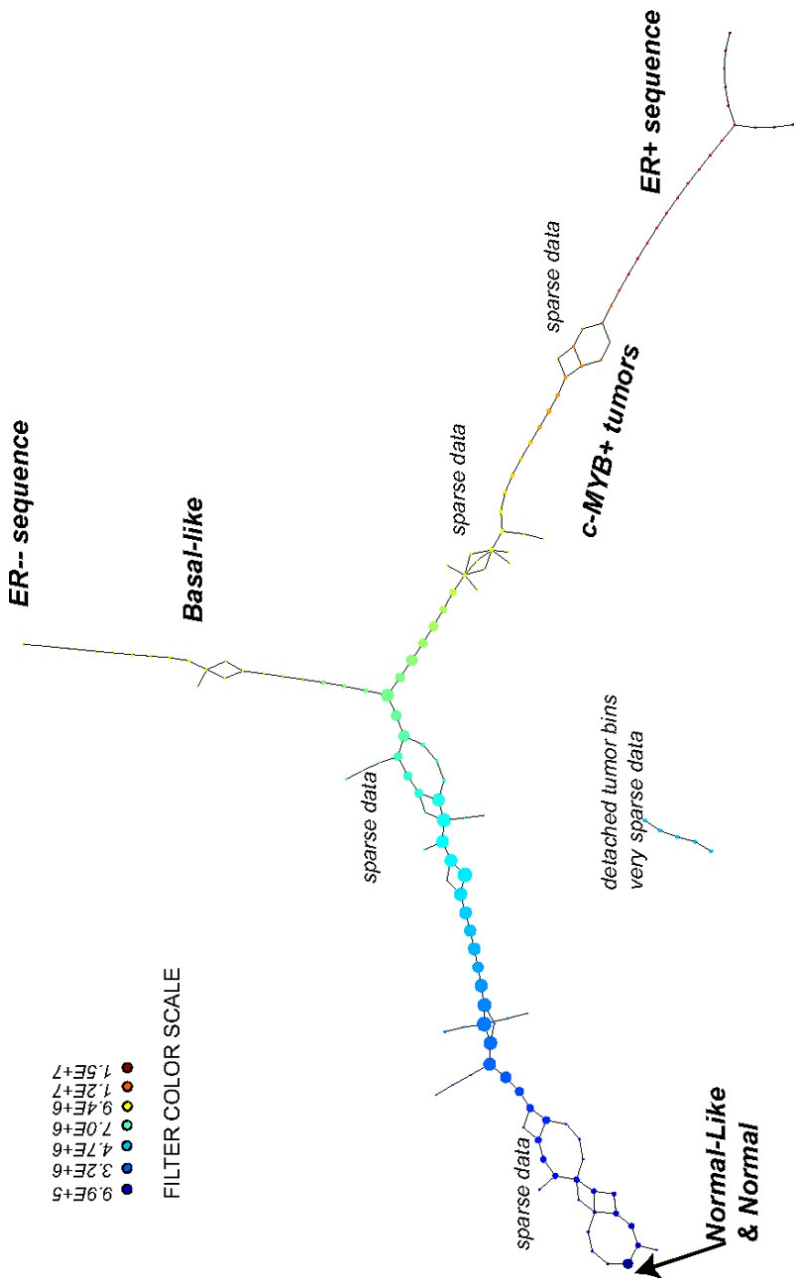
spaces with higher dimensions and richer topology, so as to preserve more features in the original data sets.

2.4.1. *Examples.* Mapper has been proven useful in many examples. Below we outline two of them, one is geometric, and the other is an application to biology/medicine.

In section 5.2 of [**?**], the authors explains the application of Mapper to a sample of 1500 points from a two dimensional torus embedded in $\mathbb{R}^3$. This data set was embedded into $\mathbb{R}^30$ with a random rotation. They then built a graph whose vertices are the data points. Each pair of distinct vertices is connected by an edge and labelled by a weight which only depends on the distances between the data points. The graph Laplacian is the 1500×1500 matrix whose entries are given by the (normalized) weight. For the filters, they used the first two non-trivial eigenfunctions of the graph Laplacian, which we touched upon in section 1. Each entry of an eigenfunction is the image of a data point. These eigenfunctions carry interesting geometric information of the data set. At the end, with appropriate choices, the authors obtained a four-dimensional simplicial complex whose Betti numbers agree with that of the torus.

In the PNAS paper [4], a new group of breast cancer was discovered using Mapper. The initial data set came in the form of a real matrix M, where the columns are the genomic data of individual patients, and the rows are different genomic variable types. Then the authors produced a new matrix M_d from M whose columns represent differences between breast cancer data and normal data for individual patients. The columns of M_d is the data set X used for Mapper analysis.

The filter functions are the $k \in \mathbb{Z}^+$ powers of L^p norms of the vectors in X. These filters measure deviations from the normal tissues by real numbers. For clustering individual components, a notion of distance between individual data vectors is needed. For that the authors treated the genes of individual patients as random variables and the coordinates for each gene vector in X as data samples. Then given two vectors(finite collections of real numbers representing different aspects of genes) in X, the authors defined their distance to be their statistical correlation. At the end, the new group of breast cancer, c-MYB+, is shown as a distinct segment below.



References

[1] Ery Arias-Castro, Gilad Lerman, and Teng Zhang. Spectral clustering based on local pca. *The Journal of Machine Learning Research*, 18(1):253–309, 2017.

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[3] Herbert Edelsbrunner and John Harer. Persistent homology-a survey. *Contemporary mathematics*, 453:257–282, 2008.

[4] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 454–463. IEEE, 2000.
[5] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
[6] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.
[7] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.
[8] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.
[9] Xu Wang, Konstantinos Slavakis, and Gilad Lerman. Multi-manifold modeling in non-euclidean spaces. In *Artificial Intelligence and Statistics*, pages 1023–1032, 2015.
[10] Larry Wasserman. Topological data analysis. *Annual Review of Statistics and Its Application*, 5:501–532, 2018.

Authors:
Casey Jao, Department of Mathematics, UC Berkeley, cjao@math.berkeley.edu
Qiao Zhou, Perimeter Institute for Theoretical Physics, qzhou@perimeterinstitute.ca