SPECTRAL CLUSTERING FROM A GEOMETRICAL VIEWPOINT

TYRUS BERRY * AND TIMOTHY SAUER †

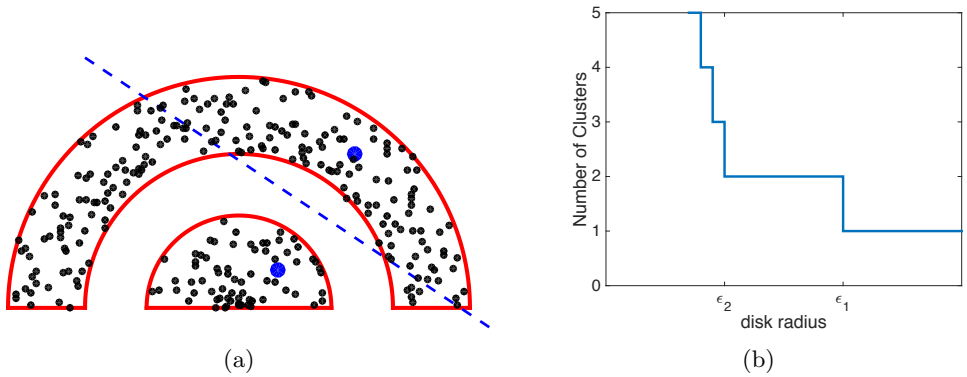
Abstract. Spectral methods have received attention as powerful theoretical and practical approaches to a number of machine learning problems. The methods are based on the solution of the eigenproblem of a similarity matrix formed from distance kernels. In this article we discuss three problems that are endemic in current implementations of spectral clustering: (1) the need to use another clustering method such as k-means as a final step, (2) the determination of the number of clusters, and (3) the failure of spectral clustering on multi-scale examples. These three problems are manifest even when the clusters are separated connected components. We advocate the use of the LU-factorization to solve (1), and treat clustering as a geometry problem to attack the second two problems. Specifically, the ideas of persistence and reconstruction of the Laplace-Beltrami operator are introduced as solutions to (2) and (3). We show that these suggested solutions are robust in a series of illustrative examples.

1. Introduction. Division of a set of points into clusters is a fundamental machine learning problem. Clustering underlies segmentation problems in network theory, image analysis, graph theory, and many other areas. In recent years, the clustering problem has attracted the attention of many researchers using spectral methods [25, 30, 14, 15, 10, 11, 29, 28, 13, 12]. These methods apply a kernel function $W_{ij} = W(x_i, x_j)$ to all pairs of data points, forming a square "affinity" matrix. In the typical *spectral clustering* approach, the data is projected onto an eigenspace of the kernel matrix, and a more conventional clustering algorithm is applied to the data in the new coordinates. The reasoning behind spectral methods is that they are matrix versions of the maximization of graph cuts, which compare pairwise distances within and outside the assigned cluster. Comprehensive introductions to spectral clustering can be found in the tutorials of Chung [7] and Von Luxborg [27].

In this article, we argue that a more geometric treatment of spectral clustering can better illuminate the underlying workings of the method, which in turn motivates a more powerful algorithm for multiscale problems. Our contribution has three parts: (1) a new approach to the last step of spectral clustering, the unmixing of the eigenspace basis that results in indicator functions for cluster assignment; (2) the systematic use of persistence as a way to simultaneously choose an appropriate global scaling and the correct number of clusters; and (3) the use of a geometry-motivated local scaling to solve the problem of varying sampling densities between and within clusters.

The idea of the geometric view of spectral clustering is to assume that the data points are sampled from a manifold with multiple connected components. Finding these connected components and assigning membership is the

*Dept. of Mathematics, Pennsylvania State University, University Park, PA 16802
†Dept. of Mathematical Sciences, George Mason University, Fairfax, VA 22030



(a) (b)

FIG. 2.1. (a) The k-means method identifies clusters as the set of points nearest to designated centroids; because of the non-convexity, a set like the one shown cannot be correctly clustered into two sets. (b) Spectral methods can easily divide the two clusters, if the critical parameter ϵ is chosen appropriately. The persistence diagram shows the dependence of the number of calculated clusters on this parameter.

first step toward any clustering objective. Finer distinctions that are based on probability distribution or other notions may further divide a connected component, depending on the goals of the analysis, although we view these as a separate problem that will not be pursued in this article.

By introducing this geometric assumption, it is possible to develop a spectral clustering algorithm (including unmixing and persistence) which provably yields the correct clustering in the limit of large data. Moreover, the geometric assumption will show that the local scaling algorithm introduced in [2] is an appropriate method to reduce the dependence of the clustering results on the sampling measure.

Our results in this article were motivated by previous work. Attempts to find a more convenient final step than k-means were discussed by Zelnick-Manor and Perona in [31], as were methods of handling multiscale point sets. Coifman, Nadler, Keverekidis et al. [18, 9, 16, 17] introduced diffusion maps as a preferred means of reconstructing the Laplace-Beltrami operator for clustering problems. Persistence-based clustering was discussed from a non-spectral point of view in [6]. Here we take a geometric approach that will extend the suggestions of these authors and put them in a natural framework.

In Section 2, contributions (1) and (2) are developed in the general spectral clustering context. These contributions will be interpreted in the geometric context in Section 3, where we introduce contribution (3). There we further develop the geometry-motivated solution to varying sampling densities and demonstrate our improved spectral clustering approach on multiscale examples.

2. Improved spectral clustering. Spectral clustering is motivated by the failure of k-means clustering on examples such as that shown in Figure 2.1(a). The k-means approach attempts to choose k cluster centroids in data

space such that points are sorted by determining the nearest centroid. However, any choice of two centroids in Figure 2.1(a) implicitly defines a line of equidistance that must separate the clusters. It is not possible to separate the points into the obvious two connected components, due to the non-convex shape of one of the clusters.

In Section 2.1 we briefly review some key results of spectral clustering from the graph theory perspective, and we show how to interpret global scaling in terms of persistence. In Section 2.2, we introduce a novel algorithm for the final 'unmixing' step motivated by the normalization of the eigenfunctions produced by the eigensolver in the spectral clustering approach. Finally, in Section 2.3, we demonstrate the persistence approach and our new unmixing on some illustrative example data sets.

2.1. Spectral methods and persistence. Spectral methods arose as matrix translations of methods for maximizing graph cuts (see [24, 19, 27] and references therein). The methods have in common the construction of a *graph Laplacian* matrix, of which there are several versions. Given a set of n points, let W be an $n \times n$ symmetric weight matrix that describes the affinity between pairs of points. For example, one could set $W_{ij} = 1$ if $||x_i - x_j|| < \epsilon$ and 0 otherwise, for some fixed $\epsilon > 0$. Let D be the diagonal matrix of row sums of W, that is $D = W\mathbf{1}$. Some examples of graph Laplacians are the unnormalized, the symmetric, the random walk, and the diffusion maps Laplacian [8], denoted by:

$$\begin{aligned} L_{\text{un}} &= D - W \\ L_{\text{sym}} &= I - D^{-1/2} W D^{-1/2} \\ L_{\text{rw}} &= I - D^{-1} W \\ L_{\text{dm}} &= I - \hat{D}^{-1} \hat{W} \end{aligned}$$

respectively, where $\hat{W} = D^{-1} W D^{-1}$ and $\hat{D} = \hat{W}\mathbf{1}$.

For the symmetric matrix W, let the *W-connected components* denote the equivalence classes of points x_i where $x_i \sim x_j$ if $W_{ij}^p \neq 0$ for some integer $p > 0$. The following theorem is central to spectral clustering, and applies to each of the graph Laplacians L above. A proof can be found in the tutorial [27].

THEOREM 2.1. *For any of the four graph Laplacians L defined above, the multiplicity k of the eigenvalue 0 of L is equal to the number of W-connected components.*

Moreover, the associated eigenvectors determine the assignment of points into k clusters, as follows: For $L_{\text{un}}, L_{\text{rw}}$ and L_{dm}, the eigenspace associated to eigenvalue 0 is spanned by the indicator functions $\mathbf{1}_{S_i}$ of the W-connected components. For L_{sym}, the eigenspace is spanned by the vectors $D^{1/2}\mathbf{1}_{S_i}$.

Notice that the non-symmetric Laplacians L_{rw} and L_{dm} are conjugate to symmetric matrices, for example $D^{1/2} L_{\text{rw}} D^{-1/2} = L_{\text{sym}}$. In order to find the

eigendecomposition of a non-symmetric Laplacian, it is numerically preferable to find the eigenvectors v of the symmetric matrix L_{sym}; then the eigenvectors of L_{rw} are $D^{-1/2}v$. Likewise, the eigenvectors of the nonsymmetric matrix L_{dm} are $\hat{D}^{-1/2}v$, where v represents an eigenvector of the symmetric matrix $I - \hat{D}^{-1/2}\hat{W}\hat{D}^{-1/2}$.

Examples like Fig. 2.1(a), which cause the k-means algorithm to fail, are easily divided into connected components with any of the above graph Laplacians, as long as an appropriate ϵ is chosen. Here we want to point out that ϵ serves as a global scale parameter. The user will choose it depending on the context of the data, meaning that it depends on information beyond the data points themselves. Assume that we are using the weight matrix as defined above, so that W_{ij} is given by the kernel

$$W_{ij} = h\left(\frac{||x_i - x_j||^2}{\epsilon^2}\right), \quad \text{where} \quad h(x) = \begin{cases} 1 & \text{if } x < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

For the set of points in Fig. 2.1(a), it is clear that if ϵ is chosen to be larger than one-half of the distance ϵ_1 between the two clusters, then according to Theorem 2.1, the spectral method with any of the four graph Laplacians will group the points into a single component. If ϵ is smaller than ϵ_1, and larger than ϵ_2, defined to be one-half of the the maximum radius of a circle centered at a data point containing no other points, then the graph Laplacians will have two-dimensional zero-eigenspaces, verifying that there are two components. This may be the intuitive choice – but it is a choice. As ϵ decreases beyond ϵ_2, the two large clusters will gradually break up into subclusters, and for sufficiently small ϵ there will be n single-point clusters. These facts are shown schematically in Fig. 2.1(b).

Fig. 2.1(b) is an example of a *persistence diagram* [5, 32, 4], which clarifies the point that the number of clusters is dependent on a parameter of the spectral method. This choice is inescapable, and in multiscale problems, the *global scaling parameter* ϵ will typically be chosen based on the goals of the investigation. From another point of view, the persistence diagram is a polite way of dealing with ϵ, which would otherwise be called a nuisance parameter.

The "shape function" h is also open to the user's choice. An alternative to the sharp cutoff used in (2.1) is an exponential decay function, such as

$$h(x) = e^{-x/4}. \tag{2.2}$$

The ϵ parameter plays a similar role for this shape function, in that smaller ϵ localizes the affinity of nearby points. A smooth shape function is more appropriate for data that is measured with uncertainty. Many other shapes could be used, and in Section 3 below we will revisit this question and introduce an even more general class of "local" kernels.

It is worth pointing out that in numerical calculation, the similarity of the sharp cutoff and exponential decay functions is more pronounced than

may be obvious at first glance. With the exponential decay shape function, all weights are nonzero, implying that there is only one cluster, according to Thm. 2.1. However, in finite precision computation, when the ratios of weights are greater than the reciprocal of machine epsilon $1/\epsilon_{\text{mach}}$, the computation of eigenvalues will effectively treat extremely small weights as zero. Although there is one component in theory, exponential decay shape functions can give numerical results that resemble those from finite support shape functions, and they are routinely used in practice.

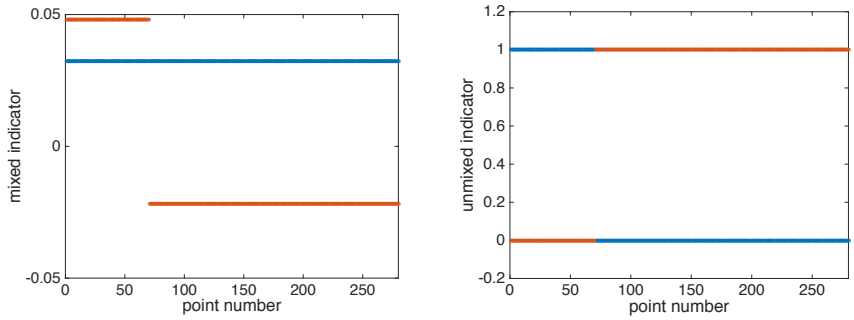
Thm. 2.1 identifies the vector space of indicator functions of the connected components as a single eigenspace. A numerical eigensolver can be employed to construct a basis for this space. However, there is still some work to do. Each indicator function is a linear combination of the basis. The step that remains is to "unmix" this set of eigenvectors to extract the indicator functions of the individual components. The standard approach is to project each point onto the eigenbasis returned by the eigensolver, and to apply the k-means algorithm on the results to affix a component label to each point. Instead of outsourcing this final step to another, non-spectral method, we develop and illustrate a simpler endgame in the next section.

2.2. Unmixing. The goal of this section is to recover the cluster labels for each data point from the eigenvectors of a graph Laplacian returned by an eigensolver. Specifically, we will be given eigenvectors $\{\varphi_j\}_{j=1}^c$ corresponding to eigenvalue 0 from which we want to extract indicator functions $\{\mathbf{1}_{S_l}\}_{l=1}^c$ of the connected components S_i. We refer to this procedure as *unmixing* the eigenvectors.

Suppose we consider a graph Laplacian whose zero-eigenspace is spanned by indicators functions of the components, such as $L_{\text{un}}, L_{\text{rw}}$, or L_{dm} discussed above. The dimension of the zero-eigenspace is exactly the number of W-connected components and the cluster indicator functions $\{\mathbf{1}_{S_l}\}_{l=1}^c$ are a basis for the zero-eigenspace, and therefore there is a linear change of variables A such that each eigenvector φ_j can be written as $\varphi_j(x_i) = \sum A_{lj}\mathbf{1}_{S_l}(x_i)$. We will refer to the matrix A as the *mixing* matrix. The eigenvectors φ_j are considered *mixed* and the eigenvectors $\mathbf{1}_{S_l}$ are *unmixed*.

As mentioned above, for computational stability reasons, we will always obtain the spanning set of eigenvectors from a symmetric Laplacian matrix. If a non-symmetric Laplacian is desired, the eigenvectors can be obtained from those of the conjugate symmetric Laplacian by multiplying by a symmetric matrix. As a result, the spanning set will be given by the columns of an $N \times c$ matrix $\Phi = B^{-1}\tilde{\Phi}$, where $\tilde{\Phi}$ has orthonormal columns from the (symmetric) eigensolver and B is an invertible diagonal matrix. (Namely, $B = I$ for L_{un}, $B = D^{1/2}$ for L_{rw}, and $B = \hat{D}^{1/2}$ for L_{dm}.) Let C be the $n \times c$ matrix whose column l is the indicator function $\mathbf{1}_{S_l}$. With this notation the mixing is represented by $\Phi = CA$, and the goal of unmixing is to recover C from Φ.

In fact, one can see that the mixing matrix A is the product of a diagonal



(a) (b)

FIG. 2.2. Unmixing for the nonconvex example in Fig. 2.1. (a) Two eigenvectors returned by a standard eigensolver (b) Unmixed indicator functions from the LU method.

matrix and an orthogonal matrix. Define the $c \times c$ matrix N by

$$N = (BC)^T BC = A^{-T}\tilde{\Phi}^T\tilde{\Phi}A^{-1} = A^{-T}A^{-1} \tag{2.3}$$

and note that $N = C^T B^2 C$ is diagonal since $N_{ij} = \sum_k C_{ki} B_{kk}^2 C_{kj}$ is only nonzero when $i = j$. For example, in the case of L_{un} we have $B = I$, which implies that $N_{ii} = \sum_k C_{ki}^2$ is simply the number n_i of points in the i-th component. Then (2.3) implies that $Q \equiv N^{1/2}A$ is an orthogonal $c \times c$ matrix, and so the mixing matrix $A = N^{-1/2}Q$.

In Fig. 2.2 we show the (a) mixed and (b) unmixed eigenfunctions for the nonconvex example introduced in Fig. 2.1(a) where we have chosen $\epsilon_2 < \epsilon < \epsilon_1$ from the persistence diagram in Fig. 2.1(b). Each eigenvector is an n-vector, where $n = 280$ is the number of data points. Two eigenvectors are shown, whose entries are plotted versus point number. Here we have presented the points in sorted order from left to right, starting with the smaller component, to clarify the resulting plot.

Since the number of points in each cluster cannot be determined prior to the unmixing, we cannot easily remove the bias of the unknown diagonal matrix N, and if the numbers of points in the various components are nonuniform, it will not suffice to unmix the eigenvectors only with an orthogonal matrix. We will introduce a simple method for determining A directly from the eigenvectors $\{\varphi_j\}_{j=1}^c$ that will not be biased by the number of points in the various clusters.

Since the mixed eigenvectors φ_j are linear combinations of the indicator functions, all pairs of points x, y in a given component S_l have the same values: $(\varphi_1(x), ..., \varphi_c(x)) = (\varphi_1(y), ..., \varphi_c(y))$. So for each component S_l there is a unique *barcode*, which is a row vector $(\varphi_1(x), ..., \varphi_c(x))$, where x is any point in S_l. These barcodes are shown visually in Figure 2.2(a) where for clarity we have artificially sorted the data so that all the points in a given cluster are grouped together, making the barcode structure stand out clearly. Of course, if the data points were randomly organized, the barcode structure would not

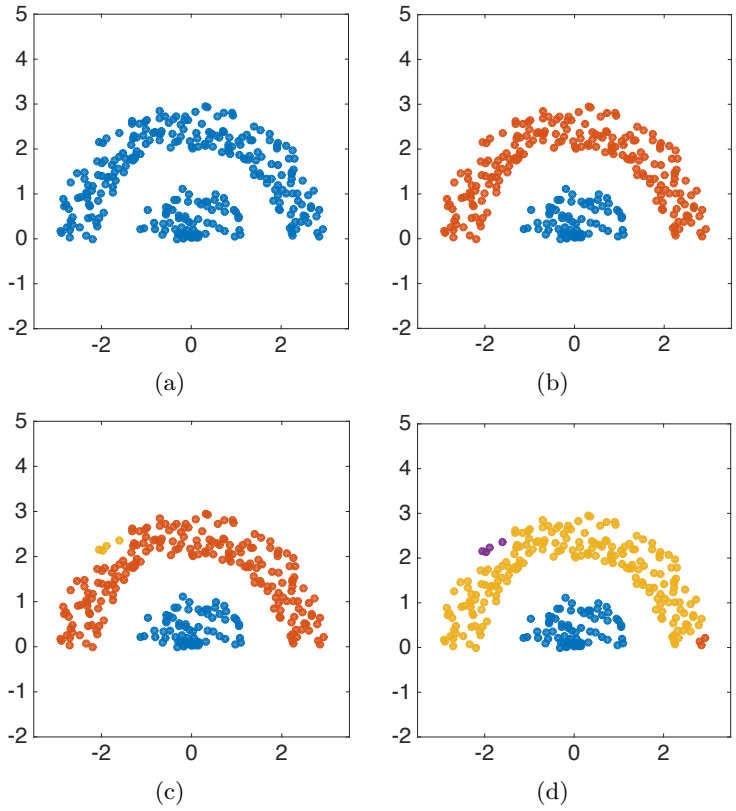


FIG. 2.3. Clustering of the nonconvex example in Fig. 2.1 using LU unmixing. Results are shown for global parameter ϵ resulting in (a) 1 component (b) 2 components (c) 3 components (d) 4 components.

be visually obvious. In fact, the barcodes are simply the rows of the mixing matrix A, since for $x \in S_l$, all the indicator functions are zero except for 1_{S_l} so $\varphi_j(x) = A_{jl}$ and $(\varphi_1(x), ..., \varphi_c(x)) = (A_{l1}, ..., A_{lc})$. In fact, since the cluster indicator functions C which we wish to recover have such a simple form, the row of the mixing matrix A are simply rows of Φ. Of course, each row of A will appear many times in Φ, once for each point in the corresponding cluster. Moreover, the barcodes could appear in any order, corresponding to the order of the points, so we cannot simply select the first c row of Φ. So in order to build the matrix A, we only need to select c linearly independent rows from Φ.

There are many methods which can select c linearly independent rows from the rows of Φ. We have used a simple technique based on the partial pivoting approach used in the LU-factorization algorithm [22]. First, apply the $P\Phi = LU$ matrix factorization to the $n \times c$ matrix Φ; here, P is an $n \times n$ permutation matrix, L is an $n \times c$ lower triangular matrix, and U is $c \times c$ upper triangular. Then define the matrix $T = P\Phi$, so that the square matrix formed by the top c rows of T is the mixing matrix A (up to a permutation of the

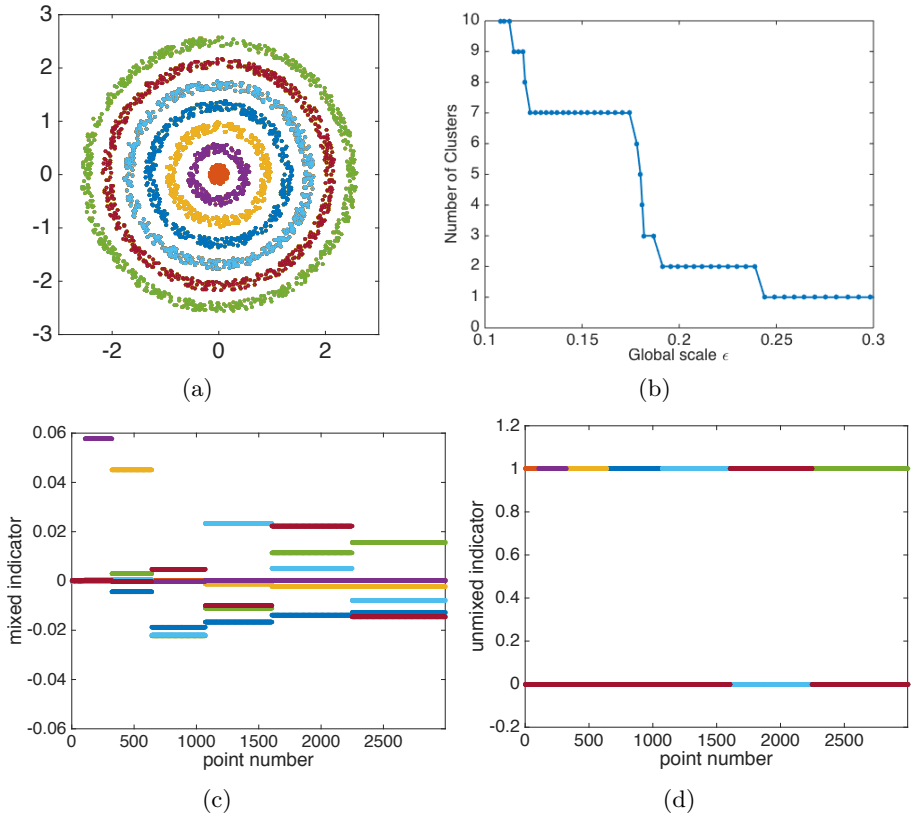


FIG. 2.4. (a) Target example. (b) Persistence diagram in the global scaling parameter ϵ. (c) The 7 mixed eigenvectors associated to eigenvalue 0, each plotted in a different color as a function of data points. (d) Unmixed indicator functions for the 7 components.

rows of A). This is because the permutation P selects c linearly independent rows of Φ, which are exactly the rows of A. We can then recover the unmixed eigenvectors as the columns of $C = \Phi A^{-1}$.

Figures 2.2(b) shows the unmixed eigenvectors C recovered from the mixed eigenvectors Φ shown in Figures 2.2(a). Of course, the key to this unmixing approach is the barcode structure of the mixed eigenvectors Φ, which requires that the columns of Φ are eigenvectors with eigenvalue zero.

In Fig. 2.3 we show the results of our LU unmixing algorithm on the example data set from Fig. 2.1. The clustering result in Fig. 2.3(b) corresponds to the unmixed indicator functions shown in Fig. 2.2(b), where each indicator function is plotted in a separate color. In Fig. 2.3 we also show the clusterings found by the LU unmixing with three other values of ϵ, chosen from the persistence diagram to correspond to (a) one, (c) three and (d) four clusters. For each value of ϵ, we find a basis for the zero-eigenspace, using the tolerance $N\epsilon_{\text{mach}}$ to verify that an eigenvalue is numerically zero.

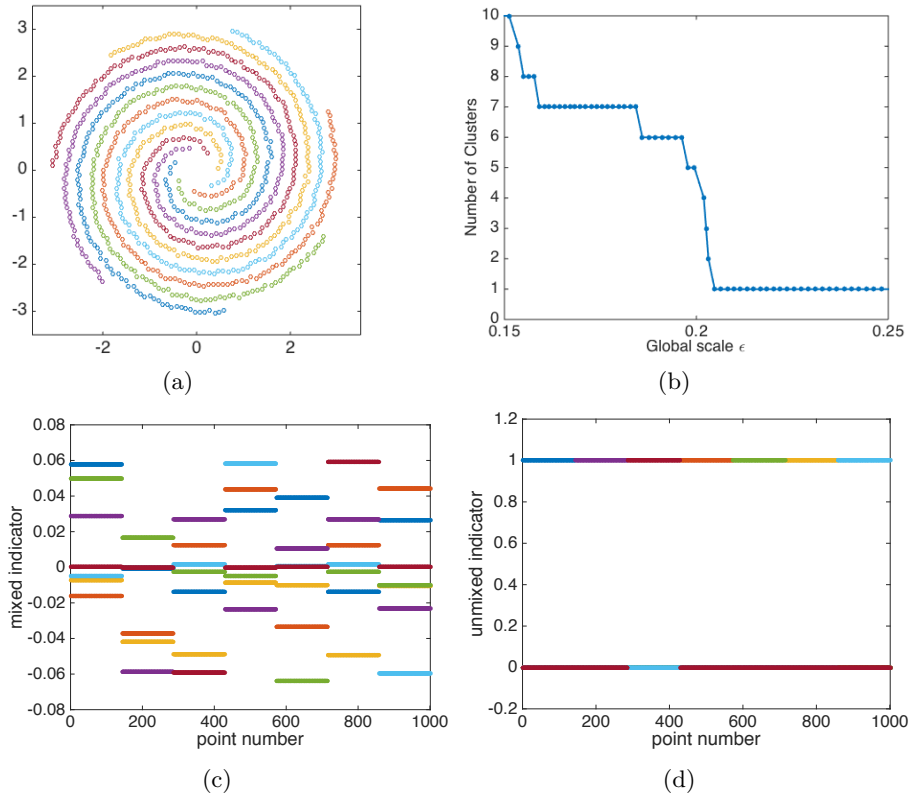


FIG. 2.5. (a) Example of seven interleaved spirals, consisting of 1000 points. (b) The persistence diagram has a range of ϵ where there are 7 components. (c) Mixed eigenvectors, and (d) unmixed eigenvectors / indicator functions.

2.3. Examples. In this section we demonstrate the complete spectral clustering algorithm on three challenging toy examples. These examples reveal the complex structure that can arise in the persistence diagrams, as well as the significantly complicated mixing that can occur. Both examples exhibit significant persistence at a particular 'correct' number of clusters, and the LU unmixing is successful at labeling these clusters.

The first example is a data set with a target like structure where the clusters are nested annuli in the plane as shown in Fig. 2.4(a). This is an extended version of a classical clustering example which typically contains fewer nested annuli. The additional annuli exacerbate the differences in the number of points between the clusters, which can be a problem for some unmixing algorithms. There are $N = 3000$ points shown, chosen randomly to fill the seven connected components. The numbers of points in each component satisfy the ratios $1 : 2 : 3 : 4 : 5 : 6 : 7$ from inside to outside, so that the density of points is approximately constant. (Later, in Fig. 3.4, we will explore this example with varying density.) The exponential shape function (2.2) was used to build the weight matrix. The persistence diagram in Fig. 2.4(b) shows a substantial

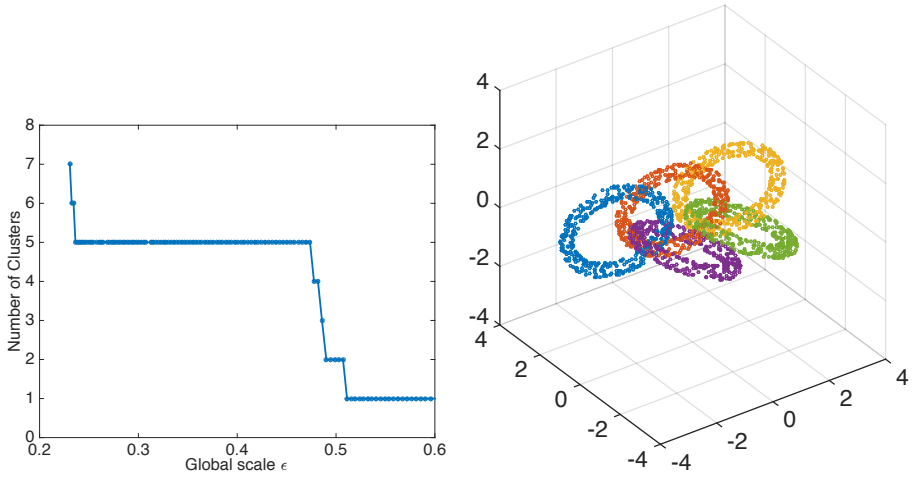


FIG. 2.6. Example with five interlocking tori. (a) The persistence diagram shows a large range of ϵ for which there are 5 effectively zero eigenvalues. (b) The assignment of clusters due to the LU unmixing.

set of ϵ for which the zero-eigenspace is 7-dimensional. The mixed eigenfunctions, which form a basis of the 7-dimensional zero-eigenspace, are plotted in Fig. 2.4(c), where the points have been sorted from inside to outside along the horizontal axis for clarity. The unmixed linear combinations are plotted in Fig. 2.4(d), which are used to color part (a) of the figure.

The second example is a data set made up of seven interleaved spirals. In this example, the seven components contain approximately equal numbers of points, shown in Fig. 2.5(a). The persistence diagram in Fig. 2.5(b) has a significant range of the global scale parameter ϵ for which there are 7 components. Fig. 2.5(c) and (d) display the mixed and unmixed eigenvectors, respectively, where the LU-method introduced above is used for unmixing. The results are obtained in the same manner as in Fig. 2.4.

Finally, Fig. 2.6 shows an example of two-dimensional manifolds in three dimensional ambient space, consisting of points sampled uniformly from five interlocking tori. The persistence diagram locates 5 clusters, whose resulting assignment is shown in Fig. 2.6(b).

3. Non-constant Density and the Large Data Limit. The example in Fig. 3.1 shows a more complicated situation, which uncovers a weakness of the spectral methods discussed thus far. Nominally, there are three components. Two components are densely sampled and a third is more sparsely sampled. Consider the radius ϵ indicated by the circles around the data points, and for simplicity assume that the sharp cutoff function (2.1) is used. At this radius, many of the points in the sparse cluster are not connected to any other points in that component. This ϵ is too small for the sparse component to

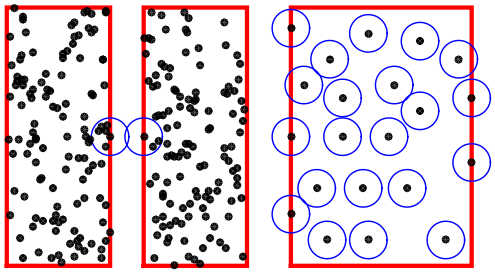


FIG. 3.1. An example with varying densities. Any spectral method that relies on a single global bandwidth (denoted by the circles) cannot properly divide the example into three clusters.

be seen as one unit, but at the same time is too large to distinguish the two densely sampled clusters. With the available data, concluding that there are three components from the zero-eigenspace of the graph Laplacian is impossible; moreover, concluding that there are two components is also impossible. This leads to a serious failure: Neither two nor three will be found in the persistence diagram.

The flaw in the framework of standard spectral clustering methods that is revealed by examples like Fig. 3.1(a) is the following: Definitions of clustering built on a fixed global scale parameter ϵ lack the flexibility to deal with point sets of varying density. In this article we argue that the pathway out of this paradox, as in many mathematical areas, is to look to the continuous case for guidance.

Imagine that the data in Fig. 3.1 were generated from an independently and identically distributed sequence. Now suppose more data could be added to the three clusters, according to the same underlying distribution. Given enough data, the number of points in the sparsely sampled cluster would become large enough that they could be connected with a smaller ϵ. This ϵ could further be made small enough to simultaneously separate the two densely sampled clusters.

In other words, the persistence diagram *in the large data limit* would successfully find ranges of ϵ for both two and three clusters. In addition, one cluster, and N clusters, would be found, for sufficiently large and small ϵ, respectively. However, far from the large data limit, we may not have the luxury of populating the data set sufficiently to identify the correct number, or any appropriate number, of components with a single ϵ.

In order to handle multiscale problems like this one with a fixed, finite data set, it would be helpful to replace the global scale parameter ϵ^2 in the kernel function (2.1) with a quantity $\epsilon(x)\epsilon(y)$ that varies with the points x and y. In this way, local variations in density could be accounted for. For example, kernels were introduced by Zelnick-Manor and Perona [31] that rely on local scaling. Such kernels are not covered by the diffusion maps theory in [8], which requires a constant global ϵ. That presents us with the question of

how to justify such kernels mathematically.

The goal of this section is to provide a mathematical justification for using local scaling to accomplish spectral clustering in a way that is invariant to the sampling density. To do this, the focus must be shifted from the point set to the geometry underlying the point set. Formally, we will make the assumption that the point sets are finite realizations of probability distributions lying on manifolds. We view this assumption as establishing a "geometric prior" for the problem. By appealing to geometry, we will see that spectral methods can be realized as methods for representing function spaces on a manifold. In this interpretation, density variations and other epiphenomena of the point set can be dealt with more readily. Along the way, we will revisit the original role of the Laplacian in spectral clustering and generalize it. In particular, we will find that replacing ϵ with a carefully chosen nonconstant $\epsilon(x)$ can in many cases reconstruct the correct geometry without the necessity of approaching the large data limit.

The main goal in Section 3 is to approximate the Laplace-Beltrami operator with as little data as possible. Most of the groundwork already exists, and is the natural extension of ideas developed by Belkin and Niyogi [1] and Coifman and collaborators [8, 18, 9, 17, 16]. In Section 3.1 we briefly survey the theory of local kernels [3] for describing the geometry of data, and in Section 3.2 we present the topological grounds for using the Laplace-Beltrami operator. Section 3.3 shows examples of the use of the local kernel idea for clustering.

3.1. Geometry of Data. The geometric prior assumes that the subset of positive sampling density is a smooth manifold $\mathcal{M} \subset \mathbb{R}^n$. The assumption of this smooth structure gives us a natural volume form $d\text{vol}$ that $\mathcal{M}$ inherits from the ambient space, and we will consider the sampling density q, to be taken relative to this volume form (rather than relative to the standard measure on $\mathbb{R}^n$). If the sampling measure is uniform relative to the volume form (meaning $q \equiv 1/\text{vol}(\mathcal{M})$), it was shown in [1] that the symmetric normalized Laplacian matrix is a discrete approximation to the Laplace-Beltrami operator on the manifold $\mathcal{M}$. This was the first re-interpretation of the central spectral clustering construction in terms of differential geometry. The very restrictive assumption of uniform sampling was overcome with the introduction of diffusion maps [8], by deriving the bias introduced by the sampling density, and using a kernel density estimate of the sampling density along with a new normalization technique to control and even remove the bias.

The diffusion maps algorithm depends on a global kernel of form $W_{ij} = h\left(||x_i - x_j||^2/\epsilon^2\right)$ where $h(x)$ is a radial basis function with exponential decay, such as $h(x) = e^{x/4}$. The row sums $D_{ii} = \sum_{j=1}^{N} W_{ij}$ can be viewed as a kernel density estimate of the sampling measure $q(x_i)$. In order to remove the sampling bias, the diffusion maps algorithm first forms the normalized kernel $\hat{W} = D^{-1}WD^{-1}$, and then forms the normalized graph Laplacian matrix

$I - \hat{D}^{-1}\hat{W}$ from the normalized matrix $\hat{W}$, with $\hat{D}_{ii} = \sum_{j=1}^{N} \hat{W}_{ij}$. In the limit of large data, the diffusion maps algorithm was shown in [8] to estimate the Laplace-Beltrami operator Δ on the manifold $\mathcal{M}$, independently of the sampling measure q. Notice that by using the non-symmetric graph Laplacian, we guarantee the vector of all ones is an eigenvector with eigenvalue zero. A noted earlier, the symmetric graph Laplacian matrix $I - \hat{D}^{-1/2}\hat{W}\hat{D}^{-1/2}$ can be used for computing the eigenvectors, which should then be multiplied by $\hat{D}^{-1/2}$ to find the eigenvectors of the non-symmetric graph Laplacian matrix. The theory of diffusion maps in [8] is applicable to the radial basis functions most commonly used in spectral clustering, and requires fixed ϵ.

In [3], a broad generalization of the diffusion maps theory was introduced that allows *local* kernels. A local kernel is any nonzero function $W_\epsilon : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ such that there exists constants $\epsilon, c, \sigma > 0$ and a vector field $b : \mathbb{R}^n \to \mathbb{R}^n$ independent of ϵ that satisfy

$$0 \leq W_\epsilon(x, x + \epsilon z) \leq ce^{-\sigma||z-\epsilon b(x)||^2}$$

for all $x, z \in \mathbb{R}^n$. In particular, taking $b = 0$ and $y = x + \epsilon z$, we see that any kernel that satisfies $W_\epsilon(x, y) \leq c \exp(-\sigma||x - y||^2/\epsilon^2)$ is a local kernel. The requirement that the kernel is bounded above by an exponentially decaying function is rather weak, so any compactly supported kernel is local, and any kernel which decays exponentially in any non-Euclidean norm is also local. The theory of local kernels developed in [3] showed that applying the diffusion maps algorithm to the symmetric kernel $\overline{W}_\epsilon(x, y) = W_\epsilon(x, y) + W_\epsilon(y, x)$ yields (in the limit of large data and $\epsilon \to 0$) a Laplacian operator with respect to an *intrinsic geometry* which depends on both the data set and the functional form of the local kernel W_ϵ. Since the choice of local kernel determines a geometry on the manifold, topological properties (such as the connected components) are independent of the choice of local kernel in the limit of large data.

Local kernels are designed to alleviate a weakness of the standard diffusion maps approach, that it uses a globally-scaled estimate of the sampling density q (contained in the diagonal matrix D). As a result, its estimate of the Laplace-Beltrami operator is invariant to the true sampling density q only in the limit of large data. Local kernels allow us to use more powerful, empirical estimates of q, which results in more accurate approximations of the Laplace-Beltrami operator in finite data circumstances.

A choice of local kernel that is particularly relevant to clustering was suggested in [2]. Let $q(x)$ represent the sampling density of the data on the manifold $\mathcal{M} \subset \mathbb{R}^n$. For a fixed global parameter ϵ, define $\epsilon(x) = \epsilon q(x)^\beta$ for some real number β. Then consider the symmetric variable bandwidth kernel

$$W_\epsilon(x, y) = h\left(\frac{||x - y||^2}{\epsilon(x)\epsilon(y)}\right) = h\left(\frac{||x - y||^2}{\epsilon^2 q(x)^\beta q(y)^\beta}\right) \tag{3.1}$$

for any shape function $h : [0, \infty) \to [0, \infty)$ that has exponential decay. To connect variable bandwidth kernels to the Laplace-Beltrami operator, define

the functions

$$F_i(x_j) = \frac{W_\epsilon(x_i, x_j) f(x_j)}{q_\epsilon(x_i)^\alpha q_\epsilon^S(x_j)^\alpha}, \qquad G_i(x_j) = \frac{W_\epsilon(x_i, x_j)}{q_\epsilon(x_i)^\alpha q_\epsilon^S(x_j)^\alpha},$$

where

$$q_\epsilon(x_i) = \sum_{l=1}^{n} W_\epsilon(x_i, x_l)/q(x_i)^{d\beta}, \tag{3.2}$$

and where d is the manifold dimension. These normalizations are necessary to control the bias that the sampling density has on the resulting operator, as is shown in the following result.

THEOREM 3.1. [2] *Let $\{x_i\}_{i=1}^N$ be sampled independently with distribution q. Let $W_\epsilon(x, y)$ be a variable bandwidth kernel with bandwidth function $\rho = q^\beta + \mathcal{O}(\epsilon^2)$. Then, with high probability,*

$$\begin{aligned} L_{\epsilon,\alpha,\beta} f(x_i) &\equiv \frac{1}{\epsilon^2 m \rho(x_i)^2} \left(\frac{\sum_j F_i(x_j)}{\sum_j G_i(x_j)} - f(x_i) \right) \\ &= \mathcal{L}_{\alpha,\beta} f(x_i) + \mathcal{O}\left(\epsilon^2, \frac{q(x_i)^{(1-d\beta)/2}}{\sqrt{N}\epsilon^{4+d/2}}, \frac{||\nabla f(x_i)||q(x_i)^{-c_2}}{\sqrt{N}\epsilon^{1+d/2}} \right) \end{aligned} \tag{3.3}$$

for a finite valued constant m, where

$$\mathcal{L}_{\alpha,\beta} f \equiv \Delta f + c_1 \nabla f \cdot \frac{\nabla q}{q}, \tag{3.4}$$

$c_1 = 2 - 2\alpha + d\beta + 2\beta$ and $c_2 = 1/2 - 2\alpha + 2d\alpha + d\beta/2 + \beta$.

The special case $\alpha = 1, \beta = 0$ of Theorem 3.1 corresponds to the diffusion map approximation of the Laplace-Beltrami operator in [8]. In particular, for each ϵ, the eigenvectors of the discrete linear operator $L_{\epsilon,1,0}$ form an approximate basis for functions on the manifold, where the approximation error is controlled in terms of ϵ and N.

Since the theorem gives a two-parameter family of choices, there are other ways to achieve the Laplace-Beltrami operator, including ways that allow use of a nonhomogeneous kernel with variable local scaling. We will focus on the choice $\alpha = 1/2 - d/4$ and $\beta = -1/2$. This choice yields $c_1 = 0$, so that according to (3.4), the operator $\mathcal{L}_{\alpha,\beta}$ is independent of the sampling density. Also, according to (3.3), $c_2 = -1 + 5d/4 - d^2/2 < 0$, implying that the error is bounded even for q arbitrarily close to zero. We note that the dimension d is the intrinsic dimension of the manifold $\mathcal{M}$ and can be determined automatically as part of the numerical algorithm (see Appendix A for details).

Theorem 3.1 allows us license to apply the whole range of kernel density estimation theory to approximate the sampling measure q. In the Appendix we show how to bootstrap an approximation to q from the data set. We generate

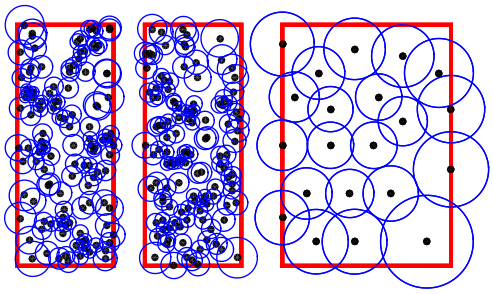


FIG. 3.2. The example with varying densities from Fig. 3.1. The variable bandwidth kernel with $\beta = -1/2$ and an appropriate ϵ from the persistence diagram divides the example into three clusters.

an initial approximation of local neighbor distance, and use it to build an exponential kernel to approximate q.

According to Theorem 3.1, the construction of the discretized approximation to the Laplace-Beltrami operator begins with the kernel matrix W in (3.1). Define the diagonal matrix D by $D_{ii} = (\sum_j W_{ij})/q(x_i)^{d\beta}$ to represent the $q_\epsilon(x_i)$ in (3.2), where d is the manifold dimension. Define $W_\alpha = D^{-\alpha} W D^{-\alpha}$, and define the diagonal matrix $D_\alpha = W_\alpha \mathbf{1}$. Then the discrete version of the Laplace-Beltrami operator is the $n \times n$ matrix

$$L = \epsilon^{-2} R^{-2} (D_\alpha^{-1} W_\alpha - I)$$

where R is a diagonal matrix with $R_{ii} = \rho(x_i) = q(x_i)^\beta$. (Here we have neglected the constant factor m in Theorem 3.1, since it does not affect the $\lambda = 0$ eigenspace.)

To find eigenvectors of L, we instead compute eigenvectors of a similar matrix that is symmetric. Define the diagonal matrix $B = \epsilon R D_\alpha^{1/2}$. Then $BL = B^{-1} W_\alpha - \epsilon^{-2} R^{-2} B = SB$ where $S = B^{-1} W_\alpha B^{-1} - \epsilon^{-2} R^{-2}$ is symmetric. If $\tilde{\varphi}$ is an eigenvector of S, then $\varphi = B^{-1}\tilde{\varphi}$ is an eigenvector of L with the same eigenvalue.

Now we can revisit the example of Fig. 3.1 with the kernel

$$W_\epsilon(x, y) = h\left(\frac{||x - y||^2}{\epsilon^2 (q(x) q(y))^{-1/2}}\right),$$

where q is the approximated sampling measure. If we can develop a reasonably accurate approximation to q, then due to Theorem 3.1, we know that this method reconstructs the Laplace-Beltrami operator on the underlying manifold and has significantly removed the bias of the sampling measure.

Fig. 3.2 shows the result of the locally-scaled kernel. The data set is the same as in Fig. 3.1. Each point x is the center of a circle of radius $\epsilon(x) = \epsilon q(x)^{-1/2}$, where ϵ is an appropriate choice from the persistence diagram that delivers the desired clustering into three components. With this local scaling,

we can use any convenient shape function h, such as the sharp cutoff h in (2.1), or the standard exponential $h(x) = e^{-x/4}$, to successfully separate the three connected components.

3.2. Topological clustering. In order to rigorously define the clustering problem, we assume that the data set lies on a compact Riemannian manifold $\mathcal{M} = \bigoplus_{l=1}^{c} \mathcal{M}_l$ which consists of c connected components $\{\mathcal{M}_l\}_{l=1}^{c}$. We define the *topological clusters* of the data set to be the connected components of the manifold. The goal of topological clustering is to develop an algorithm which provably identifies the topological clusters in the limit of large data. The connected components of a manifold are a topological, not a geometric property: changing the way we measure local distances between points does not change the connected components of the manifold.

The key to topological clustering is a result from Hodge theory, which connects topological features of a manifold to the geometric Laplacian operator Δ on the manifold. In particular, every connected component of a manifold corresponds to a unique *harmonic* function $\Delta f = 0$, meaning that f is an eigenfunction of the Laplacian with eigenvalue zero. Classical Hodge theory shows that for closed manifolds (compact without boundary) the only harmonic functions are constant functions [20]. This fact can be extended to compact manifolds with boundary by taking Neumann boundary conditions [23]. Of course, when there are multiple connected components a harmonic function can take a different constant value on each connected component. This shows that every harmonic function satisfying Neumann boundary conditions can be written as a linear combination of the indicator functions $\{1_{\mathcal{M}_l}\}$ of the connected components. The indicator functions are the natural harmonic representatives of the connected components, and they span the zero-eigenspace of the Laplacian with Neumann boundary conditions. Since the topological clusters are defined to be the connected components, the indicator function $1_{\mathcal{M}_l}$ is exactly the cluster function which identifies the cluster $\mathcal{M}_l$.

The topological clustering approach is to approximate a basis for the space of harmonic functions using a discrete approximation to the Laplacian operator. While there are many different geometries, which correspond to many different Laplacian operators on the manifold $\mathcal{M}$, all of these geometries have the same topology and the same topological clusters. As mentioned in Section 3.1, a large class of *local* kernels can be used to estimate the various Laplacian operators on the manifold. Since any of these Laplacian operators can be used to recover the topological clusters, any local kernel can be used for topological clustering in the limit of large data. The topological clustering approach is in fact a re-interpretation of spectral clustering for local kernels, which provably recovers the topological clusters in the limit of large data.

3.3. Examples with variable bandwidth kernel. We begin with the three box example of Fig. 3.1. Using the variable bandwidth kernel with $\alpha = 0, \beta = -1/2$ results in the persistence diagram shown in Fig. 3.3(a).

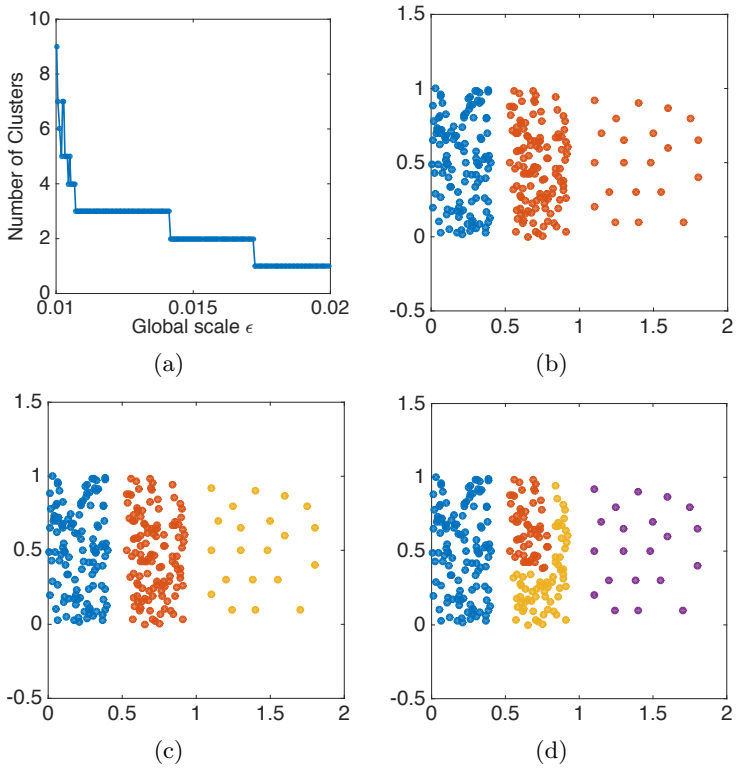


FIG. 3.3. (a) Persistence diagram for the three box example of Fig. 3.1 using the variable bandwidth kernel. For various global scale ϵ there are (b) two clusters (c) three clusters (d) four clusters.

There are substantial ranges of the global scale ϵ for which the approximate Laplace-Beltrami operator has two and three zero eigenvalues (within tolerance $N\epsilon_{\text{mach}}$), respectively. The ϵ used in Fig. 3.2, for example, was chosen from the latter range. The clusters derived from the LU method mentioned above are shown in Fig. 3.3(b-c). A short range of even smaller ϵ divides the set into four clusters, shown in Fig. 3.3(d).

Generically, as the global scaling is varied, every possible number of clusters is attained. In order to determine both the proper scaling and the number of clusters simultaneously, we will follow the philosophy of [32, 4] and look for multiplicities that are persistent across a nontrivial range of global scalings. In the limit of large data, the true number of clusters will persist from a maximum scale to any arbitrarily small scale, so in an appropriate sense, the 'most' persistent multiplicity gives the true number of clusters. However, for a fixed finite data set, there can easily be several multiplicities which persist, in which case there are multiple scales of clustering in the data. Rather than selecting one scale, we advocate viewing the persistence diagram as a useful tool for understanding the multiscale nature of the data set.

Next we revisit the target example of Fig. 2.4, but with variable densities.

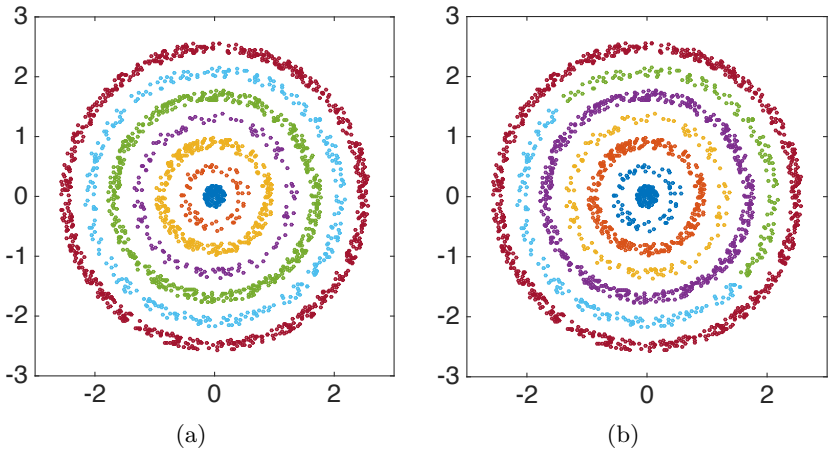


FIG. 3.4. Target example with varying densities. (a) Seven clusters with variable bandwidth (b) Same with fixed bandwidth kernel given by the standard diffusion map construction. The variable density causes the fixed bandwidth method to merge the inner two components and split one of the outer components into two.

Using the same parameters as in the previous example, we obtain a persistence diagram (not shown) with a range of ϵ having a 7-dimensional eigenspace corresponding to the eigenvalue zero, and a clustering shown in Fig. 3.4(a). For comparison, a diffusion map (fixed bandwidth) kernel results in a less compelling clustering shown in Fig. 3.4(b).

Fig. 3.5 is an example where density of points varies within connected components. The variable bandwidth kernel (3.1) successfully divides the points into four components, shown in Fig. 3.5(a). On the other hand, the fixed bandwidth kernel, used by the diffusion maps Laplacian L_{dm}, fails completely on this example, since the outlier points tend to form their own components for any bandwidth small enough to separate the main four in the denser part of the region. The indicator functions colored in Fig. 3.5(a) are easily found with either the finite cutoff kernel (2.1) or the exponential kernel (2.2), as long as the variable bandwith kernel (3.1) is used. Fig. 3.5(b) shows the approximate $\epsilon(x)$ derived from the kernel density estimate $q(x)$.

While our results indicate that the variable bandwidth kernel of works robustly for several nontrivial examples, we emphasize that further improvement may still be possible in terms of adaptive kernels. For example, kernel density estimation in high dimensions can adapt not only the bandwidth, but also the ellipsoidal shape of the local similarity function [26, 21]. Another possible area of exploration for the current kernels is for manifolds with boundary, since the error bounds of [8] on the boundary are not as strong as in the interior, and this issue is not addressed in [2, 3].

By adding a global bandwidth parameter, the kernel of [31] can be seen to be a local kernel; so in the limit of large data the kernels of [31] and [2] are equivalent in terms of clustering on compact manifolds. Empirically, we

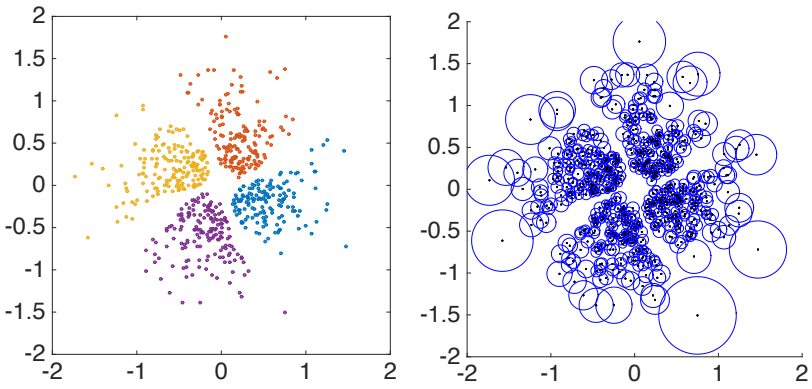


FIG. 3.5. Four sets with varying densities within the sets. (a) Assignment of clusters by the variable bandwidth kernel. (b) The variable bandwidth kernel uses the bootstrapped density estimate to build an appropriate weight matrix.

have had the best results with the kernel of [2], most likely due to the kernel density estimate being less sensitive to the sampling realization than the ad hoc scaling function of [31]. We advocate the variable bandwidth kernel of [2] over that of [31] due to the rigorous error bounds of Thm. 3.1 as well as empirically better results on numerical examples. In Fig. 3.6 we show the results of applying the kernel of [31] to the three box data set from Fig. 3.1.

4. Conclusion. Our aim is to clarify and extend the use of spectral clustering, principally by exploiting the assumption of a geometric manifold underlying the observed data. This assumption is not new; several authors beginning with Belkin and Niyogi [1] pursued the idea of reconstructing the Laplace-Beltrami operator on the manifold. However, with the recent discovery of a much more general set of "local" kernels that will generate that operator [3], and in particular kernels that can cope with arbitrarily variable bandwidth [2], it has become practical to reconstruct the Laplace-Beltrami operator, far from the large data limit, even when sampling densities fluctuate significantly throughout the data set.

We also emphasize the imperative of a scaling parameter in clustering problems, and argued for the viewpoint of persistence in making a decision about the number of clusters. In essence, we localize the idea of scaling by replacing the global scaling ϵ (with units of distance) with a localized $\epsilon(x) = \epsilon q(x)^{\beta}$, where $q(x)$ is the (unitless) sampling density. In addition, we advocate for the use of the LU-factorization as a dependable "final step" in the spectral clustering algorithm, to circumvent the need for another clustering algorithm to finish the process.

In this paper we restrict our attention to compact manifolds because the local kernel theory of [3] is restricted to compact manifolds. In fact, the variable bandwidth kernels introduced in [2] and advocated here are also applicable

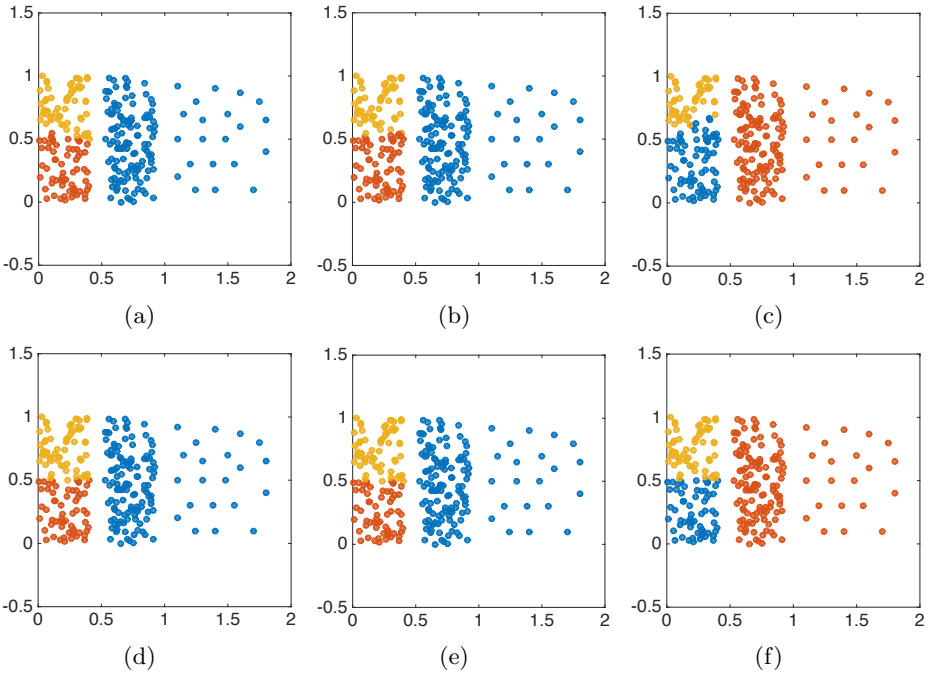


FIG. 3.6. Method of [31], assuming 3 clusters, for number of neighbors k set to values from 3 to 8 in parts (a) - (f), respectively.

to non-compact manifolds assuming that the sampling density has fast decay at infinity. So clustering on non-compact manifolds is also possible with the variable bandwidth kernel, although the Hodge theory also changes on non-compact manifolds. In particular, not all Neumann-harmonic functions are constant on non-compact manifolds. For example, the function $f(x) = x$ is harmonic on $\mathbb{R}$. However, if we restrict our attention to Neumann-harmonic functions which never change sign, we once again recover linear combinations of indicator functions. This suggests that topological clustering may be possible on non-compact manifolds, although a more complex unmixing algorithm would be required, in order to insure that the span of the unmixed eigenfunctions does not include any functions which change sign.

5. Acknowledgements. This research was partially supported by NSF grants DMS-1216568, DMS-1250936, and CMMI-130007.

Appendix A. Numerical Algorithm.

The numerical clustering algorithm has two main steps. First, we construct the discrete approximation L to the operator $\mathcal{L}_{\alpha,\beta}$ of Theorem 3.1 and find the eigenvectors of L. Second, we unmix the eigenvectors to approximate indicator functions which will define the clusters. We assume that the data set consists of n points $\{x_i\}_{i=1}^N \subset \mathcal{M} \subset \mathbb{R}^m$ sampled according to a density q on a d-dimensional manifold $\mathcal{M}$ relative to the volume form on $\mathcal{M}$.

Theorem 3.1 requires knowledge of the density function q within an error proportional to ϵ^2. The first task is to bootstrap an approximation to q from the data set. We will begin with a rough approximation of local neighbor distance $\hat{\rho}$, and use an exponential kernel to approximate q. The propagation of errors in the bootstrap estimate of q into the final kernel are accounted for in the second term of the error estimate in Theorem 3.1 as shown in [2].

According to Theorem 3.1, the construction of the discretized approximation to the Laplace-Beltrami operator begins with the kernel matrix W in (3.1). Define the diagonal matrix D by $D_{ii} = (\sum_j W_{ij})/q(x_i)^{d\beta}$ to represent the $q_\epsilon(x_i)$ in (3.2), where d is the manifold dimension. Define $W_\alpha = D^{-\alpha} W D^{-\alpha}$, and define the diagonal matrix $D_\alpha = W_\alpha \mathbf{1}$. Then the discrete version of the Laplace-Beltrami operator is the $n \times n$ matrix

$$L = \epsilon^{-2} R^{-2} (D_\alpha^{-1} W_\alpha - I)$$

where R is a diagonal matrix with $R_{ii} = \rho(x_i) = q(x_i)^\beta$. (Here we have neglected the constant factor m in Theorem 3.1, since it does not affect the $\lambda = 0$ eigenspace.)

To find eigenvectors of L, we instead compute eigenvectors of a similar matrix that is symmetric. Define the diagonal matrix $B = \epsilon R D_\alpha^{1/2}$. Then $BL = B^{-1} W_\alpha - \epsilon^{-2} R^{-2} B = SB$ where $S = B^{-1} W_\alpha B^{-1} - \epsilon^{-2} R^{-2}$ is symmetric. If $\tilde{\varphi}$ is an eigenvector of S, then $\varphi = B^{-1}\tilde{\varphi}$ is an eigenvector of L with the same eigenvalue.

The shape function h can be any function with exponential decay at infinity, and in all our examples we used the shape function $h(x) = \exp(-x/4)$. Note that in Theorem 3.1 the formula for $L_{\epsilon,\alpha,\beta}$ has a constant term m in the denominator which is equal to half of the second moment of the shape function. The constant 4 in the shape function $h(x)$ is chosen to result in $m = 1$, but even if a shape function is chosen with $m \neq 1$, this would only result in scaling the eigenvalues λ_j by this constant factor.

In the numerical implementation the matrix W can typically be represented as a sparse matrix due to the exponential decay in the weights W_{ij}. This is often accomplished by only allowing nonzero weights between each point and its nearest neighbors, which will also reduce the amount of memory required by the algorithm.

The algorithm contains a single nuisance parameter k, which is used in the initial density estimation. Namely, it is used to construct the ad hoc scaling function $\hat{\rho}$, which is used to obtain the kernel density estimate $q(x_i)$. We have found the algorithm to be quite robust to choices $4 \leq k \leq 64$, and in all of our examples we used $k = 8$.

Clustering with Variable Bandwidth Local Kernels

Inputs: Data set $\{x_i\}_{i=1}^n \subset \mathcal{M} \subset \mathbb{R}^m$, global scale parameter $\epsilon > 0$ and number of nearest neighbors k for ad hoc bandwidth.

Outputs: Number of clusters c at scale ϵ, and $n \times c$ indicator function matrix C

1. Find the pairwise distances $||x_i - x_j||$.
2. Find a kernel density estimate $q(x_i)$. For example:
   (a) Define the ad hoc bandwidth function $\hat{\rho}_i = \sqrt{\sum_{j=1}^k ||x_i - x_{I(i,j)}||^2}$ where $I(i,j)$ is the index of the j-th nearest neighbor of x_i.
   Tune the bandwidth for the kernel density estimate in steps (b)-(f).
   (b) Let $\delta_l = 2^l$ for $l = -30, -29.9, ..., 9.9, 10$.
   (c) Compute $T_l = \sum_{i,j=1}^N \exp\left(\frac{-||x_i - x_j||^2}{4\delta_l^2 \hat{\rho}_i \hat{\rho}_j}\right)$.
   (d) Estimate the local power law $T_l = \delta_l^a$ at each l by $a_l = \frac{\log T_l - \log T_{l-1}}{\log \delta_l - \log \delta_{l-1}}$.
   (e) Estimate the intrinsic dimension $d = \max_{\delta_l}\{a_l\}$ and set $\delta = \text{argmax}_{\delta_l}\{a_l\}$.
   (f) Estimate the density $q_i = q(x_i) = (4\pi\delta^2\hat{\rho}_i^2)^{-d/2} N^{-1} \sum_{j=1}^N \exp\left(\frac{-||x_i - x_j||^2}{4\delta^2 \hat{\rho}_i \hat{\rho}_j}\right)$.
3. Approximate the discrete Laplacian.
   (a) Define the local scaling $\epsilon_i = \epsilon q_i^\beta$. Set $\beta = -\frac{1}{2}$ and $\alpha = \frac{1}{2} - \frac{d}{4}$.
   (b) Form the kernel matrix $W_{ij} = h\left(\frac{||x_i - x_j||^2}{\epsilon_i \epsilon_j}\right)$.
   (c) Form the diagonal normalization matrix $D_{ii} = \sum_{j=1}^N W_{ij}/q_i^{d\beta}$.
   (d) Form normalized matrix $W_\alpha^S = D^{-\alpha} W D^{-\alpha}$.
   (e) Form the diagonal matrices $(D_\alpha)_{ii} = \sum_{j=1}^N (W_\alpha)_{ij}$, $R_{ii} = q_i^\beta$ and $B = \epsilon R D_\alpha^{1/2}$.
   (f) Form the symmetric matrix $S = B^{-1} W_\alpha B^{-1} - \epsilon^{-2} R^{-2}$.
4. Find the eigenvalues λ_j and eigenvectors $\tilde{\varphi}_j$ of S. The number of clusters c is the number of $|\lambda_j| < n\epsilon_{\text{mach}}$, where ϵ_{mach} is machine precision. The zero-eigenspace of the Laplacian is spanned by the columns of the $n \times c$ matrix $\Phi = B^{-1}\tilde{\Phi}$, where $\tilde{\Phi}$ holds the orthonormal zero-eigenvectors of S.
5. Unmix the eigenvectors.
   (a) Compute the $P\Phi = LU$ factorization of Φ.
   (b) Set A to be the inverse of the upper $c \times c$ block of $P\Phi$. Then C=ΦA^{-1}.

REFERENCES

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[2] T. Berry and J. Harlim. Variable bandwidth diffusion kernels. *Appl. Comp. Harmonic Anal.*, 39(doi:10.1016/j.acha.2015.01.001), 2014.

[3] T. Berry and T. Sauer. Local kernels and the geometric structure of data. *Appl. Comp. Harmonic Anal.*, 39(doi:10.1016/j.acha.2015.03.002), 2015.

[4] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[5] Gunnar Carlsson, Afra Zomorodian, Anne Collins, and Leonidas J Guibas. Persistence barcodes for shapes. *International Journal of Shape Modeling*, 11(02):149–187, 2005.

[6] Frédéric Chazal, Leonidas J Guibas, Steve Y Oudot, and Primoz Skraba. Persistence-based clustering in riemannian manifolds. *Journal of the ACM (JACM)*, 60(6):41, 2013.

[7] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[8] R. Coifman and S. Lafon. Diffusion maps. *Appl. Comp. Harmonic Anal.*, 21:5–30, 2006.

[9] R. Coifman, S. Lafon, B. Nadler, and I. Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Appl. Comp. Harmonic Anal.*, 21:113–127, 2006.

[10] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556. ACM, 2004.

[11] Chris HQ Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SDM*, volume 5, pages 606–610. SIAM, 2005.

[12] Matthias Hein and Simon Setzer. Beyond spectral clustering-tight relaxations of balanced graph cuts. In *Advances in Neural Information Processing Systems*, pages 2366–2374, 2011.

[13] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[14] F Jordan and F Bach. Learning spectral clustering. *Adv Neural Inf Process Syst*, 16:305–312, 2004.

[15] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM (JACM)*, 51(3):497–515, 2004.

[16] Boaz Nadler and Meirav Galun. Fundamental limitations of spectral clustering methods. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.

[17] Boaz Nadler, Stephane Lafon, Ronald Coifman, and Ioannis G Kevrekidis. Diffusion maps-a probabilistic interpretation for spectral embedding and clustering algorithms. In *Principal manifolds for data visualization and dimension reduction*, pages 238–260. Springer, NY, 2008.

[18] Boaz Nadler, Stephane Lafon, Ioannis Kevrekidis, and Ronald R Coifman. Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In *Advances in Neural Information Processing Systems*, pages 955–962. MIT Press, 2005.

[19] AY Ng, M Jordan, and Y Weiss. On spectral clustering: Analysis and an algorithm. *Neur. Inf. Proc. Soc.*, 2002.

[20] S. Rosenberg. *The Laplacian on a Riemannian manifold*. Cambridge University Press, 1997.

[21] Stephan R. Sain and David W. Scott. On locally adaptive density estimation. *Journal of the American Statistical Association*, 91(436):1525–1534, 1996.

[22] T. Sauer. *Numerical Analysis*. Pearson Education, 2nd edition, 2012.

[23] G. Schwarz. *Hodge decomposition—a method for solving boundary value problems*, volume 1607 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1995.

[24] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22:888–905, 2000.

[25] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[26] D. G. Terrell and D. W. Scott. Variable kernel density estimation. *Annals of Statistics*, 20:1236–1265, 1992.

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

[28] Ulrike Von Luxburg, Mikhail Belkin, and Olivier Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, pages 555–586, 2008.

[29] Scott White and Padhraic Smyth. A spectral clustering approach to finding communities in graph. In *SDM*, volume 5, pages 76–84. SIAM, 2005.

[30] Stella X Yu and Jianbo Shi. Multiclass spectral clustering. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 313–319. IEEE, 2003.

[31] L Zelnick-Manor and P Perona. Self-tuning spectral clustering. *Neur. Inf. Proc. Soc.*, 2004.

[32] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete and Computational Geometry*, 33(2):249–274, 2005.