Iterated Diffusion Maps for Feature Identification

Tyrus Berry[a,*], John Harlim[a,b]

[a]*Department of Mathematical Sciences, George Mason University, 4400 Exploratory Hall, Fairfax, Virginia 22030, USA*
[b]*Department of Meteorology, the Pennsylvania State University, 503 Walker Building, University Park, PA 16802-5013, USA*

Abstract

Recently, the theory of diffusion maps was extended to a large class of *local kernels* with exponential decay which were shown to represent various Riemannian geometries on a data set sampled from a manifold embedded in Euclidean space. Moreover, local kernels were used to represent a diffeomorphism $\mathcal{H}$ between a data set and a feature of interest using an anisotropic kernel function, defined by a covariance matrix based on the local derivatives $D\mathcal{H}$. In this paper, we generalize the theory of local kernels to represent degenerate mappings where the intrinsic dimension of the data set is higher than the intrinsic dimension of the feature space. First, we present a rigorous method with asymptotic error bounds for estimating $D\mathcal{H}$ from the training data set and feature values. We then derive scaling laws for the singular values of the local linear structure of the data, which allows the identification the tangent space and improved estimation of the intrinsic dimension of the manifold and the bandwidth parameter of the diffusion maps algorithm. Using these numerical tools, our approach to feature identification is to iterate the diffusion map with appropriately chosen local kernels that emphasize the features of interest. We interpret the iterated diffusion map (IDM) as a discrete approximation to an intrinsic geometric flow which smoothly changes the geometry of the data space to emphasize the feature of interest. When the data lies on a manifold which is a product of the feature manifold with an irrelevant manifold, we show that the IDM converges to the quotient manifold which is isometric to the feature manifold, thereby eliminating the irrelevant dimensions. We will also demonstrate empirically that if we apply the IDM to features which are not a quotient of the data manifold, the algorithm identifies an intrinsically lower-dimensional set embedding of the data which better represents the features.

Keywords: diffusion maps, local kernel, iterated diffusion map, dimensionality reduction, feature identification

1. Introduction

Often, for high-dimensional data and especially for data lying on a nonlinear subspace of Euclidean space, the variables of interest do not lie in the directions of largest variance and this makes them difficult to identify. The *features* (variables of interest) may be nonlinear functions of the ambient Euclidean coordinates. Moreover, other nonlinear combinations of the ambient coordinates may be independent of the variables of interest, and should be eliminated; we call these quantities the *irrelevant variables*. For example, consider the annulus shown in Figure 1, where the feature of interest is the radius as indicated by the color. The feature of interest is a nonlinear function of the ambient coordinates, namely $r = \sqrt{x^2 + y^2}$, and is completely independent of the irrelevant variable $\theta = \tan^{-1}(y/x)$. We should mention that a related direction which is being explored in the current research attempts to discover features which are common in multiple 'views' [9, 6, 18] using cross-diffusion between views and nonlinear canonical correlation analysis [10]. In this paper, we will consider the case when the desired feature is known on a training data set and we wish to learn the feature map in order that it may be extended to new data points. In other words, we consider the supervised learning problem, that is, to learn the underlying map that takes the data space to the feature space using a training data set that includes the feature values. In particular, we are seeking a representation of the feature map which can be extended to new data points.

*Corresponding author
Email addresses: tberry@gmu.edu (Tyrus Berry), jharlim@psu.edu (John Harlim)

Throughout this manuscript we will assume that the training data set consists of data points which lie near a d-dimensional manifold $\mathcal{M} \subset \mathbb{R}^m$ embedded in an m-dimensional Euclidean space; we refer to $\mathcal{M}$ as the *data space* or *data manifold* and we refer to $\mathbb{R}^m$ as the *ambient data space*. We also assume that we have a set of feature values corresponding to each training data point, and these feature values are assumed to lie near a $d_{\mathcal{N}}$-dimensional manifold $\mathcal{N} \subset \mathbb{R}^n$ embedded in an n-dimensional Euclidean space; we refer to $\mathcal{N}$ as the *feature space* or *feature manifold* and we refer to $\mathbb{R}^n$ as the *ambient feature space*. We do not assume any knowledge of the structure of the manifolds $\mathcal{M}, \mathcal{N}$ or the feature map $\mathcal{H} : \mathcal{M} \to \mathcal{N}$, we only assume that the feature map is differentiable.

When the feature manifold is intrinsically lower-dimensional than the data manifold, the data manifold contains information which is irrelevant to the feature, and we refer to this information broadly as the 'irrelevant variables' or the 'irrelevant space'. In some contexts it is possible to identify the irrelevant space explicitly, for example the data manifold may simply be a product manifold of the feature manifold and an irrelevant manifold. This is exactly the case with the annulus, which is a product manifold of the feature space $[0, 1] \ni r$ with the irrelevant space $[0, 2\pi) \ni \theta$. However, more complex relationships between the data manifold, feature manifold, and irrelevant variables are possible.

In this paper, we generalize a method introduced in [4], which was developed for representing diffeomorphisms to more general maps which are differentiable but not necessarily invertible. In [4], a diffeomorphism is represented using a *local kernel* to pull back the Riemannian metric from one manifold onto the other. With respect to the intrinsic geometry of the local kernel, the manifolds are isometric, and the isometry can be represented by a linear map between the eigenfunctions of the respective Laplacian operators. In this paper, we consider the more difficult case when the manifolds are not diffeomorphic, so that one manifold may even be higher dimensional than the other. This is typically the case with feature maps, since the data space may contain irrelevant variables. This implies that the data manifold dimension, d, may be greater than the feature manifold dimension, $d_{\mathcal{N}}$. In the annulus example the data space is two dimensional and both the feature (radius, r) and the irrelevant variable (angle, θ) are one dimensional.

The challenge of having irrelevant variables is that it violates the fundamental assumption of differential geometry, namely that it is local. This is because data points which differ only in the irrelevant variables will be far away in the data space and yet have the same feature values. This fundamental issue is independent of the amount of data available and is illustrated in Figure 1. Namely, if the feature of interest is the radius of an annulus, then points on opposite sides of the annulus are closely related with respect to this feature of interest. Conversely, points which are far away in the feature space may appear relatively close in data space; this can occur when many of the irrelevant variables are very close. Of course, in the limit of large data, points being close in data space implies that they are close in feature values. However, a large number of irrelevant variables can easily overwhelm any finite data set due to the curse-of-dimensionality. The presence of irrelevant variables makes it difficult to determine the true neighbors.

Intuitively our goal is to determine the true neighbors of every point in the data space, meaning the points in the data space which have similar feature values regardless of the values of the irrelevant variables. The key difficulty is that we need a method which can be extended to new data points, since the goal of representing the map $\mathcal{H}$ is to be able to apply this map to new points in data space. In order to learn the map $\mathcal{H}$ we will assume that we have a training data set for which the feature values are known. Of course, for the training data set, we could easily find the true neighbors of the training data points by using the known feature values. However, finding the neighbors using the feature values cannot be used for determining the true neighbors of a new data point, since the goal is to determine the feature values of the new data point.

In this paper, we introduce the Iterated Diffusion Map (IDM) which is an iterative method that smoothly distorts the data space in a way that locally contracts in the directions of the irrelevant variables and expands in the directions of the features. Crucially, this iterative mapping can be smoothly extended to new data points for which the feature values are unknown, allowing us to extend the feature map to these new data points. We illustrate our method for the purposes of intuitive explanation in Figure 1. In this example, the original data set is an annulus in the plane, but the variable of importance (represented by color in the top images) is the radial component of the annulus, meaning that the angular component is an irrelevant dimension of the manifold. The initial neighborhood is simply an Euclidean ball in the plane as shown in the bottom row of images. In the subsequent images we apply the diffusion map multiple times to evolve the data set in a way that biases it towards the desired feature. Notice that as the geometry evolves, the notion of neighborhood evolves. In the bottom right image, we see that after four iterations of the diffusion map, the notion of neighbor has grown to include any points which have the same radius, independent of their angle. Moreover, after four iterations, we see that points that were initially very close neighbors, namely points that have the same angle

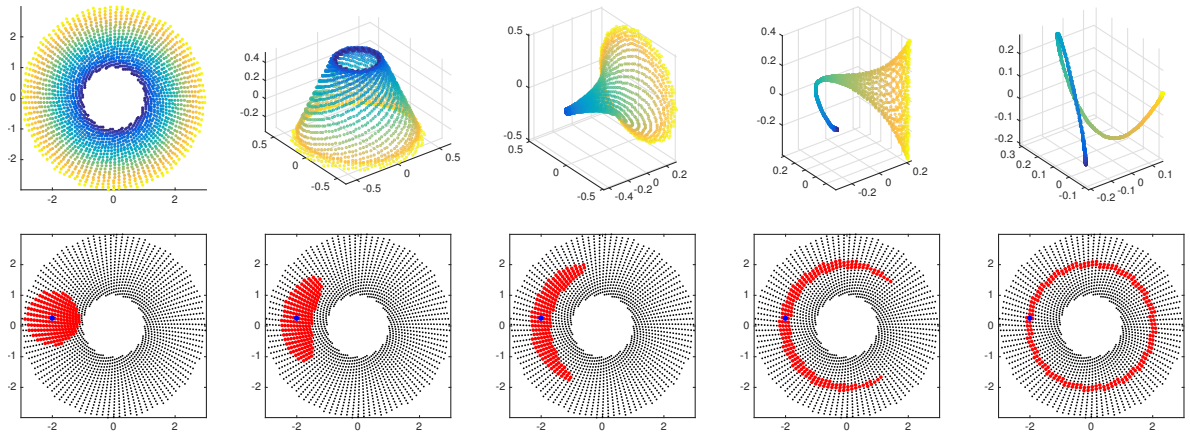

Figure 1: Top: Original data set colored according to the desired feature (leftmost) followed by four iterations of the diffusion map using a local kernel defined in Section 4. Bottom: Original data set showing the 200 nearest neighbors (red) of the blue data point, where the neighbors are found in the corresponding iterated diffusion map space. Notice that as the iterated diffusion map biases the geometry towards the desired feature (the radius), the neighbors evolve towards the *true* neighbors with respect to the desired feature.

but slightly different radii, are no longer neighbors. So after applying the IDM, the notion of neighbor becomes very sensitive to the feature (radius) and independent of the irrelevant variable (angle).

The construction of the IDM requires several tools. In Section 2.1, we will show that iterating the standard diffusion map of [7] has no effect (after the first application of the diffusion map, subsequent applications will approximate the identity map when appropriately scaled). This is because the isotropic kernel used in the standard diffusion map yields a canonical isometric embedding of the manifold. In Section 2.2, we review how *local kernels*, can change the geometry of the manifold and obtain an isometric embedding with respect to the new geometry. Local kernels are a broad generalization of the isotropic kernels used in [7] and were shown in [4] to be capable of representing any geometry on a manifold.

To construct a local kernel that emphasizes the feature directions, we will need to estimate the derivative of the feature map, $D\mathcal{H}$. In Section 3, we give a rigorous method of estimating $D\mathcal{H}$, including asymptotic error bounds, based on a weighted local linear regression. One complication of estimating $D\mathcal{H}(x)$ is that our only representation of the data and feature manifolds is in the ambient Euclidean spaces. Thus, we will actually estimate a map $D\hat{\mathcal{H}}(x) : \mathbb{R}^m \to \mathbb{R}^n$ which is equal to $D\mathcal{H}$ when restricted to the respective tangent spaces. Fortunately, the geometry of the local kernel which we will construct with $D\hat{\mathcal{H}}$ is only influenced by the restriction to the tangent spaces.

The estimation of $D\mathcal{H}(x)$ is accomplished with a weighted linear regression which localizes the regression around the base point x. Previously, in [5], a similar weighted linear regression was analyzed and was shown to converge to a matrix valued function of x, however the matrix valued function was not identified. The important contribution of Section 3 is that we assume that the feature map is described by an underlying differentiable function $\mathcal{H}$ and we are able to identify the matrix valued function as the derivative $D\mathcal{H}$. Our result requires a special empirical normalization beyond the standard regression of [5]. A significant result of [5] is that the variance of their estimator depends only on the intrinsic dimension of the manifold, rather than the dimension of the ambient space where the manifold is embedded. While we believe a similar result will hold for our estimator (following the methods of [15, 3]), due to the required normalization the variance of our regression is more complicated to compute and is beyond the scope of this manuscript. Reducing the variance of the estimator typically determines the data requirements, which will grow exponentially in the dimension of manifold as in [3]. To assist in finding this low dimensional structure, we will show that the scaling in our estimation of $D\hat{\mathcal{H}}$ allows one to identify tangent space of a manifold near a point (see Section 3.1). We also devise more robust criteria for estimating the intrinsic dimension of the manifold and the local bandwidth parameter of the diffusion maps algorithm (interested readers may check Appendix B).

While Section 3 provides us with a local description of the feature map $\mathcal{H}$ via $D\mathcal{H}(x)$, the main goal of this paper is to provide a global representation of $\mathcal{H}$. This global description will be built by tying together the local descriptions

at each data point using a kernel function. This is a nonparametric regression of the feature map $\mathcal{H}$ since we do not assume any parametric form. A related idea was introduce in [11], which looks for a linear map between Hilbert spaces defined on the manifold. In fact, such a linear map only exists for isometric manifolds, meaning the feature map $\mathcal{H}$ would have to be an isometry. However, in [4] it was shown that if $\mathcal{H}$ is a diffeomorphism, an anisotropic kernel could be constructed using the estimated derivatives $D\mathcal{H}(x)$ which would allow a linear representation of $\mathcal{H}$. The anisotropic kernel of [4] pulls back the geometry of the feature manifold onto the data manifold, making the manifolds isometric. Of course, assuming $\mathcal{H}$ to be a diffeomorphism implies that the feature manifold and data manifold to have the same intrinsic dimension, which is still overly restrictive.

In Section 4 we introduce the IDM as a discrete approximation of a geometric flow that contracts the irrelevant variables and expands the feature variables on the manifold. The goal of the IDM is to find a similar linear representation of a more general differentiable feature map which may not be invertible (so the feature manifold may be lower dimensional). The IDM will iteratively evolve the data manifold geometry towards the feature space geometry as shown in Figure 1. If the geometries become equal, a linear representation of $\mathcal{H}$ will again be possible following [4]. However, while the estimation of $D\mathcal{H}$ in Section 3 applies to any differentiable feature map, we will only be able represent certain classes of differentiable functions with the IDM. Moreover, the construction of the IDM requires that the data manifold is sufficiently well sampled, and our extension to new data points will be based on the Nyström extension [12] which also applies only to new data points which are near the training data.

In Section 4.2 we show that when the data manifold is the product of the feature manifold with irrelevant variables, this geometric flow will recover the quotient map from the data manifold to the feature manifold. In Section 4.3 we give several numerical examples demonstrating the IDM and we also include the IDM numerical algorithm in Appendix C. We close the paper with a short summary, highlighting the advantages and limitations, .

2. Background

In this section, we review recent key results that are relevant to the method developed in this paper. First, we remind the readers that, up to a scalar factor, a diffusion map is an isometric embedding of the manifold represented by a data set. Second, we briefly review the recently developed method for representing diffeomorphism between manifolds [4], which we will use as a building block.

2.1. The Diffusion Map as an Isometric Embedding

A natural distance that respects the nonlinear structure of the data is the geodesic distance, which can be approximated as the shortest path distance. However, the shortest path distance is very sensitive to small perturbations of a data set. A more robust metric that also respects the nonlinear structure of the data is the diffusion distance which is defined by an integral over transition probabilities through intermediate points. This metric can be approximated by the Euclidean distance of the data points in the embedded space constructed by the diffusion maps algorithm [7].

For a Riemannian manifold $\mathcal{M}$ with associated heat kernel $k(t, x, y)$, we can define the diffusion distance as,

$$D_t(x, y)^2 = \|k(t, x, \cdot) - k(t, y, \cdot)\|^2_{L^2(\mathcal{M})} = \int_{\mathcal{M}} (k(t, x, u) - k(t, y, u))^2 dV(u),$$

for $x, y \in \mathcal{M}$ where dV is the volume form on $\mathcal{M}$ associated to the Riemannian metric which corresponds to k. The heat kernel $k(t, x, u)$ represents the probability of a Brownian particle transitioning from x to u in time t. The diffusion distance averages the difference of these transition probabilities (starting from x and y) over all intermediate points u.

We can write the heat kernel as $k(t, x, y) = \left(e^{t\Delta}\delta_x\right)(y)$ where δ_x is the Dirac delta function and Δ is the (negative definite) Laplace-Beltrami operator on $\mathcal{M}$ with eigenfunctions $\Delta\varphi_i = \lambda_i\varphi_i$, where $0 = \lambda_0 > \lambda_1 > \lambda_2 > \ldots$. Using the Plancherel equality the diffusion distance becomes,

$$D_t(x, y)^2 = \|e^{t\Delta}\delta_x - e^{t\Delta}\delta_y\|^2_{L^2(\mathcal{M})} = \sum_{i=1}^{\infty} \left\langle e^{t\Delta}\delta_x - e^{t\Delta}\delta_y, \varphi_i \right\rangle^2 = \sum_{i=1}^{\infty} e^{2t\lambda_i}(\varphi_i(x) - \varphi_i(y))^2,$$

where the term $i = 0$ is zero since φ_0 is constant. Defining the diffusion map by,

$$\Phi_t(x) = (e^{t\lambda_1}\varphi_1(x), ..., e^{t\lambda_M}\varphi_M(x))^\top,$$

for M sufficiently large, the diffusion distance is well approximated by the Euclidean distance in the diffusion coordinates, $D_t(x,y) \approx \|\Phi_t(x) - \Phi_t(y)\|$. The key to making this idea practical is the algorithm of [7] which uses the data set $\{x_i\}$ sampled from a manifold $\mathcal{M} \subset \mathbb{R}^m$ to construct a sparse graph Laplacian L that approximates the Laplace-Beltrami operator, Δ on $\mathcal{M}$.

Of course, the dimension M of the diffusion coordinates will depend on the parameter t, which is intuitively a kind of coarsening parameter. For small t we have,

$$\left\langle e^{t\Delta}\delta_x, \delta_y \right\rangle = (4\pi t)^{-d/2} e^{-d_g(x,y)^2/(4t)}(u_0(x,y) + O(t)),$$

where $d_g(x,y)$ is the geodesic distance and d is the intrinsic dimension of $\mathcal{M}$ (see for example [14]). The function $u_0(x,y)$ is the first term in the heat kernel expansion. In [14] the following formula is derived for $u_0(x,y)$,

$$u_0(x,y) = |d(\exp_x^{-1})(y)|^{1/2} = |I_{d\times d} + O(d_g(x,y)^2)|^{1/2} = 1 + O(d_g(x,y)^d)$$

where $\exp_x$ is the exponential map based at x, and the expansion follows from noting that $\exp_x$ is a smooth map with first derivative equal to the identity at x and second derivative orthogonal to the tangent plane. Using the expansion of $u_0(x,y)$, for $d_g(x,y)$ sufficiently small, we have the following expansion of the heat kernel,

$$\left\langle e^{t\Delta}\delta_x, \delta_y \right\rangle = (4\pi t)^{-d/2} e^{-d_g(x,y)^2/(4t)}(1 + O(t, d_g(x,y)^d)), \tag{1}$$

and below we will bound the error by the worst case of the intrinsic dimension, namely $d = 1$. Using the heat kernel expansion (1), we can expand the diffusion distance as,

$$\begin{aligned}
D_t(x,y)^2 &= \left\langle e^{2t\Delta}\delta_x, \delta_x \right\rangle + \left\langle e^{2t\Delta}\delta_y, \delta_y \right\rangle - 2\left\langle e^{2t\Delta}\delta_x, \delta_y \right\rangle = (8\pi t)^{-d/2}\left(2 - 2e^{-d_g(x,y)^2/(8t)}\right)(1 + O(t, d_g(x,y))) \\
&= (8\pi t)^{-d/2}\left(2 - 2\left(1 - d_g(x,y)^2/(8t) + O(d_g(x,y)^4/t^2)\right)\right)(1 + O(t, d_g(x,y))) \\
&= (8\pi t)^{-d/2}(4t)^{-1} d_g(x,y)^2 (1 + O(d_g(x,y)^2/t))(1 + O(t, d_g(x,y))) \\
&= \frac{d_g(x,y)^2}{(2\pi)^{d/2}(4t)^{d/2+1}}\left(1 + O(t, d_g(x,y), d_g(x,y)^2/t)\right).
\end{aligned} \tag{2}$$

Based on (2) we define the rescaled diffusion map by,

$$\hat{\Phi}_t(x) = (2\pi)^{d/4}(4t)^{d/4+1/2}\Phi_t(x),$$

and the rescaled diffusion distance, $\hat{D}_t(x,y) \equiv (2\pi)^{d/4}(4t)^{d/4+1/2} D_t(x,y)$, which is approximated by,

$$\hat{D}_t(x,y)^2 \approx \|\hat{\Phi}_t(x) - \hat{\Phi}_t(y)\|^2 \approx (2\pi)^{d/2}(4t)^{d/2+1} D_t(x,y)^2 = d_g(x,y)^2 + O(t\, d_g(x,y)^2, d_g(x,y)^3, d_g(x,y)^4/t),$$

so that for $d_g(x,y)^2 \ll t \ll 1$ the rescaled diffusion distance approximates the geodesic distance. The purely geometric fact that $\hat{\Phi}_t$ is an approximate isometry was realized as early as [2] outside of the context of the diffusion maps algorithm.

Notice that the rescaled diffusion distance only approximates the geodesic distance when the geodesic distance is small. In particular, the diffusion distance should not be thought of as an approximate geodesic distance, as is clearly shown in Figure 2 below. For small distances, the rescaled diffusion distance and the geodesic distance also agree very closely with the Euclidean distance in any isometric embedding, as was shown in [7]. In fact, the diffusion map provides a canonical embedding of the manifold $\mathcal{M}$ up to a rotation in the following sense: For any isometric image $\mathcal{N} = \iota(\mathcal{M})$ of $\mathcal{M}$ where ι is an isometry, the diffusion map embeddings of $\mathcal{N}$ and $\mathcal{M}$ with the same parameter t will differ by at most an orthogonal linear map. This is because the eigenfunctions of the Laplacian depend only on the geometry of the manifold, which is preserved by an isometric map, and for the eigenfunctions corresponding to repeated eigenvalues may differ only by an orthogonal transformation. Moreover, the fact that the rescaled diffusion map preserves small geodesic distances implies that the rescaled diffusion map is approximately an isometric embedding (this was shown previously in [13] which provides more detailed bounds).

The approximate isometry means that the Laplace-Beltrami operator on $\Phi_t(\mathcal{M})$ is very close to the Laplace-Beltrami operator on $\mathcal{M}$ in the sense that $(\Delta_{g_\mathcal{N}} f) \circ \Phi_t \approx \Delta_{g_\mathcal{M}}(f \circ \Phi_t)$, for sufficiently smooth functions $f : \Phi_t(\mathcal{M}) \to$

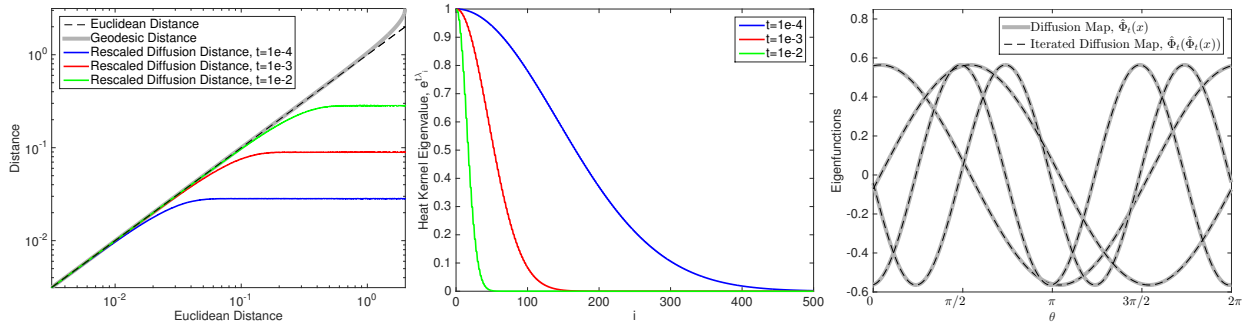

Figure 2: For 2000 data points equally spaced on a unit circle in $\mathbb{R}^2$, we compare the Euclidean distance, geodesic distance, and rescaled diffusion distances for $t \in \{10^{-4}, 10^{-3}, 10^{-2}\}$ (left). We also show the spectra of the heat kernel $e^{t\lambda_i}$ for the corresponding values of t (middle) and the results of iterating the standard diffusion map compared to the original diffusion map eigenfunctions (right).

$\mathbb{R}$ and the corresponding inherited Riemannian metrics, $g_{\mathcal{M}}, g_{\mathcal{N}}$. This implies that the eigenfunctions of the new Laplace-Beltrami operator will be very close to the eigenfunctions of the original Laplace-Beltrami operator. Since the diffusion map is defined by these eigenfunctions, if we iterate the rescaled diffusion map, the results should not change.

To demonstrate these facts numerically, we generated $N = 2000$ points $\{x_j\}_{j=1}^N$ equally spaced on a unit circle in $\mathbb{R}^2$. We applied the diffusion maps algorithm to estimate the eigenvalues λ_i and eigenfunctions $\varphi_i(x_j)$ of the Laplace-Beltrami operator on the unit circle. We should emphasize that it is crucial to correctly normalize the eigenfunctions φ_i using a kernel density estimate $q(x_j)$, that is, we require,

$$1 = \frac{1}{N}\sum_{j=1}^{N} \frac{\varphi_i(x_j)^2}{q(x_j)} \approx \int_{\mathcal{M}} \varphi_i(x)^2 dV(x),$$

where dV is the volume form on $\mathcal{M}$ inherited from the ambient space. See [3] for details on the Monte-Carlo integral above and a natural density estimate implicit to the diffusion maps construction. We then evaluated the rescaled diffusion map $\hat{\Phi}_t(x_j)$ for $t \in \{10^{-4}, 10^{-3}, 10^{-2}\}$ and compared the resulting diffusion distances $\hat{D}_t(x_i, x_j) \approx \|\hat{\Phi}_t(x_i) - \hat{\Phi}_t(x_j)\|$ to the Euclidean distances $\|x_i - x_j\|$ and the geodesic distances $d_g(x_i, x_j)$ in Figure 2. Notice that for distances less than $t^{1/2}$ all the distances agree as shown above. We also show the spectra of the heat kernels, $e^{t\lambda_i}$, which are the weights of the various eigenfunctions in the diffusion map embedding. Notice that for t large, the spectrum decays much faster, so fewer eigenfunctions are required for the diffusion distance to be well approximated by the Euclidean distance in the diffusion mapped coordinates. Finally, for $t = 10^{-2}$, we performed an 'iterated' diffusion map, by computing the (rescaled) diffusion map of the data set $\hat{x}_j \equiv \hat{\Phi}_t(x_j)$, in effect finding $\hat{\Phi}_t(\hat{\Phi}_t(x))$. We then compared the eigenfunctions $\varphi_i(x_j)$ from the first diffusion map with those $\hat{\varphi}_i(\hat{x}_j) = \hat{\varphi}_i(\hat{\Phi}_t(x))$. Due to the symmetry of the unit circle, the eigenfunctions corresponding to repeated eigenvalues differed by an orthogonal linear map (meaning a phase shift in this case). After removing the phase shift, the eigenfunctions are compared in Figure 2.

As a second numerical example, we generated $N = 20000$ points with a standard embedding of the torus in $\mathbb{R}^3$

$$(\theta, \phi) \mapsto ((2 + \cos\theta)\cos\phi, (2 + \cos\theta)\sin\phi, \sin\theta)^\top.$$

In order to obtain more uniformly spaced points we generated a grid which is equally spaced in θ and with the number of points in the ϕ direction proportional to $\sqrt{5 + 4\cos\theta}$. This non-uniform grid in (θ, ϕ) results in a more uniform spacing of the points on the embedded torus, which allows the asymptotics to be obtained with fewer points. We applied the diffusion maps algorithm to this data set to estimate the first 2000 eigenfunctions and eigenvalues of the Laplacian and then computed the rescaled diffusion map $\hat{\Phi}_t$ on each of the data points. In Figure 3 we compare the diffusion distance for $t \in \{10^{-3}, 10^{-2}\}$ to the original Euclidean distance for the 500 nearest neighbors of each data point in the original data set. For $t = 10^{-2}$ we then compute the diffusion map of the 2000-dimensional embedding $\hat{\Phi}_t$ and in Figure 3 we compare the first non-trivial eigenfunction of this iterated diffusion map to that of the original

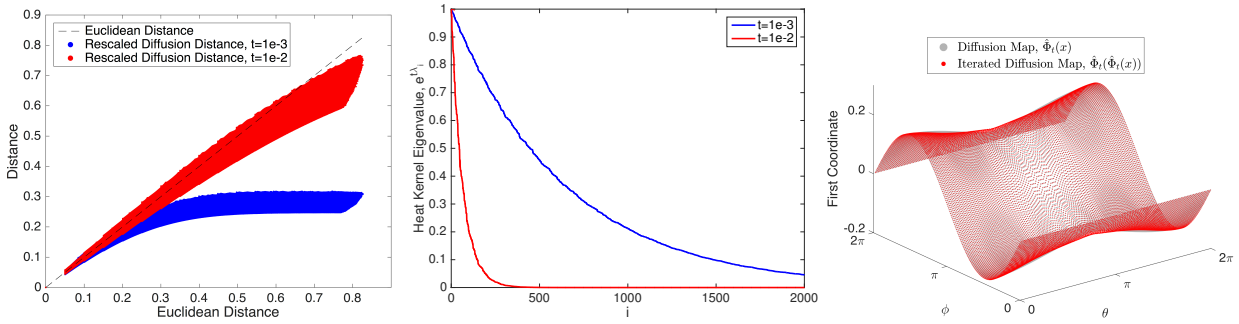

Figure 3: For 20000 data points on a torus in $\mathbb{R}^3$, we compare the Euclidean distance and rescaled diffusion distances for $t \in \{10^{-3}, 10^{-2}\}$ (left). We also show the spectra of the heat kernel $e^{t\lambda_i}$ for the corresponding values of t (middle) and the results of iterating the standard diffusion map compared to the original diffusion map for the first non-trivial eigenfunction (right). Notice that the mapping in the Euclidean and diffusion distances are not one-to-one especially when the diffusion distance is large.

diffusion map as a function of the intrinsic coordinates θ and ϕ. Figure 3 shows very good agreement between the initial and iterated diffusion map eigenfunctions, however even for this large data set the agreement is not exact. This illustrates that the rescaled diffusion map is only an asymptotically an isometry, and for small data sets with large values of t iterating the diffusion map will distort the geometry. However, as shown above, in the limit as $t \to 0$ (which requires $N \to \infty$) the rescaled diffusion map is an isometry and so iteration the diffusion map will have no effect in this limit.

Since the diffusion map Φ_t differs from the rescaled diffusion map $\hat{\Phi}_t$ by a scalar factor, the eigenfunction from iterating the standard diffusion map will also agree. The only purpose of the rescaled diffusion map $\hat{\Phi}_t$ is to exactly recover the local distances in the data set, and thereby to also find the same eigenvalues (since rescaling the manifold will change the spectrum of the Laplacian). Finally, if the standard diffusion map is used, the nuisance parameter ϵ will have to be retuned in order to iterate the diffusion map, since the diffusion distances will be scaled differently than the original distances. We emphasize that the goal of this section is to show that iterating the standard diffusion map algorithm is not a useful method. However, in [4] it was shown that a generalization of diffusion maps to *local kernels* can be used to construct the Laplace-Beltrami operator with respect to a different metric. In the remainder of the paper we will see that when the new metric is induced by a feature map on the data set, iterating the diffusion map has a nontrivial effect which can be beneficial.

2.2. *Local Kernels and the Pullback Geometry*

The connection between kernel functions and geometry was introduced by Belkin and Niyogi in [1] and generalized by Coifman and Lafon in [7]. Assuming that a data set $\{x_i\}$ is sampled from a density $p(x)$ supported on a d-dimensional manifold $\mathcal{M} \subset \mathbb{R}^m$, summing a function $\sum_i f(x_i)$ approximates the integral $\int_{\mathcal{M}} f(y)p(y)\,dV(y)$ where $V(y)$ is the volume form on $\mathcal{M}$ inherited from the ambient space $\mathbb{R}^m$. The central insight of [1, 7] is that by choosing a kernel function $K(x_i, x_j)$ which has exponential decay, the integral $\sum_j K(x_i, x_j)f(x_j) \approx \int_{\mathcal{M}} K(x_i, y)f(y)p(y)\,dV(y)$ is localized to the tangent space $T_{x_i}\mathcal{M}$ of the manifold.

The theory of [1, 7] was recently generalized in [4] to a wide class of kernels called *local kernels* which are assumed only to have decay that can be bounded above by an exponentially decaying function of distance. The results of [4] generalized an early result of [17] to a much wider class of kernels and connected these early results to their natural geometric interpretations. In this paper we will use the following prototypical example of a local kernel, since it was shown in [4] that every operator which can be obtained with a local kernel can also be obtained with a prototypical kernel. Let $C(x) \in \mathbb{R}^{m\times m}$ be a matrix valued function on the manifold $\mathcal{M} \subset \mathbb{R}^m$ such that each $C(x)$ is a symmetric positive definite $m \times m$ matrix. Define the prototypical kernel with covariance C (and first moment of zero)

$$K(\epsilon, x, y) = \exp\left(-\frac{(x-y)^T C(x)^{-1}(x-y)}{2\epsilon}\right). \tag{3}$$

The theory of local kernels [4] uses a method closely related to the method of Diffusion Maps of [7] to construct matrices L_ϵ and L_ϵ^* which are discrete approximations to the following operators,

$$\mathcal{L}f = \sum_{i,j=1}^{d} \frac{1}{2} c_{ij} \nabla_i \nabla_j f \qquad \mathcal{L}^* f = \sum_{i,j=1}^{d} \frac{1}{2} \nabla_j \nabla_i (c_{ij} f), \tag{4}$$

where ∇_i refers to the covariant derivative in the direction of the i-th basis element for $T_x\mathcal{M}$. The matrices L_ϵ, L_ϵ^* are consistent estimators of the operators in (4) in the sense that that in the limit of large data and as $\epsilon \to 0$ it is shown in [4] that $L_\epsilon \to \mathcal{L}$ and $L_\epsilon^* \to \mathcal{L}^*$. Notice that the matrix valued function $C(x)$ acts on the ambient space, whereas the tensor $c(x)$ in the limiting operator $\mathcal{L}$ is only defined on the tangent planes of $\mathcal{M}$. As shown in [4], only the projection of $C(x)$ onto the tangent space $T_x\mathcal{M}$ will influence the operator $\mathcal{L}$. Thus, we introduce the linear map $\mathcal{I}(x) : \mathbb{R}^m \to T_x\mathcal{M}$ which acts as the identity on the tangent plane as a subspace of $\mathbb{R}^m$ and sends all vectors originating at x which are orthogonal to $T_x\mathcal{M}$ to zero. The map $\mathcal{I}$ projects the ambient space onto the tangent space so that $\mathcal{I}(x)$ is a $d \times m$ matrix and we define

$$c(x) = \mathcal{I}(x) C(x) \mathcal{I}(x)^\top \tag{5}$$

so that $c(x)$ will be the same for the equivalence class of matrices $C(x)$ which are equal when projected onto the tangent space.

To build a global map $\mathcal{H} : \mathcal{M} \to \mathcal{N} = \mathcal{H}(\mathcal{M})$ between data sets, we need to estimate the local linear maps $D\mathcal{H}(x_i)$ between the tangent spaces. Notice that,

$$D\mathcal{H}(x) : T_{x_i}\mathcal{M} \to T_{\mathcal{H}(x_i)}\mathcal{N},$$

is a $d_\mathcal{N} \times d$ matrix, where d is the intrinsic dimension of $\mathcal{M}$ and $d_\mathcal{N}$ is the intrinsic dimension of $\mathcal{N}$. However, since the data lies in the ambient Euclidean spaces we can only compute a local linear regression in the ambient spaces. Therefore, we will represent $D\mathcal{H}$ with a map,

$$D\hat{\mathcal{H}}(x) : \mathbb{R}^m \supset T_x\mathcal{M} \to \mathbb{R}^n \supset T_{\mathcal{H}(x)}\mathcal{N},$$

between the ambient spaces. Similar to $\mathcal{I}(x)$, we introduce the notation $\mathcal{I}_\mathcal{N}(\mathcal{H}(x))$ for the $d_\mathcal{N} \times n$ matrix valued function which projects orthogonally from $\mathbb{R}^n$ onto the tangent space $T_{\mathcal{H}(x)}\mathcal{N} \subset \mathbb{R}^n$. With this notation we have the following relationship between $D\mathcal{H}$ and $D\hat{\mathcal{H}}$,

$$D\hat{\mathcal{H}}(x) = \mathcal{I}_\mathcal{N}(\mathcal{H}(x))^\top D\mathcal{H}(x) \mathcal{I}(x). \tag{6}$$

In Section 3 we will provide a consistent estimator for $D\hat{\mathcal{H}}(x) \in \mathbb{R}^{n \times m}$, and crucially in Theorem 2.1 only the restriction to $D\mathcal{H}$ will influence the intrinsic geometry defined by the kernel.

Given data sampled from a d-dimensional manifold $\mathcal{M}$ embedded in Euclidean space $\mathbb{R}^m$ the manifold $\mathcal{M}$ naturally inherits a Riemannian metric, $g_\mathcal{M}$, from the ambient space. The standard Diffusion Maps algorithm uses an isotropic kernel (where the covariance matrix is a multiple of the identity matrix) to estimate the Laplace-Beltrami operator corresponding to the metric $g_\mathcal{M}$. It was shown in [4] that local kernels such as (3) can be used to approximate the Laplace-Beltrami operator corresponding to a new Riemannian metric $\tilde{g} = c^{-1/2} g_\mathcal{N} c^{-1/2}$, where c is defined in (5) and $g_\mathcal{N}$ is the Riemannian metric which the manifold $\mathcal{N} = \mathcal{H}(\mathcal{M})$ inherits from the ambient space $\mathbb{R}^n$. The key to this result is to define the local kernel using a covariance matrix $C(x)$ such that $c(x)$ defined in (5) satisfies $c^{-1} = D\mathcal{H}^\top D\mathcal{H}$. Formally, we summarize this result as follows:

Theorem 2.1 (Pullback geometry of local kernels, with nonuniform sampling)**.** *Let $(\mathcal{M}, g_\mathcal{M})$ be a Riemannian manifold and let $\{x_i\}_{i=1}^N \subset \mathcal{M}$ be sampled according to any smooth density on $\mathcal{M}$. Let $\mathcal{H} : \mathcal{M} \to \mathcal{N}$ be a diffeomorphism and let $y_i = \mathcal{H}(x_i)$ and $c(x_i)^{-1} = D\mathcal{H}(x_i)^\top D\mathcal{H}(x_i)$. For the local kernel K in (3) with any $C(x)$ which restricts to $c(x) = \mathcal{I}(x) C(x) \mathcal{I}(x)^\top$, define the symmetric kernel $\overline{K}(\epsilon, x, y) = K(\epsilon, x, y) + K(\epsilon, y, x)$. Then for any smooth function f on $\mathcal{M}$,*

$$\lim_{N\to\infty} \frac{2}{\epsilon} \left(\frac{\sum_{j=1}^{N} \overline{K}(\epsilon, x_i, x_j) f(x_j) / \sum_l \overline{K}(\epsilon, x_j, x_l)}{\sum_{j=1}^{N} \overline{K}(\epsilon, x_i, x_j) / \sum_l \overline{K}(\epsilon, x_j, x_l)} - f(x_i) \right) = \Delta_{\tilde{g}} f(x_i) + O(\epsilon) = \Delta_{g_\mathcal{N}} (f \circ \mathcal{H}^{-1})(y_i) + O(\epsilon)$$

where $\tilde{g}(u,v) = g_{\mathcal{N}}(D\mathcal{H}u, D\mathcal{H}v)$ *in other words* $\tilde{g} = D\mathcal{H}^\top g_{\mathcal{N}} D\mathcal{H} = c^{-1/2} g_{\mathcal{N}} c^{-1/2}$.

Theorem 2.1 follows directly from Theorem 4.7 of [4]. This result was used by [4] to represent a diffeomorphism between two manifolds. We assume we are given a training data set $x_i \in \mathcal{M} \subset \mathbb{R}^m$ sampled from the data manifold $\mathcal{M}$ along with the true feature values, $y_i = \mathcal{H}(x_i)$, where y_i lie on $\mathcal{N} = \mathcal{H}(\mathcal{M})$. When $\mathcal{H}$ is a diffeomorphism, we can use a local kernel to pullback the Riemannian metric from $\mathcal{N}$ onto $\mathcal{M}$ via the correspondence between the data sets. With this metric on $\mathcal{M}$, the two manifolds are isometric, which implies that the Laplacians ($\Delta_{\tilde{g}}$ on $\mathcal{M}$ and $\Delta_{g_{\mathcal{N}}}$ on $\mathcal{N}$) have the same eigenvalues, and that the associated eigenfunctions of any eigenvalue are related by an orthogonal transformation [14].

In Section 3 we will give a rigorous method to approximate $c(x_i)^{-1} = D\mathcal{H}(x_i)^\top D\mathcal{H}(x_i)$ by finding $D\hat{\mathcal{H}}(x_i)$ using the training data. With this approximation, numerically we evaluate the local kernel

$$K(\epsilon, x_i, x_j) = \exp\left(-\frac{\|D\hat{\mathcal{H}}(x_i)(x_j - x_i)\|^2}{2\epsilon}\right). \tag{7}$$

Notice that by properties of orthogonal projection $\mathcal{I}(x)\mathcal{I}(x)^\top = I_{d\times d}$ and similarly for $\mathcal{I}_{\mathcal{N}}$ so that

$$\mathcal{I}(x) D\hat{\mathcal{H}}(x_i)^\top D\hat{\mathcal{H}}(x_i) \mathcal{I}(x)^\top = D\mathcal{H}(x_i)^\top D\mathcal{H}(x_i)$$

which shows that (7) satisfies the assumptions of Theorem 2.1.

By Theorem 2.1, using the kernel (7), we approximate the Laplacian $\Delta_{\tilde{g}} = (\mathcal{H}^{-1})^* \Delta_{g_{\mathcal{N}}}$ on $\mathcal{M}$. Simultaneously, using the standard diffusion maps algorithm (with $\alpha = 1$) we approximate the Laplacian $\Delta_{g_{\mathcal{N}}}$ on $\mathcal{N}$. Since $(\mathcal{M}, \tilde{g})$ and $(\mathcal{N}, g_{\mathcal{N}})$ are isometric, the eigenvalues of $\Delta_{\tilde{g}}$ and $\Delta_{g_{\mathcal{N}}}$ will be the same and the corresponding eigenfunctions will be related by an orthogonal transformation. By taking sufficiently many eigenfunctions $\Delta_{\tilde{g}}\varphi_l = \lambda_l \varphi_l$ and $\Delta_{g_{\mathcal{N}}}\tilde{\varphi}_l = \tilde{\lambda}_l \tilde{\varphi}_l$ on the respective manifolds, the eigenfunctions can be considered as coordinates of an embeddings $\Phi(x) = (\varphi_1(x), ..., \varphi_M(x))^\top$ and $\tilde{\Phi}(y) = (\tilde{\varphi}_1(y), ..., \tilde{\varphi}_M(y))^\top$. We can now project the diffeomorphism $\mathcal{H}$ into these coordinates as,

$$\begin{array}{ccc} \mathcal{M} & \xrightarrow{\mathcal{H}} & \mathcal{N} \\ \downarrow \Phi & & \downarrow \tilde{\Phi} \\ L^2(\mathcal{M}, \tilde{g}) \approx \mathbb{R}^M & \xrightarrow{\tilde{\Phi}\circ\mathcal{H}\circ\Phi^{-1}} & L^2(\mathcal{N}, g_{\mathcal{N}}) \approx \mathbb{R}^M \end{array}$$

where $\tilde{\Phi} \circ \mathcal{H} \circ \Phi^{-1}$ is linear and can be estimated using linear least squares. Finally, to extend the diffeomorphism to a new data point $x \in \mathcal{M}$ we need only extend the map Φ to this new data point using the Nyström extension formula [12],

$$\varphi_l(x) = \frac{1}{\lambda_l} \sum_{j=1}^{N} J(x, x_j) \varphi_l(x_j) \tag{8}$$

where

$$J(x, x_j) = \frac{2}{\epsilon}\left(\frac{\overline{K}(\epsilon, x, x_j)/\sum_l \overline{K}(\epsilon, x_j, x_l)}{\sum_{k=1}^{N} \overline{K}(\epsilon, x, x_k)/\sum_l \overline{K}(\epsilon, x_k, x_l)} - 1\right)$$

is the evaluation of the normalized local kernel from Theorem 2.1 on the new data point x paired with each original data point x_j. The Nyström extension is a standard technique in kernel methods and follows from the fact that

$$\sum_{j=1}^{N} J(x, x_j)\varphi_l(x_j) \approx (\Delta_{\tilde{g}}\varphi_l)(x) = \lambda_l \varphi_l(x)$$

where the first equality follows from Theorem 2.1 and the second is the definition of the eigenfunction φ_l.

Now that we have extended the embedding Φ to the new point x, we can map the point $\Phi(x)$ to a point $\tilde{y}$ using the linear map $\tilde{\Phi} \circ \mathcal{H} \circ \Phi^{-1}$ (which is estimated using a linear regression on the training data points) by

$$\tilde{y} = \tilde{\Phi} \circ \mathcal{H}(x) = \left(\tilde{\Phi} \circ \mathcal{H} \circ \Phi^{-1}\right)(\Phi(x)).$$

At this point we have mapped the point x into the new embedding space $\tilde{\Phi}(\mathcal{N})$ which is an isometric copy of the target space $\mathcal{N}$. Since $\tilde{\Phi}$ is an invertible map, there is a unique point $y \in \mathcal{N}$ such that $\tilde{y} = \tilde{\Phi}(y)$, however identifying this point y would require inverting the Nyström extension. Instead, in Section 4 we will work directly with the new coordinates $\tilde{\Phi}(\mathcal{N})$, since this new embedding is isometric to $\mathcal{N}$.

Notice that the key to the existence of the linear map $\tilde{\Phi} \circ \mathcal{H} \circ \Phi^{-1}$ is that the diffeomorphism $\mathcal{H}$ induces a new metric on $\mathcal{M}$ that is isometric to the metric on $\mathcal{N}$. In Section 4 we will make use of this theorem for identifying feature in $\mathcal{M}$ that is relevant to the data in $\mathcal{N}$, even when $\mathcal{H}$ is not a diffeomorphism, but simply a differentiable mapping. However, we will first give rigorous results in Section 3 for approximating the tangent plane $T_x\mathcal{M}$ and the derivative $D\mathcal{H}$ from data.

3. Tangent Spaces and Derivatives

In this section we improve and make rigorous a method originally introduced in [4] that estimates the local linear maps from data using a weighted regression. For a differentiable map $\mathcal{H} : \mathcal{M} \subset \mathbb{R}^m \to \mathcal{N} \subset \mathbb{R}^n$ recall that we define $D\hat{\mathcal{H}}(x)$ in (6) to be the orthogonal lifting of $D\mathcal{H}$ to the ambient space. In other words, $D\hat{\mathcal{H}}$ is a $n \times m$ matrix which has rank equal to the minimum of the dimensions d and $d_{\mathcal{N}}$ of $\mathcal{M}$ and $\mathcal{N}$ respectively, and $D\hat{\mathcal{H}}$ is equal to $D\mathcal{H}$ when restricted to the tangent spaces of $\mathcal{M}$ and $\mathcal{N}$.

To estimate $D\hat{\mathcal{H}}(x_i)$, we take the nearest neighbors $\{x_j\}$ of x_i and use the fact that we are given the value of the feature map on the training data to find $y_i = \mathcal{H}(x_i)$ and the neighbors $y_j = \mathcal{H}(x_j)$. Note that the set $\{y_j\}$ may not be the nearest neighbors of y_i. In [4] they construct the weighted vectors

$$dx_j = \exp\left(-\|x_j - x_i\|^2/(4\epsilon)\right)(x_j - x_i) \qquad dy_j = \exp\left(-\|x_j - x_i\|^2/(4\epsilon)\right)(y_j - y_i),$$

and define $D\hat{\mathcal{H}}(x_i)$ to be the matrix which minimizes $\sum_j \|dy_j - D\hat{\mathcal{H}}(x_i)dx_j\|^2$. Intuitively, the exponential weight is used to localize the vectors; otherwise the linear least squares problem would try to preserve the longest vectors $x_j - x_i$, which do not represent the tangent space well. This method of localization was used in [4] for estimating $D\hat{\mathcal{H}}(x_i)$, and it is also closely related to a method of determining the tangent space of a manifold which was introduced in [16]. Using the foundational theory developed in [7] we will now make this method of finding tangent spaces and derivatives rigorous.

Theorem 3.1. *Let x_i be samples from $\mathcal{M} \subset \mathbb{R}^m$ with density $p(x)$ and $y_i = \mathcal{H}(x_i)$ where $\mathcal{H} : \mathcal{M} \to \mathcal{N} \subset \mathbb{R}^n$. Let $x \in \mathcal{M}$ with $y = \mathcal{H}(x)$ and define the normalization factor,*

$$D(x) = \sum_{i=1}^{N} \exp\left(-\frac{\|x_i - x\|^2}{2\epsilon}\right).$$

Let X to be a matrix with columns $X_j = D(x)^{-1/2} \exp\left(-\frac{\|x_j - x\|^2}{4\epsilon}\right)(x_j - x) = D(x)^{-1/2} dx_j$ and let Y be a matrix with columns $Y_j = D(x)^{-1/2} \exp\left(-\frac{\|x_j - x\|^2}{4\epsilon}\right)(y_j - y) = D(x)^{-1/2} dy_j$, then

$$\lim_{N \to \infty} \frac{1}{\epsilon} Y X^{\top} = D\hat{\mathcal{H}}(x) + O(\epsilon), \tag{9}$$

with $D\hat{\mathcal{H}}(x)$ as in (6).

Proof. See Appendix A. □

Theorem 3.1 shows that the correlation $\frac{1}{\epsilon}YX^\top$ is a consistent estimator of $D\hat{\mathcal{H}}$, meaning that in the limit of infinite data and $\epsilon \to 0$ we recover the true derivative of the feature map. Obtaining this consistent estimator for data and features lying on submanifolds embedded in Euclidean space is a novel result which requires both the exponential weighting and the correct normalization factor, $D(x)$. Due to the presence of the normalizing factor $D(x)$, the variance of this estimator cannot be computed using independence of samples (which is the standard approach in finding the variance of a kernel density estimate for example). Instead the probability of a large error can be estimated using the method in Singer [15] which was generalized to nonuniform sampling in [3]. While this computation is beyond the scope of this paper, the results of [15, 3], as well as the variance of an unnormalized weighted linear regression in [5], suggest that the variance of our estimator should depend only on the dimension d of $\mathcal{M}$ and not the dimension m of the ambient Euclidean space.

In the remainder of this section, we will discuss several consequences of Theorem 3.1 in more details. In particular, we shall see that the scaling law established in this theorem provides systematic methods to identify tangent spaces and estimate derivative $D\mathcal{H}$. In Appendix B, we also show that this scaling law can be used as a guideline to estimate the kernel bandwidth parameter ϵ, which is crucial for accurate numerical approximation.

3.1. Identifying Tangent Spaces with the Singular Value Decomposition

The first method of leveraging Theorem 3.1 is by applying the singular value decomposition (SVD) to the weighted vectors. As we will show below, the singular vectors will naturally be sorted into tangent vectors, with singular values of order $\sqrt{\epsilon}$, and orthogonal vectors, with singular values of order ϵ. To see this we state the following corollary to Theorem 3.1.

Corollary 3.2. *Let x_i be samples from $\mathcal{M} \subset R^m$ with density $p(x)$. Define X to be a matrix with columns $X_j = D(x)^{-1/2}\exp\left(-\frac{\|x_j-x\|^2}{4\epsilon}\right)(x_j - x) = D(x)^{-1/2}dx_j$, where $D(x)$ is defined as in Theorem 3.1. Then,*

$$\lim_{N\to\infty}\frac{1}{\epsilon}XX^\top = \mathcal{I}(x)^\top\mathcal{I}(x) + O(\epsilon). \tag{10}$$

Proof. The proof follows from Theorem 3.1 with $\mathcal{H}(x) = x$ so that $D\mathcal{H}(x) = I_{d\times d}$ and $D\hat{\mathcal{H}}(x) = \mathcal{I}(x)^\top D\mathcal{H}(x)\mathcal{I}(x) = \mathcal{I}(x)^\top\mathcal{I}(x)$. □

Recall that $\mathcal{I}(x) : \mathbb{R}^m \to T_x\mathcal{M}$ is the projection onto the tangent space at x viewed as a subspace of $\mathbb{R}^m$. Corollary 3.2 shows that if $v \in T_x\mathcal{M}$, then $\lim_{N\to\infty} v^\top XX^\top v = \epsilon\|v\|^2 + O(\epsilon^2)$, whereas for $v \in T_x\mathcal{M}^\perp$ we find $\lim_{N\to\infty} v^\top XX^\top v = O(\epsilon^2\|v\|^2)$ (see Appendix A for details). This shows that if v is a singular vector, the associated singular value,

$$\sigma_v = \lim_{N\to\infty}\frac{\sqrt{v^\top XX^\top v}}{\|v\|},$$

will either be order-$\sqrt{\epsilon}$ if v is in the tangent space, or order-ϵ if v is orthogonal to the tangent space. Since the singular value decomposition of X finds v which maximizes σ_v, when ϵ is sufficiently small (and N sufficiently large) the first d (largest) singular values will all be order-$\sqrt{\epsilon}$ and the remaining $m-d$ singular values will be order-ϵ. This fact gives us a way to identify the tangent vectors of the manifold by defining the scaling law, α_l, of a singular value, σ_l, to be the exponential power such that $\sigma_l \propto \epsilon^{\alpha_l}$. When $\alpha_l \approx 1/2$ then the associated singular vector is a tangent vector and when $\alpha_l \geq 1$ then the associated singular vector is orthogonal to $T_x\mathcal{M}$.

For discrete data, this power law will change as a function of the bandwidth parameter ϵ. Numerically, we can estimate this power law by computing $\sigma_l(\epsilon)$ for discrete values ϵ_i and then approximating,

$$\alpha_l = \frac{d\log(\sigma_l)}{d\log(\epsilon)} \approx \frac{\log(\sigma_l(\epsilon_i)) - \log(\sigma_l(\epsilon_{i-1}))}{\log(\epsilon_i) - \log(\epsilon_{i-1})}$$

We now demonstrate this numerically by sampling 10000 points $(\theta_i, \phi_i) \in [0, 2\pi]^2$ from a uniform grid and mapping them onto a torus embedded in $\mathbb{R}^3$ by $(x, y, z)^\top = ((2 + \cos(\theta))\cos(\phi), (2 + \cos(\theta))\sin(\phi), \sin(\theta))^\top$. We chose a point $x = (1.996, 0.126, 1.000)^\top$ and constructed the weighed vectors $X_j = D(x)^{-1/2}\exp\left(-\frac{\|x_j-x\|^2}{4\epsilon}\right)(x_j - x)$ for $\epsilon_l = 2^{-13+l/10}$

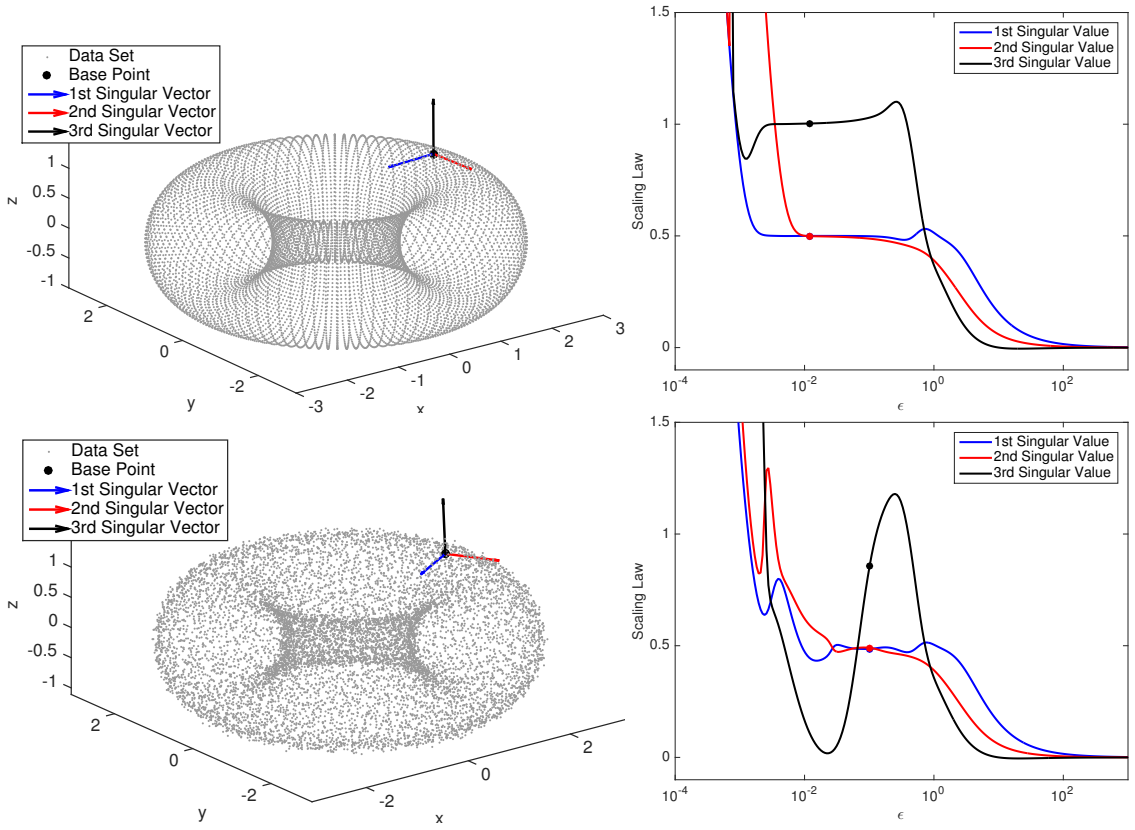

Figure 4: Data set sampled from a Torus embedded in $\mathbb{R}^3$ (top) and with noise added (bottom). Singular vectors are shown (left) that correspond to the optimal choice of ϵ (shown above with the solid dot in the scaling law curves, see Appendix B) based on the empirical scaling laws (right) for the various singular values and the determinant of the weighted vectors X at the base point $(1.996, 0.126, 1.000)^\top$.

where $l = 1, ..., 230$. For each value of ϵ_l we compute the three singular values of X_j and then we compute the scaling laws for each singular value. These scaling laws are shown in Figure 4. We selected the optimal value of ϵ using the method that we will describe in Appendix B, which are highlighted by a solid dot in the scaling law curves, and we plot the associated singular vectors in Figure 4.

To demonstrate the robustness of this methodology to small noise in the ambient space, we repeated the experiment adding a three dimensional Gaussian random perturbation with mean zero and variance $0.04I_{3\times3}$ to each point. In the noisy case, the theoretical scaling laws are obtained for a much smaller range of values of ϵ as shown in Figure 4. In fact, when analyzed at a small scale ($\epsilon < 0.15$) all three singular values have scaling law $\alpha_l \approx 1/2$, which represents the three dimensional nature of the manifold after the addition of the noise. However, the scaling laws also capture the approximate two-dimensional structure, as shown by the scaling law of the third singular vector being very close to 1 for $0.22 < \epsilon < 0.4$. This suggests that the scaling laws are robust for perturbations of magnitude less than ϵ, however, the singular vectors are more sensitive as shown by the slight tilt in the tangent plane defined by the first two singular vectors in Figure 4.

3.2. *Estimating Derivatives with the Linear Regressions*

We now return to the problem of estimating the derivative of a nonlinear mapping $\mathcal{H} : \mathcal{M} \subset \mathbb{R}^m \to \mathcal{N} \subset \mathbb{R}^n$ where we assume that we know the values of $\mathcal{H}$ on our training data set $y_i = \mathcal{H}(x_i)$. As mentioned above, the approach of [4] was to use a linear regression to estimate $D\hat{\mathcal{H}}(x_i)$ as the matrix which minimizes $\sum_j \|dy_j - D\hat{\mathcal{H}}(x_i)dx_j\|^2$. Notice that the linear regression minimizes the residual $r = Y - D\hat{\mathcal{H}}(x_i)X$ by setting $D\hat{\mathcal{H}}(x_i) \equiv YX^\top(XX^\top)^{-1}$ (where the additional factor of $D(x)$ from Theorem 3.1 cancels making this equivalent to the approach of [4]).

For convenience in this section we will assume that $XX^\top$ is invertible, which will generally be the case for embeddings with extrinsic curvature. For certain 'flat' embeddings (for example a cylinder which is flat in one direction) it is possible that $XX^\top$ will not be full rank, and in this case we define the inverse of $XX^\top$ by inverting the restriction to the non-zero singular directions and we define $(XX^\top)^{-1}$ to be the identity in the singular directions. Numerically, the standard least squares regression algorithms which are used to estimate $D\hat{\mathcal{H}}(x_i)$ will use the singular value decompositions rather than explicitly forming and inverting $XX^\top$, so singularity should not produce any issues in practice.

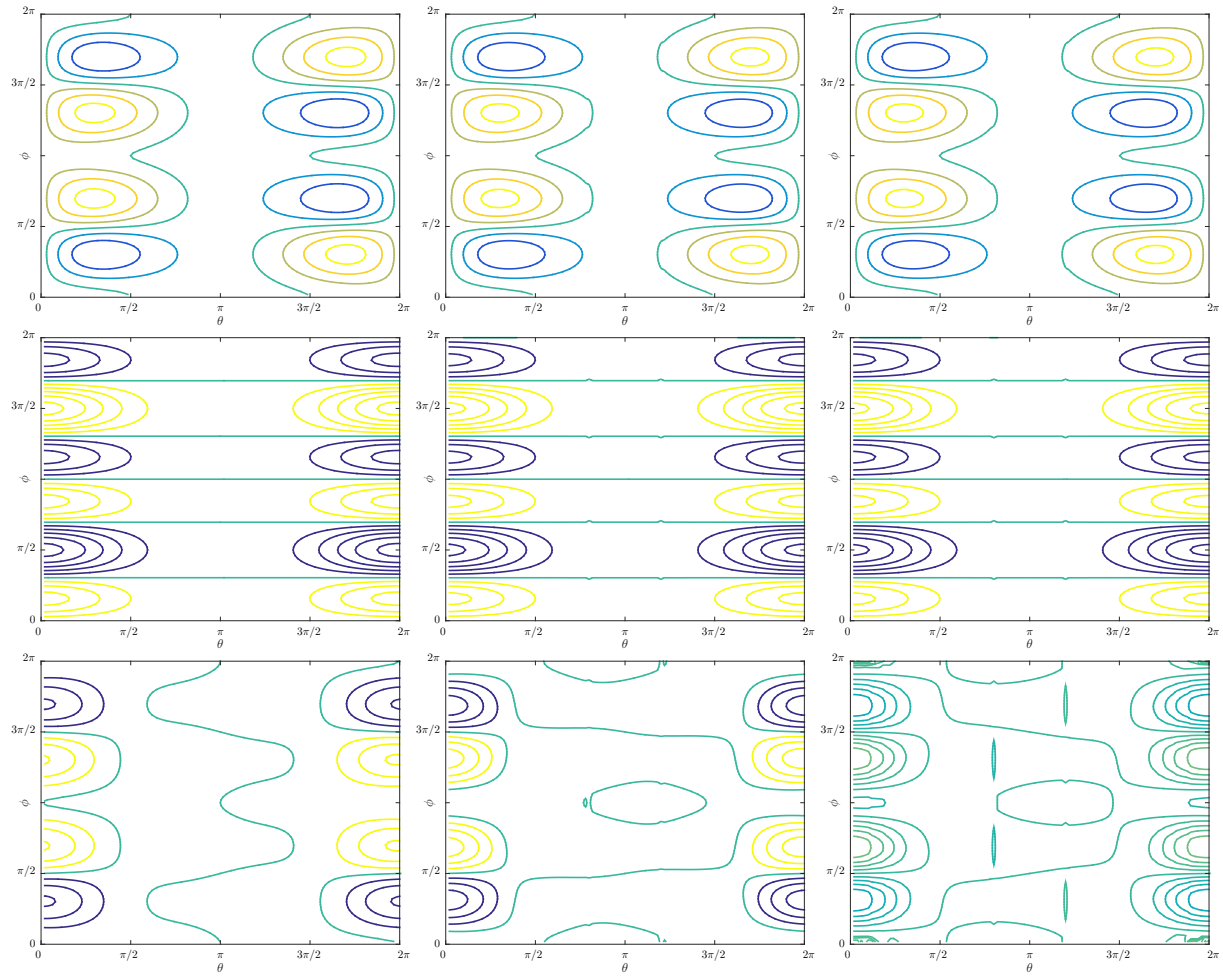

Figure 5: Contour plots of derivatives $D\hat{\mathcal{H}}(x,y,z)\frac{d(x,y,z)}{d\theta}$ (top), $D\hat{\mathcal{H}}(x,y,z)\frac{d(x,y,z)}{d\phi}$ (middle), and $D\hat{\mathcal{H}}(x,y,z)\left(\frac{d(x,y,z)}{d\theta}\times\frac{d(x,y,z)}{d\phi}\right)$ (bottom) are shown for the analytical computation (first column), regression estimate (second column) and covariance estimate (third column).

Theorem 3.1 showed that a simple method of estimating the derivative $D\hat{\mathcal{H}}(x)$ is with the correlation matrix $\frac{1}{\epsilon}YX^\top$. We now show that a related estimator of this derivative is the linear regression $YX^\top(XX^\top)^{-1}$. From Corollary 3.2 we have $\lim_{N\to\infty} XX^\top = \epsilon\mathcal{I}(x)^\top\mathcal{I}(x) + O(\epsilon^2)$, which implies that in the limit of large data,

$$\lim_{N\to\infty}(XX^\top)^{-1} = \frac{1}{\epsilon}((\mathcal{I}(x)^\top\mathcal{I}(x))^\dagger + O(\epsilon)),$$

where † denotes the pseudo-inverse. Combining this equation and Theorem 3.1 we have,

$$\begin{aligned}\lim_{N\to\infty} YX^\top(XX^\top)^{-1} &= (D\hat{\mathcal{H}}(x) + O(\epsilon))((\mathcal{I}(x)^\top \mathcal{I}(x))^\dagger + O(\epsilon)) \\ &= D\hat{\mathcal{H}}(x)(\mathcal{I}(x)^\top \mathcal{I}(x))^\dagger + O(\epsilon) \\ &= \mathcal{I}_{\mathcal{N}}(\mathcal{H}(x))^\top D\mathcal{H}(x)\mathcal{I}(x)(\mathcal{I}(x)^\top \mathcal{I}(x))^\dagger + O(\epsilon) \\ &= \mathcal{I}_{\mathcal{N}}(\mathcal{H}(x))^\top D\mathcal{H}(x)(\mathcal{I}(x)^\top)^\dagger + O(\epsilon).\end{aligned}$$

This implies that the regression based estimate of $D\mathcal{H}(x)$ can have large errors in directions orthogonal to the tangent space $T_x\mathcal{M}$. However, these large errors are not important when $D\mathcal{H}(x)$ is used in constructing a local kernel, since the local kernel construction only depends on the projection of $D\mathcal{H}(x)$ onto the tangent space.

Note that the columns in Y require the value of $y = \mathcal{H}(x)$, which is assumed to be known in the training data set, but will not be known if we wish to extend the map $D\hat{\mathcal{H}}$ to a new point x^*. This is particularly important for the Nyström extension described in Section 2.2 which requires evaluating the local kernel on the new data point. However, even when $y^* \equiv \mathcal{H}(x^*)$ is unknown, we can still estimate $D\hat{\mathcal{H}}(x^*)$ by converting from a linear regression to an affine regression, which will implicitly estimate a weighted linear regression for y^*. More explicitly, we are unable to construct

$$Y_j = D(x^*)^{-1/2} \exp\left(-\|x_j - x^*\|/\epsilon\right)(y_j - y^*)$$

since y^* is unknown. Notice that we can write the matrix $Y = \hat{Y} - y^*W$ where $\hat{Y}_j = D(x^*)^{-1/2} \exp\left(-\|x_j - x^*\|/\epsilon\right) y_j$ is a known $n \times N$ matrix and $W_j \equiv D(x^*)^{-1/2} \exp\left(-\|x_j - x^*\|/\epsilon\right)$ is a known $1 \times N$ vector, and only the $n \times 1$ vector y^* is unknown. The linear regression described above finds $D\hat{\mathcal{H}}(x^*) = YX^\dagger$ so that $Y \approx D\hat{\mathcal{H}}(x^*)X$. This linear regression will not produce an exact equality in general due to the fact that the relationship between the coordinates Y and X is only asymptotically linear as shown in Theorem 3.1. Notice that substituting $Y = \hat{Y} - y^*W$ we can rewrite this approximation as

$$\hat{Y} \approx D\hat{\mathcal{H}}(x^*)X + y^*W = (D\hat{\mathcal{H}}(x^*) \;\; y^*)\begin{pmatrix} X \\ W \end{pmatrix}$$

which shows that we can estimate $D\hat{\mathcal{H}}(x^*)$ and y^* simultaneously with the affine regression

$$(D\hat{\mathcal{H}}(x^*) \;\; y^*) = \hat{Y}\begin{pmatrix} X \\ W \end{pmatrix}^\dagger.$$

We emphasize that the estimate of y^* from the affine regression is not used directly, it is only included to properly center the regression so that we can estimate $D\hat{\mathcal{H}}(x^*)$. The key point here is that the unknown matrix Y depends linearly on the unknown value y^* (the only nonlinear dependence is on the known value x^*).

We demonstrate this method of estimating derivatives by defining the function,

$$\hat{\mathcal{H}}(x, y, z) = xy^2 + z,$$

which restricted to the torus can be written in the coordinates (θ, ϕ) as,

$$\hat{\mathcal{H}}(x, y, z) = \mathcal{H}(\theta, \phi) = (2 + \cos(\theta))^3 \cos(\phi) \sin^2(\phi) + \sin(\theta).$$

We evaluate $\hat{\mathcal{H}}$ on the data set lying exactly on the torus example in Section 3.1. We will evaluate the derivative $D\hat{\mathcal{H}}(x, y, z) = (y^2, x, 1)$ by projecting onto the two tangent directions $\frac{d(x,y,z)}{d\theta}$ and $\frac{d(x,y,z)}{d\phi}$ and the orthogonal direction $\frac{d(x,y,z)}{d\theta} \times \frac{d(x,y,z)}{d\phi}$. In Figure 5, we compare the contour plot of the analytical derivatives (first column) to the corresponding estimates obtained by the linear regression (second column) and the covariance matrix (third column). Notice that the correlation matrix estimate $\frac{1}{\epsilon} YX^\top \approx D\hat{\mathcal{H}}$ is approximately zero when projected in the direction orthogonal to the tangent plane, whereas the linear regression estimate $YX^\top(XX^\top)^{-1}$ recovers the analytic derivative even in this orthogonal direction. We re-emphasize that when used in a local kernel, the behavior in the orthogonal direction is irrelevant to the limiting operator. A possible reason that the linear regression, $YX^\top(XX^\top)^{-1}$, gives better results than the correlation estimate, $\frac{1}{\epsilon} YX^\top$, is that the errors arising from the approximation of the continuous integrals by the finite summations in $YX^\top$ and $XX^\top$ may be correlated, similar to the result found in [15].

4. Iterated Diffusion Map (IDM)

In this section we consider representing general differentiable maps $\mathcal{H}$ that can take data in high-dimensional spaces to lower-dimensional spaces, generalizing the result in [4] that was reviewed in Section 2.2. In particular, we will make use of Theorem 2.1 to find an isometric embedding of $\mathcal{M}$ with respect to an appropriate geometry such that these new embedded coordinates emphasize the feature of interest $\mathcal{H}(\mathcal{M}) = \mathcal{N}$. In analogy to the diagram in Section 2.2, we shall see that the proposed method represents $\mathcal{H}$ with a linear map between the iterated diffusion mapping of the data manifold $\mathcal{M}$ and the rescaled diffusion coordinates of the feature space $\mathcal{N}$.

One of the challenges is that the result in [4] is not immediately applicable since $\mathcal{H}$ is not assumed to be a diffeomorphism, and therefore the kernel constructed in (7) from $D\mathcal{H}$ is not necessarily a local kernel. To see this, we can define a covariance matrix $C(x)^{-1} = D\hat{\mathcal{H}}(x)^\top D\hat{\mathcal{H}}(x)$, where $D\hat{\mathcal{H}}(x)$ is the local derivative in the ambient space estimated by linear regression as discussed in Section 3.2. If we naively form the kernel $K(\epsilon, x, y)$ from (3) with covariance matrix $C(x)$, then this will not be a local kernel. The problem is that the restriction of $C(x)^{-1}$ to the tangent plane, $c(x)^{-1} = \mathcal{I}(x)C(x)^{-1}\mathcal{I}(x)^\top = D\mathcal{H}(x)^\top D\mathcal{H}(x)$, may not be full rank since the map $\mathcal{H}$ may take the manifold $\mathcal{M}$ to a lower-dimensional manifold $\mathcal{H}(\mathcal{M})$. If $c(x)^{-1}$ is not full rank, then there exists a nontrivial vector $u \in T_x\mathcal{M}$ such that $u^\top c(x)^{-1}u = 0$ (in fact $c(x)^{-1}u = 0$), so if $y - x = (u, q(u))$ we find $K(\epsilon, x, y) = O(1)$, which means that K does not have the exponential decay, so K is not a local kernel (see Section 2.2 and [4]).

Often the kernel K is constructed using the k nearest neighbors, so that $K(\epsilon, x, y) \equiv 0$ by definition when y is not in the list of the k nearest neighbors of x, and vice-versa. When the k nearest neighbor algorithm is used, technically the kernel K constructed with a rank deficient covariance matrix is still a local kernel since the kernel still has an implicit decay that can be bounded above by an exponential function. This is a discrete effect caused by the k nearest neighbors cutoff. However, the localization caused by the k nearest neighbor algorithm has a very sharp cutoff such that the corresponding operator approximated by the kernel is very sensitive to the choice of k.

In order to use the local kernels theory to represent the feature map $\mathcal{H}$, we propose a novel algorithm called the iterated diffusion map (IDM). The IDM will make use of local kernels which use small perturbations of identity covariance matrices such that Theorem 2.1 is applicable on each iteration. In Section 4.1, we present the IDM and show that it is a discrete approximation of an intrinsic geometric flow. In Section 4.2, we show that if the data space $\mathcal{M}$ is a product of the feature space and the irrelevant space, then IDM will produce a quotient manifold that is isometric to the feature space, eliminating the irrelevant dimension. Finally, we will show numerical results with IDM in Section 4.3 and a nontrivial application. The numerical algorithm of the IDM is outlined in Appendix C.

4.1. IDM as an Intrinsic Geometric Flow

We now introduce the IDM algorithm for feature identification. The method assumes the availability of a pair of data sets $x_i \in \mathcal{M} \subset \mathbb{R}^m$ and $y_i = \mathcal{H}(x_i) \in \mathcal{N} \subset \mathbb{R}^n$, where $\mathcal{H}$ is not assumed to be a diffeomorphism and $\mathcal{N}$ may even be lower dimension than $\mathcal{M}$. With this training data, we apply the linear regression method in Section 3.2 to approximate the local derivative $D\hat{\mathcal{H}}$ in the ambient space which is subsequently used to define a new covariance,

$$C_{\mathcal{H}^{(0)}}(x) = \left((1-\tau)\mathrm{I}_{m\times m} + \tau D\hat{\mathcal{H}}(x)^\top D\hat{\mathcal{H}}(x)\right)^{-1}, \tag{11}$$

where $\mathrm{I}_{m\times m}$ is the $m \times m$ identity matrix. The use of subscript '(0)' will become clear below.

With this construction, $C_{\mathcal{H}^{(0)}}(x)$ is guaranteed to be positive definite, even when $D\mathcal{H}(x)^\top D\mathcal{H}(x)$ is not a full rank matrix (where the relation of $D\mathcal{H}$ and $D\hat{\mathcal{H}}$ is defined in (6)). With the definition in (11), we implicitly define a map $\mathcal{G} : \mathcal{M} \to \mathcal{M}$ such that $D\mathcal{G}(x)^\top D\mathcal{G}(x) = C_{\mathcal{H}^{(0)}}(x)^{-1}$. Notice that the matrix $D\mathcal{G}(x)$ always exists as the matrix square root of the symmetric and positive definite matrix $C_{\mathcal{H}^{(0)}}(x)^{-1}$ and the square root can be chosen to vary smoothly with x since the covariance matrix is always full rank and also varies smoothly with x. Since the matrices $D\mathcal{G}(x)$ depend smoothly on x and are always full rank they define a diffeomorphism $\mathcal{G}$ by the implicit function theorem such that the Jacobian of $\mathcal{G}$ at each x is exactly $D\mathcal{G}(x)$. When $\tau \ll 1$, intuitively, $\mathcal{G}$ is a small perturbation of an identity map on $\mathcal{M}$ since, by asymptotic expansion,

$$D\mathcal{G}(x) = \sqrt{C_{\mathcal{H}^{(0)}}(x)^{-1}} = \sqrt{(1-\tau)\mathrm{I}_{m\times m} + \tau D\hat{\mathcal{H}}(x)^\top D\hat{\mathcal{H}}(x)} = \mathrm{I}_{m\times m} - \frac{1}{2}\tau\left(D\hat{\mathcal{H}}(x)^\top D\hat{\mathcal{H}}(x) - \mathrm{I}_{m\times m}\right) + O(\tau^2).$$

Unlike the sharp decay due to the k nearest neighbor cutoff, a kernel (3) constructed using $C_{\mathcal{H}^{(0)}}(x)$ achieves a smooth decay even in directions where $D\mathcal{H}(x)D\mathcal{H}(x)^\top$ is rank deficient, this is because $C_{\mathcal{H}^{(0)}}(x)$ is always full rank. Using the prototypical kernel,

$$K(\epsilon, x, y) = \exp\left(-(y-x)^\top C_{\mathcal{H}^{(0)}}(x)^{-1}(y-x)/2\right),$$

along with the construction in Theorem 2.1, we approximate the operator $\Delta_{g_{\mathcal{H}^{(0)}}}$ which is the Laplace-Beltrami operator with respect to the Riemannian metric,

$$g_{\mathcal{H}^{(0)}} = c_{\mathcal{H}^{(0)}}^{-1/2} g_{\mathcal{M}} c_{\mathcal{H}^{(0)}}^{-1/2} = ((1-\tau)\mathrm{I}_{d\times d} + \tau D\mathcal{H}^\top D\mathcal{H})^{1/2} g_{\mathcal{M}} ((1-\tau)\mathrm{I}_{d\times d} + D\mathcal{H}^\top D\mathcal{H})^{1/2}, \tag{12}$$

where $c_{\mathcal{H}^{(0)}}(x) = \mathcal{I}(x) C_{\mathcal{H}^{(0)}}(x) \mathcal{I}(x)^\top$. Notice that if we build a diffusion map $\Phi_s^{(0)}(x) = (e^{s\lambda_1}\varphi_1(x), ..., e^{s\lambda_M}\varphi_M(x))^\top \equiv x^{(1)}$ using the eigenfunctions of $\Delta_{g_{\mathcal{H}^{(0)}}}$ (approximated by the local kernel construction) this gives an approximately isometric embedding of $\mathcal{M}$ with respect to the metric $g_{\mathcal{H}^{(0)}}$, for small enough parameter s. Moreover, the new metric $g_{\mathcal{H}^{(0)}}$ in (12) puts a larger weight on directions in which $D\mathcal{H}$ are large, which are the direction associated with the range space of $\mathcal{H}$.

The key point that makes the iterated diffusion map useful is that the local kernels with covariance defined below (cf. (15)), change the geometry, as opposed to iterating the diffusion maps using identity covariance, $C(x) = I_{m\times m}$, as discussed in Section 2.1. In particular, the ℓ-th iteration is performed on the coordinate

$$x^{(\ell-1)} = \Phi_s^{(\ell-2)}(x^{(\ell-2)}), \quad \ell = 2, 3, \ldots, \tag{13}$$

where $x^{(0)} \equiv x$, with induced feature maps $\mathcal{H}^{(\ell-1)} : \mathbb{R}^M \to \mathcal{N}$ defined as follows,

$$\mathcal{H}^{(\ell-1)}(x^{(\ell-1)}) \equiv \mathcal{H}(x), \quad \ell = 2, 3, \ldots \tag{14}$$

which simply says that as a point is mapped through the IDM, the value of the feature map is always the original feature values for that data point. Numerically, we approximate the local derivative $D\hat{\mathcal{H}}^{(\ell-1)}$ of $\mathcal{H}^{(\ell-1)}$ in the ambient space by the linear regression method in Section 3.2. In this particular implementation, $X_j^{(\ell-1)}$ and $Y_j^{(\ell-1)}$ in Theorem 3.1 are defined as,

$$X_j^{(\ell-1)} = D(x^{(\ell-1)})^{-1/2} \exp\left(-\frac{\|x_j^{(\ell-1)} - x^{(\ell-1)}\|^2}{4\epsilon}\right)(x_j^{(\ell-1)} - x^{(\ell-1)}),$$
$$Y_j^{(\ell-1)} = D(x^{(\ell-1)})^{-1/2} \exp\left(-\frac{\|x_j^{(\ell-1)} - x^{(\ell-1)}\|^2}{4\epsilon}\right)(y_j - y),$$

where $x_j^{(\ell-1)} := (\Phi_s^{(\ell-2)} \circ \Phi_s^{(\ell-3)} \circ \ldots \circ \Phi_s^{(0)})(x_j)$ for $\ell \geq 2$. Given $D\hat{\mathcal{H}}^{(\ell-1)}$, we define local kernels induced by covariance matrices,

$$C_{\mathcal{H}^{(\ell-1)}}(x^{(\ell-1)}) = \left((1-\tau)\mathrm{I}_{m\times m} + \tau D\hat{\mathcal{H}}^{(\ell-1)}(x^{(\ell-1)})^\top D\hat{\mathcal{H}}^{(\ell-1)}(x^{(\ell-1)})\right)^{-1}, \quad \ell = 2, 3, \ldots. \tag{15}$$

We can now repeat the local kernel construction above using the covariance $C_{\mathcal{H}^{(\ell-1)}}(x^{(\ell-1)})$ to produce eigenfunctions and eigenvalues of $\Delta_{g_{\mathcal{H}^{(\ell-1)}}}$ and obtain $x^{(\ell)} = \Phi_s^{(\ell-1)}(x^{(\ell-1)})$. For s sufficiently small, the new coordinates $x^{(\ell)}$ will be an approximately isometric embedding of $\mathcal{M}$ with respect to the metric,

$$g_{\mathcal{H}^{(\ell-1)}} = c_{\mathcal{H}^{(\ell-1)}}^{-1/2} g_{\mathcal{H}^{(\ell-1)}} c_{\mathcal{H}^{(\ell-1)}}^{-1/2} = c_{\mathcal{H}^{(\ell-1)}}^{-1/2} \cdots c_{\mathcal{H}^{(0)}}^{-1/2} g_{\mathcal{M}}\, c_{\mathcal{H}^{(0)}}^{-1/2} \cdots c_{\mathcal{H}^{(\ell-1)}}^{-1/2}, \tag{16}$$

where $c_{\mathcal{H}^{(j)}}(x) = \mathcal{I}(x) C_{\mathcal{H}^{(j)}}(x) \mathcal{I}(x)^\top$ for $j = 0, \ldots, \ell-1$. Intuitively speaking, each iteration of the diffusion map further emphasizes the directions on the manifold $\mathcal{M}$, which are important to the function $\mathcal{H}$ (this will be made more rigorous in Section 4.2 for certain classes of manifolds and features).

As further motivation for the IDM construction, we note that the map $\mathcal{H}(x)$ is a fixed point of the iterated diffusion map process. To see this, assume that for some k we have $x^{(k)} = \mathcal{H}(x)$, then we find

$$\mathcal{H}^{(k)}(\mathcal{H}(x)) = \mathcal{H}^{(k)}(x^{(k)}) = \mathcal{H}(x),$$

so $D\mathcal{H}^{(k)}D\mathcal{H} = D\mathcal{H}$ which implies that $D\mathcal{H}^{(k)}$ acts as the identity on the range of $D\mathcal{H}$, which is the tangent space, $T_{\mathcal{H}(x)}\mathcal{N}$. On the other hand, we can deduce that,

$$\begin{aligned} c^{-1}_{\mathcal{H}^{(k)}}(x^{(k)}) &= \mathcal{I}(x^{(k)})C^{-1}_{\mathcal{H}^{(k)}}(x^{(k)})\mathcal{I}(x^{(k)})^\top \\ &= \mathcal{I}(x^{(k)})\left((1-\tau)\mathrm{I}_{m\times m} + \tau D\hat{\mathcal{H}}^{(k)}(x^{(k)})^\top D\hat{\mathcal{H}}^{(k)}(x^{(k)})\right)\mathcal{I}(x^{(k)})^\top \\ &= (1-\tau)\mathrm{I}_{d\times d} + \tau D\mathcal{H}^{(k)}(x^{(k)})^\top D\mathcal{H}^{(k)}(x^{(k)}), \end{aligned} \tag{17}$$

where we have used (15), equality $\mathcal{I}\mathcal{I}^\top = I_{d\times d}$, and the definition of $D\mathcal{H}$ as the restriction of $D\hat{\mathcal{H}}$ in the tangent space. This means that for any $y \in T_{\mathcal{H}(x)}\mathcal{N}$,

$$y^\top c^{-1}_{\mathcal{H}^{(k)}} y = (1-\tau)y^\top y + \tau\, y^\top D\mathcal{H}^{(k)}(x^{(k)})^\top D\mathcal{H}^{(k)}(x^{(k)})y = (1-\tau)y^\top y + \tau y^\top y = y^\top y,$$

since $D\mathcal{H}^{(k)}y = y$. Since $c^{-1}_{\mathcal{H}^{(k)}} = I_{d\times d}$, it is clear that $g_{\mathcal{H}^{(k+1)}} = g_{\mathcal{H}^{(k)}}$ and therefore $x^{(k+1)} = x^{(k)} = \mathcal{H}(x)$. Similarly, any isometric embedding $\iota_{\mathcal{N}}(\mathcal{H}(x))$ is a fixed point of the iterated diffusion map. To see this, assume $x^{(k)} = \iota_{\mathcal{N}}(\mathcal{H}(x))$ and note that $\mathcal{H}^{(k)}(\iota_{\mathcal{N}}(\mathcal{H}(x))) = \mathcal{H}(x)$ so that $D\mathcal{H}^{(k)}D\iota_{\mathcal{N}}D\mathcal{H} = D\mathcal{H}$. Since $\iota_{\mathcal{N}}$ is an isometric embedding, we have $D\iota_{\mathcal{N}}$ acts an orthogonal matrix in the tangent space which implies that $D\mathcal{H}^{(k)}$ acts as an ortthogonal transformation on the range of $D\mathcal{H}$ and similar argument follows as above. It remains an open question whether this fixed point is attracting in the general case, and further analysis is needed to understand this issue.

One possible interpretation of the IDM is as a discretization of a geometric flow. Motivated by (17), we define,

$$c(x, t+\tau) = \left((1-\tau)\mathrm{I}_{d\times d} + \tau D\mathcal{H}(x(t))^\top D\mathcal{H}(x(t))\right)^{-1},$$

as a continuous analog of (15), where $x(t)$ is an isometric embedding of $(\mathcal{M}, g(t))$ where $g(0) = g_{\mathcal{M}}$ and with a feature map defined continuously $\mathcal{H}(x(t)) = \mathcal{H}(x(0))$, where $x(0) = x$ to mimic the discrete setting in (14). The new metric introduced by $c(x, t+\tau)$ would be,

$$\begin{aligned} g(t+\tau) &= c(x,t+\tau)^{-1/2} g(t) c(x,t+\tau)^{-1/2} \\ &= (\mathrm{I} + \tau(D\mathcal{H}^\top D\mathcal{H} - \mathrm{I}))^{1/2} g(t) (\mathrm{I} + \tau(D\mathcal{H}^\top D\mathcal{H} - \mathrm{I}))^{1/2} \\ &= g(t) + \frac{\tau}{2}\Big(D\mathcal{H}^\top D\mathcal{H} g(t) + g(t) D\mathcal{H}^\top D\mathcal{H} - 2g(t)\Big) + O(\tau^2). \end{aligned} \tag{18}$$

Rewriting the previous equation we find,

$$\frac{dg}{dt} = \lim_{\tau\to 0} \frac{g(t+\tau) - g(t)}{\tau} = -g + \frac{1}{2}\Big(D\mathcal{H}^\top D\mathcal{H} g + g D\mathcal{H}^\top D\mathcal{H}\Big), \tag{19}$$

which is an equation describing an intrinsic geometric flow. Notice again that if $D\mathcal{H} = I$, then g is an equilibrium solution of (19). We should note that the geometric flow in (19) is nonlinear since the map $\mathcal{H}$ depends on the metric g in nontrivial fashion (since $\mathcal{H}$ maps an isometric embedding of $(\mathcal{M}, g(t))$ to the feature of interest in $\mathcal{H}(\mathcal{M})$). This is the reason why it is not straightforward to see whether there are other equilibrium solutions or even to determine the stability of any equilibrium solution. The standard linear stability analysis suggests that if g^* is the fixed point of (19), then g^* is locally attracting when the real part of all of the eigenvalues of the linearized operator $D_g\Big(D\mathcal{H}^\top D\mathcal{H} g + g D\mathcal{H}^\top D\mathcal{H}\Big)|_{g=g^*}$ is less than 2.

4.2. IDM for Product Manifolds

In the previous section we introduced the IDM as an approximation of an intrinsic geometric flow. While we are not able to describe the limit of this flow in every case, in this section we will derive the limit for manifolds and features that have a simple structure. In particular we will consider manifolds $\mathcal{M}$ which are isometric embeddings of product manifolds where the feature map is simply selecting an diffeomorphic copy of one of the features.

For clarity, we first consider the simplest case where the manifold $\mathcal{M}$ is exactly a product space $\mathcal{M} = \mathcal{N} \times \mathcal{P}$, such that $\mathcal{N}$ is the feature space and $\mathcal{P}$ contains variables we wish to ignore. In this case, the map $\mathcal{H} : \mathcal{M} \to \mathcal{N}$ has a particularly simple structure. In each local neighborhood we can find coordinates $x = (y, z) \in \mathcal{M}$ where y

are coordinates on $\mathcal{N}$ and z are coordinates on $\mathcal{P}$. In these coordinates the metric g will naturally decompose into a block diagonal matrix. The first block represents the metric $g_{\mathcal{N}}$ on $\mathcal{N}$ and this block is $d_{\mathcal{N}} \times d_{\mathcal{N}}$ and the second block represents the metric $g_{\mathcal{P}}$ on $\mathcal{P}$ and this block is $d_{\mathcal{P}} \times d_{\mathcal{P}}$. Since $\mathcal{H}(\mathcal{M}) = \mathcal{N}$ maps each point to the feature of interest, in these local coordinates, the feature map will take the form $\mathcal{H}(x) = \mathcal{H}(y, z) = y$. Moreover, in these coordinates, $D\mathcal{H}(x)$ is a block diagonal matrix where the first $d_{\mathcal{N}} \times d_{\mathcal{N}}$ submatrix is the identity matrix and the remaining entries are all zero. So we find that $D\mathcal{H}^\top D\mathcal{H} - I$ is again a block diagonal matrix, where the bottom $d_{\mathcal{P}} \times d_{\mathcal{P}}$ block is equal to $-I$ and the remaining entries are zero. Writing the geometric flow (19) in these coordinates we find,

$$\dot{g} = \frac{1}{2}\left((D\mathcal{H}^\top D\mathcal{H} - I)g + g(D\mathcal{H}^\top D\mathcal{H} - I)\right) = \frac{1}{2}\left(\begin{pmatrix} 0 & 0 \\ 0 & -I \end{pmatrix}\begin{pmatrix} g_{\mathcal{N}} & 0 \\ 0 & g_{\mathcal{P}} \end{pmatrix} + \begin{pmatrix} g_{\mathcal{N}} & 0 \\ 0 & g_{\mathcal{P}} \end{pmatrix}\begin{pmatrix} 0 & 0 \\ 0 & -I \end{pmatrix}\right), \tag{20}$$

which implies that $\dot{g}_{\mathcal{N}} = 0$ and $\dot{g}_{\mathcal{P}} = -g_{\mathcal{P}}$. This shows that for product manifolds the geometric flow (19) will contract the irrelevant variables to zero and leave the features of interest unchanged. So for sufficiently small discretization τ and in the limit of sufficiently many iterations, the IDM will construct the quotient map from the product manifold to an isometric copy of the feature space. At this point, the data can easily be mapped to the feature space using the method of Section 2.2. In fact, since the quotient manifold is already isometric to the feature space, one could simply estimate a linear map between the rescaled diffusion coordinates of the quotient manifold and those of the feature space (since these coordinates are canonical up to rotation as shown in Section 2.1). In analogy to the diagram in Section 2.2 which represents a diffeomorphism, we can summarize the IDM construction of the quotient map with the following diagram,

$$\begin{array}{ccc} \mathcal{M} = \mathcal{N} \times \mathcal{P} & \xrightarrow{\quad\mathcal{H}\quad} & \mathcal{N} \\ \Big\downarrow{\scriptstyle \Psi \equiv \lim_{\ell\to\infty, s\to 0} \Phi_s^{(\ell)} \circ \cdots \circ \Phi_s^{(0)}} & & \Big\downarrow{\scriptstyle \tilde{\Phi}} \\ L^2(\mathcal{N}, \tilde{g}) \approx \mathbb{R}^M & \xrightarrow{\quad H\quad} & L^2(\mathcal{N}, g_{\mathcal{N}}) \approx \mathbb{R}^M \end{array}$$

where Ψ represents the iterated diffusion map, $\tilde{\Phi}$ are the rescaled diffusion coordinates of $\mathcal{N}$. The above diagram shows how $\mathcal{H}$ is represented by an orthogonal linear transformation H via $\mathcal{H} = \tilde{\Phi}^{-1} \circ H \circ \Psi$.

Next, consider the case when $\mathcal{M} = \mathcal{N} \times \mathcal{P}$, but the feature of interest is $\mathcal{F}(\mathcal{N})$, where $\mathcal{F}$ is a diffeomorphism. In this case, the block diagonal structure $D\mathcal{H}$ and of (20) will still hold, and in particular we still find $\dot{g}_{\mathcal{P}} = -g_{\mathcal{P}}$. This shows that the flow still contracts the irrelevant variables $\mathcal{P}$ to zero, and the only difference is that we will find $\dot{g}_{\mathcal{N}} = \frac{1}{2}\left((D\mathcal{F}^\top D\mathcal{F} - I)g_{\mathcal{N}} + g_{\mathcal{N}}(D\mathcal{F}^\top D\mathcal{F} - I)\right)$. Notice that the fixed point for this flow satisfies $D\mathcal{F} = I$, so we expect in the limit to obtain an isometric copy of $\mathcal{N}$. However, even if the flow on $g_{\mathcal{N}}$ has not converged, once the IDM has contracted the irrelevant variables $\mathcal{P}$, we can use the construction in Theorem 2.1 to represent the final diffeomorphism between $\Psi(\mathcal{M})$ and the feature space $\mathcal{F}(\mathcal{N})$. In the next section we demonstrate the IDM on two product manifolds, namely the annulus and the torus. We will also attempt to apply the IDM to manifolds which are not product manifolds and report the empirical results.

Finally, in practice the embedded manifold $\mathcal{M}$ is rarely exactly a product space, and many of the examples considered below are isometric embeddings of product manifolds so that $\mathcal{M} = \iota(\mathcal{N} \times \mathcal{P})$ where $D\iota(y, z)$ is an orthogonal matrix for each $(y, z) \in \mathcal{N} \times \mathcal{P}$. In this case the feature map may be a diffeomorphism $\mathcal{F}$ applied to the variables $\mathcal{N}$ which are obtained by inverting the isometry ι, namely

$$\mathcal{H} = \mathcal{F}(y) = \mathcal{F} \circ \pi_{\mathcal{N}} \circ \iota^{-1}(x)$$

where $\pi_{\mathcal{N}} : \mathcal{N} \times \mathcal{P} \to \mathcal{N}$ is the projection map. In this case we find that $D\mathcal{H} = D\mathcal{F} D\pi_{\mathcal{N}} D\iota^{-1}$ and since $D\iota$ is an orthogonal matrix we have

$$\dot{g} = \frac{1}{2}\left(D\iota^{-\top}(D\pi_{\mathcal{N}}^\top D\mathcal{F}^\top D\mathcal{F} D\pi_{\mathcal{N}} - I)D\iota^{-1}g + gD\iota^{-\top}(D\pi_{\mathcal{N}}^\top D\mathcal{F}^\top D\mathcal{F} D\pi_{\mathcal{N}} - I)D\iota^{-1}\right).$$

We can then make the change of variables $\hat{g} = D\iota^{-1} g D\iota^{-\top}$ so that the geometric flow (20) becomes

$$\dot{\hat{g}} = \frac{1}{2}\left((D\pi_{\mathcal{N}}^\top D\mathcal{F}^\top D\mathcal{F} D\pi_{\mathcal{N}} - I)\hat{g} + \hat{g}(D\pi_{\mathcal{N}}^\top D\mathcal{F}^\top D\mathcal{F} D\pi_{\mathcal{N}} - I)\right)$$

and in these coordinates we recover the splitting of (20). This shows that the IDM also recovers the quotient map in the limit when the manifold $\mathcal{M}$ is an isometric embedding of a product manifold and the feature map is a diffeomorphism of one of the quotient spaces.

4.3. Examples

In this section we will demonstrate how the iterated diffusion map is able to contract a manifold onto a lower-dimensional feature of interest. All the examples use $M = 250$ rescaled diffusion coordinates. We found the results to be robust down to around $M = 100$ rescaled diffusion coordinates and no improvement above $M = 250$. In the examples below we adjusted the parameter $\tau \in (0, 1)$, which defines the discretization of the geometric flow in Section 4.1, in order to achieve the desired feature in about four iterations of the diffusion map. In principle, one would like to take τ as small a possible, however this requires many iterations that are computationally intensive. Also, we have found that numerical errors can accumulate over large numbers of iterations, which we discuss in the Section 5. For a compact description of the numerical algorithm, see Appendix C.

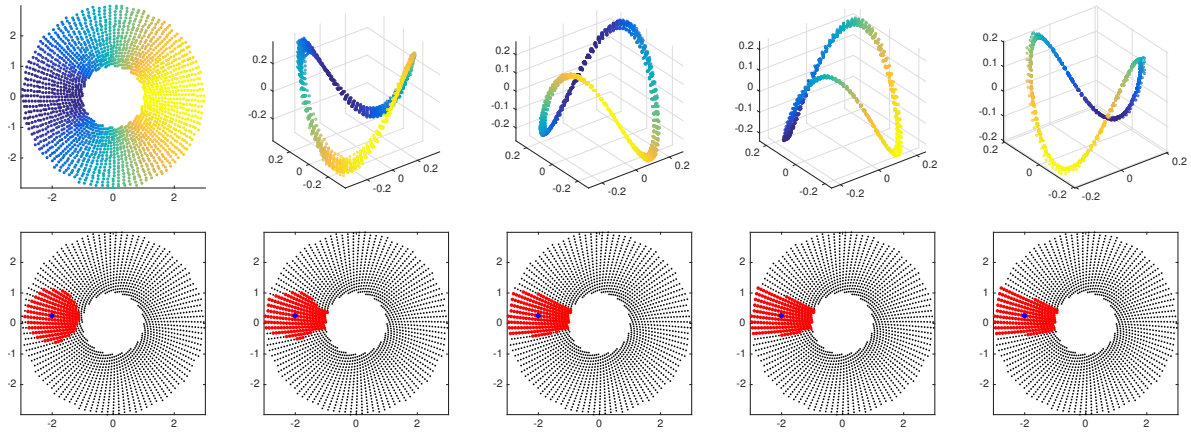

Figure 6: Top: Original annulus data set colored according to the feature of interest (leftmost), followed by four iterations of the diffusion map using the local kernel defined in Section 4 with $\tau = 0.3$. Each diffusion map shows the first three rescaled diffusion coordinates colored according to the feature of interest (the angle of the data point in the original annulus). However, 250 rescaled diffusion coordinates are maintained at each step. Bottom: Original data set showing the 200 nearest neighbors (red) of the blue data point, where the neighbors are found in the corresponding iterated diffusion map embedding.

The first example is the annulus described in Section 1 which used $\tau = 0.65$. The annulus is a product space $A = S^1 \times [1, 3]$ and in Figure 1 we show the iterated diffusion map recovering the radial component. If we parameterize the annulus with polar coordinates $(\theta, r) \in [0, 2\pi) \times [1, 3]$, then the feature of interest in Figure 1 was the coordinate r, which shows that the iterated diffusion map is able to change the topology of a manifold (both the dimension and the number of holes are changed in Figure 1). Notice that although both the source and target manifolds are less than three dimensional, the iterated diffusion map must move through a three-dimensional embedding in order to transition between these very different geometries. Indeed, the first application of the diffusion map (with the local kernel described in Section 4.1) shown in Figure 1 transforms the geometry from an annulus to a cylinder. Intuitively the cylinder introduces a new variable, height, to represent the feature of interest. This is shown by the coloring in Figure 1 which represents the radius of each point on the original annulus, and varies only with the height of the cylinder. As the diffusion map is iterated, the geometry evolves as described in Section 4, intuitively putting more emphasis on the direction (namely the height) which contains the radial information. This is manifested as the circle component of the cylinder contracting until the data set becomes a line, thereby representing only the feature of interest as shown by the coloring.

We now show that the IDM can also contract the annulus onto the other natural feature of interest, namely the angle. However, note that the single parameter $\theta \in [0, 2\pi)$ is not an embedding of the circle, since the periodic boundary conditions cannot be satisfied in $\mathbb{R}^1$. Instead, to recover the circle from the annulus, the feature of interest

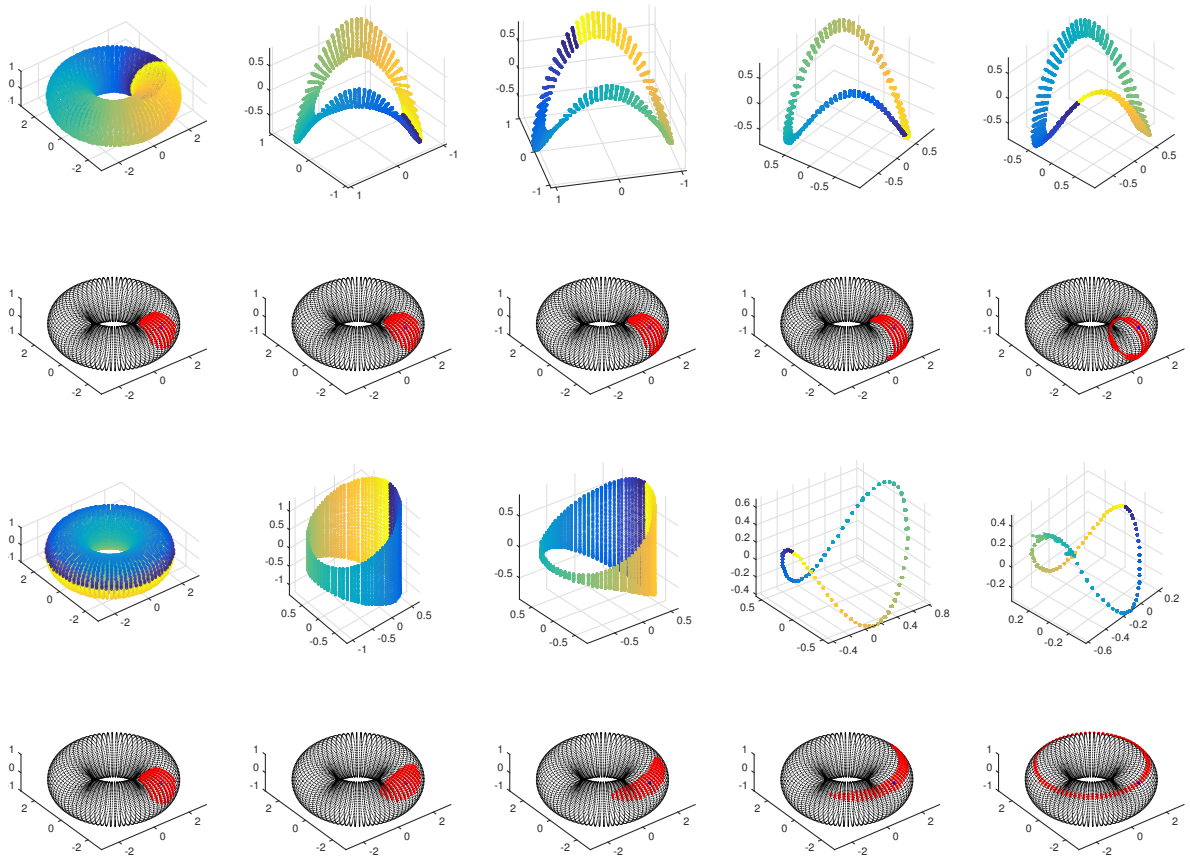

Figure 7: Top Row: Original data set colored according to the desired feature (leftmost) followed by four iterations of the IDM with the feature of interest given by $\mathcal{H}(x, y, z) = (\sin(\phi), \cos(\phi))^\top$ and $\tau = 0.4$. Second Row: Original data set showing the 200 nearest neighbors (red) of the blue data point, where the neighbors are found in the corresponding iterated diffusion map space from the top row. Third Row: Original data set colored according to the desired feature (leftmost) followed by four iterations of the IDM with the feature of interest given by $\mathcal{H}(x, y, z) = (\sin(\theta), \cos(\theta))^\top$ and $\tau = 0.65$. Bottom Row: Original data set showing the 200 nearest neighbors (red) of the blue data point, where the neighbors are found in the corresponding iterated diffusion map space from the third row.

is the two-dimensional feature $(\sin(\theta), \cos(\theta))^\top$, which is an embedding of the circle. In Figure 6 we show the results of applying the iterated diffusion map to the annulus with the feature $\mathcal{H}(\theta, r) = (\sin(\theta), \cos(\theta))^\top$.

Next we consider a simple example where the manifold is a torus $T^2 = S^1 \times S^1$ with intrinsic coordinates $(\theta, \phi) \in [0, 2\pi)^2$ with periodic boundary conditions. Since the torus is a product of two circles, parameterized by θ and ϕ, respectively, we can consider either of these circles as a lower dimension feature of interest. For example, when the desired feature is the circle parameterized by θ, the feature valued function is $\mathcal{H}(x, y, z) = (\sin(\theta), \cos(\theta))^\top$. As shown in Figure 7, when the feature of interest on the torus is either of the circles in the product structure, the IDM evolves the manifold by contracting the irrelevant circle until only the feature of interest remains. Notice that the IDM is able to destroy topological features such as holes in pursuit of the feature of interest.

We also consider a manifold that is not a product space, namely a 2-dimensional unit sphere, and we first choose the feature of interest to be simply the x-coordinate of the sphere and the results for this example are shown in Figure 8. We also consider a sphere with a more complex feature that twists up the sphere, as shown in the third row of Figure 8. While in these cases the geometric flow cannot be described by the simple product formula in (20), the flow still emphasizes the feature of interest and seems to contract the manifold onto a lower dimensional manifold that better represents the feature.

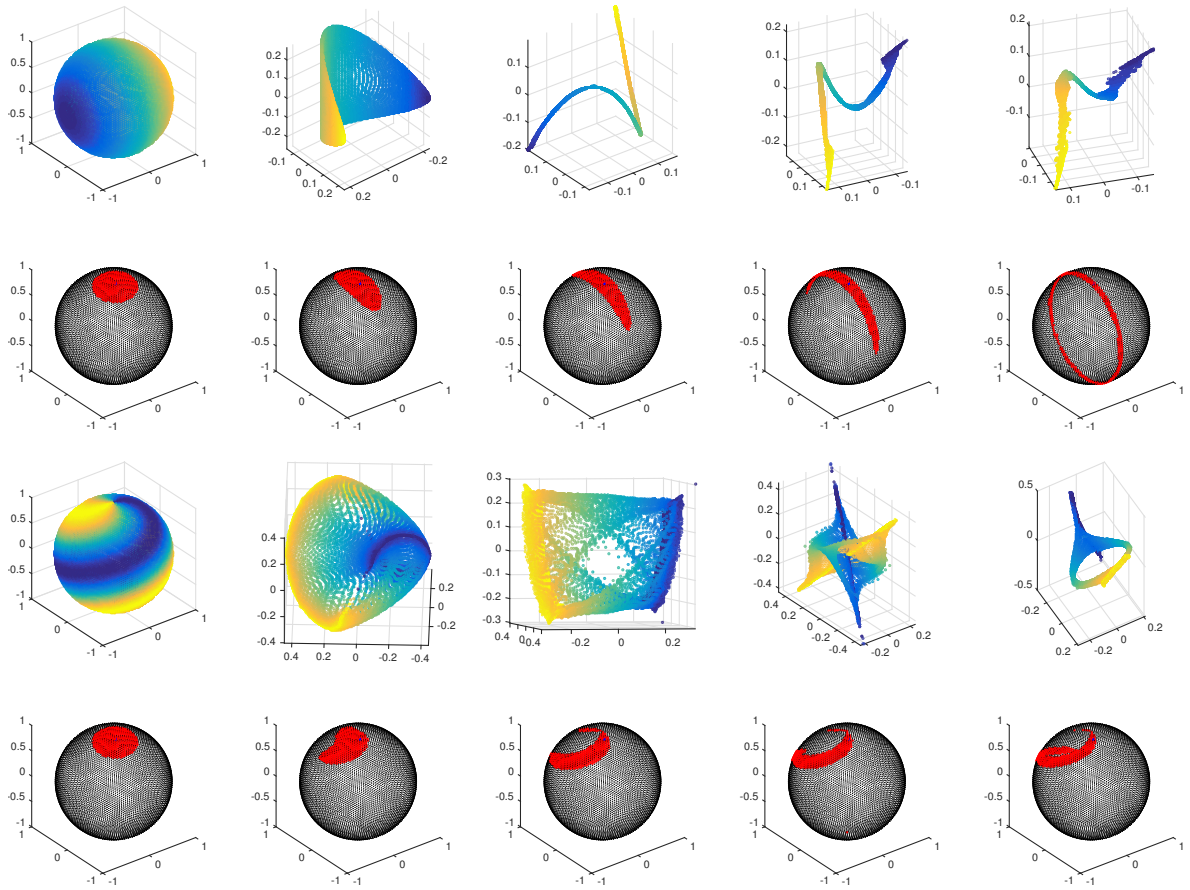

Figure 8: Top: Original data set colored according to the desired feature (leftmost), followed by four iterations of the diffusion maps is the local kernel defined in Section 4 with $\tau = 0.7$. Second Row: Original data set showing the 400 nearest neighbors (red) of the blue data point, where the neighbors are found in the corresponding iterated diffusion map space. Third Row: Original data set colored according to a more complex desired feature (leftmost), followed by four iterations of the IDM with the feature of interest given by $\mathcal{H}(x, y, z) = \sin(\pi z/2 + \tan^{-1}(y/x))$ and $\tau = 0.6$. Bottom Row: Original data set showing the 400 nearest neighbors (red) of the blue data point, where the neighbors are found in the corresponding iterated diffusion map space from the third row.

4.4. Application to Image Data with Incomplete Feature Description

In this section we consider an example of a data set of images which contain two nonlinear modes of variation which are localized in certain regions of the images. By considering a single pixel as a feature on the set of images and applying the IDM we will construct a basis which will include all the pixels which are related to the chosen pixel.

We embed a torus into $\mathbb{R}^{900}$ where each point on the torus is mapped to a 30-pixel-by-30-pixel image. The torus can be described by the two intrinsic variables $(\theta, \phi) \in [0, 2\pi)^2$, so to map the torus into the space of images we constructed two wave fronts which are shown in Figure 9. To construct these images we define a function $\text{Image}(x, y)$ where $(x, y) \in \{1, ..., 30\}^2$ are the pixel coordinates. When the pixel coordinates satisfy $y > x + 5\sin((x + y)/3)$ we set $\text{Image}(x, y) = \sin(r_1(x, y) + \theta)$ where $r_1(x, y) = \sqrt{x^2 + y^2}$ is the distance from the upper left corner. Otherwise, when $y \leq x + 5\sin((x + y)/3)$ we set $\text{Image}(x, y) = \sin(r_2(x, y) + \phi)$ where $r_2(x, y) = \sqrt{(x - 30)^2 + (y - 30)^2}$ is the distance from the lower right corner. Each image is thus separated into to segments by the curve $y = x + 5\sin((x + y)/3)$ which starts near the top left of the image and ends near the bottom right of the image. Above the curve, each image contains stripes which oscillate according to their distance from the upper left corner $(x, y) = (0, 0)$ and the phase of these stripes is defined by the first intrinsic coordinate θ. Below the curve, each image contains stripes which oscillate according to their distance from the bottom right corner $(x, y) = (30, 30)$ and the phase of these stripes is defined by

the second intrinsic coordinates ϕ. This gives a nontrivial embedding of the intrinsic coordinates (θ, ϕ) into the space of 30×30 images. Moreover, information about the coordinate θ is only contained in pixels above the curve and conversely only pixels below the curve contain information about ϕ.

Using a grid of θ and ϕ values we generated 2500 images and in Figure 10 we show examples of five images chosen randomly from this data set. By construction this image set is intrinsically low dimensional, with only two modes of variability, namely the phase θ of the waves in the upper right and the phase ϕ of the waves in the lower left. These two modes of variability are independent over the whole data set because the 2500 images include all possible combinations of θ and ϕ on a discrete two dimensional grid.

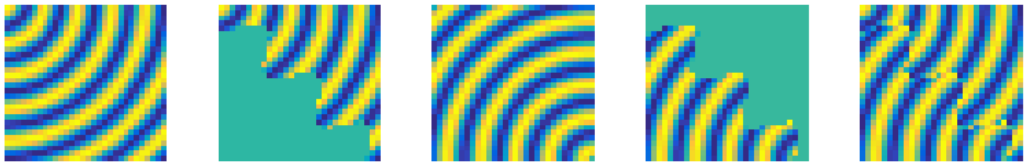

Figure 9: Construction of an image from (θ, ϕ) from left to right. First: Image which encodes θ as the phase of a wavefront. Second: Contribution of first image to the final image. Third: Images which encodes ϕ as the phase of another wavefront in the opposite direction. Fourth: Contribution of third image to the final image. Fifth: Final image encoding both θ and ϕ by combining second and fourth images.

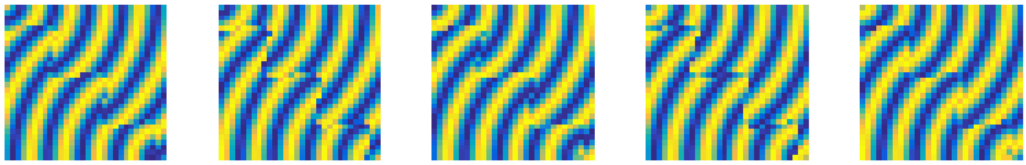

Figure 10: Example images constructed using the method described in Figure 9.

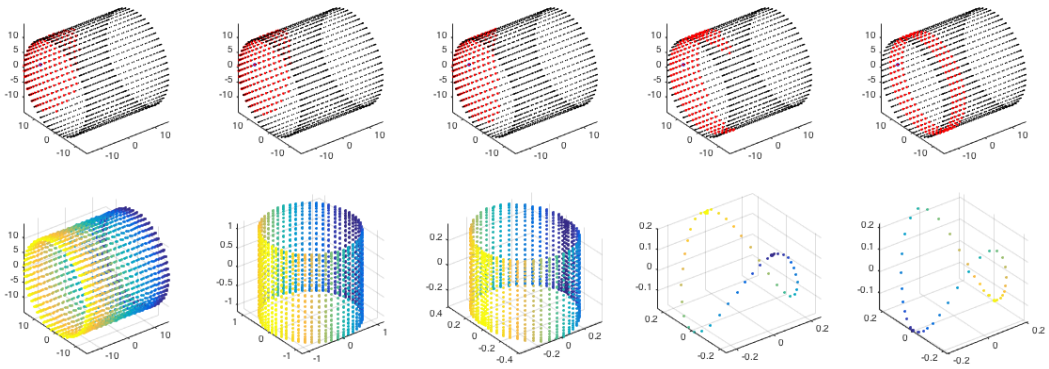

Figure 11: The evolution of the IDM on the image data set where the feature is a single pixel chosen from the bottom left portion of the image.

The goal of this example is to use the IDM to identify the variables in the image which are related to a feature of interest. The feature of interest in this example will be a single pixel which is identified by the user. This is a very practical feature of interest which would be common in many applications. Notice that a single pixel from the bottom left portion of the image will only contain information from ϕ, and it will only contain the partial information

Figure 12: The images $\{c_l\}_{l=2}^{6}$ which represent the eigenfunctions of the feature space learned by the IDM (φ_1 is a constant function so c_1 is not shown).

$\sin(r_2 + \phi)$. In order to correctly identify the feature, the IDM must eliminate all the pixels which are functions of θ. However, the IDM must also use other pixels in the bottom left portion of the image to identify $\sin(r_2 + \phi + \pi/2) = \cos(r_2 + \phi)$ in order identify the intrinsic variable ϕ. Using the full images as our data set and this single pixel as our training feature, we applied the IDM with $\tau = 0.7$ and four iteration and the first three IDM coordinates are shown in Figure 11.

Using the final eigenfunctions $\varphi_l(X_i)$ which are functions defined on the original data set $\{X_i\}_{i=1}^{2500}$ after applying the IDM, we projected the images X_i onto this basis by setting,

$$c_l = \sum_{i=1}^{2500} X_i \varphi_l(X_i).$$

Notice that each c_l is an image which represents the l-th eigenfunction of the feature space learned by the IDM, and these images are shown in Figure 12. We then reconstructed the images as linear combinations of the first five eigenfunctions which span the learned feature space by,

$$\tilde{X}_i = \sum_{l=1}^{5} c_l \varphi_l(X_i).$$

The reconstructed images represent the projection of the data set onto the feature space learned by the IDM and are shown in Figure 13. Notice that the feature space learned by the IDM detects all the pixels which are nonlinearly related to the single pixel chosen as the feature of interest (which was chosen from the bottom left of the image). Moreover, the feature space learned by the IDM is independent of all the pixels in the top right portion of the image which are independent of the chosen feature. We also applied the IDM using a single pixel chosen from the top right portion of the image and the projection onto this feature space correctly isolated the upper right portion of the image.

While this is an idealized example with nice sampling and no noise, it demonstrates a potential application domain of the IDM. This example shows that even in a high-dimensional ambient space the IDM is able to remove variables which are independent of the feature of interest. Significantly, the IDM also identifies variables which are nonlinearly related to the feature of interest which was specified.

Figure 13: The reconstructed images $\tilde{X}_i$ corresponding to the images shown in Figure 10.

5. Conclusion

The above results show that for intrinsically low-dimensional data sets, an iterated diffusion map can help identify features that are hidden in the geometric structure of the data. From a geometric point of view, the IDM approximates a geometric flow that stretches directions on the manifold that are locally correlated to the desired feature and contracts directions that are locally uncorrelated with the desired feature. When the data manifold is a product of the feature manifold and the irrelevant variables, the geometric flow (19) reduces to (20), which stably contracts the irrelevant variables to zero. So for product manifolds, the IDM constructs the quotient map from the product manifold to the feature manifold. For more general manifolds, this geometric flow appears empirically to converge to a lower-dimensional manifold that better represents the feature of interest.

Several key tools are necessary for the construction of the IDM. First, as shown in Section 3, one needs to be able to estimate the local derivatives of a nonlinear map between manifolds embedded in Euclidean space using only empirical data. Second, the construction of Section 4 is necessary to form a local kernel that satisfies the requirements of Theorem 2.1. Finally, the rescaled diffusion mapping of Section 2 is needed to give an isometric embedding of the new geometry introduced by the local kernel. With these three pieces in place, it is possible to iterate the diffusion map in a way that approximates a geometric flow and emphasizes the variable of interest.

However, several important problems remain unsolved. First, we found empirically that applying too many iterations of the diffusion map lead to apparent numerical instability. We suspect that this problem arises from accumulated numerical error in the repeated eigensolves required to find the diffusion mappings. Second, the exact criterion for the convergence of the iterated diffusion map to the desired feature requires a better theoretical understanding of the geometric flow of Section 4, such as its attracting set and the stability of the equilibrium points. Finally, the current algorithm requires the entire manifold to be well-sampled, which means that the data requirements depend on the dimension of the data set and not simply the feature set. Intuitively, it may be possible to extend the IDM to points that lie in sparsely-sampled regions of the data set, as long as these regions are well-sampled in the feature space. Ideally, this could reduce the data requirements to only depend on the dimensionality of the very low-dimensional feature set. However, it is unclear how to extrapolate the IDM into these sparsely-sampled regions of data space, and currently the need to estimate the local derivatives requires fairly dense sampling everywhere on the data manifold.

Acknowledgments

The research of J.H. is partially supported by the Office of Naval Research Grants N00014-11-1-0310, N00014-13-1-0797, MURI N00014-12-1-0912 and the National Science Foundation DMS-1317919. T. B. was supported under the ONR MURI grant N00014-12-1-0912.

References

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

[2] Pierre Bérard, Gérard Besson, and Sylvain Gallot. Embedding riemannian manifolds by their heat kernel. *Geometric & Functional Analysis GAFA*, 4(4):373–398, 1994.

[3] Tyrus Berry and John Harlim. Variable bandwidth diffusion kernels. *Applied and Computational Harmonic Analysis*, 2015.

[4] Tyrus Berry and Timothy Sauer. Local kernels and the geometric structure of data. *Applied and Computational Harmonic Analysis*, 2015.

[5] Peter J. Bickel and Bo Li. *Local polynomial regression on unknown manifolds*, volume Volume 54 of *Lecture Notes–Monograph Series*, pages 177–186. Institute of Mathematical Statistics, Beachwood, Ohio, USA, 2007.

[6] Kamalika Chaudhuri, Sham M Kakade, Karen Livescu, and Karthik Sridharan. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th International Conference on Machine Learning (ICML-09)*, page 17, 2009.

[7] R. Coifman and S. Lafon. Diffusion maps. *Appl. Comput. Harmon. Anal.*, 21:5–30, 2006.

[8] Ronald R Coifman, Yoel Shkolnisky, Fred J Sigworth, and Amit Singer. Graph laplacian tomography from unknown random projections. *Image Processing, IEEE Transactions on*, 17(10):1891–1899, 2008.

[9] Ofir Lindenbaum, Arie Yeredor, Moshe Salhov, and Amir Averbuch. Multiview diffusion maps. *arXiv preprint arXiv:1508.05550*, 2015.

[10] Tomer Michaeli, Weiran Wang, and Karen Livescu. Nonparametric canonical correlation analysis. *arXiv preprint arXiv:1511.04839*, 2015.

[11] Jens Nilsson, Fei Sha, and Michael I. Jordan. Regression on manifolds using kernel dimension reduction. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 697–704, New York, NY, USA, 2007. ACM.

[12] E.J. Nyström. Über die praktische auflösung von integralgleichungen mit anwendungen auf randwertaufgaben. *Acta Mathematica*, 54:185–204, 1930.

[13] Jacobus W Portegies. Embeddings of riemannian manifolds with heat kernels and eigenfunctions. *Communications on Pure and Applied Mathematics*, 2015.
[14] S. Rosenberg. *The Laplacian on a Riemannian manifold*. Cambridge University Press, 1997.
[15] A. Singer. From graph to manifold laplacian: The convergence rate. *Appl. Comp. Harmonic Anal.*, 21:128–134, 2006.
[16] A. Singer and H.-T. Wu. Vector diffusion maps and the connection laplacian. *Communications on Pure and Applied Mathematics*, 65(8):1067–1144, 2012.
[17] Amit Singer and Ronald R. Coifman. Non-linear independent component analysis with diffusion maps. *Applied and Computational Harmonic Analysis*, 25(2):226 – 239, 2008.
[18] Bo Wang, Jiayan Jiang, Wei Wang, Zhi-Hua Zhou, and Zhuowen Tu. Unsupervised metric fusion by cross diffusion. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2997–3004. IEEE, 2012.

Appendix A. Proof of Theorem 3.1

Proof. Following Appendix B of [7], let $x, y \in \mathcal{M}$ with $\|y - x\| < \sqrt{\epsilon}$ with ϵ sufficiently small so that there is a unique geodesic $\gamma : [0, s] \to \mathcal{M}$ with $\gamma(0) = x$ and $\gamma(s) = y$. Let $\{e_i\}$ be a basis for the tangent space $T_x\mathcal{M}$ and define the projection of the geodesic onto the tangent plane by $u_i = \langle y - x, e_i \rangle = \langle \gamma(s) - \gamma(0), e_i \rangle$. Locally, we can parameterize the manifold using a function $q : T_x\mathcal{M} \to T_x\mathcal{M}^\perp$ so that $y - x = (u, q(u))$. We now use the Taylor expansion $\gamma(s) = \gamma(0) + s\gamma'(0) + s^2\gamma''(0)/2 + O(s^3)$, where $\gamma'(0) \in T_x\mathcal{M}$ and $\gamma''(0)$ is orthogonal to the tangent space. Combining the previous lines yields,

$$(u, q(u)) = y - x = \gamma(s) - \gamma(0) = s\gamma'(0) + s^2\gamma''(0)/2 + O(\epsilon^{3/2})$$

and since u and $\gamma'(0)$ are in the tangent plane and $q(u)$ and $\gamma''(0)$ are orthogonal to the tangent plane, we have $u = s\gamma'(0) + O(\epsilon^{3/2})$ and $q(u) = s^2\gamma''(0)/2 + O(\epsilon^{3/2})$. From Equation (B.2) in [7], we have $\|y - x\|^2 = \|u\|^2 + O(\epsilon^2)$. For $v \in T_x\mathcal{M}$ and $w \in T_x\mathcal{M}^\perp$ we have,

$$\langle y - x, v \rangle = s \langle \gamma'(0), v \rangle + O(\epsilon^{3/2}) \qquad \langle y - x, w \rangle = s^2/2 \langle \gamma''(0), w \rangle + O(\epsilon^{3/2})$$

This shows that taking the inner product with vectors $y - x$ in the $\sqrt{\epsilon}$ neighborhood of x, vectors in the tangent space are of order- $\sqrt{\epsilon}$ and vectors in the orthogonal complement are of order-ϵ.

Let $\{x_i\}$ be discrete data points sampled from $\mathcal{M}$. Recall from [7] we have,

$$\begin{aligned}\lim_{N\to\infty} \frac{1}{N} D(x) \equiv \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \exp\left(-\frac{\|x_i - x\|^2}{2\epsilon}\right) &= \int_{\mathcal{M}} \exp\left(-\frac{\|y - x\|^2}{2\epsilon}\right) p(y)\, dV(y) \\ &= \int_{T_x\mathcal{M}} \exp\left(-\frac{\|u\|^2}{2\epsilon}\right) p(x)(1 + O(\epsilon))\, du = (2\pi\epsilon)^{d/2} p(x) + O(\epsilon^{d/2+1}),\end{aligned} \tag{A.1}$$

where the continuous integral is a result of taking Monte-Carlo limit over data sampled from the sampling density $p(y)$ with respect to the volume form dV that $\mathcal{M}$ inherits from the ambient space. The restriction of the integral to the tangent plane $T_x\mathcal{M}$ was shown in [7] and follows from the exponential decay of the integrand and we also use the fact from [7] that $dV(y) = (1 + O(\epsilon))du$. Finally, the change of variables in (A.1) drops all the odd order terms due to the symmetry of the kernel.

Recall that X was the matrix with columns $X_j = D(x)^{-1/2} \exp\left(-\frac{\|x_j - x\|^2}{4\epsilon}\right)(x_j - x) = D(x)^{-1/2} dx_j$ and Y is the matrix

with columns $Y_j = D(x)^{-1/2}\exp\left(-\frac{\|x_j-x\|^2}{4\epsilon}\right)(y_j - y) = D(x)^{-1/2}dy_j$. For any vectors $v \in \mathbb{R}^m$ and $w \in \mathbb{R}^n$ we have,

$$\begin{aligned}
\lim_{N\to\infty} w^\top Y X^\top v &= \lim_{N\to\infty} D(x)^{-1}\sum_{j=1}^{N}\exp\left(-\frac{\|x_j-x\|^2}{2\epsilon}\right)\left\langle \mathcal{H}(x_j)-\mathcal{H}(x), w\right\rangle\left\langle x_j - x, v\right\rangle \\
&= \lim_{N\to\infty}\left(\frac{D(x)}{N}\right)^{-1}\frac{1}{N}\sum_{j=1}^{N}\exp\left(-\frac{\|x_j-x\|^2}{2\epsilon}\right)\left\langle \mathcal{H}(x_j)-\mathcal{H}(x), w\right\rangle\left\langle x_j - x, v\right\rangle \\
&= (2\pi\epsilon)^{-d/2}p(x)^{-1}(1+O(\epsilon))\int_{\mathcal{M}}\exp\left(-\frac{\|y-x\|^2}{2\epsilon}\right)\langle \mathcal{H}(y)-\mathcal{H}(x), w\rangle\langle y-x, v\rangle\, p(y)\, dV(y) \\
&= (2\pi\epsilon)^{-d/2}\int_{T_x\mathcal{M}}\exp\left(-\frac{\|u\|^2}{2\epsilon}\right)\left\langle D\mathcal{H}(x)u + \frac{1}{2}u^\top H(\mathcal{H})(x)u + O(\epsilon^2), w\right\rangle(\langle u, v\rangle + \langle q(u), v\rangle)\,(1+O(\epsilon))\, du
\end{aligned} \tag{A.2}$$

where $H(\cdot)$ is the Hessian operator and the last equality follows from using the exponential decay of the integrand to restrict the integral to the tangent plane (see [7] for details). For $w \in T_{\mathcal{H}(x)}\mathcal{H}(\mathcal{M})$ and $v \in T_x\mathcal{M}$ we reduce (A.2) to,

$$\begin{aligned}
\lim_{N\to\infty} w^\top Y X^\top v &= (2\pi\epsilon)^{-d/2}\int_{T_x\mathcal{M}}\exp\left(-\frac{\|u\|^2}{2\epsilon}\right)\sum_{i,j,k} D\mathcal{H}(x)_{ij}u_j w_i u_k v_k\, du + O(\epsilon^2) = \epsilon\sum_{i,j} D\mathcal{H}(x)_{ij}w_i v_j + O(\epsilon^2) \\
&= \epsilon w^\top D\mathcal{H}(x)v + O(\epsilon^2)
\end{aligned} \tag{A.3}$$

On the other hand, for $w \in \mathbb{R}^n$ and $v \in T_x\mathcal{M}^\perp$ we reduce (A.2) to,

$$\begin{aligned}
\lim_{N\to\infty} w^\top Y X^\top v &= (2\pi\epsilon)^{-d/2}\int_{T_x\mathcal{M}}\frac{1}{2}\exp\left(-\frac{\|u\|^2}{2\epsilon}\right)\sum_{i,j,k,l}[H(\mathcal{H}_l)(x)]_{ij}u_i u_j w_l q_k(u)v_k(1+O(\epsilon))\, du \\
&= (2\pi\epsilon)^{-d/2}\int_{T_x\mathcal{M}}\frac{1}{4}\exp\left(-\frac{\|u\|^2}{2\epsilon}\right)\sum_{i,j,k,l,a,b}[H(\mathcal{H}_l)(x)]_{ij}u_i u_j u_a u_b w_l [H(q_k)(0)]_{ab}v_k\, du + O(\epsilon^3) \\
&= \epsilon^2\sum_{k,l} v_k w_l R_{\mathcal{H}}(x)_{lk} + O(\epsilon^3) = \epsilon^2 w^\top R_{\mathcal{H}}(x)v + O(\epsilon^3),
\end{aligned} \tag{A.4}$$

where we have used the expansion $q_k(u) = u^\top H(q_k)(0)u$ (as shown in [7], since $q(0) = 0$ and $q'(0) = 0$ the first term in the Taylor expansion is quadratic) and we define

$$R_{\mathcal{H}}(x)_{lk} = \frac{1}{4}\Big(\sum_{i,j}[H(\mathcal{H}_l)(x)]_{ii}[H(q_k)(0)]_{jj} + [H(\mathcal{H}_l)(x)]_{ij}[H(q_k)(0)]_{ij} + [H(\mathcal{H}_l)(x)]_{ij}[H(q_k)(0)]_{ji}\Big). \tag{A.5}$$

Finally, it is easy to see that for $w \in T_{\mathcal{H}(x)}\mathcal{H}(\mathcal{M})^\perp$ and $v \in T_x\mathcal{M}$ all the terms appearing in (A.2) will be polynomials of degree 3 or higher in the coordinates of u, and since the degree 3 terms are all odd, by the symmetry of the domain of integration we have $\lim_{N\to\infty} w^\top Y X^\top v = O(\epsilon^3)$. Together with (A.3) and (A.4), we now see that the only order-ϵ terms in $w^\top Y X^\top v$ are in the tangent directions and correspond to $D\mathcal{H}$. The remaining terms are all order-ϵ^2 so that,

$$\frac{1}{\epsilon}w^\top Y X^\top v = w^\top \mathcal{I}_{\mathcal{N}}(x)^\top D\mathcal{H}(x)\mathcal{I}(x)v + O(\epsilon) = w^\top D\hat{\mathcal{H}}(x)v + O(\epsilon).$$

since for v, w in the tangent spaces the projection maps act as the identity and for v, w orthogonal to the tangent spaces the projection is zero. □

We note that the above proof can easily be generalized on kernels of the form $K(\epsilon, x, y) = h\left(\frac{\|y-x\|^2}{\epsilon}\right)$ for $h : [0,\infty) \to [0,\infty)$ having exponential decay by following [7]. In order to find the variance of the estimator, one can apply the method of Singer [15] which was generalized to nonuniform sampling in [3], however this is a lengthy computation which is beyond the scope of this manuscript.

Note that in Corollary 3.2 the Hessian $H(\mathcal{H})$ in the definition of $R_{\mathcal{H}}$ in (A.5) is with respect to the coordinates $u \in T_x\mathcal{M}$, so in general $R_{\mathcal{I}}$ is not necessarily zero. In fact, by repeating the argument in the derivation of (A.5) one can show that,

$$R_{\mathcal{I}}(x)_{lk} = \frac{1}{4}(\mathcal{I}^{\perp}(x))^{\top}\Big(\sum_{i,j}[H(q_l)(0)]_{ii}[H(q_k)(0)]_{jj} + [H(q_l)(0)]_{ij}[H(q_k)(0)]_{ij} + [H(q_l)(0)]_{ij}[H(q_k)(0)]_{ji}\Big)\mathcal{I}^{\perp}(x), \quad \text{(A.6)}$$

where $\mathcal{I}^{\perp}(x) : \mathbb{R}^m \to T_x\mathcal{M}^{\perp}$ is a projection operator that is identity in the directions orthogonal to $T_x\mathcal{M}$ and maps all vectors originating at x to zero when they are in $T_x\mathcal{M}$. In particular this shows that for $v \in T_x\mathcal{M}^{\perp}$ we have $\lim_{N\to\infty} v^{\top}XX^{\top}v = \epsilon^2 v^{\top}R_I(x)v + O(\epsilon^3) = O(\epsilon^2\|v\|^2)$ as mentioned in Section 3.1.

Appendix B. Tuning the Local Bandwidth via SVD

A significant challenge in applying kernel-based methods such as diffusion maps and local kernels is tuning the bandwidth parameter ϵ. The algorithms of [7, 4] are based on a global bandwidth parameter, meaning that the same value of ϵ is used for all data points. In [8] a method was introduced for tuning the global bandwidth parameter based on the scaling law in (A.1). As pointed out in [8], when ϵ is well chosen, the kernel $\exp\left(-\frac{\|y-x\|^2}{2\epsilon}\right)$ will localize the integral over the whole manifold onto the tangent plane. This localization is made rigorous up to an error of order-$\epsilon^{3/2}$ in Lemma 8 of [7]. Thus, when ϵ is well-tuned we expect to see the scaling law $D(x) \propto \epsilon^{d/2}$. On the other hand, in the limit as $\epsilon \to 0$ we find $\frac{1}{N}D(x) \approx \frac{1}{N}\sum_{i=1}^{N} 0 = 0$ and in the limit as $\epsilon \to \infty$ we find $\frac{1}{N}D(x) \approx \frac{1}{N}\sum_{i=1}^{N} 1 = 1$. When using a global bandwidth, the approach advocated in [8] was to average $D(x)$ over the dataset, and to choose bandwidth parameter ϵ so that

$$\overline{D}(\epsilon) \equiv \frac{1}{N}\sum_{j=1}^{N}\frac{D(x_j)}{N} = \frac{1}{N^2}\sum_{i,j=1}^{N}\exp\left(-\frac{\|y-x\|^2}{2\epsilon}\right) \propto \epsilon^{d/2}.$$

In [8] they advocated choosing ϵ such that $\log(\overline{D}(\epsilon)) \approx \frac{d}{2}\log(\epsilon) + c$ is approximately linear as a function of $\log(\epsilon)$. However, since $\overline{D}(\epsilon)$ is a continuous curve and there is no single linear fit which will extract d. For each value of ϵ a local linear regression will give a different value of d, and vice-versa. Thus, this method of tuning the bandwidth parameter implicitly requires either knowing the intrinsic dimension d of the manifold $\mathcal{M}$ or an ad hoc choice of a 'linear' region of the curve $\overline{D}(\epsilon)$.

In [3], an extension of the method of [8] was advocated that simultaneously determines the bandwidth parameter ϵ and the intrinsic dimension d. The approach of [3] is based on the scaling law $S(\epsilon)$ defined by,

$$S(\epsilon) \equiv \frac{d\log(\overline{D})}{d\log(\epsilon)},$$

and noting that when $\epsilon \to 0$ and $\epsilon \to \infty$ we have $S(\epsilon) \to 0$. We should note that the limit $S(\epsilon) \to 0$ as $\epsilon \to 0$ applies only to the biased estimate $D(x_j)$ where the summation includes $i = j$, meaning that the largest summand is always 1. The largest summand being 1 implies that the other summands will lose numerical significance as $\epsilon \to 0$, meaning D converges to a constant and $S(\epsilon) \to 0$. If the unbiased summation of $D(x_j)$ were used (for example in a Kernel Density Estimation) then as $\epsilon \to 0$ the summand corresponding to the shortest distance would dominate, so that $D \propto \exp(-c/\epsilon)$ and $S(\epsilon) = \frac{d\log(D)}{d\log(\epsilon)} = \epsilon\frac{d(-c/\epsilon)}{d\epsilon} \propto \epsilon^{-1}$ in the limit as $\epsilon \to 0$. However, in this paper we restrict our attention to the biased estimate, as required by the diffusion maps and related algorithms, so that as $\epsilon \to 0$ we have $S(\epsilon) \to 0$. This implies that $S(\epsilon)$ has a unique maximum, and in [3] they chose ϵ to maximize $S(\epsilon)$ and then set the dimension by, $d = 2S(\epsilon)$. The approach of [3] was found to be ineffective for kernels with a global bandwidth parameter, especially when there are large variations in the sizes of local neighborhoods due to the sampling of the data set. However, the method of [3] was found to be very robust for a variable bandwidth kernel of the form $\exp\left(-\frac{\|y-x\|^2}{\epsilon\rho(x)\rho(y)}\right)$ where the bandwidth function $\rho(x)$ was chosen to be inversely proportional to a power of the sampling density, namely $\rho(x) \propto p(x)^{\beta}$ for $\beta < 0$.

From (A.1), we should have a scaling law $D(x) \propto \epsilon^{d/2}$ in each local region. We can now connect this fact to the scaling laws of the singular values shown above. Recall that X has d singular values equal to $\sigma_l = \epsilon^{1/2} + O(\epsilon)$, $l = 1, ..., d$ and the remaining $n - d$ singular values are order-ϵ. Thus, we have $\text{trace}(XX^{\top}) = \sum_l \sigma_l^2 = d\epsilon + O(\epsilon^2)$ so

that $\frac{1}{\epsilon}\text{trace}(XX^\top) = d + O(\epsilon)$. Since the trace is independent of the order of multiplication, we can define $\nu = (2\epsilon)^{-1}$ so that $d\log\nu = \frac{d\nu}{\nu} = -\frac{d\epsilon}{\epsilon} = -d\log\epsilon$ and write,

$$\frac{1}{\epsilon}\text{trace}(XX^\top) = 2\nu\,\text{trace}(X^\top X) = \frac{2\nu}{D(x)}\sum_i \exp\left(-\nu\|x_i - x\|^2\right)\|x_i - x\|^2 = \frac{-2\nu}{D(x)}\sum_i \frac{d}{d\nu}\exp\left(-\nu\|x_i - x\|^2\right)$$
$$= \frac{-2\nu}{D(x)}\frac{d}{d\nu}\sum_i \exp\left(-\nu\|x_i - x\|^2\right) = \frac{-2\nu}{D(x)}\frac{dD(x)}{d\nu} = 2\frac{d\log D(x)}{d\log\epsilon}.$$

The previous equation confirms that the scaling law of $D(x)$, given by,

$$S_1(\epsilon) \equiv \frac{d\log(D(x))}{d\log(\epsilon)}$$

should be equal to $d/2$, so one method of estimating the dimension for a given value of ϵ would be,

$$d_1(\epsilon) = 2S_1(\epsilon),$$

and this formula uses the singular values by implicitly taking the trace of the matrix $XX^\top$. This formula was previously known based on the fact that $D(x) \propto \epsilon^{d/2}$, which comes from the normalization factor for a Gaussian on $T_x\mathcal{M}$. However, the connection to the sum of the singular values reveals that when the ambient space dimension, m, is large, the singular values σ_l for $l > d$ can lead to overestimation since,

$$d_1(\epsilon) = 2S_1(\epsilon) = \frac{1}{\epsilon}\text{trace}(XX^\top) = d + \sum_{l=d+1}^{m}\sigma_l^2/\epsilon.$$

Of course, each σ_l is order-ϵ^2 for $l > d$, however, when m is large enough, this summation can lead to significant overestimation. We note that the coefficients of these order-ϵ^2 singular values depend on the curvature of the manifold at the point x, and these coefficients can be large for complex geometries. This shows how the value of ϵ, which maximizes the local scaling law $S_1(\epsilon)$, as suggested in [3], can overestimate the dimension.

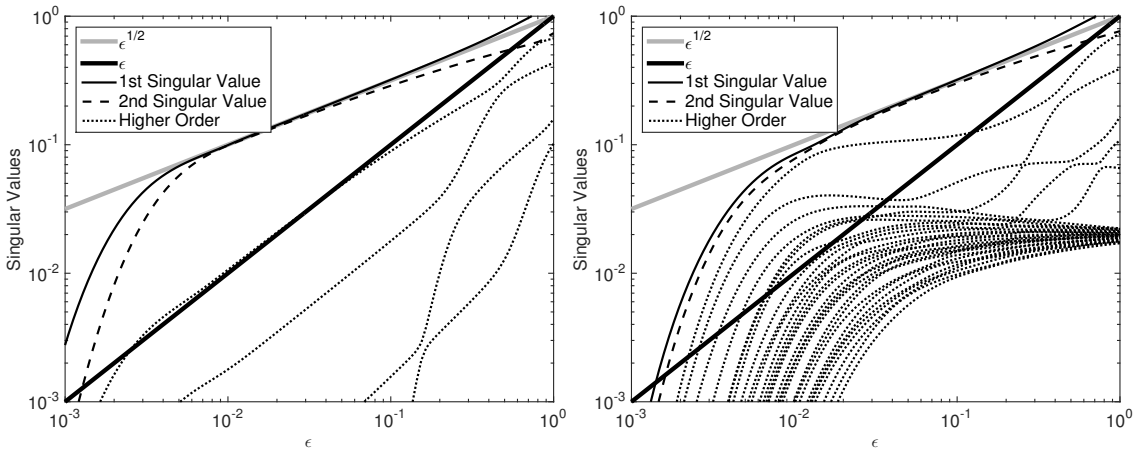

Figure B.14: Singular values as a function of ϵ for a high curvature embedding of a torus into $\mathbb{R}^{30}$ (left) and the same data set perturbed by 30-dimensional additive Gaussian noise with mean zero and covariance matrix $\frac{1}{50}I_{30\times 30}$ (right).

Here, we introduce a new method that combines the ideas of [8, 3] with the local SVD in order to tune ϵ in each local region and improve approximation of the tangent space. Recall from Section 3.1, in a local region of $x \in \mathcal{M}$, we define the matrix of weighted vectors, X, with columns,

$$X_i = D(x)^{-1/2}\exp\left(-\frac{\|x_i - x\|^2}{4\epsilon}\right)(x_i - x).$$

Letting σ_l be the singular values of X, when ϵ is well tuned the first d singular values obey the scaling law $\sigma_l \propto \sqrt{\epsilon}$ and the remaining $m-d$ singular values (where m is the ambient space dimension) are higher order, namely $\sigma_l = O(\epsilon)$. Notice that the $m-d$ singular values which are $O(\epsilon)$ are not necessarily proportional to ϵ; indeed they can be exactly zero in the case of a linear manifold such as a plane embedded in $\mathbb{R}^3$. One strategy would be to threshold the singular values, however, by adding a small amount of noise to the data set in the ambient space, we can easily produce singular values which are greater than ϵ. We illustrate these issues in Figure B.14 by embedding a torus into $\mathbb{R}^{30}$ where the first three coordinates are the standard embedding of the torus and the remaining 27 coordinates results from applying a randomly-generated orthogonal transformation to the first three coordinates raised to the third power and divided by 30. Cubing the coordinates results in a high curvature embedding, which leads to large constants on the $O(\epsilon)$ bound on the singular values corresponding to singular vectors that are orthogonal to the manifold. The orthogonal transformation generates a nontrivial embedding into $\mathbb{R}^{30}$ and the data is then perturbed by independent Gaussian noise with variance $1/50$ added to all 30 coordinates. This results in a highly complex embedding of an intrinsically simple data set. In Figure B.14 we show the singular values for the clean and noisy 30-dimensional embeddings. Notice that thresholding singular values less than ϵ may be effective when the data lies exactly on the manifold (left), however the high curvature can result in nontrivial constants in the $O(\epsilon)$ bound. The addition of noise implies that the dimension of the manifold is greater than two for some values of ϵ (for example $\epsilon \approx 10^{-2}$). For $\epsilon \in [2\times 10^{-2}, 10^{-1}]$ the third largest singular value is larger than ϵ but does not obey the scaling law $\epsilon^{1/2}$. While thresholding alone cannot detect the two-dimensional structure, the scaling laws reveal the true dimension of the manifold.

To incorporate the scaling laws of the singular values into the tuning of ϵ and the dimension estimation, we introduce the following measure of dimension,

$$d_2(\epsilon) \equiv 2 \sum_{l=1}^{\text{floor}(d_1)} \frac{d\log(\sigma_l)}{d\log(\epsilon)} + 2(d_1 - \text{floor}(d_1)) \frac{d\log(\sigma_{\text{floor}(d_1)+1})}{d\log(\epsilon)}.$$

Notice that when d_1 is an integer, the second term is zero, and the summation is simply the sum of the first d_1 scaling laws. The second term prevents $d_2(\epsilon)$ from jumping when $\text{floor}(d_1)$ jumps between integer values (since the first term is a discrete sum it would jump). If the first d_1 singular values correspond to tangent vectors, then the associated scaling laws should be $1/2$, and in this case, we would find $d_2 = 2\sum_{l=1}^{d_1} 1/2 = d_1$. More generally, we can see that the summation can be rewritten as,

$$2 \sum_{l=1}^{\text{floor}(d_1)} \frac{d\log(\sigma_l)}{d\log(\epsilon)} = 2\frac{d}{d\log(\epsilon)} \log\left(\prod_{l=1}^{d_1} \sigma_l\right),$$

which reveals this second dimension to be related to the determinant since it comes from a product of singular values (as opposed to d_1, which comes from a summation of singular values). The final term is included so that d_2 is a smooth function of ϵ.

For each value of ϵ we now have two estimates the dimension, and when ϵ is well-tuned these two estimates of the intrinsic dimension should agree, so we choose ϵ to minimize the relative disagreement $\left|\frac{d_1(\epsilon)-d_2(\epsilon)}{d_{\text{ave}}(\epsilon)}\right|$ where we set the intrinsic dimension to be,

$$d_{\text{ave}}(\epsilon) \equiv (d_1(\epsilon) + d_2(\epsilon))/2.$$

A slight complication is that the curves $d_1(\epsilon)$ and $d_2(\epsilon)$ can intersect multiple times, as shown in Figure B.15. In order to ensure that the scaling laws are stationary at the intersection point, we would also like to minimize the derivatives $\left|\frac{d\log d_1}{d\log\epsilon}\right|$ and $\left|\frac{d\log d_2}{d\log\epsilon}\right|$. Thus, as a practical method of choosing ϵ, we minimize the metric,

$$M(\epsilon) \equiv \left|\frac{d_1(\epsilon) - d_2(\epsilon)}{d_{\text{ave}}(\epsilon)}\right| + \left|\frac{d\log d_1}{d\log\epsilon}\right| + \left|\frac{d\log d_2}{d\log\epsilon}\right|$$

where the derivatives are numerically discretized.

We demonstrate this method of tuning the bandwidth ϵ on the example in Section 3.1 and the results are shown in the top row of Figure B.15. We also applied this method of tuning the bandwidth to the 30-dimensional high curvature embedding from Figure B.14, and the results are shown in the bottom row of Figure B.15. The optimal bandwidth shown in the top row of Figure B.15 was used to plot the singular vectors in Figure 4 above.

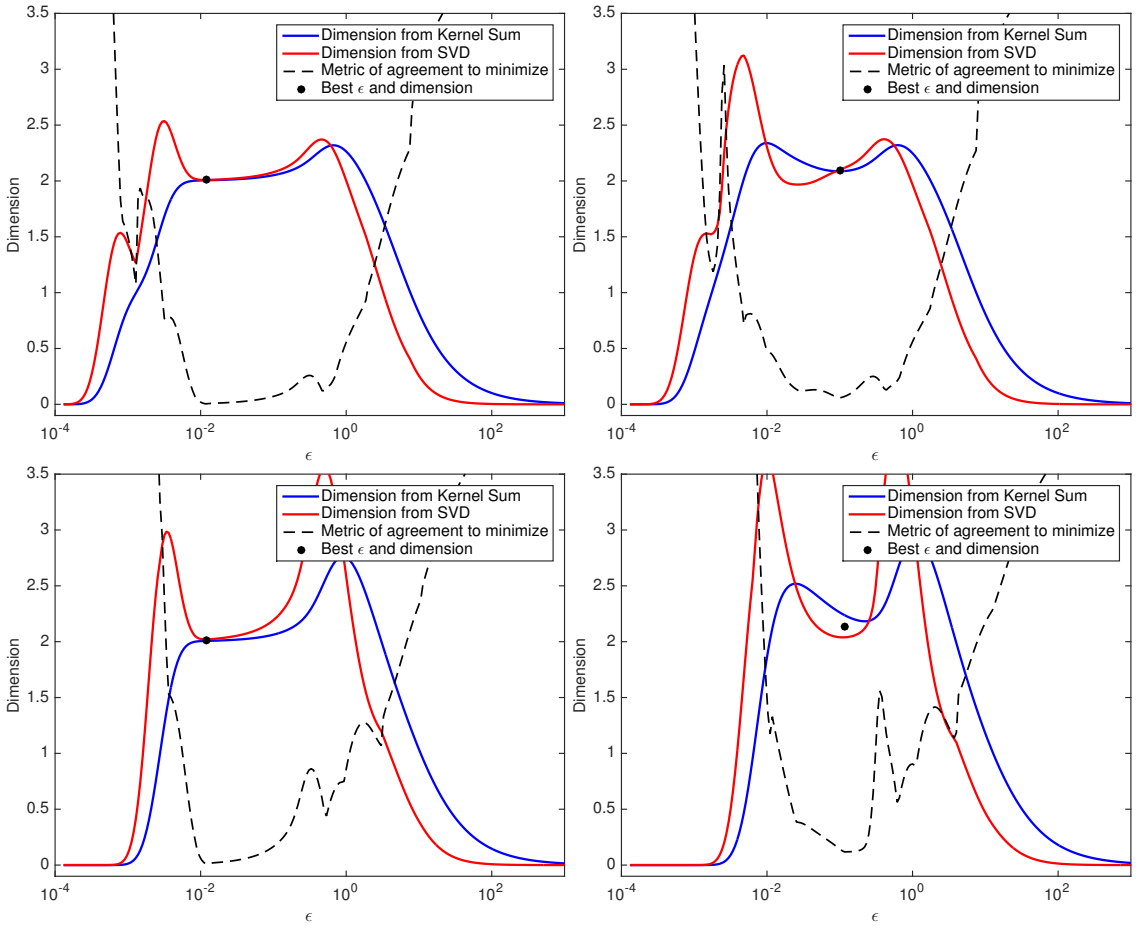

Figure B.15: Top Row: Dimension measures d_1 (blue) and d_2 (red) as functions of the bandwidth ϵ at the base point $(1.996, 0.126, 1.000)^\top$ corresponding to the data set sampled from the torus (left) and the noisy torus (right) shown in Figure 4 of Section 3.1. The metric of agreement, $M(\epsilon)$, is shown as the dotted black curve. The solid black dot represents the bandwidth that minimizes the metric along with the average dimension at the optimal ϵ. Bottom Row: Same curves for the 30-dimensional high-curvature embedding used in Figure B.14 (left) and with 30-dimensional Gaussian noise (right)

We should note that there are many other approaches one could take to estimate the intrinsic dimension of manifolds using the facts introduced in this section. In particular, there are many thresholding methods that could be applied to find integer dimensions. Motivated by applications to noisy and fractal data sets (which fall outside of the current theory) we have developed a non-integer measure of dimension based on scaling laws. Moreover, notice that in Figure 1, different parts of the manifold contract at different rates, so that the dimension of the manifold does not appear constant. This may be intrinsic to the geometric flow or a discretization effect, the theory of geometric flows does not immediately tell us if this can happen intrinsically. As a result of this, in Section 4 and in Appendix C we will use the rescaled diffusion mapping $\hat{\Phi}$ of Section 2.1 with a locally determined dimension $d(x_i)$. Whichever method is used to estimate dimensions, the examples in this section show that both the magnitudes and the scaling laws of the singular values should be incorporated.

A significant drawback of the method of tuning the bandwidth ϵ, introduced in this section, is that computing $d_2(\epsilon)$ requires computing the singular value decomposition of the weighted vectors X, for every base point and a large range of bandwidth parameters. Due to the increase in computational complexity, in all of the examples in this paper (and in the algorithm of Appendix C) we use the simple method of maximizing d_1 to choose the bandwidth. We suspect that this method of choosing the bandwidth is sufficient for the examples below due to low curvature embeddings with

small noise, so we included this new method of tuning ϵ to demonstrate a robust tuning method for more complex data sets.

Appendix C. Numerical Algorithm for the Iterated Diffusion Map

Given a data set $\{x_i\}_{i=1}^N \subset \mathcal{M} \subset \mathbb{R}^m$ and a feature $\{y_i = \mathcal{H}(x_i)\}_{i=1}^N \subset \mathcal{N} \subset \mathbb{R}^n$, the algorithm of this section can be used to construct an embedding $\Psi(x_i)$ which emphasizes the feature of interest. We assume that x_i are points in $\mathbb{R}^m$ which are sampled on (or very close to) a d-dimensional manifold $\mathcal{M}$, and that the feature of interest $y_i = \mathcal{H}(x_i)$ lives on a manifold $\mathcal{N}$ which has dimension less than or equal to d. The algorithm also produces a basis of eigenfunctions $\{\psi_j(x_i)\}_{j=1}^M$ that can be used to represent the map $\mathcal{H}$ and extend this mapping to out-of-sample data points x with standard methods such as the Nyström extension.

The first part of the IDM is a generic algorithm for estimating the derivative $D\mathcal{H}(x_i)$ at each point. The step-by-step algorithm is outlined in the first box below. Optionally, if the embedding of $\mathcal{M}$ has high curvature or the data is noisy, a more robust method of tuning the local bandwidth parameter may be used by adding the following substeps to Step 4. and then replacing Step 5. with "Set $\ell = \text{argmin}(M)$ and $\dim(i) = d_{\text{ave}}(\ell)$."

(e) Form the $k \times m$ matrix $\tilde{X}$ of weighted vectors with j-th column $\tilde{X}_j = \sqrt{\frac{w_j}{D(\ell)}}(x_{I(j)} - x_i)$

(f) Compute the singular values $\sigma_1, ..., \sigma_m$ of X and store them in $s(\ell, j) = \sigma_j$ for $j = 1, ..., m$

(g) If $\ell > 1$ Compute the scaling law of each singular value $\alpha(\ell - 1, j) = \frac{\log(s(\ell,j)) - \log(s(\ell-1,j))}{\log(\epsilon(\ell)) - \log(\epsilon(\ell-1))}$

(h) If $\ell > 1$ Set $d_0 = \text{floor}(d_1(\ell - 1))$ and compute $d_2(\ell - 1) = 2\sum_{j=1}^{d_0} \alpha(\ell - 1, j) + 2(d_1 - d_0)\alpha(\ell - 1, d_0 + 1)$

(i) If $\ell > 1$ set $d_{\text{ave}}(\ell - 1) = (d_1(\ell - 1) + d_2(\ell - 1))/2$

(j) If $\ell > 1$ set $M(\ell - 1) = \left|\frac{d_1(\ell-1) - d_2(\ell-1)}{d_{\text{ave}}(\ell-1)}\right| + \left|\frac{\log(d_1(\ell)) - \log(d_1(\ell-1))}{\log(\epsilon(\ell)) - \log(\epsilon(\ell-1))}\right| + \left|\frac{\log(d_2(\ell)) - \log(d_2(\ell-1))}{\log(\epsilon(\ell)) - \log(\epsilon(\ell-1))}\right|$

Algorithm 1: Estimating $D\hat{\mathcal{H}}(x_i)$ with Auto-tuned Bandwidth

Inputs: Data sets $\{x_i\}_{i=1}^N \subset \mathbb{R}^m$ and $\{y_i\}_{i=1}^N \subset \mathbb{R}^n$. Parameters: number of nearest neighbors, k, and number of discrete values of the bandwidth to consider, L.

Outputs: At each point x_i the algorithm returns the local estimates of the derivative $D\hat{\mathcal{H}}(x_i)$ and the intrinsic dimension $\dim(i)$ and the sampling density $q(i) = q(x_i)$.

For each $i = 1, ..., N$

1. Find the k-nearest neighbors of x_i in $\mathbb{R}^m$, let their indices be $I(j)$ (where $I(1) = i$) for $j = 1, ..., k$ ordered by increasing distance and let $d(i, j) = \|x_i - x_{I(j)}\|$. In Section 4.3 we used $k = 500$.
2. Tune the local bandwidth ϵ using steps (3)-(6)
3. Define $\epsilon_{\min} = d(i, 2)/(2 \log \epsilon_{\text{MACH}})$ and $\epsilon_{\max} = 10d(i, k)$
4. For $\ell = 1, ..., L$
   (a) Let $\epsilon(\ell) = \exp\left(\log(\epsilon_{\min}) + (\ell/L)(\log(\epsilon_{\max}) - \log(\epsilon_{\min}))\right)$
   (b) Compute the weights $w_j = \exp\left(-d(i, j)^2/(2\epsilon(\ell))\right)$
   (c) Compute the sum $D(\ell) = \sum_{j=1}^k w_j$
   (d) If $\ell > 1$, compute $d_1(\ell - 1) = 2\frac{\log(D(\ell)) - \log(D(\ell-1))}{\log(\epsilon(\ell)) - \log(\epsilon(\ell-1))}$
5. Set $\ell = \text{argmax}(d_1)$ and $\dim(i) = d_1(\ell)$
6. Set $\epsilon = \epsilon(\ell + 1)$
7. Compute the weights $w_j = \exp\left(-d(i, j)^2/(2\epsilon)\right)$
8. Compute the sum $D = \sum_{j=1}^k w_j$
9. Set $q(i) = \frac{(2\pi\epsilon)^{\dim(i)/2}}{N} D$
10. Form the $k \times m$ matrix X of weighted vectors with j-th column $X_j = \sqrt{\frac{w_j}{D}}(x_{I(j)} - x_i)$
11. Form the $k \times n$ matrix Y of weighted vectors with j-th column $Y_j = \sqrt{\frac{w_j}{D}}(y_{I(j)} - y_i)$
12. Compute the $m \times n$ matrix $D\hat{\mathcal{H}}(i)$ using the linear least squares regression $D\hat{\mathcal{H}}(i) = (X^\top X)^{-1} X^\top Y$

To compute the Iterated Diffusion Map (IDM), we will iteratively construct x_i^ℓ, where $x_i^0 = x_i$ and ℓ runs up to the desired number of iterations, t. Determining a good stopping criterion is a difficult problem. If the goal is to estimate the map $\mathcal{H}$ using the method of Section 4.2, then a promising approach is to use cross-validation. This approach would first compute the rescaled diffusion coordinates of the feature y_i and then attempt a linear regression from x_i^ℓ to these diffusion coordinates and iterate until the residual ceases to decrease. Another significant issue is that the theory of [4] has not yet been extended to use the variable bandwidth kernels of [3]. So even though we have an estimate of the optimal bandwidth at each point, we can only apply Theorem 3.1 with a fixed global bandwidth. In our examples we found the best choice of global bandwidth was a simple average of the squared distances to the $k = 32$ nearest neighbors of each point, averaged over the whole data set. One reason this ad hoc bandwidth is required is due to the large variations in the dimension that occur as the diffusion map iterates, see for example Figure 1 where some parts of the annulus contract to a line before others. These variations of the dimension also require us to use a locally rescaled diffusion mapping. Notice that Step 8 (a)-(d) are the standard diffusion map algorithm using the local kernel defined by $C_{\mathcal{H}}$ and the associated distances $d_{\mathcal{H}}$. However, to normalize the eigenfunctions in Step 9, we use the locally estimated sampling density $q(i)$. Also, to form the rescaled diffusion mapping in Step 11, we use the locally estimated dimension $\dim(i)$. We found that when a global kernel density estimate and a globally estimated dimension were used, the distances were scaled very differently in different parts of the data set and this distortion led to numerical problems after several iterations.

Algorithm 2: The Iterated Diffusion Map (IDM)

Inputs: Data sets $\{x_i\}_{i=1}^N \subset \mathbb{R}^m$ and $\{y_i\}_{i=1}^N \subset \mathbb{R}^n$. Parameters: number of nearest neighbors, k, number of discrete values of the bandwidth to consider, L, number of nearest neighbors to use to estimate the global bandwidth, k_2, number of eigenfunctions to use in the diffusion map, M, geometric flow discretization parameter, τ, and number of iterations, t.

Outputs: The M-dimensional IDM embedding $x_i^t = \Psi(x_i) = \Phi^{(t)} \circ \cdots \circ \Phi^{(1)}(x_i)$ which represent the feature y_i.

Set $x_i^0 = x_i$
For each $\ell = 0, ..., t$

1. Use Algorithm 1, with inputs $\{x_i^\ell\}$ and $\{y_i\}$ to estimate $D\hat{\mathcal{H}}(x_i^\ell)$, the local dimension $\dim(i) = \dim(x_i^\ell)$ and density $q(i) = q(x_i^\ell)$
2. Let $I(i, j)$ be the index of the j-th nearest neighbor of x_i^ℓ and let $d(i, j) = \|x_{I(i,j)}^\ell - x_i^\ell\|$
3. Define the distance $d_{\mathcal{H}}(i, j) = (1 - \tau)d(i, j) + \tau\|D\hat{\mathcal{H}}^{(\ell)}(x_i^\ell)(x_{I(i,j)}^\ell - x_i^\ell)\|$ with respect to $C_{\mathcal{H}}(x_i)$ from (11)
4. Use the ad hoc global bandwidth estimate $\epsilon = \frac{1}{Nk_2}\sum_{i=1,j=1}^{i=N,j=k_2} d_{\mathcal{H}}(i, j)^2$
5. Build the local kernel $J(i, j) = \exp\left(-d_{\mathcal{H}}(i, j)^2/2\epsilon\right)$
6. Build a sparse $N \times N$ matrix $\tilde{J}$ with $\tilde{J}_{i,I(i,j)} = J(i, j)$
7. Symmetrize $\hat{J} = (\tilde{J} + \tilde{J}^\top)/2$
8. Apply the standard diffusion maps normalizations as in [7, 4]
   (a) Right Normalization: Set $D_i = \sum_j \hat{J}_{ij}$ and $K = \hat{J}_{ij}/(D_i D_j)$
   (b) Left Normalization: Set $\hat{D}_i = \left(\sum_j K_{ij}\right)^{1/2}$ and $\hat{K} = K_{ij}/(\hat{D}_i \hat{D}_j)$
   (c) Compute the $M + 1$ largest eigenvalues ξ_r and associated eigenvectors $\tilde{\varphi}_r$ of $\hat{K}$ for $r = 0, ..., M$
   (d) Define $\hat{\varphi}_r(x_i^\ell) = \tilde{\varphi}_\ell(x_i^\ell)/\hat{D}_i$
9. Normalize the eigenvectors with respect to the sampling density $\varphi_r = \hat{\varphi}_r / \sqrt{\frac{1}{N}\sum_{i=1}^N \hat{\varphi}_r(x_i^\ell)^2/q(i)}$
10. Set $s = 10\epsilon$ and $\lambda_r = \log(\xi_r)/\epsilon$
11. Define the rescaled diffusion mapping $x_i^{\ell+1} = \Phi_s^{(\ell)}(x_i^\ell) = (2\pi)^{\dim(i)/4}(4s)^{\dim(i)/4+1/2}(e^{\lambda_1 s}\varphi_1(x_i^\ell), ..., e^{\lambda_M s}\varphi_M(x_i^\ell))^\top \in \mathbb{R}^M$