

High-Dimensional Pattern Recognition using Low-Dimensional Embedding and Earth Mover's Distance

Linh Lieu^{*,a}, Naoki Saito^a

^a*Department of Mathematics, University of California, Davis, CA 95616, USA*

Abstract

We propose an algorithm that combines existing techniques in a novel way to do classification of datasets consisting of high-dimensional data (e.g., sets of signals or images). Furthermore, our algorithm sets up a framework for application of the Earth Mover's Distance (EMD) [1, 2] as a discriminant measure between datasets. We show how to prepare a compact representation – a *signature* – for each dataset so that computation of EMD between datasets can be done efficiently. This signature-construction step requires the tasks of dimension reduction, automatic determination of the data's intrinsic dimensionality, out-of-sample extension, and point clustering. We will show how to apply some existing methods (which include Laplacian eigenmaps [3, 4, 5], diffusion maps framework [6, 7, 8], and *elongated K*-means [9]) to perform these tasks successfully. We will also provide two examples of applications of our proposed algorithm.

Key words: diffusion maps, Laplacian eigenmaps, principal component analysis, Earth Mover's Distance, Hausdorff distance

1. Introduction

Many problems in pattern recognition require comparison between sets instead of points. For example, in a visual speech recognition problem, two or more clips of recorded video are compared for similar patterns. Typically a video clip consists of a sequence of images. We can view an image as a point and a video clip as a set of points in the image space. Thus, comparing two video clips is the same as comparing two sets of points in the image space. For a second example, consider the task of identifying an object on the ocean floor given a set of sonar waveforms

*Corresponding author

Email addresses: `llieu@math.ucdavis.edu` (Linh Lieu),
`saito@math.ucdavis.edu` (Naoki Saito)

reflected from the object. This requires comparing the set of waveforms reflected from the unknown object to those reflected from known objects. Then the unknown object is identified when a match is made.

In this paper, we propose an algorithm that utilizes existing techniques in a novel way to solve classification problems where the data corresponding to an object consist of a set of points in a high dimensional space. An important idea in our proposed algorithm is the application of the Earth Mover’s Distance (EMD) as a discriminant measure of sets to the classification problems. In [1, 2], the authors have successfully applied EMD to do image retrieval from databases. However, to the best of our knowledge, application of EMD to classification of objects characterized by sets of signals has not been proposed by other authors.

Our proposed method consists of two main stages. The first stage constructs for each dataset a compact representation called a *signature* – a high-dimensional histogram – that carries all important features of the dataset. This can be achieved by utilizing either diffusion maps [6, 7] or Laplacian eigenmaps [3, 4, 5] for dimension reduction and the *elongated K*-means algorithm [9] for automatic determination of the data’s intrinsic dimensionality and point clustering. Each signature is then consisted of the cluster centers in the reduced (low-dimensional) space, and each center is associated with a weight. The second stage applies EMD to compare the signature of each unlabeled dataset to the signatures of the labeled datasets and then classify by the smallest EMD value.

The very first task in our proposed algorithm is to perform dimension reduction on the data. This is necessary because modern technologies often generate data of extremely high dimension. For example, a small 128×128 gray-scale image has dimension 16384. High dimensionality makes data analysis inefficient and inaccurate. Fortunately, the data that we encounter often have low intrinsic dimensionality. For example, consider the set of all $n \times n$ gray-scale images taken of an object under fixed lighting by a moving camera. This is a subset of \mathbb{R}^{n^2} possessing the natural structure of a low-dimensional manifold with its dimensionality defined by the degrees of freedom of the camera [4]. In short, it is often possible to find a low-dimensional representation of the data.

Many nonlinear methods for dimension reduction have been proposed (see [10, 11, 12] for some examples). Unlike the classical methods such as Principal Component Analysis (PCA) and Multidimensional Scaling, nonlinear methods in general offer the advantage of preserving local geometry while achieving dimension reduction. Giving the example of a moving camera in the paragraph above for motivation, M. Belkin and P. Niyogi [3, 4, 5] were among the first group of scientists who proposed a nonlinear dimension reduction algorithm that explicitly considers the manifold structure that may very well be the intrinsic geometry of the data. They proposed using *Laplacian eigenmaps* (LE) constructed from gener-

alized eigenvectors of the (unnormalized) graph Laplacian defined on a weighted graph constructed from the data to embed the data points into a low-dimensional space that preserves the local geometry in the data.

In the case when the data arise from non-uniform sampling of a manifold, embedding via a Laplacian eigenmap may result in distortion of the manifold embedded into the reduced space (see [7] for examples). Sensitivity to sampling densities may be a serious drawback in certain cases. For this reason, R. R. Coifman and S. Lafon [6, 7] proposed a density-invariant normalization of the weights on the graph before computing the graph Laplacian. This would eliminate sensitivity to sampling densities of the Laplacian eigenmaps. Furthermore, the authors defined *diffusion maps* from the eigenvalues and eigenvectors of the diffusion operator (defined in Eq. (2) below) and provided an intuitive interpretation of how point clustering in a diffusion coordinate system is linked to a Markov random walk on the weighted graph (see also [13]).

Although Laplacian eigenmaps and diffusion maps are related, they do have their distinctive differences. However, we shall not delve into detailed comparison between them (interested readers are referred to [6, 7]). They present to us a nice way to prepare the data for appropriate and efficient application of EMD. This preparation step requires clustering of the (embedded) points in the reduced space to form a signature for each dataset. To do this, we propose to apply an existing algorithm called the elongated K -means algorithm [9]. As we briefly mentioned above, the elongated K -means algorithm allows us to simultaneously determine the intrinsic dimensionality of the data and the clusters of points within the reduced space. It falls directly under the framework of the Laplacian eigenmaps and the diffusion maps. In our proposed method, the elongated K -means algorithm is the connecting bridge between Laplacian eigenmaps or diffusion maps and Earth Mover’s Distance.

We will present two examples of application for our proposed algorithm in Section 5. One example involves classification of underwater objects from sonar data provided to us by the Naval Surface Warfare Center, Panama City (NSWC-PC), FL. Such classification problems are of high interests at NSWC-PC. Another example of application is a small lip-reading experiment in which we try to identify the word spoken from a sequence of images extracted from a video clip. No audio is involved. We will also apply PCA for dimension reduction in our numerical experiments. However, in this case we will have to make an educated guess of the number of significant principal components based on the decay of the eigenvalues of the covariance matrix.

Finally, we note that utilizing diffusion maps for dimension reduction and the out-of-sample extension scheme (the GHME scheme discussed in Section. 3.3 below) is also part of the Lafon-Keller-Coifman (LKC) method proposed in [8] for

datasets matching problems similar to the ones in our consideration. Their algorithm uses diffusion maps to embed the data into a reduced space and then use the Hausdorff distance (HD) to measure the difference between the point sets in the reduced space. One important difference between our proposed algorithm and the LKC method is using EMD instead of HD. Moreover, our proposed method sets up a framework for application of EMD to classification of datasets (datasets matching). The techniques we employ in the signature-construction stage is not limited to the framework of the diffusion maps. Furthermore, the advantage that EMD has over HD is robustness to outliers. As we shall see in Section 5 below, HD is very sensitive to outliers. In contrast, the EMD between the signatures measures the differences between the distributions of points within the reduced space. Therefore it is less affected by outliers and hence yields more reliable results.

2. Earth Mover’s Distance

The definition of the Earth Mover’s Distance (EMD) is based on the solution to a discrete *optimal mass transportation problem*. EMD represents the minimum cost of moving earth (or sand) from some source locations to fill up holes at some sink locations. In other words, given any two probability distributions, one of them can be viewed as a distribution of earth and the other a distribution of holes, then the EMD between the two distributions is the minimum cost of rearranging the mass in one distribution to obtain the other. In the continuous setting, this problem is known as the *Monge-Kantorovich optimal mass transfer* problem and has been well studied over the past 100 years (see [21] for an introductory reading on the problem). The importance here is that EMD can be applied to measure the discrepancy between two multidimensional distributions.

In the discrete setting, the optimal mass transfer problem can be formulated as a linear optimization problem as follows [1, 2]. Suppose we have a source (earth) distribution $P = \{(\mathbf{p}_1, w_{\mathbf{p}_1}), \dots, (\mathbf{p}_m, w_{\mathbf{p}_m})\}$ and a sink (hole) distribution $Q = \{(\mathbf{q}_1, w_{\mathbf{q}_1}), \dots, (\mathbf{q}_n, w_{\mathbf{q}_n})\}$ in a high-dimensional space \mathbb{R}^s . In this setting, P and Q are called *signatures* and can be viewed as two distributions of feature vectors representing two objects. P is a signature of one object that consists of m clusters in \mathbb{R}^s , where \mathbf{p}_i is the center of the i th cluster and $w_{\mathbf{p}_i}$ is the proportion of the object’s feature vectors that belongs to the i th cluster. Similarly, Q is a signature of another object that consists of n clusters with the cluster center and the weight pairs $(\mathbf{q}_j, w_{\mathbf{q}_j})$, $j = 1, \dots, n$.

Suppose the cost of moving one unit of mass from \mathbf{p}_i to \mathbf{q}_j is $c(\mathbf{p}_i, \mathbf{q}_j)$, and f_{ij} denotes the amount of mass flow from \mathbf{p}_i to \mathbf{q}_j . Then, the transportation cost is

defined as:

$$\text{COST}(P, Q, \mathbf{F}) \triangleq \sum_{i=1}^m \sum_{j=1}^n c(\mathbf{p}_i, \mathbf{q}_j) f_{ij},$$

where $\mathbf{F} \triangleq [f_{ij}] \in \mathbb{R}^{m \times n}$. The optimal mass transfer problem seeks the flow \mathbf{F}^* that transfers the maximum allowable amount of earth to fill up the holes with minimum total transportation cost, i.e.,

$$\mathbf{F}^* = \arg \min_{\mathbf{F} \in S} \text{COST}(P, Q, \mathbf{F}),$$

where $\mathbf{F} \in S \subset \mathbb{R}^{m \times n}$ means that \mathbf{F} must satisfy the following constraints:

- (i) $f_{ij} \geq 0$, for all i, j ;
- (ii) $\sum_{j=1}^n f_{ij} \leq w_{\mathbf{p}_i}$, for all $1 \leq i \leq m$;
- (iii) $\sum_{i=1}^m f_{ij} \leq w_{\mathbf{q}_j}$, for all $1 \leq j \leq n$; and
- (iv) $\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min \left(\sum_{i=1}^m w_{\mathbf{p}_i}, \sum_{j=1}^n w_{\mathbf{q}_j} \right)$.

The constraint (i) ensures that one can only move earth from P to Q , not vice versa; (ii) that the amount of earth moved from P is no more than the sum of the weights $w_{\mathbf{p}_i}$; (iii) that the amount of earth received at Q is no more than the sum of the weights $w_{\mathbf{q}_j}$; and (iv) that the maximum allowable amount of earth is moved.

Once the optimal flow \mathbf{F}^* from P to Q is found, EMD is then defined as the total cost normalized by the total flow:

$$\text{EMD}(P, Q) \triangleq \frac{\text{COST}(P, Q, \mathbf{F}^*)}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}^*} = \frac{\sum_{i=1}^m \sum_{j=1}^n c(\mathbf{p}_i, \mathbf{q}_j) f_{ij}^*}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}^*}. \quad (1)$$

Notice that the normalization factor is the total weight of the smaller signature due to the constraint (iv). This normalization ensures that smaller signatures are not favored in the case when two signatures have different total weights. Furthermore, EMD is symmetric, i.e., $\text{EMD}(P, Q) = \text{EMD}(Q, P)$ for any two distributions P and Q .

3. Diffusion Maps and Laplacian Eigenmaps

In this section, we review the construction of diffusion maps and Laplacian eigenmaps on the data and the properties that allow us to achieve meaningful dimensionality reduction. We will also review an algorithm proposed in [8] for extension of these embedding maps from the training data to the test data.

3.1. Diffusion maps and diffusion distances

Diffusion maps are constructed from the eigenvectors of an averaging operator – the *diffusion operator*. We assume that the given data $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ lie in a space having a natural dissimilarity measure δ that gives a sense of affinity between any two data points. This is a reasonable assumption to make in practice. For example, if X is a database of image patches of size 32×32 , then the image patches can be viewed as points in \mathbb{R}^{1024} , and δ may be the ℓ^2 norm on \mathbb{R}^{1024} . Or, if X lies on a submanifold in \mathbb{R}^n , then δ may be the usual Euclidean distance. Following the work of Coifman and Lafon [6, 7], we construct the diffusion operator on X as follows. View the data points $\mathbf{x}_1, \dots, \mathbf{x}_N$ as nodes of a weighted symmetric graph, commonly referred to as *similarity graph*. Any two nodes \mathbf{x}_i and \mathbf{x}_j are connected by an edge with weight $w_\varepsilon(\mathbf{x}_i, \mathbf{x}_j) \triangleq e^{-(\delta(\mathbf{x}_i, \mathbf{x}_j)/\varepsilon)^2}$, $\varepsilon > 0$. The weight function w_ε gives the notion of local geometry to X . That is, it defines the notion of a local neighborhood at each point $\mathbf{x} \in X$ via the affinity between \mathbf{x} and other points, and the value of the parameter ε specifies the size of this neighborhood. Moreover, as explained in [4], when the dataset X approximately lies on a submanifold, using the weights w_ε on the graph corresponds to an approximation of the heat kernel on the submanifold.

Applying the so-called graph-Laplacian normalization to w_ε yields the *diffusion kernel*

$$k(\mathbf{x}, \mathbf{y}) \triangleq \frac{w_\varepsilon(\mathbf{x}, \mathbf{y})}{d_\varepsilon(\mathbf{x})}, \quad (2)$$

where $d_\varepsilon(\mathbf{x}) \triangleq \sum_{\mathbf{y} \in X} w_\varepsilon(\mathbf{x}, \mathbf{y})$. The corresponding diffusion operator is

$$Af(\mathbf{x}) \triangleq \sum_{\mathbf{y} \in X} k(\mathbf{x}, \mathbf{y})f(\mathbf{y}). \quad (3)$$

The kernel k is non-negative and row-stochastic (i.e., $\sum_{\mathbf{y} \in X} k(\mathbf{x}, \mathbf{y}) = 1$ for all $\mathbf{x} \in X$). Hence, it can be viewed as a transition matrix of a Markov process on X . The operator A is an averaging operator, since it is positivity-preserving (i.e., $Af \geq 0$ for any $f \geq 0$) and preserves constant functions. We can interpret the action of the operator A as ‘diffusion’ of information throughout the graph, and the Markov chain dictates the directions of fast and slow information propagation.

An important idea in the diffusion framework is to take larger powers of the operator A . For $t > 0$, raising the operator A to a power t is equivalent to running the Markov process forward by time t , which can be interpreted as letting information diffuse for a period of time t . The information propagates more easily and quickly among the regions of high affinity than those of low affinity. This is essentially how we can capture the local geometry of the data.

Let $k^{(t)}$ denotes the kernel of the operator A^t – the t th power of the operator A . (Note that $k^{(t)}(\mathbf{x}, \mathbf{y})$ represents the probability of transition from \mathbf{x} to \mathbf{y} in t steps.) The graph is connected by construction, therefore as $t \rightarrow +\infty$ the Markov process approaches a unique stationary distribution ϕ_0 [6], i.e, for any $\mathbf{x}, \mathbf{y} \in X$,

$$\lim_{t \rightarrow +\infty} k^{(t)}(\mathbf{x}, \mathbf{y}) = \phi_0(\mathbf{y}).$$

In practice we always work in the discrete setting, therefore we may view the operator A as a matrix whose rows are indexed by \mathbf{x} and columns are indexed by \mathbf{y} . Then the stationary distribution $\phi_0 = [\phi_0(\mathbf{x}_1), \dots, \phi_0(\mathbf{x}_N)]$ is the left eigenvector of A corresponding to the top eigenvalue 1, i.e., $\phi_0 A = \phi_0$. It can be easily derived from equations (2) and (3) and the symmetry of w_ε that

$$\phi_0(\mathbf{x}) = \frac{d_\varepsilon(\mathbf{x})}{\sum_{\mathbf{z} \in X} d_\varepsilon(\mathbf{z})}.$$

With this, the *diffusion distance* $D_t(\mathbf{x}, \mathbf{y})$ between any two data points \mathbf{x} and \mathbf{y} at the t th time step is given by

$$\begin{aligned} D_t(\mathbf{x}, \mathbf{y})^2 &\triangleq \left\| k^{(t)}(\mathbf{x}, \cdot) - k^{(t)}(\mathbf{y}, \cdot) \right\|_{L^2(X, \frac{1}{\phi_0})}^2 \\ &= \sum_{\mathbf{z} \in X} \frac{(k^{(t)}(\mathbf{x}, \mathbf{z}) - k^{(t)}(\mathbf{y}, \mathbf{z}))^2}{\phi_0(\mathbf{z})}. \end{aligned} \quad (4)$$

This is simply the weighted L^2 distance between $k^{(t)}(\mathbf{x}, \cdot)$ and $k^{(t)}(\mathbf{y}, \cdot)$. We observed earlier that $k^{(t)}(\mathbf{x}, \mathbf{z})$ is the probability of transition from \mathbf{x} to \mathbf{z} in t steps. Therefore it is easy to see that the diffusion distance between \mathbf{x} and \mathbf{y} measures the difference in how much connected or how strong in affinity these two nodes are to the rest of the graph at time (or step) t . In its definition, the diffusion distance $D_t(\mathbf{x}, \mathbf{y})$ takes into account all incidences relating \mathbf{x} and \mathbf{y} . Consequently, it is robust to noise perturbations and hence a great tool for extracting the underlying geometry in the dataset X , especially when X is a low dimensional manifold lying in a high-dimensional space.

The diffusion distance is directly related to the eigenvalues and eigenvectors of the matrix A . In practice, we approximate $D_t(\cdot, \cdot)$ by using eigenvalues and eigenvectors of A . To see this, let us first do some preprocessing: conjugate the kernel k with $\sqrt{\phi_0}$ to obtain the symmetric kernel

$$\begin{aligned} \tilde{k}(\mathbf{x}, \mathbf{y}) &\triangleq \sqrt{\phi_0(\mathbf{x})} k(\mathbf{x}, \mathbf{y}) \frac{1}{\sqrt{\phi_0(\mathbf{y})}} \\ &= \frac{w_\varepsilon(\mathbf{x}, \mathbf{y})}{\sqrt{d_\varepsilon(\mathbf{x})} \sqrt{d_\varepsilon(\mathbf{y})}}. \end{aligned} \quad (5)$$

Let \tilde{A} be the operator with \tilde{k} as its kernel, i.e.,

$$\tilde{A}f(\mathbf{x}) \triangleq \sum_{\mathbf{y} \in X} \tilde{k}(\mathbf{x}, \mathbf{y})f(\mathbf{y}). \quad (6)$$

It shares the same spectrum as A , and eigenvectors of A can be obtained from those of \tilde{A} via conjugation by $\sqrt{\phi_0}$. Suppose $\{\lambda_\ell\}$ are the eigenvalues (with $|\lambda_0| \geq |\lambda_1| \geq \dots$) and the corresponding eigenvectors of \tilde{A} are $\{\tilde{\phi}_\ell\}$, then the left and right eigenvectors of A corresponding to λ_ℓ are $\phi_\ell = \tilde{\phi}_\ell \cdot \sqrt{\phi_0}$ and $\psi_\ell = \tilde{\phi}_\ell / \sqrt{\phi_0}$, respectively.

The advantage of the operator \tilde{A} is that it is symmetric, positive semi-definite, and compact. Hence it has a discrete, non-increasing, non-negative spectra: $\lambda_0 = 1 > \lambda_1 \geq \lambda_2 \geq \dots \geq 0$, and the orthonormal eigenvectors $\{\tilde{\phi}_\ell\}$ form a basis for $L^2(X)$ (the eigenvector corresponding to top eigenvalue $\lambda_0 = 1$ is $\tilde{\phi}_0 = \sqrt{\phi_0}$). The kernel \tilde{k} has spectral decomposition

$$\tilde{k}(\mathbf{x}, \mathbf{y}) = \sum_{j \geq 0} \lambda_j \tilde{\phi}_j(\mathbf{x}) \tilde{\phi}_j(\mathbf{y}).$$

Hence,

$$k(\mathbf{x}, \mathbf{y}) = \sum_{j \geq 0} \lambda_j \psi_j(\mathbf{x}) \phi_j(\mathbf{y})$$

and

$$k^{(t)}(\mathbf{x}, \mathbf{y}) = \sum_{j \geq 0} \lambda_j^t \psi_j(\mathbf{x}) \phi_j(\mathbf{y}). \quad (7)$$

Now, $\{\phi_\ell\}$ and $\{\psi_\ell\}$ are biorthogonal (i.e., $\sum_{\mathbf{z} \in X} \phi_j(\mathbf{z}) \psi_\ell(\mathbf{z}) = \delta_{j\ell}$, where $\delta_{j\ell}$ is the Kronecker delta), and

$$\phi_\ell(\mathbf{x}) = \phi_0(\mathbf{x}) \psi_\ell(\mathbf{x}).$$

Thus,

$$\sum_{\mathbf{z} \in X} \frac{\phi_j(\mathbf{z}) \phi_\ell(\mathbf{z})}{\phi_0(\mathbf{z})} = \sum_{\mathbf{z} \in X} \phi_j(\mathbf{z}) \psi_\ell(\mathbf{z}) = \delta_{j\ell}.$$

That is, $\{\phi_\ell\}$ is an orthonormal basis in $L^2(X, 1/\phi_0)$. Therefore, for fixed \mathbf{x} , the formula (7) can be interpreted as the expansion of the function $k^{(t)}(\mathbf{x}, \cdot)$ in this basis, and the expansion coefficients are $\{\lambda_j^t \psi_j(\mathbf{x})\}$. Consequently, the formula (4) for the diffusion distance reduces to

$$D_t(\mathbf{x}, \mathbf{y})^2 = \sum_{j \geq 1} \lambda_j^{2t} (\psi_j(\mathbf{x}) - \psi_j(\mathbf{y}))^2. \quad (8)$$

Note that $\psi_j(\mathbf{x}) = \tilde{\phi}_j(\mathbf{x})/\tilde{\phi}_0(\mathbf{x})$. In other words, the proper diffusion distance can be obtained by the eigenanalysis of the symmetrized operator \tilde{A} with the kernel \tilde{k} instead of the original averaging operator A with the transition kernel k . The summation in (8) starts at index $j = 1$ because $\psi_0 \equiv 1$.

In practice, we approximate the diffusion distance formula (8) by the following consideration. Since the eigenvalues λ_j 's are non-increasing, the diffusion distance can be approximated to a relative accuracy $\tau > 0$ specified by the user by

$$D_t(\mathbf{x}, \mathbf{y})^2 \approx \sum_{j=1}^{s(\tau,t)} \lambda_j^{2t} (\psi_j(\mathbf{x}) - \psi_j(\mathbf{y}))^2, \quad (9)$$

where

$$s(\tau, t) \triangleq \arg \max_{j \in \mathbb{N}} \{|\lambda_j|^t > \tau |\lambda_1|^t\}. \quad (10)$$

From this, the *diffusion map* is defined as

$$\Psi_t : \mathbf{x} \mapsto \left(\lambda_1^t \psi_1(\mathbf{x}), \lambda_2^t \psi_2(\mathbf{x}), \dots, \lambda_{s(\tau,t)}^t \psi_{s(\tau,t)}(\mathbf{x}) \right)^T. \quad (11)$$

It can be viewed as coordinates in a $s(\tau, t)$ -dimensional Euclidean space characterized by the parameters ε , t , and τ . We shall refer to this space a *diffusion space*.

We use Ψ_t to embed our dataset into a diffusion space denoted by $\mathbb{R}^{s(\tau,t)}$. Note that *the usual Euclidean distance in this diffusion space is an approximation to the diffusion distance*. The key point here is that the diffusion map Ψ_t produces a low-dimensional representation of the data that highlights the underlying intrinsic local geometry in the data.

The final important thing to mention is the density-invariant normalization of the edge weights $w_\varepsilon(\mathbf{x}, \mathbf{y})$ if the data X approximately lies on a submanifold \mathcal{M} of \mathbb{R}^n [6, 7, 8]. In this case, we replace w_ε with a normalized version

$$w_\varepsilon(\mathbf{x}, \mathbf{y}) \longleftarrow \frac{w_\varepsilon(\mathbf{x}, \mathbf{y})}{d_\varepsilon(\mathbf{x})d_\varepsilon(\mathbf{y})}. \quad (12)$$

Then proceed to construct diffusion kernel as described in (2) above. In other words, we normalize the weights twice to construct diffusion kernel: first, the above density-invariant normalization, and second, the graph Laplacian normalization. When the data points are sampled from \mathcal{M} nonuniformly, this normalization makes the transition matrix A approximate the Laplace-Beltrami diffusion operator on \mathcal{M} and the embedding of the data points via diffusion maps invariant to the density distribution of the sampled data. In short, the density-invariant normalization produces a spectral embedding that depends only on the geometry of \mathcal{M} and not the density of the sampled data points.

3.2. Laplacian eigenmaps

Laplacian eigenmaps [3, 4] and diffusion maps are constructed from the same eigenvectors, therefore many of the discussions in the previous section are easily transferred. However, we stress that LE and DM have distinctive differences. For example, the clustering property of diffusion maps can be derived from considering a random walk forwarded in time while that of Laplacian eigenmaps is derived from the local-neighborhood-preserving property of the generalized eigenvectors (see below).

As before, we start with constructing from the data a weighted (similarity) graph, such as the Gaussian weights $w_\varepsilon(\mathbf{x}_i, \mathbf{x}_j) = e^{-(\delta(\mathbf{x}_i, \mathbf{x}_j)/\varepsilon)^2}$, $\varepsilon > 0$. With $d_\varepsilon(\mathbf{x}) = \sum_{\mathbf{y} \in X} w_\varepsilon(\mathbf{x}, \mathbf{y})$ as before, the unnormalized graph Laplacian [14] is defined as follows.

$$L(\mathbf{x}, \mathbf{y}) \triangleq \begin{cases} d_\varepsilon(\mathbf{x}) - w_\varepsilon(\mathbf{x}, \mathbf{x}) & \text{if } \mathbf{x} = \mathbf{y}, \\ -w_\varepsilon(\mathbf{x}, \mathbf{y}) & \text{otherwise.} \end{cases}$$

Then for any real-valued function f ,

$$Lf(\mathbf{x}) \triangleq \sum_{\mathbf{y}} (f(\mathbf{x}) - f(\mathbf{y}))w_\varepsilon(\mathbf{x}, \mathbf{y}). \quad (13)$$

(Here we use the notation L for both the kernel and the corresponding operator). Also, the normalized graph Laplacian is defined as

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) \triangleq \frac{L(\mathbf{x}, \mathbf{y})}{\sqrt{d_\varepsilon(\mathbf{x})}\sqrt{d_\varepsilon(\mathbf{y})}} = \begin{cases} 1 - \frac{w_\varepsilon(\mathbf{x}, \mathbf{x})}{d_\varepsilon(\mathbf{x})} & \text{if } \mathbf{x} = \mathbf{y}, \\ \frac{-w_\varepsilon(\mathbf{x}, \mathbf{y})}{\sqrt{d_\varepsilon(\mathbf{x})}\sqrt{d_\varepsilon(\mathbf{y})}} & \text{otherwise.} \end{cases}$$

And,

$$\mathcal{L}f(\mathbf{x}) \triangleq f(\mathbf{x}) - \sum_{\mathbf{y}} \frac{w_\varepsilon(\mathbf{x}, \mathbf{y})}{\sqrt{d_\varepsilon(\mathbf{x})}\sqrt{d_\varepsilon(\mathbf{y})}} f(\mathbf{y}). \quad (14)$$

Both L and \mathcal{L} are symmetric positive semi-definite. The constant vector $f \equiv 1$ is the unique eigenvector of L and $g(\mathbf{x}) = \sqrt{d_\varepsilon(\mathbf{x})}$ is the unique eigenvector of \mathcal{L} corresponding smallest eigenvalue $\lambda = 0$, since the graph is connected. More properties of L can be found in [4] and of \mathcal{L} in [15].

In [4], the authors consider the solutions $f_i(\mathbf{x})$ of the generalized eigenvector problem

$$Lf_i(\mathbf{x}) = \lambda_i d_\varepsilon(\mathbf{x}) f_i(\mathbf{x}), \quad (15)$$

with eigenvalues ordered in ascending order $0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \dots$. When the graph is connected, $f \equiv 1$ is the unique solution corresponding to $\lambda_0 = 0$.

Owing to the Rayleigh principle $\langle f, Lf \rangle = \frac{1}{2} \sum_{\mathbf{x}, \mathbf{y}} (f(\mathbf{x}) - f(\mathbf{y}))^2 w_\varepsilon(\mathbf{x}, \mathbf{y})$, the first non-trivial generalized eigenvector $f_1(\mathbf{x})$ satisfies

$$f_1 = \arg \min_{\substack{\langle f, d_\varepsilon f \rangle = 1 \\ \langle f, d_\varepsilon \rangle = 0}} \langle f, Lf \rangle. \quad (16)$$

This implies that f_1 provides an optimal one-dimensional embedding of the data X that preserves local neighborhood. This property generalizes to any s -dimensional embedding via the first non-trivial generalized eigenvectors f_1, \dots, f_s , ($s \geq 1$). Thus, for a given $s \geq 1$, the *Laplacian eigenmap* defined as

$$\Psi : \mathbf{x} \mapsto (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_s(\mathbf{x}))^T \quad (17)$$

is a local-neighborhood-preserving embedding map from the data set X into \mathbb{R}^s .

Proposition 3.1. *[Relationship between L and the diffusion operator A in Eq. (3)] Suppose the unnormalized graph Laplacian L and the diffusion operator A (as in Eq. (3)) are defined on the same connected symmetric graph with weights $w_\varepsilon(\mathbf{x}, \mathbf{y})$. Let $d_\varepsilon(\mathbf{x}) = \sum_{\mathbf{y} \in X} w_\varepsilon(\mathbf{x}, \mathbf{y})$. Then, λ and $f(\mathbf{x})$ solve the generalized eigenvalue problem $Lf(\mathbf{x}) = \lambda d_\varepsilon(\mathbf{x})f(\mathbf{x})$ if and only if $f(\mathbf{x})$ is eigenvector of A corresponding to eigenvalue $(1 - \lambda)$.*

Proposition 3.2. *[Relationship between \mathcal{L} and the symmetric operator \tilde{A} in Eq. (6)] Suppose the normalized graph Laplacian \mathcal{L} and the symmetric operator \tilde{A} (as in Eq. (6)) are defined on the same connected symmetric graph with weights $w_\varepsilon(\mathbf{x}, \mathbf{y})$. Then,*

$$\mathcal{L}f(\mathbf{x}) = f(\mathbf{x}) - \tilde{A}f(\mathbf{x}).$$

Hence, $f(\mathbf{x})$ is eigenvector of \mathcal{L} corresponding to eigenvalue λ if and only if $f(\mathbf{x})$ is eigenvector of \tilde{A} corresponding to eigenvalue $1 - \lambda$.

Proofs of these propositions follow directly from the definitions. Furthermore, from the definition of the diffusion map Ψ_t and Proposition 3.1, we have the following

$$\Psi_t(\mathbf{x}) = ((1 - \lambda_1)^t f_1(\mathbf{x}), (1 - \lambda_2)^t f_2(\mathbf{x}), \dots, (1 - \lambda_s)^t f_s(\mathbf{x}))^T, \quad (18)$$

where λ_i and f_i are as in (15) above, provided the weighted graph are same in both cases.

Let us end this section by mentioning that eigenvectors of \mathcal{L} are used for dimension reduction in the spectral clustering algorithm proposed by Ng, Jordan, and Weiss in [16] and also in the elongated K -means algorithm [9]. Similarly, the eigenvectors of L are used in the Shi-Malik spectral clustering algorithm [17].

3.3. Out-of-sample multiscale extension via geometric harmonics

Our goal is in classifying newly obtained unlabeled data (or *test* data) based on a classification rule learned from the known labeled data (or *training* data). In order to make meaningful inference from the training data to the unlabeled data, we need to have the same low-dimensional representation for all datasets. That is, we need to embed test data into the same reduced space as the training data. Hence, it becomes necessary for us to extend the embedding map (which can be the diffusion map or the Laplacian eigenmap) computed on the training dataset to the test data. To perform this task, we employ the multiscale extension scheme proposed in [8], which is based on “geometric harmonics” originally introduced in [7, Chap. 3] and [18]. Let us call this scheme GHME (geometric harmonics multiscale extension) for short. We now review the GHME scheme.

The GHME scheme is an improvement of the Nyström extension method proposed in [19, 20]. Let X and Y denote the training set and the unlabeled test set, respectively. First consider the eigenvalues $\{\mu_\ell\}$ and orthonormal eigenfunctions $\{\varphi_\ell\}$ of a (symmetric) Gaussian kernel of width $\sigma > 0$ on the training set X :

$$\mu_\ell \varphi_\ell(\mathbf{x}) = \sum_{\mathbf{z} \in X} e^{-\|\mathbf{x}-\mathbf{z}\|^2/\sigma^2} \varphi_\ell(\mathbf{z}), \quad \mathbf{x} \in X, \quad (19)$$

where the nonnegative eigenvalues $\{\mu_\ell\}$ are sorted in decreasing order. From Equation (19), the Nyström extension of φ_ℓ from X to $\mathbf{y} \in Y$ is defined as

$$\bar{\varphi}_\ell(\mathbf{y}) \triangleq \frac{1}{\mu_\ell} \sum_{\mathbf{z} \in X} e^{-\|\mathbf{y}-\mathbf{z}\|^2/\sigma^2} \varphi_\ell(\mathbf{z}). \quad (20)$$

Since the eigenfunctions $\{\varphi_\ell\}$ form an orthonormal basis for $L^2(X)$, any function $f \in L^2(X)$ can be expanded as

$$f(\mathbf{x}) = \sum_{\ell} \langle f, \varphi_\ell \rangle \varphi_\ell(\mathbf{x}), \quad \mathbf{x} \in X.$$

Thus the Nyström extension of f from X to $\mathbf{y} \in Y$ can be defined as

$$\bar{f}(\mathbf{y}) \triangleq \sum_{\ell} \langle f, \varphi_\ell \rangle \bar{\varphi}_\ell(\mathbf{y}).$$

We observe that the range of the extension in (20) is proportional to σ . If the ratio $\|\mathbf{y} - \mathbf{z}\|/\sigma$ is large for all $\mathbf{z} \in X$, then $\bar{\varphi}_\ell(\mathbf{y})$ will be numerically small and hence may not be meaningful. Hence the extension scale σ should be as large as possible. However, for large enough σ , the Gaussian kernel in (19) becomes ill-conditioned, i.e., μ_ℓ tends to 0 more quickly compared to the case where σ is

small. Thus the Nyström extension in (20) will blow up. Furthermore, it is well known that the extension range depends on the smoothness of the function to be extended [7, Chap. 3], [18]. If f is fairly smooth, it can be extended far away from the training set. On the other hand, if f varies wildly on X , then it has limited extension range. To address the ill-condition issue, the GHME scheme considers the following approximate extension for f :

$$\bar{f}(\mathbf{z}) \triangleq \sum_{\ell: \eta\mu_\ell > \mu_0} \langle f, \varphi_\ell \rangle \bar{\varphi}_\ell(\mathbf{z}), \quad (21)$$

where $\eta > 0$ is some fixed condition number and $\mathbf{z} \in X \cup Y$. This extension \bar{f} is well-defined on $X \cup Y$, but it is not equal to f on the training set X . Observe that if the value of σ decreases, the eigenvalues $\mu_\ell \rightarrow 0$ more slowly. This allows more terms in (21), making \bar{f} a better approximation of f on X . Based on this observation, the GHME iteratively searches for an extension \bar{f} that approximates f on X with a pre-set error tolerance $\varrho > 0$ by slowly decreasing the value of the extension scale σ .

The GHME scheme can be summarized as follows:

Step 1: Suppose f is a function defined on the training set X and to be extended to a new dataset Y . Fix a condition number $\eta > 0$ and an error tolerance $\varrho > 0$. Set the extension scale $\sigma = \sigma_0$, for some large value σ_0 .

Step 2: Compute eigenvalues $\{\mu_\ell\}$ and orthonormal eigenfunctions $\{\varphi_\ell\}$ of the Gaussian kernel of width σ and expand f (on the training set X) in this eigenbasis

$$f(\mathbf{x}) = \sum_{\ell} \langle f, \varphi_\ell \rangle \varphi_\ell(\mathbf{x}), \quad \mathbf{x} \in X,$$

i.e., compute the coefficients $c_\ell \triangleq \langle f, \varphi_\ell \rangle$.

Step 3: On the training set X , approximate f by \bar{f} defined in (21). Compute the approximation error

$$Err \triangleq \left(\sum_{\ell: \mu_0/\mu_\ell \geq \eta} |c_\ell|^2 \right)^{1/2}.$$

If $Err > \varrho$, set $\sigma \leftarrow \frac{1}{2}\sigma$ and return to Step 2. Otherwise, continue.

Step 4: For each ℓ such that $\mu_0/\mu_\ell < \eta$, compute the Nyström extension

$$\bar{\varphi}_\ell(\mathbf{y}) = \frac{1}{\mu_\ell} \sum_{\mathbf{x} \in X} e^{-\|\mathbf{x}-\mathbf{y}\|^2/\sigma^2} \varphi_\ell(\mathbf{x}),$$

for all $\mathbf{y} \in Y$. And finally, compute the approximate extension \bar{f}

$$\bar{f}(\mathbf{y}) \triangleq \sum_{\ell: \mu_0/\mu_\ell < \eta} c_\ell \bar{\varphi}_\ell(\mathbf{y}).$$

4. Dataset Matching using Diffusion Maps or Laplacian Eigenmaps and EMD

We now describe how diffusion maps or Laplacian eigenmaps coupled with the Earth Mover’s Distance can be applied to perform datasets matching. Our approach quantitatively determines the dissimilarity between any two sets of points in high dimensional space (each set corresponds to an object).

4.1. Signature construction

As explained in Section 2, Earth Mover’s Distance measures the discrepancy between two discrete distributions. To apply EMD as a set-discriminant measure, we need to construct a signature for each dataset. This involves the following steps: (i) Perform dimension reduction on the training data using a diffusion map or a Laplacian eigenmap as the embedding map; (ii) Perform out-of-sample extension on the embedding map to embed the unlabeled data into the same reduced space as the training data; (iii) For each dataset (training and test), cluster the corresponding points in the reduced space to form the signature $P = \{(\mathbf{p}_1, w_{\mathbf{p}_1}), \dots, (\mathbf{p}_m, w_{\mathbf{p}_m})\}$, where $\{\mathbf{p}_j\}_{j=1}^m$ are cluster centers (these are points in the reduced space) and $w_{\mathbf{p}_j}$ is the density of cluster j , that is, the percentage of points in the dataset that fall into the cluster j .

4.1.1. Elongated K -means

To determine the intrinsic dimensionality of the data (the dimension of the reduced space) and simultaneously form point clusters within the reduced space, we apply the *elongated K -means* algorithm [9]. Elongated K -means (*ekmeans*) was adapted from the spectral clustering algorithm proposed in [16] by replacing the Euclidean distance with an elongated distance in the computation of point-to-center distances. We recall that the elongated distance between two points $\mathbf{x}, \mathbf{c} \in \mathbb{R}^n$ is defined in [9] as

$$\text{e-dist}(\mathbf{x}, \mathbf{c}) = (\mathbf{x} - \mathbf{c})^T M (\mathbf{x} - \mathbf{c}), \quad (22)$$

where $M = \frac{1}{\alpha} \left(I_n - \frac{\mathbf{c}\mathbf{c}^T}{\mathbf{c}^T\mathbf{c}} \right) + \alpha \frac{\mathbf{c}\mathbf{c}^T}{\mathbf{c}^T\mathbf{c}}$. The parameter α controls the elongation of the cluster (the smaller, the more elongated the cluster). In other words, the *ekmeans* algorithm groups points lying inside a thin long ellipsoid to form a cluster,

as opposed to inside a sphere. In all of our numerical experiments, we set $\alpha = 0.2$, the value recommended by the authors in [9].

To motivate our consideration of *ekmeans*, let us repeat the analysis given in [9] which examines the ideal scenario when the data consists of K clusters widely separated from each other. In this case the matrix of the kernel $\tilde{k}(\mathbf{x}, \mathbf{y})$ in (5), (with rows re-ordered by clusters if necessary) is block diagonal with exactly K blocks. (Recall that this is the symmetric operator \tilde{A} in (6), and from Proposition 3.2 it has same eigenvectors as the normalized graph Laplacian \mathcal{L}). Thus, it has K eigenvectors associated with the largest eigenvalue 1, one eigenvector for each cluster. Each eigenvector has ones in the entries corresponding to the points in the cluster and zeros elsewhere. Suppose we perform a spectral embedding of the data (using these top K eigenvectors) into the top K eigenspace (the space spanned by the top K eigenvectors). The data would get mapped to K clusters at the K unit vectors on the coordinate axes. In general, rotations may occur, depending on the computation of the eigenvectors. In other words, any set of K mutually orthogonal vectors in the top K eigenspace is an admissible set of eigenvectors associated with eigenvalue 1. Furthermore, eigenvectors are usually normalized. These two situations lead to K elongated clusters lying along some K mutually orthogonal directions within the top K eigenspace (instead of on the coordinate axes).

We observe that when the data are embedded into the top q eigenspace with $q < K$, or equivalently, when we project the K elongated clusters down to the q -dimensional subspace spanned by the first q eigenvectors, the results are elongated clusters lying along radial directions and (possibly) some dense clusters near the origin. The clusters near the origin are the projection image of those clusters that lie elongated along the directions orthogonal to this q -dimensional subspace. On the other hand, suppose we embed the data into the top q eigenspace with $q > K$. We would find no additional cluster other than the K elongated clusters already accounted for. The reason behind this phenomenon is that each of the eigenvectors after the K th eigenvector are close to the zero vector. In other words, large separations in the data are already captured in the top K eigenspace. Consequently, increasing the dimension of the embedding spectral space does not affect the clustering behaviors in the data.

ekmeans exploits the geometric properties of the eigenvectors of $\tilde{k}(\mathbf{x}, \mathbf{y})$ to cluster the data and automatically determine the number of intrinsic clusters. That is, *ekmeans* does not require input of the number of clusters. To determine the number of intrinsic clusters automatically, *ekmeans* starts the clustering process in the top 2 eigenspace with three centers initialized, two centers at two different elongated clusters and one at the origin. If there are more than two elongated clusters, the center at the origin will be dragged to a cluster not accounted for. Then the algorithm moves the clustering process to the top 3 eigenspace, adds a

center at the origin, and repeats the process until no additional cluster is found. This clustering process stops at the top K eigenspace if there are K (intrinsic) clusters in the data.

In practice, the data we handle may not be widely separated, thus we may not have more than one eigenvector associated with the largest eigenvalue 1. However, the eigenvectors still have similar geometric properties if there are K tight clusters in the data causing a spectral gap at the $(K + 1)$ st eigenvalue (see [16]). The same geometric properties are passed on to the embedding of the data via the Laplacian eigenmaps and the diffusion maps (since these are constructed from the eigenvectors of $\tilde{k}(\mathbf{x}, \mathbf{y})$ conjugated by $\sqrt{\phi_0}$). Thus, the Laplacian eigenmap and the diffusion map in our consideration both satisfy the necessary properties under the framework of the elongated K -means algorithm.

Our accomplishment in utilizing *ekmeans* is twofold: to determine the intrinsic dimensionality of the data and to form representative clusters for the signature of each dataset. Since *ekmeans* determines the intrinsic number of clusters in the data based on geometric properties, this number can be considered as the intrinsic dimensionality of the data. We take advantage of this aspect of *ekmeans* to determine the dimension s of our reduced space \mathbb{R}^s . We explain how to select s in more details in Section 4.2.

Now, suppose we have N datasets, X^1, \dots, X^N , with each X^j containing n_j points. Suppose *ekmeans* has determined that dataset X^j has K_j clusters. This means the cluster centers for set X^j determined by *ekmeans* are vectors in the top K_j -dimensional subspace of the reduced space \mathbb{R}^s (where s is determined as described in Section 4.2). However, all centers must be in \mathbb{R}^s in order to input into EMD. Thus, to bring all centers up to \mathbb{R}^s , we re-cluster each set of embedded points in \mathbb{R}^s by running K -means with elongated distance to reform the K_j clusters. Here, we use the previous cluster memberships as a starting condition for K -means. At the end of this re-clustering process, all signatures corresponding to the datasets contain centers in the reduced space \mathbb{R}^s .

4.2. Parameters selection

There are four parameters to be determined in the first stage of our proposed method: the scale ε for the diffusion kernel $k(\mathbf{x}, \mathbf{y})$ defined in (2); the dimension s of the reduced (embedding) space \mathbb{R}^s ; the error tolerance ϱ in approximating the extension of the embedding map; and the cutoff bound η for the condition number of the extension kernel. Clearly we have to select each of these parameters wisely. Let us give some rules of thumb for selecting these parameters.

In computing the diffusion maps, the scale $\varepsilon > 0$ for the diffusion kernel should be chosen so that when we form the graph with Gaussian weight $w_\varepsilon(\mathbf{x}, \mathbf{y})$ on the edge between points \mathbf{x} and \mathbf{y} the graph is numerically connected. Connectedness of

the graph guarantees the existence and uniqueness of the stationary distribution ϕ_0 of the Markov process on the graph with transition probabilities defined in Eq. (2) (see Appendix I in [8]), and the construction of the diffusion maps depends on the existence and uniqueness of ϕ_0 . Therefore, when computing diffusion maps, the value for ε must be large enough to ensure that every point in the graph is connected to at least one other point. However, it is clear that when ε is too large any affinity or dissimilarity between the data points is obscured, since w_ε converges to 1 as ε increases to infinity. One suggestion given in [14] is to choose ε “*in the order of the mean distance of a point to its k th nearest neighbor* (where k is proportional to the logarithm of the number of points in the graph). In our numerical experiments, we select ε to be the mean of the Euclidean distances from each point to its k -nearest neighbor, where k equals to 5% of the total number of points in the training set. In other words, choose ε so that approximately 5% of all distances between pairs of points are less than or equal to ε . This means approximately 5% percent of all possible edges in the graph have weights greater than or equal to e^{-1} , the rest have smaller weights, i.e., the graph is sparse but not too sparse. The spectrum of the diffusion kernel decays relatively fast with this choice of ε . For example, in Fig. 1 we plot the largest 100 eigenvalues of the density-invariant normalized diffusion kernel taken from one trial of the lip-reading experiment. The value of ε determined by the 5% heuristic is $\varepsilon = 740$. We see that the eigenvalues decrease quickly. Fast decay of the spectrum implies that any random walk initiated on the graph converges quickly to steady state and that the diffusion distance can be approximated more accurately with a smaller number of eigenfunctions. In other words, we will be able to detect clustering behaviors in the data with a small number of time steps t . In the numerical experiments below, we use $t = 1$. We do not need to set the Markov process forward in time. In these cases, having found the value ε appropriate for the data is enough for identifying grouping patterns in the data.

When computing the Laplacian eigenmap in our numerical experiments, we start with the 5% heuristic described above, then perform exhaustive search for an optimal value for ε by cross-validation. In each iteration of the search, we divide the training datasets into halves, then run our classification algorithm. Then we increase k (described above) by 5 if the classification is improving, otherwise we decrease k by 5, then move to next iteration. In our numerical experiments, starting with the 5-percent heuristic described above, we only need on average 15 iterations of cross-validation to find an optimal value for ε .

We note that in the case of the diffusion maps, we have also tried exhaustive search for an optimal ε by cross-validation. The values for ε found by exhaustive search turned out to be very close to the values determined the 5% heuristic above.

The dimension s of the reduced (embedding) space can be determined by tak-

ing advantage of the geometrically grounded properties of the *ekmeans* algorithm. As described in Section 4.1, when we cluster each dataset using *ekmeans*, the intrinsic dimension of the dataset is automatically determined. Suppose our training set consists of a total of N datasets, X^1, \dots, X^N , belonging to C different classes, *ekmeans* will find an intrinsic dimension K_j for each set X^j . This number K_j is also the intrinsic number of clusters in the set. As discussed in Section 4.1, this intrinsic number of K_j clusters does not change when the set X^j is embedded into any “eigenspace” of dimension greater than K_j . Therefore, it is natural to set the dimension of the (embedding) reduced space for all of our data to be the maximum of K_j over all $j = 1, \dots, N$, that is,

$$s \triangleq \max_{1 \leq j \leq N} K_j.$$

When choosing a value for the error tolerance ϱ for the approximation of the out-of-sample extension of the embedding map, we should keep in mind that small error limit means small extension range. Suppose we know a priori that our training set is a good representative of a manifold or data space (that is, there are no missing gap so that we can completely capture the shape of the manifold from the training data) and the unlabeled data lie on the manifold, then the approximation of the extension is fairly accurate, thus we can set ϱ to be small. A heuristic value to set for ϱ is 1% of the size of the test data. This gives on average a bound of 0.01^2 (from **Step 3** of the GHME scheme) on the error at each point where the extension is being computed. In our numerical experiments, we use this heuristic to set a value for ϱ .

To determine a cutoff lower bound η for the condition number of the Gaussian extension kernel $e^{-\|\mathbf{x}-\mathbf{y}\|^2/\sigma^2}$ in (19), we have to keep in mind the approximation error tolerance ϱ . If ϱ is small, then η has to be large. In addition, as σ increases, the condition number of the kernel also increases. To predict how large η might get, we can take advantage of the symmetric kernel $\tilde{k}(\mathbf{x}, \mathbf{y})$, which we already computed from the Gaussian weights w_ε with ε optimally chosen for the data. Let κ be the condition number of $\tilde{k}(\mathbf{x}, \mathbf{y})$. It is easy to show that when $\sigma = \varepsilon$, the condition number of $e^{-\|\mathbf{x}-\mathbf{y}\|^2/\sigma^2}$ is proportional to κ . Furthermore, as σ grows, the condition number of the Gaussian kernel will only get worse. Thus, we can consider setting η larger than κ and inversely proportional to ϱ . In our numerical experiments, we set $\eta = \kappa/\varrho$, if $\kappa < 10^5$, and $\eta = 10^5/\varrho$ otherwise.

4.3. Algorithm for classification of datasets

We summarize our proposed method for classification of datasets in the following algorithm:

Algorithm 4.1. [Classification of datasets by DM, LE, and EMD]

0. Let X and Y denote the training data and the unlabeled data, respectively. Also, $X = \cup_i X^i$, where X^i is a set containing all signals characterizing one object, e.g., all image frames in one video sequence. Similarly, $Y = \cup_j Y^j$. There are C classes, and each X^i is known to belong to one of the C classes.

1. Signature construction in reduced (embedding) space:

- i. Construct the diffusion map Ψ_t or the Laplacian eigenmap Ψ on the training data X , then embed X into a reduced (embedding) space \mathbb{R}^s . The dimension s is determined by using ekmeans as described in Section 4.2.
- ii. Extend Ψ_t or Ψ to the unlabeled data Y (this embeds Y into \mathbb{R}^s).
- iii. For each i th set of embedded points corresponding to X^i , construct a signature $P^i = \{(\mathbf{p}_1^i, w_{\mathbf{p}_1^i}^i), \dots, (\mathbf{p}_m^i, w_{\mathbf{p}_m^i}^i)\}$. Likewise, construct a signature $Q^j = \{(\mathbf{q}_1^j, w_{\mathbf{q}_1^j}^j), \dots, (\mathbf{q}_n^j, w_{\mathbf{q}_n^j}^j)\}$ for each j th set of embedded points corresponding to Y^j .

2. Classification via EMD:

- i. Compute EMD between P^i and Q^j for all possible pairs (i, j) . Define the cost of moving one unit mass from \mathbf{p}_k^i to \mathbf{q}_ℓ^j to be

$$c(\mathbf{p}_k^i, \mathbf{q}_\ell^j) \triangleq \frac{1}{2} \|\mathbf{p}_k^i - \mathbf{q}_\ell^j\|^2,$$

where $\|\cdot\|$ is the Euclidean distance in the reduced space \mathbb{R}^s . Let $D_{ij} \triangleq \text{EMD}(P^i, Q^j)$, as defined in (1).

- ii. For each j , suppose $i_j \triangleq \arg \min_i D_{ij}$. Label Y^j with the label of X^{i_j} . That is, assign label by the nearest neighbor using EMD distance.

In Step 2.i we define the cost of moving one unit of mass from center \mathbf{p}_k^i to center \mathbf{q}_ℓ^j to be proportional to the squared (instead of to the first power) of the Euclidean distance between the two centers so as to give more preference to very close clusters.

5. Numerical Experiments and Results

We now illustrate how our proposed algorithm can be applied to classification problems where the data characterizing each object consist of a set of signals instead of a single signal. We will show two examples of application. The first example is classification of underwater objects via analyzing Synthetic Aperture Sonar

(SAS) waveforms reflected from the objects. The second example is a lip-reading application in which we identify the spoken word from a sequence of image frames extracted from a silent video segment. We will consider four ways of performing dimension reduction: (i) via diffusion map (DM) Ψ_t with $t = 1$; (ii) via Laplacian eigenmap (LE); (iii) via eigenvectors of $k(\mathbf{x}, \mathbf{y})$ ($\mathcal{L}\mathcal{E}$) – these are also eigenvectors of normalized graph Laplacian (14) and are utilized for spectral clustering in [9]; and (iv) Principal Component Analysis (PCA).

The cases (i),(ii),(iii) fall directly within the framework of our proposed classification algorithm. For PCA, we first computed the eigenvectors (PCA vectors) of the covariance matrix of the training data (see [22, Sec.2.1] for efficient computation of PCA vectors). Then, we retain the top s PCA vectors associated with the largest eigenvalues and use these as a basis for the reduced (PCA) space. The choice for $s > 0$ is made based the decay of the eigenvalues. We choose s to be the cutoff point where the s largest eigenvalues decrease most rapidly. The remaining the eigenvalues are relatively small and decrease slowly. (In the classification of underwater objects example, we found $s = 10$, and in the lip-reading example $s = 15$.) Using these s basis vectors, we project both training and test data onto the corresponding s -dimensional PCA space. Finally we apply the remaining steps in Algorithm 4.1 to construct signatures and then label the test data.

We will also compare the discriminative performance of EMD and the Hausdorff distance (HD). HD was considered in the LKC method [8]. We recall that the HD between any two sets S_1 and S_2 is defined as

$$d_H(S_1, S_2) \triangleq \max \left(\max_{\mathbf{y} \in S_2} \min_{\mathbf{x} \in S_1} \|\mathbf{x} - \mathbf{y}\|, \max_{\mathbf{x} \in S_1} \min_{\mathbf{y} \in S_2} \|\mathbf{x} - \mathbf{y}\| \right),$$

where $\|\cdot\|$ denotes the Euclidean distance.

5.1. Classification of underwater objects

The data in this example are collected from three different controlled experiments in a fresh water test pond at NSW-PC. For details of the experiments, see [23]. In each of the three experiments, two objects were placed – either buried in the sand or proud – at the bottom of the pond. One of the objects was a sphere made of an iron casing filled with a different material each time. The other object was a solid aluminum cylinder of different length in each experiment. A sinusoidal pulse was transmitted across the floor of the pond and the reflected signal was recorded over a period of time at uniform time intervals. The data obtained contain waveforms reflected from the entire area of the pond floor. Waveforms corresponding to objects are extracted and processed using an improved version of the algorithm presented in [24]. This yields one set of rectangular blocks of waveforms per object.

Our goal is to identify objects according to their material compositions regardless of their shapes. We name the sphere and the cylinder in Experiment j as S_j and C_j , for $j = 1, 2, 3$. Sphere S_1 was filled with air, so we categorize it as one class with label **IA** for iron-air. Spheres S_2 and S_3 were filled with silicone oil so we group them into another class with label **IS** for iron-silicone. All three cylinders were of the same diameter and of the same material, so we grouped them into one class with label **AI** for aluminum. However, we note that C_1 and C_2 were of the same length while C_3 was much shorter.

The waveform data is of extremely high dimension. Each rectangular block of waveforms is a 2D array of size 17 (cross range samples) by 600 (time samples), see Fig. 4(a) for some examples. The set of waveforms characterizing the spheres S_1 , S_2 , and S_3 contains 8, 8, and 16 blocks, respectively; and those characterizing the three cylinders have 32 blocks each. We treat each rectangular block as a point in $\mathbb{R}^{17 \times 600}$. Then we apply the steps in Algorithm 4.1 to identify the test object.

In our numerical experiment, we set aside one set of waveforms (corresponding to one object) to use as test data and train our algorithm on the remaining five sets. We cycle through all six objects, that is, we repeat the classification process six times. The classification results for all six runs are shown in Table 1.

Using EMD coupled with nonlinear dimension reduction (DM, LE, $\mathcal{L}E$), we consistently and correctly identify all three cylinders as objects of class **AI** and the spheres S_2 and S_3 as objects of class **IS**. Moreover, the mistake of labeling the sphere S_1 as **AI** is also consistent. Note that this error is expected since the class **IA** contains only one member S_1 . We have no training data for this class when the sphere S_1 is left out as test data. Furthermore, note that having S_1 as part of the training data does not confuse the identification of the spheres S_2 and S_3 when nonlinear dimension reduction is applied. This is not the case when PCA is used for dimension reduction. We will discuss more on this in Section 5.3 below.

Classification of the objects using HD is not so consistent among the different dimension reduction methods. The main reason for this is that HD is highly sensitive to outliers. For a closer look, let us examine the distribution of the embedded points in the *diffusion* space – the reduced space obtained via a diffusion map. Fig. 2 shows three sets of (embedded) points projected onto the first three diffusion coordinates. The diffusion map in this case is computed from the training data consisting of all three cylinders and the spheres S_1 and S_3 . The sphere S_2 is first left out as test data then embedded into the same diffusion space via the out-of-sample extension GHME scheme. In the figure, blue crosses correspond to cylinder C_3 (class **AI**), green triangles correspond to sphere S_3 (class **IS**), and red circles correspond to the unlabeled (test) object S_2 (true label is **IS**). Black stars are cluster centers – the spatial representatives in the signature of each object.

We see that the red circles (object S_2) are on average close to the green trian-

gles ($S3$), but because of the outlier at the bottom left of the plot, the HD between the $S2$ and $S3$ is smaller than that between $S2$ and $C3$. The actual values are shown in Table 2. Note that the dimension of the reduced (diffusion) space is actually 13 not 3 in this case. We can see from Table 2 that the smallest EMD value is 0.3556 corresponding to $S3$ and the smallest HD value is 1.1761 corresponding to $C3$. Thus, EMD correctly labels object $S2$ as **IS**, but HD mislabels $S2$ as **AI**.

5.2. Lip reading experiment

In this section, we present a simplified version of the lip-reading problem to illustrate how our proposed algorithm can be applied in practice. The objective of lip reading is to train a machine to automatically recognize the spoken words from the movements of the lips captured on silent video segments (no sound is involved). Much research effort has been devoted to this area. Many published algorithms involved sophisticated feature selection. In this example, we simply perform dimension reduction on the sequences of images which we treat as sets of points. We do not extract any lip feature from the images (such as those in [25, 26] and many other publications). Furthermore, the lips data we use are collected from one speaker. More sophisticated feature selection might be necessary when more speakers, i.e., more variations in the lips, are involved.

We recorded a subject speaking the first five digits (‘one’,...,‘five’) ten times using a Nikon Coolpix digital camera sampling at a rate of 60 frames per second. We then extracted the image frames from each movie clip and did some simple processing. First, we convert the images from color to gray scales ranging from 0 to 255. Then we cropped each image to a 55×70 pixels window around the lips to compensate for translations. (The speaker’s nose was marked with a color marker to facilitate automatic cropping of the image frames). Each cropped frame is treated as a point in $\mathbb{R}^{55 \times 70}$.

For each spoken digit, we randomly selected five image sequences from the ten in our collection to use as training data. This gives us a total of 25 sequences for training data and 25 for test data. We apply Algorithm 4.1 to identify the test sequences. We repeat the whole process 100 times. The total misclassification rates (averaging over 100 experimental trials) are shown in Table 3. Again, we see that using EMD gives smaller recognition errors than using HD, regardless of the dimension reduction techniques used.

Table 3 shows similar high classification errors for all local nonlinear dimension reduction method in our consideration. On the other hand, EMD coupled with PCA makes only 3.5% classification error.

5.3. PCA and local nonlinear dimension reduction

We have seen that PCA performs really well as a dimension reduction method compared to the local nonlinear methods DM, LE, and $\mathcal{L}E$ in the lip-reading experiment above. To understand why, we need to take a closer look at the variations between the image sequences. In Fig. 3 we display some images selected from two different sequences corresponding to the word ‘one’ and also the embedding into the first three coordinates of the diffusion space and the PCA space of all images belonging to the two sequences (some images are not displayed). In the first sequence (top row) the speaker simply spoke the word. In the second sequence (bottom row) the speaker was smiling while speaking. If we look at the diffusion coordinates of these two sequences, smiling while speaking the same word translates to local perturbations between the two sets of images. Diffusion maps (and the other nonlinear maps in our consideration) possess the property that preserves local neighborhood. Thus, the two image sequences that get embedded into the (reduced) diffusion space have somewhat different (although overlapping) distributions of points. On the other hand, the corresponding two distributions of points in the PCA space are not much different. As a consequence, the EMD between the two corresponding signatures constructed in the PCA space is much smaller than the two constructed in the diffusion space.

The same local-neighborhood preserving property is what makes local nonlinear dimension reduction such as diffusion maps and Laplacian eigenmaps more appropriate for application to the sonar data. The variations between sets of waveforms corresponding to different objects are small and local. In Fig. 4(a), we display some samples of waveforms reflected from the three spheres in the classification of underwater objects experiment. Each rectangular block is viewed as one data point. Our eyes can easily discern the difference between the three sets. However, the variations are small and local. In fact, PCA just treated the variations as noise. In 4(b),(c) we plot the projection of the embedding of all waveforms onto the first three coordinates of the diffusion space and the PCA space. The training data for the computation of the diffusion map and principal components in this case are the three cylinders and the spheres $S1$ and $S2$. We see that diffusion maps embed $S3$ close to $S1$ in the diffusion space, whereas PCA embeds $S3$ close to $S1$. Recall the classification results from Table 1. When we use nonlinear dimension reduction coupled with EMD, $S3$ is correctly identified with $S2$, whereas $S3$ is identified as $S1$ (and vice versa) when we use PCA.

6. Conclusion

We have proposed an algorithm for classification of objects that are characterized or described by sets of signals, as opposed to one single signal, using the Earth

Mover's Distance (EMD). Our algorithm sets up the framework for application of EMD to such classification problems.

We have shown that EMD is more robust to noise and hence more appropriate for discrimination of sets than the Hausdorff distance. Furthermore, we have illustrated how to apply local nonlinear manifold-approximation based dimension reduction methods such as the diffusion maps and the Laplacian eigenmaps to construct signatures for the datasets. We have shown that these methods give similar results when applied to reduce dimensionality in the data in order to facilitate the classification process.

We have provided two examples of practical applications for our proposed algorithm: classification of underwater objects and lip reading. The sonar signals reflected from underwater objects have local patterns – small variations among some variables that appear to characterize the spheres of class **IS**. These features are important for the correct identification of the objects. PCA misses this and thus misidentified some of the spheres. Diffusion maps and Laplacian eigenmaps on the other hand were successful at identifying all the objects. The lip data in the lip-reading experiment also contain local variations. However, these are undesirable features and should be considered as noise. In this case, PCA performs very well compared to Diffusion maps and Laplacian eigenmaps.

Finally, we would like to comment that our application of the nonlinear manifold-based dimension reduction methods in the lip-reading experiments may not do justice to these local nonlinear methods. The data of the lips we considered are limited to a tiny number of instances of the words spoken. They do not give much (if any) information about the manifold on which they may lie. This manifold can be approximated much better from data that include many possible shapes of the lips. We believe that our lip-reading experiment illustrates an incorrect way to apply diffusion maps and Laplacian eigenmaps.

Acknowledgments

This work was partially supported by the ONR grants N00014-06-1-0615, N00014-07-1-0166, N00014-09-1-0041, N00014-09-1-0318, the NSF grant DMS-0410406, and the NSF VIGRE grants DMS-0135345, DMS-0636297. A preliminary version of a subset of the material in this paper was presented at the SPIE Wavelets XII Conference, in San Diego, in August 2007 [27]. We thank Dr. Quyen Huynh and Dr. Joe Lopes of NSWC-PC for providing the experimental sonar data. We have used the SPECTRAL Toolbox version 0.1 distributed on the web by Guido Sanguinetti and Jonathan Laidler to compute Elongated Kmeans. Finally, we would like thank Bradley Marchand for his help in processing the sonar data used in the numerical experiments.

References

- [1] Y. Rubner and C. Tomasi, *Perceptual Metrics for Image Database Navigation*, Kluwer Academic Publishers, Boston, 1999.
- [2] Y. Rubner, C. Tomasi, L. J. Guibas, “The earth mover’s distance as a metric for image retrieval”, *Int. J. Comput. Vision*, **40**(2), pp. 99–121, 2000.
- [3] M. Belkin and P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering”, *Advances in Neural Information Processing Systems*, pp. 585–591, 2001.
- [4] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation”, *Neural Computation*, **15**(6), pp. 1373–1396, 2003.
- [5] M. Belkin and P. Niyogi, “Towards a theoretical foundation for Laplacian-based manifold methods”, in *Learning Theory: 18th Annual Conference on Learning Theory, COLT 2005* (P. Auer and R. Meir, eds.), pp. 486–500, 2005.
- [6] R. R. Coifman and S. Lafon, “Diffusion maps”, *Appl. Comput. Harmon. Anal.*, **21**, pp. 5–30, July 2006.
- [7] S. Lafon, “Diffusion Maps and Geometric Harmonics”, Ph.D. Dissertation, Yale University, May 2004.
- [8] S. Lafon, Y. Keller, R.R. Coifman, “Data fusion and multicue data matching by diffusion maps”, *IEEE Trans. Pattern Anal. Machine Intell.*, **28**(11), pp. 1784–1797, 2006.
- [9] G. Sanguinetti, J. Laidler, N. D. Lawrence, “Automatic Determination of the Number of Clusters Using Spectral Algorithms”, *Proc. 2005 IEEE Workshop on Machine Learning for Signal Processing*, pp. 55–60, 2008.
- [10] S. Roweis and L. Saul, “Nonlinear dimensionality reduction by locally linear embedding”, *Science*, **290**, pp. 2323–2326, 2000.
- [11] D. Donoho and C. Grimes, “Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data”, *Proc. Natl. Acad. Sci. USA*, **100**(10), pp. 5591–5596, 2003.
- [12] Z. Zhang and H. Zha, “Principal manifolds and nonlinear dimension reduction via local tangent space alignment”, *SIAM J. Sci. Comput.*, **26**(1), pp. 313–338, 2005.

- [13] S. Lafon and A. B. Lee, “Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning and data set parameterization”, *IEEE Trans. Pattern Anal. Machine Intell.*, **28**(9), pp. 1393–1403, 2006.
- [14] U. von Luxburg, “A Tutorial on Spectral Clustering”, *Statistics and Computing*, **17**(4), pp. 395-416, 2007.
- [15] F. Chung, *Spectral Graph Theory*, CBMS Series, Vol. 92, AMS, Providence, RI, 1997.
- [16] A. Y. Ng, M. I. Jordan, Y. Weiss, “On spectral clustering: Analysis and an algorithm”, *Advances in Neural Information Processing Systems 14*, MIT Press, pp. 849–856, 2002.
- [17] J. Shi and J. Malik, “Normalized cuts and image segmentation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(8), pp. 888–905, 2000.
- [18] R. R. Coifman and S. Lafon, “Geometric harmonics”, *Appl. Comput. Harmon. Anal.*, **21**, pp. 31–52, July 2006.
- [19] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, “Spectral grouping using the Nyström method”, *IEEE Trans. Pattern Anal. Machine Intell.*, **26**(2), pp. 214–225, 2004.
- [20] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet, “Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering”, *Advances in Neural Information Processing Systems*, **16**, pp. 177–184, MIT Press, 2004.
- [21] L. C. Evans, “Partial Differential Equations and Monge-Kantorovich Mass Transfer” (lecture notes), www.math.berkeley.edu/~evans/Monge-Kantorovich.survey.pdf.
- [22] N. Saito, “Image approximation and modeling via least statistically dependent bases”, *Pattern Recogn.*, **34**, pp. 1765–1784, 2001.
- [23] C. L. Nesbitt and J. L. Lopes, “Subcritical detection of an elongated target buried under a rippled interface”, In *Oceans '04, MTS/IEEE Techno-Ocean '04*, **4**, pp. 1945–1952, 2004.
- [24] B. Marchand, N. Saito, and H. Xiao, “Classification of objects in synthetic aperture sonar images”, *Proc. 14th IEEE Statistical Signal Processing Workshop*, pp. 433–437, 2007.

- [25] E. K. Patterson, S. Gurbuz, Z. Tufekci, and J. Gowdy, “Moving-talker speaker-independent feature study and baseline results using the cuave multi-modal speech corpus”, *EURASIP Journal on Applied Signal Processing*, **11**, pp. 1189-1201, 2002.
- [26] X. Zhang and R. M. Mersereau, “Lip feature extraction towards an automatic speechreading system”, *Proc. 2000 International Conference on Image Processing*, **3**, pp. 226-229, 2000.
- [27] L. Lieu and N. Saito, “Automated discrimination of shapes in high dimensions”, in *Wavelets XII* (D. Van De Ville, V. K. Goyal, and M. Papadakis, eds.), *Proc. SPIE* **6701**, Paper # 67011V, 2007.

Object		<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>S1</i>	<i>S2</i>	<i>S3</i>
True Label		AI	AI	AI	IA	IS	IS
DM	EMD	AI	AI	AI	AI	IS	IS
	HD	AI	AI	AI	AI	AI	IS
LE	EMD	AI	AI	AI	AI	IS	IS
	HD	AI	AI	AI	AI	AI	IS
$\mathcal{L}E$	EMD	AI	AI	AI	AI	IS	IS
	HD	AI	AI	AI	IS	IS	IS
PCA	EMD	AI	AI	AI	IS	IS	IA
	HD	AI	AI	AI	IS	IS	IA

Table 1: Identification of Underwater Objects

Object	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>S1</i>	<i>S3</i>
EMD	0.4297	0.4904	0.4654	0.9338	0.3556
HD	2.8636	2.5512	1.1761	1.3619	1.7352

Table 2: EMD and HD values between sphere object *S2* and all other objects.

DM		LE		$\mathcal{L}E$		PCA	
EMD	HD	EMD	HD	EMD	HD	EMD	HD
25.8%	26.9%	24.1%	30.2%	21.8%	25.6%	3.5%	8.7%

Table 3: Lip-Reading total recognition errors. The errors are averaged over 100 experimental trials.

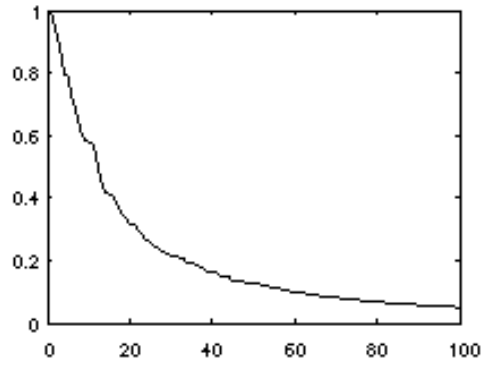


Figure 1: Largest 100 eigenvalues of the diffusion kernel in one trial of the lip-reading experiment. ($\varepsilon = 740$).

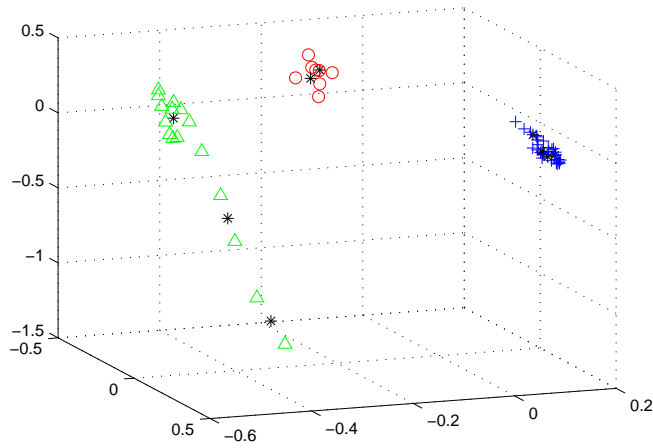
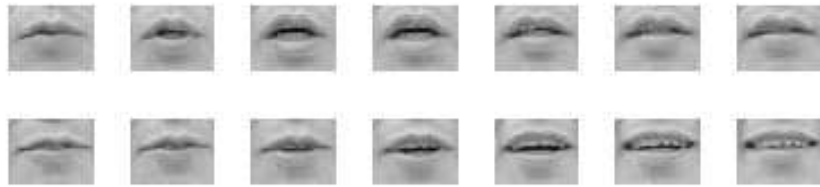
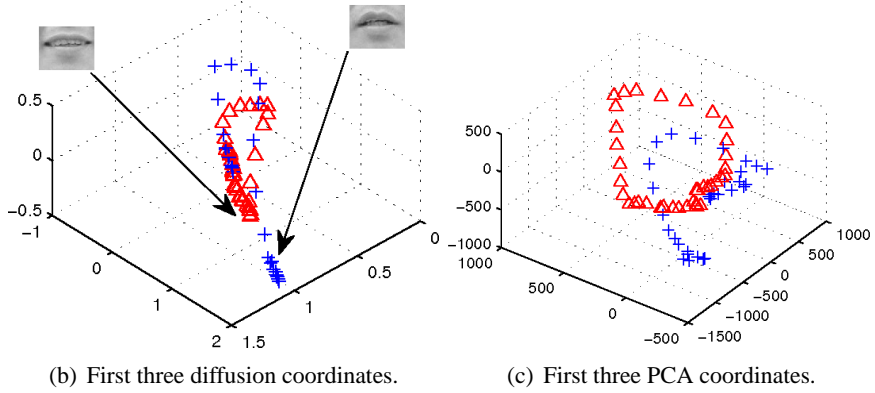


Figure 2: First three diffusion coordinates of three underwater objects. Blue crosses: $C3$. Green triangles: $S3$. Red circles: unlabeled (test) object $S2$. Black stars are cluster centers. The diffusion maps in this case is computed from the training data consisting of all three cylinders and the spheres $S1$ and $S3$.



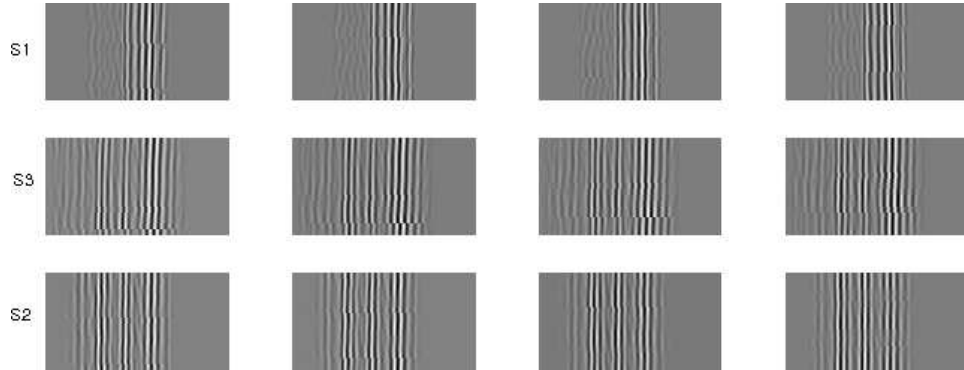
(a) Some images from two sequences (instances) of the word 'one'.



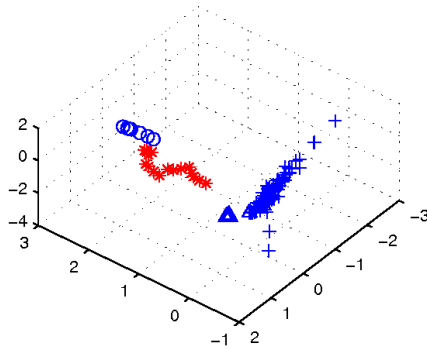
(b) First three diffusion coordinates.

(c) First three PCA coordinates.

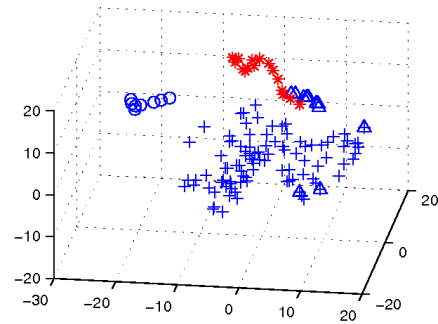
Figure 3: Two different instances of the word 'one'. In (b), (c): embedding of the two sets of images into the diffusion space and PCA space. Red triangles corresponding to the first sequence (top), blue crosses correspond to second sequence (bottom).



(a) Some waveforms from the three spheres. Top to bottom: $S1$, $S3$, $S2$.



(b) First three diffusion coordinates.



(c) First three PCA coordinates.

Figure 4: Selected waveforms corresponding to the three spheres in the underwater object experiment. In (b), (c): embedding of all waveforms corresponding to all six objects into the diffusion space and the PCA space. In this case, the training data consist of all three cylinders and the spheres $S1$ and $S2$. The test data is sphere $S3$. Blue crosses: all cylinders. Blue triangles: sphere $S1$. Blue circles: sphere $S2$. Red stars: sphere $S3$.