Appl. Comput. Harmon. Anal. 38 (2015) 399–419

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha

Approximately-isometric diffusion maps

Moshe Salhov [a,b], Amit Bermanis [a], Guy Wolf [a], Amir Averbuch [a,*]

[a] School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel
[b] Department of Mathematical Information Technology, University of Jyväskylä, Finland

ARTICLE INFO

Article history:
Received 8 July 2013
Received in revised form 25 May 2014
Accepted 30 May 2014
Available online 4 June 2014
Communicated by Amit Singer

Keywords:
Dimensionality reduction
Manifold learning
Kernel PCA
Diffusion maps
Diffusion distance
Distance preservation

ABSTRACT

Diffusion Maps (DM), and other kernel methods, are utilized for the analysis of high dimensional datasets. The DM method uses a Markovian diffusion process to model and analyze data. A spectral analysis of the DM kernel yields a map of the data into a low dimensional space, where Euclidean distances between the mapped data points represent the diffusion distances between the corresponding high dimensional data points. Many machine learning methods, which are based on the Euclidean metric, can be applied to the mapped data points in order to take advantage of the diffusion relations between them. However, a significant drawback of the DM is the need to apply spectral decomposition to a kernel matrix, which becomes infeasible for large datasets.

In this paper, we present an efficient approximation of the DM embedding. The presented approximation algorithm produces a dictionary of data points by identifying a small set of informative representatives. Then, based on this dictionary, the entire dataset is efficiently embedded into a low dimensional space. The Euclidean distances in the resulting embedded space approximate the diffusion distances. The properties of the presented embedding and its relation to DM method are analyzed and demonstrated.

1. Introduction

Recent methods process massive amounts of high dimensional data by utilizing a manifold structure on which data points are assumed to lie. This manifold is immersed in the ambient space that is defined by observable/measurable parameters. Kernel methods are designed to support data analysis tasks by utilizing the intrinsic manifold geometry. These methods are based on a kernel matrix that is designed to quantify the similarity between data points on the manifold. Spectral analysis of the kernel in these methods reveals the internal geometric structure of the data [10]. This analysis decomposes the designed kernel and generates eigenvectors that map the data from the ambient space into an embedded space that is usually

* Corresponding author. Fax: +972 3 6422020.
E-mail address: amir@math.tau.ac.il (A. Averbuch).

http://dx.doi.org/10.1016/j.acha.2014.05.002
1063-5203/

low dimensional. Spectral kernel methods have an impact on a wide range of optimization problems from graph coloring [4,3,2] to image segmentation [27] and web search [8].

Kernel methods extend the classic Multi Dimensional Scaling (MDS) method [21] by replacing its Gram matrix with a kernel matrix whose spectral decomposition preserves similarities between data points instead of preserving the inner products that MDS preserves. Some examples for kernel methods are: LLE [24], Isomaps [32], Laplacian eigenmaps [5], Hessian eigenmaps [15], local tangent space alignment [34,35] and Diffusion Maps [11].

For a sufficiently small dataset, kernel methods can be implemented and executed on relatively standard computing devices. However, even for moderate size datasets, the necessary computational requirements to process them are unreasonable and, in many cases, impractical. For example, a segmentation of a medium size image with 512×512 pixels requires a $2^{18} \times 2^{18}$ kernel matrix. The size of such a matrix necessitated about 270 GB of memory assuming double precision. Furthermore, the spectral decomposition procedure applied to such a matrix will be a formidable slow task. Hence, there is a growing need to have more computationally efficient methods that are practical for processing large datasets.

The main computational load associated with kernel methods is generated by the application of a spectral decomposition to a kernel matrix. Sparsification by a sparse eigensolver such as Lanczos, which computes the relevant eigenvectors [12] of the kernel matrix, is widely used to reduce the computational load involved in processing a kernel matrix. Another sparsification approach is to transform the dense kernel matrix into a sparse matrix by selectively truncating elements outside a given neighborhood radius of each dataset member. Other approaches to achieve matrix sparsification are described in [33]. Given a dataset with n data points, common methods including the one described in this paper for processing kernel methods require at least $O(n^2)$ operations to determine which entries to either calculate or to threshold. While there are methods to alleviate these computational complexities [1], kernel sparsification might result in a significant loss of intrinsic geometric information such as distances and similarities.

A prominent approach to reduces the discussed computational load is based on the Nyström extension method [17], which estimates the eigenvectors needed for an embedding. This approach is based on three phases:

1. The dataset is subsampled uniformly over the set of indices that are randomly chosen without repetition.
2. The subsamples define a smaller (than the dataset size) kernel. SVD is applied to the small kernel.
3. Spectral decomposition of a small kernel is extended by the application of the Nyström extension method to the entire dataset.

This three-phase approach reduces the computational load, but the approximated spectral decomposition output suffers from several major problems. Subsampling affects the quality of the spectral approximation. In addition, the Nyström extension method exhibits ill-conditioned behavior that also affects the spectral approximation [6]. Uniform subsampling of a sufficient number of data points captures most of the data probability distribution. However, rare events, compared to the subsampled size, might get lost. The results from this loss of information degrades the quality of the estimated embedded distances.

The Nyström extension method is based on inverting a kernel matrix that was derived from a uniform sampling. This kernel does not necessarily has a full rank. Therefore, a direct kernel matrix inversion is ill-conditioned. The Moor–Penrose pseudo-inverse operator can overcome the ill-conditioned effect in Nyström extension. However, this solution may generate an inaccurate extension. Therefore, combining Nyström extension with random sampling can result in inaccurate approximations of spectral decomposition.

Recently, a multiscale scheme, which is called multiscale extension (MSE), was suggested in [6]. The scheme, which samples scattered data and extends functions defined on sampled data points, overcomes some of the limitations of the Nyström method. The MSE method is based on mutual distances between

data points. It uses a coarse-to-fine hierarchy of a multiscale decomposition of a Gaussian kernel to overcome the ill-conditioned phenomenon and to speed the computations.

In this paper we focus on alleviating the computational complexity of the Diffusion Maps (DM) method and enabling its application for large datasets. This kernel method utilizes a Markovian diffusion process to define and represent nonlinear relations between data points. It provides a diffusion distance metric that correlates with the intrinsic geometry of the data. Unlike the geodesic distance metric of manifolds, the diffusion distance metric is very robust to noise. This diffusion distance metric can be explained in terms of the transition probabilities of the Markovian DM diffusion process. Namely, it is defined by the pairwise connectivity of the data points in the DM diffusion process [23], and the DM kernel that is designed to capture this connectivity. The diffusion distance metric was proved useful in clustering [14], parametrization of linear systems [31] and even shape recognition [9].

The DM kernel represents a graph in which each data point corresponds to a vertex. The weight of each edge between any pair of vertices reflects the similarity between the corresponding data points on the manifold and in the diffusion process. The eigenvalues and the corresponding eigenvectors of this kernel matrix reveal many properties and connections in the graph. These eigenvalues and eigenvectors are used to obtain the DM embedding of the data. The diffusion distances are preserved by this embedding and are expressed as the Euclidean distances in the DM embedded space, whose dimensionality is usually significantly lower than the dimensionality of the original ambient space of the data.

The DM embedding was utilized in a wide variety data and pattern analysis techniques. For example it was used to improve audio quality by suppressing transient interference [30]. In [26] it was utilized for detecting moving vehicles. Additionally, DM was proposed for scene classification [19], gene expression analysis [25] and source localization [29]. Furthermore, the DM method can be utilized for fusing different sources of data [23,20].

The application of DM to a given dataset depends on the kernel size of the dataset. The size imposes severe limitations on the physical computational abilities to process it. In this paper, we efficiently approximate the DM method by modifying the Nyström extension. This approximation, called μIDM, guaranties that the difference between the diffusion distances in DM embedding and the Euclidean distances in μIDM embedding, is preserved isometrically up to a given controllable error μ. The μIDM utilizes the low dimensional geometry from the DM embedding to constructively design a dictionary that approximates the geometry of the entire DM embedding. The members of this dictionary are tailored to reduce the worst case approximation errors between the different embeddings. Additionally, we prove the convergence of the μIDM spectrum to the respective DM spectrum. We bound the spectral convergence error as a function of the controllable error μ.

The paper has the following structure. Section 2 describes the general setup of the problem that includes a review of DM. Section 3 shows how a subset of distances in the DM space can be exactly computed via a spectral decomposition of a small kernel. Section 4 presents a variant of the Nyström method and analyzes the conditions that are required for the resulting mapping to preserve the diffusion distances of the relevant subset. Section 5 presents the dictionary construction and the μ-isometric approximation. In addition, this section analyzes the resulting approximation accuracy, its spectral convergence to DM spectrum and provides a computation complexity estimation as a function of the dataset and of the dictionary size. Finally, Section 6 examines the proposed method on data.

2. Problem formulation

Let $\mathcal{M}$ be a low-dimensional manifold that lies in the high-dimensional Euclidean ambient space $\mathbb{R}^m$ and let $d \ll m$ be its intrinsic dimension. Let $M \subseteq \mathcal{M}$ be a dataset of $|M| = n$ data points that are sampled from this manifold. The DM method [11,22] analyzes datasets such as $\mathcal{M}$ by exploring the geometry of the manifold $\mathcal{M}$ from which they are sampled. DM embeds the data into a space where the Euclidean distances

between data points in the embedded space correspond to diffusion-based distances on the manifold $\mathcal{M}$. A detailed construction of the DM is given in Section 2.1.

DM is a kernel method, which is based on the spectral analysis of an $n \times n$ kernel matrix that holds the affinities between all the data points in M. For large datasets, derivation of the exact spectral decomposition of such a kernel is impractical due to the $O(n^3)$ operations required by SVD. One way to reduce the computational complexity is to approximate this spectral decomposition such as in [1,33]. However, such SVD-based distances approximations in the embedded space is in general inaccurate and does not allow a direct control of the incurred approximation error.

In this paper, we efficiently approximate the DM embedding $\Phi : M \to \mathbb{R}^{\delta}$ by a map $\widehat{\Phi} : M \to \mathbb{R}^{\hat{\delta}}$. In order to quantify the error between the two maps, we introduce the notion of μ-isometric maps, which is given in Definition 2.1.

Definition 2.1 *(μ-isometric maps).* The maps $\Phi : M \to \mathbb{R}^{\delta}$ and $\widehat{\Phi} : M \to \mathbb{R}^{\hat{\delta}}$ are *μ-isometric* if for every $x, y \in M$, $|\|\widehat{\Phi}(x) - \widehat{\Phi}(y)\| - \|\Phi(x) - \Phi(y)\|| \leq \mu$. The notation $\|\cdot\|$ denotes Euclidean norm in the respective space.

The proposed method identifies a dictionary of data points in M that are sufficient to describe the pairwise distances between DM embedded data points. Then, the approximated map $\widehat{\Phi}$ is computed by an out-of-sample extension that preserves the pairwise diffusion distances in the dictionary. This is a modified version of Nyström extension that is used to compute the μ-Isometric maps.

2.1. Diffusion map

The DM method is based on an isotropic kernel K, whose elements are

$$k(x, y) \triangleq e^{-\frac{\|x-y\|^2}{\varepsilon}}, \quad x, y \in M, \tag{2.1}$$

where ε is a meta-parameter. This kernel represents the affinities between data points in the manifold. The kernel can be viewed as a construction of a weighted graph on the dataset M. The data points in M are used as vertices and the weights of the edges are defined by the kernel K (Eq. (2.1)). The degree of each data point (i.e., vertex) $x \in M$ in this graph is

$$q(x) \triangleq \sum_{y \in M} k(x, y). \tag{2.2}$$

Kernel normalization with this degree produces a row-stochastic transition matrix P whose elements for $x, y \in M$ are $p(x, y) = k(x, y)/q(x)$. This defines a Markov process over the data points in M. If the manifold was not sampled uniformly, one can use the normalized kernel

$$\tilde{k}(x, y) \triangleq \frac{k(x, y)}{q(x)q(y)},$$

instead of $k(x, y)$ in order to separate the geometry of the manifold from the density of the data, as shown in [11,22].

The DM method embeds data points from the manifold $\mathcal{M}$ into a Euclidean space whose dimensionality is lower than the original data dimensionality. It is preferable to work with a symmetric conjugate matrix to P, which is denoted by A, whose entries are

$$[A]_{(x,y)} = a(x, y) \triangleq \frac{k(x, y)}{\sqrt{q(x)q(y)}} = \sqrt{q(x)}p(x, y)\frac{1}{\sqrt{q(y)}}, \quad x, y \in M. \tag{2.3}$$

We will refer to A as the diffusion affinity kernel or as the symmetric diffusion kernel. Since the spectral radius of P is 1 [11,22], then A also has a unit spectral radius. In addition, as long as the data points in M are distinct, A is strictly positive definite, due to the positivity of K.

The eigenvalues of A, $1 = \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$, and their corresponding eigenvectors $\phi_1, \phi_2, \ldots, \phi_n$, are used to construct the diffusion map $\Phi : M \to \mathbb{R}^\delta$

$$\Phi(x) = \left[q^{-1/2}(x)\big(\sigma_1\phi_1(x)\big), \ldots, q^{-1/2}(x)\big(\sigma_\delta\phi_\delta(x)\big)\right] \tag{2.4}$$

for a sufficiently small δ, which is the dimension of the embedded space that depends on the decay of the spectrum of A. This construction is also known as the graph Laplacian constructed by the diffusion kernel [10].

Typically, the application of DM to a dataset M of size n involves the following steps:

1. Use Eq. (2.1) to construct the $n \times n$ kernel K;
2. Compute a diagonal matrix Q that holds for the data points in M the degrees $q_i \triangleq \sum_{j=1}^{n} K_{ij}$ for all $i = 1, \ldots, n$;
3. Normalize K by Q to get an $n \times n$ symmetric diffusion affinity kernel $A = Q^{-1/2}KQ^{-1/2}$ by using Eq. (2.3);
4. Obtain the eigenvalues and the eigenvectors of A by the application of SVD to $A = \phi\Sigma\phi^T$ to get the matrices

$$\Sigma = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n \end{bmatrix}, \qquad \phi = \begin{bmatrix} | & & | \\ \phi_1 & \cdots & \phi_n \\ | & \cdots & | \end{bmatrix}.$$

   that hold the eigenvalues and the eigenvectors of A, respectively;
5. Use the matrix $Q^{-1/2}\phi\Sigma$ to embed each data point $x_i \in M$, $i = 1, \ldots, n$, to the i-th row of this matrix. In [6,7] it was proved that the spectrum of the matrix A decays exponentially and only a small number of eigenvectors are required to obtain a reliable low dimensional embedding space.

The diffusion distances between data points $x, y \in M$ are defined by $\|p(x,\cdot) - p(y,\cdot)\|_{l_2(1/q)}$, where $p(x,\cdot)$ and $p(y,\cdot)$ are the transition probabilities that are defined by the stochastic transition matrix P. The use of the spectral theorem in [11] shows that the Euclidean distances in the embedded space of DM correspond to the diffusion distances in the manifold. Namely, $\|\Phi(x) - \Phi(y)\| = \|p(x,\cdot) - p(y,\cdot)\|_{l_2(1/q)}$.

3. Diffusion maps of a partial set

The computation of the DM embedding from Eq. (2.4) requires the spectral decomposition of the full $n \times n$ symmetric diffusion kernel. Performing this decomposition on large datasets is computationally expensive. In this section, we describe an efficient method to compute the pairwise diffusion distances between data points of a partial dataset $S \subset M$. We assume that, without loss of generality, $M = \{x_1, \ldots, x_n\}$ and $S = \{x_1, \ldots, x_s\}$, $s < n$.

Define the partial kernel $\widetilde{K}$ as the upper $s \times n$ submatrix of the Gaussian kernel K from Eq. (2.1). Also let $\widetilde{Q}$ be the $s \times s$ diagonal matrix whose diagonal entries are the degrees $\widetilde{q}(x_i) = \sum_{j=1}^{n} \widetilde{k}(x_i, x_j)$, $i = 1, 2, \ldots, s$. Finally, we define the $s \times n$ diffusion affinity kernel $\widetilde{A}$ between the partial set S and the dataset M by

$$\widetilde{A} \triangleq \widetilde{Q}^{-1/2}\widetilde{K}Q^{-1/2}. \tag{3.1}$$

Let $\widetilde{\phi}\widetilde{\Sigma}^2\widetilde{\phi}^T$ be the SVD of the $s \times s$ symmetric matrix $\widetilde{A}\widetilde{A}^T$. The eigenvalues of the matrix $\widetilde{\Sigma}^2$, which are located on its diagonal, are $\widetilde{\sigma_1}^2 \geq \ldots \geq \widetilde{\sigma_s}^2$ while its eigenvectors $\widetilde{\phi}_1, \ldots, \widetilde{\phi}_s$ are located as its columns in $\widetilde{\phi}$. Definition 3.1 uses the SVD-based decomposition to define a Partial Diffusion Map (PDM) on the partial set S. In what follows, the notation $\widetilde{q}(x)$ and $\widetilde{\phi}_j(x)$ will stand for q_i and the i-th coordinate of $\widetilde{\phi}_j$, respectively, where $x = x_i$.

Definition 3.1 *(Partial diffusion map).* The Partial Diffusion Map (PDM) $\widetilde{\Phi} : S \to \mathbb{R}^s$ of the partial set S is

$$\widetilde{\Phi}(x) \triangleq \left[\widetilde{q}^{-1/2}(x)\left(\widetilde{\sigma}_1\widetilde{\phi}_1(x)\right), \ldots, \widetilde{q}^{-1/2}(x)\left(\widetilde{\sigma}_s\widetilde{\phi}_s(x)\right)\right].$$

Definition 3.1 takes into consideration the entire spectrum of the decomposed partial kernel. In the rest of the paper, we will assume that DM also considers the entire spectrum (i.e., $\delta = n$ in Eq. (2.4)). However, for practical purposes, we can modify Definition 3.1 so that PDM will only use a small number $\widetilde{\delta} \ll s < n$ of eigenvalues, similarly to the truncation of the number of eigenvalues as done in the DM embedding by Eq. (2.4). Theorem 3.1 shows that the geometry of S under the DM embedding is preserved by the PDM embedding.

Theorem 3.1. *The geometry of S under the DM embedding is preserved by the PDM applied to S. Formally, for every $x, y \in S$, $\|\widetilde{\Phi}(x) - \widetilde{\Phi}(y)\| = \|\Phi(x) - \Phi(y)\|$ and $\langle\widetilde{\Phi}(x), \widetilde{\Phi}(y)\rangle = \langle\Phi(x), \Phi(y)\rangle$.*

Due to Theorem 3.1, an embedding that preserves the diffusion distances of a partial set of size s can be computed by decomposing only an $s \times s$ matrix instead of using a much bigger $n \times n$ matrix. Lemma 3.2 is needed for the proof of Theorem 3.1.

Lemma 3.2. *The matrix $\widetilde{Q}^{-1/2}\widetilde{A}\widetilde{A}^T\widetilde{Q}^{-1/2}$ is the $s \times s$ upper left matrix of $Q^{-1/2}A^2Q^{-1/2}$, i.e., for every $x, y \in S$, $(Q^{-1/2}A^2Q^{-1/2})_{(x,y)} = (\widetilde{Q}^{-1/2}\widetilde{A}\widetilde{A}^T\widetilde{Q}^{-1/2})_{(x,y)}$.*

Proof. According to Eq. (2.3), $Q^{-1/2}A^2Q^{-1/2} = Q^{-1}KQ^{-1}KQ^{-1}$. Due to Eq. (3.1) and definitions of $\widetilde{Q}$ and $\widetilde{K}$, the restriction of the matrix $Q^{-1/2}A^2Q^{-1/2}$ to the $s \times s$ upper left matrix yields for every $x, y \in S$, $(Q^{-1/2}A^2Q^{-1/2})_{(x,y)} = (\widetilde{Q}^{-1}\widetilde{K}Q^{-1}\widetilde{K}^T\widetilde{Q}^{-1})_{(x,y)} = (\widetilde{Q}^{-1/2}\widetilde{A}\widetilde{A}^T\widetilde{Q}^{-1/2})_{(x,y)}$. □

Lemma 3.2 shows the relation between the partial affinities and the full affinities and their associated degrees. The proof of Theorem 3.1 uses this relation.

Proof of Theorem 3.1. By Definition 3.1, for any $x, y \in S$,

$$\left\langle\widetilde{\Phi}(x), \widetilde{\Phi}(y)\right\rangle = \sum_{j=1}^{s} q^{-1/2}(x)\widetilde{\sigma}_j\widetilde{\phi}_j(x) \cdot q^{-1/2}(y)\widetilde{\sigma}_j\widetilde{\phi}_j(y).$$

By using the spectral theorem, we get $(\widetilde{A}\widetilde{A}^T)_{(x,y)} = \sum_{j=1}^{s}\widetilde{\sigma}_j^2\widetilde{\phi}_j(x)\widetilde{\phi}_j(y)$. Since the diagonal matrix $\widetilde{Q}$ holds the partial degrees $\widetilde{q}(\cdot)$, we get

$$\left\langle\widetilde{\Phi}(x), \widetilde{\Phi}(y)\right\rangle = \left[\widetilde{Q}^{-1/2}\widetilde{A}\widetilde{A}^T\widetilde{Q}^{-1/2}\right]_{(x,y)}.$$

Finally, we use Lemma 3.2 to replace $\widetilde{Q}^{-1/2}\widetilde{A}\widetilde{A}^T\widetilde{Q}^{-1/2}$ with $Q^{-1/2}A^2Q^{-1/2}$, thus

$$\left\langle\widetilde{\Phi}(x), \widetilde{\Phi}(y)\right\rangle = \left[Q^{-1/2}A^2Q^{-1/2}\right]_{(x,y)}.$$

On the other hand, by the DM definition we have

$$\langle \Phi(x), \Phi(y) \rangle = \sum_{j=1}^{n} q^{-1/2}(x)\sigma_j \phi_j(x) \cdot q^{-1/2}(y)\sigma_j \phi_j(y) = \left[Q^{-1/2} A^2 Q^{-1/2}\right]_{(x,y)}.$$

Therefore, $\langle \widetilde{\Phi}(x), \widetilde{\Phi}(y) \rangle = \langle \Phi(x), \Phi(y) \rangle$ as the theorem states. Distance preservation in the theorem follows immediately since $\|u - v\|^2 = \langle u, u \rangle - 2\langle u, v \rangle + \langle v, v \rangle$ for every u, v in both embedded spaces. □

4. An out-of-sample extension that preserves the PDM geometry

PDM provides an embedding $\widetilde{\Phi} : S \to \mathbb{R}^s$ of a partial dataset S where $s = |S|$. In order to extend this embedding to the entire dataset M, an out-of-sample extension method is applied such that $\widetilde{\Phi}$ is preserved over S. This is called an extended map. In this section, we utilize the Nyström extension [1,16] to compute the extended map for the entire dataset. In addition, we will constrain the extended map to have the same pairwise distances as PDM has. Therefore, the extended map will preserve the diffusion distances in S.

Given a partial set $S \subset M$ of size s and its complement $\bar{S} = M \setminus S$ of size $n - s$, then a diffusion affinity kernel A (Eq. (2.3)) can be described as having the following block structure

$$A = \begin{bmatrix} A_{(S,S)} & A_{(S,\bar{S})} \\ A^T_{(S,\bar{S})} & A_{(\bar{S},\bar{S})} \end{bmatrix}, \tag{4.1}$$

where the block $A_{(S,S)} \in \mathbb{R}^{s \times s}$ holds the diffusion affinities between data points in S, the block $A_{(\bar{S},\bar{S})} \in \mathbb{R}^{(n-s)\times(n-s)}$ holds the affinities between data points in $\bar{S}$, and the block $A_{(S,\bar{S})} \in \mathbb{R}^{s\times(n-s)}$ holds the affinities between data points in S and data points in $\bar{S}$. Under this formulation, Eq. (3.1) becomes

$$\widetilde{A} = \begin{bmatrix} A_{(S,S)} & A_{(S,\bar{S})} \end{bmatrix}. \tag{4.2}$$

Let $A_{(S,S)} = \widetilde{\psi}\widetilde{\Lambda}\widetilde{\psi}^T$ be the spectral decomposition of the positive-definite upper left block of A, where $\widetilde{\Lambda}$ is a diagonal matrix that contains the eigenvalues $\widetilde{\lambda}_1 \geq \widetilde{\lambda}_2 \geq \ldots \geq \widetilde{\lambda}_s$, and $\widetilde{\psi}$ contains their corresponding eigenvectors as its columns. To extend this decomposition to the entire dataset M, the Nyström extension uses the property that every eigenvector $\widetilde{\psi}$ and every eigenvalue $\widetilde{\lambda}$ satisfy $\widetilde{\psi} = A_{(S,S)}\widetilde{\psi}\widetilde{\lambda}^{-1}$ in the following way:

$$\widehat{\psi} = \begin{bmatrix} \widetilde{\psi} \\ A^T_{(S,\bar{S})}\widetilde{\psi}\widetilde{\Lambda}^{-1} \end{bmatrix}.$$

It results in an $n \times n$ approximated affinity matrix

$$\widehat{A} = \widehat{\psi}\widetilde{\Lambda}\widehat{\psi}^T = \begin{bmatrix} A_{(S,S)} & A_{(S,\bar{S})} \\ A^T_{(S,\bar{S})} & A^T_{(S,\bar{S})}(A_{(S,S)})^{-1}A_{(S,\bar{S})} \end{bmatrix}. \tag{4.3}$$

Therefore, the diffusion affinity matrix can be approximated by the extension given in Eq. (4.3).

The DM embedding is based on the spectral decomposition of the diffusion affinity matrix A, which is approximated by $\hat{A}$. Therefore, in order to approximate DM embedding using the discussed extension, the matrix $\widehat{A}$ has to be decomposed as

$$\widehat{A} = \widehat{\phi}\Lambda\widehat{\phi}^T, \tag{4.4}$$

where $\widehat{\phi}$ is an $n \times s$ matrix with orthonormal columns and Λ is an $s \times s$ diagonal matrix. A numerically efficient scheme for obtaining such a decomposition is presented in Section 4.1. Definition 4.1 presents the corresponding Nyström-based approximated DM based on this discussion.

Definition 4.1. Let $\widehat{\phi}$ and Λ be the matrices from Eq. (4.4). The Orthogonal Nyström-based Map (ONM) is the map $\widehat{\Phi}: M \to \mathbb{R}^s$, given by

$$\widehat{\Phi}(x) \triangleq \left[q^{-1/2}(x)\lambda_1\widehat{\phi}_1(x), \ldots, q^{-1/2}(x)\lambda_s\widehat{\phi}_s(x)\right],$$

where $\widehat{\phi}_1, \ldots, \widehat{\phi}_s \in \mathbb{R}^n$ are the columns of the matrix $\widehat{\phi}$ and λ_i, $i = 1, \ldots, s$, is the ith diagonal elements in Λ. In other words, ONM embeds each data point in M into $\mathbb{R}^s$ by the corresponding row of the matrix $Q^{-1/2}\widehat{\phi}\Lambda$.

The ONM in Definition 4.1 embeds the entire dataset M into $\mathbb{R}^s$. As described in Section 4.1, ONM requires a spectral decomposition of a $s \times s$ matrix rather than performing a spectral decomposition of $n \times n$ matrix. Proposition 4.1 shows that the geometry of S under the PDM is preserved by the ONM embedding.

Proposition 4.1. *Let $\widetilde{\Phi}$ and $\widehat{\Phi}$ be the PDM and the ONM embedding functions, respectively. Then, for every $x, y \in S$, $\|\widetilde{\Phi}(x) - \widetilde{\Phi}(y)\| = \|\widehat{\Phi}(x) - \widehat{\Phi}(y)\|$ and $\langle\widetilde{\Phi}(x), \widetilde{\Phi}(y)\rangle = \langle\widehat{\Phi}(x), \widehat{\Phi}(y)\rangle$.*

Proof. Since $\widehat{A} = \widehat{\phi}\Lambda\widehat{\phi}^T$ (Eq. (4.4)) and $\widehat{\phi}^T\widehat{\phi} = I$, we have $\widehat{A}^2 = \widehat{\phi}\Lambda^2\widehat{\phi}^T$, thus, the inner products in the embedded space of the ONM satisfy for every $x, y \in M$

$$\left\langle\widehat{\Phi}(x), \widehat{\Phi}(y)\right\rangle = \left[Q^{-1/2}\widehat{\phi}\Lambda^2\widehat{\phi}^TQ^{-1/2}\right]_{(x,y)} = q^{-1/2}(x)\left[\widehat{A}^2\right]_{(x,y)}q^{-1/2}(y).$$

Furthermore, due to the structure of $\widehat{A}$ in Eq. (4.3), the upper left block of $\widehat{A}^2$ is $\widetilde{A}\widetilde{A}^T$, thus, we have for every $x, y \in S$

$$\left\langle\widehat{\Phi}(x), \widehat{\Phi}(y)\right\rangle = q^{-1/2}(x)\left[\widetilde{A}\widetilde{A}^T\right]_{(x,y)}q^{-1/2}(y) = \left\langle\widetilde{\Phi}(x), \widetilde{\Phi}(y)\right\rangle.$$

The equality holds due to the spectral decomposition of $\widetilde{A}\widetilde{A}^T$ and Definition 3.1. Since the inner products in both embedded spaces are equal, the distance preservation follows immediately. □

Recall that according to Theorem 3.1, the geometry of S under the DM embedding is preserved by the PDM embedding. By combining Theorem 3.1 with Proposition 4.1, we get Corollary 4.2.

Corollary 4.2. *The geometry of S under DM embedding is preserved by the ONM embedding, i.e., for every $x, y \in S$, $\|\widehat{\Phi}(x) - \widehat{\Phi}(y)\| = \|\Phi(x) - \Phi(y)\|$ and $\langle\widehat{\Phi}(x), \widehat{\Phi}(y)\rangle = \langle\Phi(x), \Phi(y)\rangle$.*

4.1. An efficient computation of the SVD of $\widehat{A}$

Recall that the upper left submatrix $A_{(S,S)}$ is positive definite [22]. Hence, it can be used to formulate an alternative Nyström approximation, which was presented in [17]. It can be verified that for every orthogonal $s \times s$ matrix Ψ and for any $s \times s$ non-singular matrix Λ, the matrix

$$\widehat{\phi} = \begin{bmatrix} A_{(S,S)} \\ A^T_{(S,\bar{S})} \end{bmatrix} A^{-1/2}_{(S,S)}\Psi\Lambda^{-1/2} \tag{4.5}$$

satisfies $\widehat{A} = \widehat{\phi}\Lambda\widehat{\phi}^T$. Furthermore, according to [17], the matrix $\widehat{\phi}$ can be designed to decompose the matrix $\widehat{A}$ in Eq. (4.3) as $\widehat{A} = \widehat{\phi}\Lambda\widehat{\phi}^T$ while having orthogonal columns (i.e., $\widehat{\phi}^T\widehat{\phi} = I$).

In our case, the extension is aimed to preserve the pairwise diffusion distances, and the related inner products, according to the PDM of S. Technically, the matrices $\widehat{\phi}$ and Λ are required to satisfy

$$A_{(S,S)}^{1/2}\psi\Lambda\psi^T A_{(S,S)}^{1/2} = A_{(S,S)}A_{(S,S)} + A_{(S,\bar{S})}A_{(S,\bar{S})}^T, \tag{4.6}$$

where the LHS consists of the inner products of the Nyström approximation (Eq. (4.5)) of S and the RHS consists of the inner products $\widetilde{A}\widetilde{A}^T$ of the PDM of the same set S. This formulation dictates the definition of $\widehat{\phi}$ and Λ as the SVD

$$C = \psi\Lambda\psi^T, \tag{4.7}$$

where C is defined as

$$C \triangleq A_{(S,S)} + A_{(S,S)}^{-1/2}A_{(S,\bar{S})}A_{(S,\bar{S})}^T A_{(S,S)}^{-1/2}. \tag{4.8}$$

In order to prevent numerical instabilities due to inversion of $A_{(S,S)}$, one can use the stochastic matrix $(P + 2I)/3$, for example, instead of P in Eq. (2.3). While this matrix contains the same connectivity information, its spectrum lays on the interval $[1/3, 1]$, due to Gershgorin's theorem (Theorem 2.1 in [28]). Since A and P are conjugate matrices they have the same spectra, and according to Cauchy's interlacing theorem (Theorem 4.2 in [28]), the spectrum of $A_{(S,S)}$ also lays on the same interval.

5. The μ-isometric construction

In this section, we describe a constructive method to choose a partial set $S \subset M$ such that the resulting ONM from Definition 4.1 will be μ-isometric to DM, which utilizes the full diffusion kernel. The proposed method uses a single scan of the entire dataset M and optimizes the dictionary selected set S for each processed data point. The construction of S is designed such that the geometry of M under the DM embedding is approximated by the ONM embedding applied to S.

The proposed algorithm is iterative and it gradually constructs the dictionary subset S and the associated ONM. For its description, the following notations are used: The dataset M is assumed to be enumerated such that $M = \{x_1, \ldots, x_n\}$. Since the algorithm scans M only once, where in each iteration it examines a unique data point, the indices of the data points will indicate the current iteration number. That is, in iteration j $(j = 1, \ldots, n)$ of the algorithm the j-th data point is examined. In the j-th iteration, the algorithm holds a subdictionary $S_j = \{y_1, y_2, \ldots, y_{n_j}\}$. The subdictionary S_j is a subset of $M_j = \{x_1, \ldots, x_j\}$ where $n_j \leq j$. Our algorithm constructs a monotonically increasing sequence of subdictionaries, i.e., $S_{j-1} \subset S_j$ for any $j = 2, \ldots, n$. The final dictionary S_n is denoted by S. The notation $\widehat{\Phi}_j$ denotes the ONM $\widehat{\Phi}_j : M \to \mathbb{R}^{n_j}$ applied to S_j.

Let $\kappa < \ell < n$, then according to Corollary 4.2, for all $x, y \in S_\kappa$, $\|\widehat{\Phi}_\kappa(x) - \widehat{\Phi}_\kappa(y)\| = \|\Phi(x) - \Phi(y)\| = \|\widehat{\Phi}_\ell(x) - \widehat{\Phi}_\ell(y)\|$, i.e., the geometry of S_κ under the DM embedding is identical to its geometry under the ONM embedding, applied to S_ℓ. Thus, there exists $T : \mathbb{R}^{n_\kappa} \to \mathbb{R}^{n_\ell}$ that maps $\widehat{\Phi}_\kappa(S_\kappa)$ onto $\widehat{\Phi}_\ell(S_\kappa)$ isometrically. Definition 5.1 defines such a map. This definition uses the invertibility of $\widehat{\Phi}_\kappa(S_\kappa)$, which is proved in Appendix A.

Definition 5.1 *(Map-to-Map (MTM) transformation).* Assume the matrices

$$\left[\widehat{\Phi}_\kappa(S_\kappa)\right] = \underbrace{\begin{bmatrix} — & \widehat{\Phi}_\kappa(y_1) & — \\ & \vdots & \\ — & \widehat{\Phi}_\kappa(y_{n_\kappa}) & — \end{bmatrix}}_{n_\kappa \times n_\kappa}, \qquad \left[\widehat{\Phi}_\ell(S_\kappa)\right] = \underbrace{\begin{bmatrix} — & \widehat{\Phi}_\ell(y_1) & — \\ & \vdots & \\ — & \widehat{\Phi}_\ell(y_{n_\kappa}) & — \end{bmatrix}}_{n_\kappa \times n_\ell}$$

Algorithm 5.1: The μ-isometric DM (μIDM).

Input: data points: $x_1, ..., x_n \in \mathbb{R}^m$.
Parameters: Distance error bound μ, Gaussian width ε
Output: The approximated DM coordinates $\widehat{\Phi}(x_i)$, $i = 1, \ldots, n$

1: Initialize the dictionary: $S_1 \leftarrow \{x_1\}$
2: Initialize Q (Eq. (2.2)) and $\widetilde{A}$ given S_1 (Eq. (4.2))
3: Initialize the embedding: $\widehat{\Phi} \leftarrow$ ONM (Definition 4.1) of S_1.
4: **for** $\kappa = 1$ *to* $n-1$ **do**
  Set $S' \leftarrow S_\kappa \cup \{x_{\kappa+1}\}$
  Compute $\widetilde{Q}'$ (Eq. (2.2)) and $\widetilde{A}'$ given S'
  Compute $\widehat{\Phi}' \leftarrow$ ONM (Definition 4.1) of S'
  Membership Test:
    Compute $T \leftarrow$ MTM (Definition 5.1) from $\widehat{\Phi}(\cdot)$ to $\widehat{\Phi}'(\cdot)$
    Compute $\beta \leftarrow \|T(\widehat{\Phi}(x_{\kappa+1})) - \widehat{\Phi}'(x_{\kappa+1})\|$
    If $\beta > \frac{\mu}{2}$
      Set $S_{\kappa+1} \leftarrow S'$
      Set $\widetilde{Q} \leftarrow \widetilde{Q}'$ and $\widetilde{A} \leftarrow \widetilde{A}'$
      Set $\widehat{\Phi} \leftarrow \widehat{\Phi}'$
    Else
      Set $S_{\kappa+1} \leftarrow S_\kappa$
    End if
5: Output the approximated diffusion coordinates $\widehat{\Phi}(x_1), \ldots, \widehat{\Phi}(x_n)$

hold the coordinates of data points in the dictionary S_κ according to $\widehat{\Phi}_\kappa$ and $\widehat{\Phi}_\ell$, respectively. The linear Map-to-Map (MTM) transformation $T_{\kappa,\ell} : \mathbb{R}^{n_\kappa} \to \mathbb{R}^{n_\ell}$ is defined by the application[1] of the matrix $[T_{\kappa,\ell}] \triangleq [\widehat{\Phi}_\kappa(S_\kappa)]^{-1}[\widehat{\Phi}_\ell(S_\kappa)]$ to vectors $u \in \mathbb{R}^{n_\kappa}$ such that $T_{\kappa,\ell}(u) = u[T_{\kappa,\ell}] \in \mathbb{R}^{n_\ell}$.

It is clear from Definition 5.1 that the MTM transformation of every $\widehat{\Phi}_\kappa(x)$, $x \in S_\kappa$, satisfies

$$\widehat{\Phi}_\ell(x) = T_{\kappa,\ell} \circ \widehat{\Phi}_\kappa(x). \tag{5.1}$$

Therefore, the geometry of S_κ is preserved in $\mathbb{R}^{n_\ell}$ under $T_{\kappa,\ell}$. Theorem 5.2 shows that this transformation is an isometry between $\mathbb{R}^{n_\kappa}$ and its image in $\mathbb{R}^{n_\ell}$. For data points in $S_\ell \setminus S_\kappa$, the maps $T_{\kappa,\ell} \circ \widehat{\Phi}_\kappa$ and $\widehat{\Phi}_\ell$ may provide different embeddings. For all $x \in S_\ell \setminus S_\kappa$, the error

$$\beta = \left| T_{\kappa,\ell} \circ \widehat{\Phi}_\kappa(x) - \widehat{\Phi}_\ell(x) \right| \tag{5.2}$$

evaluates how well DM embeddings of data points in the set S_ℓ are approximated by ONM applied to S_κ. We will base our dictionary membership criterion on this evaluation, and whether it is sufficiently small compared to a desired error bound.

The μIDM construction in Algorithm 5.1 sequentially scans the data points $x_1, x_2, \ldots, x_n \in M$ to check if their embeddings can be approximated by the dictionary or they have to be added to it. Initially, the dictionary is set to contain a single data point x_1. Then, at each iteration κ, data points in $M_\kappa = \{x_1, \ldots, x_\kappa\}$, which were already scanned, are approximated by the constructed dictionary $S_\kappa \subseteq M_\kappa$. The algorithm processes the next data point $x_{\kappa+1}$ and checks if the approximation of its embedding by dictionary S_κ is sufficiently accurate. If it is, then the algorithm proceeds to the next iteration and the dictionary remains unchanged (i.e., $S_{\kappa+1} = S_\kappa$). Otherwise, this data point is added to the dictionary S_κ. In the next iteration, $S_{\kappa+1} = S_\kappa \cup \{x_{\kappa+1}\}$.

In Definition 5.1 and in the accompanied discussion, we assumed without loss of generality that M contains the first κ and ℓ data points that are the sets M_κ and M_ℓ, respectively. This assumption simplifies the presentation. The dictionary membership criterion is based on comparing the approximation error

[1] Vectors in this definition are considered as row vectors and the matrix $[T_{\kappa,\ell}]$ is applied to their right hand side.

$|\widehat{\Phi}'(x_{\kappa+1}) - T \circ \widehat{\Phi}(x_{\kappa+1})|$ between the examined data point $x_{\kappa+1}$ and a given adjustable threshold μ. In Section 5.1, we will show that this criterion guaranties that at the end of the dictionary construction process, the ONM embedding of every data point in $M \setminus S$ is μ-isometric to DM. The rest of this section analyzes the accuracy of the resulting embedding and the computational complexity of its construction.

5.1. Distance accuracy of μIDM

Algorithm 5.1 constructs an optimized dictionary. Then, it uses the ONM of this dictionary to approximate the DM embedding. Corollary 4.2 guaranties that the ONM-based embedding preserves the diffusion distances between the dictionary members. Equivalently, it preserves the corresponding diffusion distances. The dictionary membership criterion guarantees that the distances from every data point not in the dictionary to the dictionary members approximate well the DM embedded distances up to the accuracy threshold μ. Theorem 5.1 shows that the resulting dictionary-based ONM embedding preserves all the DM embedded diffusion distances in M, up to accuracy μ.

Theorem 5.1. *Let Φ be the DM embedding (see Section 2.1) of M. Let $S \subseteq M$ be the dictionary constructed by Algorithm 5.1 and let $\widehat{\Phi}$ be the ONM, based on this dictionary. Then, for all $x, y \in M$, $\|\widehat{\Phi}(x) - \widehat{\Phi}(y)\| \approx \|\Phi(x) - \Phi(y)\|$ with an approximation error of at most μ.*

Theorem 5.1 shows that the parameter μ in Algorithm 5.1 dictates the worst-case error of the approximated pairwise distances of the μIDM. In order to prove this theorem, we first present Theorem 5.2 and Lemma 5.3.

Theorem 5.2. *The MTM transformation $T_{\kappa,\ell}$ from Definition 5.1 embeds $\mathbb{R}^{n_\kappa}$ isometrically in $\mathbb{R}^{n_\ell}$, i.e., it satisfies for every $u, v \in \mathbb{R}^{n_\kappa}$, $\|T_{\kappa,\ell}(u) - T_{\kappa,\ell}(v)\| = \|u - v\|$.*

The proof of Theorem 5.2 appears in Appendix B. This theorem is used to prove Lemma 5.3, which shows that the μIDM and the MTM isometry can be used to approximate the embedded diffusion coordinates of every data point up to an approximation error of $\frac{\mu}{2}$.

Lemma 5.3. *Assume we have Φ, S, $\widehat{\Phi}$ from Theorem 5.1. Let T be the MTM isometry (Definition 5.1) between the μIDM embedded space $\widehat{\Phi}(\cdot)$ and the DM embedded space $\Phi(\cdot)$. Then, every data point $x \in M$ satisfies $\|\Phi(x) - (T \circ \widehat{\Phi})(x)\| \leq \frac{\mu}{2}$.*

Proof. Recall that by Definition 5.1 of the MTM isometry, $\Phi(x) = T\widehat{\Phi}(x)$, $x \in S$. Then, we only have to consider data points that are not in the dictionary S. Consider such a data point $x' \in M \setminus S$. Let $S' = S \cup \{x'\}$ and let $\widehat{\Phi}'$ be the ONM of S'. Assume also that T' is the MTM isometry between $\Phi(\cdot)$ and $\widehat{\Phi}'(\cdot)$. Let T'' be the MTM isometry between $\widehat{\Phi}'(\cdot)$ and $\widehat{\Phi}(\cdot)$. The dictionary membership criterion in Algorithm 5.1 guarantees that for $x' \notin S$, $\|\widehat{\Phi}'(x') - (T'' \circ \widehat{\Phi})(x')\| \leq \frac{\mu}{2}$. By the application of Theorem 5.2 to the MTM isometry T' we get

$$\left\|\left(T' \circ \widehat{\Phi}'\right)(x') - \left(T' \circ T'' \circ \widehat{\Phi}\right)(x')\right\| = \left\|\widehat{\Phi}'(x') - \left(T'' \circ \widehat{\Phi}\right)(x')\right\| \leq \frac{\mu}{2}. \tag{5.3}$$

According to Definition 5.1, we have $T = (T' \circ T'')$ and $\Phi(x') = (T' \circ \widehat{\Phi}')(x')$. Introducing T and $\Phi(x')$ into Eq. (5.3)

$$\left\|\left(T' \circ \widehat{\Phi}'\right)(x') - \left(T' \circ T'' \circ \widehat{\Phi}\right)(x')\right\| = \left\|\Phi(x') - (T \circ \widehat{\Phi})(x')\right\| \leq \frac{\mu}{2}. \qquad \square$$

The dictionary construction in Algorithm 5.1 compares the ONM approximations of each data point $x_{\kappa+1}$ based on the dictionary S_κ with the PDM of $S_\kappa \cup \{x_{\kappa+1}\}$. This comparison is done by utilizing an MTM isometry (see Definition 5.1). The result in Lemma 5.3 shows that this membership criterion guarantees that the μIDM embedding followed by the MTM transformation is sufficiently close to the DM embedding (up to a perturbation of size $\mu/2$). Lemma 5.3 is used to prove Theorem 5.1, which shows that the μIDM embedding of M is μ-isometric to the application of DM embedding to M.

Proof of Theorem 5.1. Consider two data points $x, y \in M$. Then, by using Lemma 5.3 we get $\|\Phi(x) - (T \circ \widehat{\Phi})(x)\| < \frac{\mu}{2}$ and $\|\Phi(y) - (T \circ \widehat{\Phi})(y)\| < \frac{\mu}{2}$. Therefore, from the triangle inequality

$$\begin{aligned}\left\|(T \circ \widehat{\Phi})(x) - (T \circ \widehat{\Phi})(y)\right\| &\leq \left\|\Phi(x) - (T \circ \widehat{\Phi})(x)\right\| + \left\|\Phi(x) - \Phi(y)\right\| \\ &\quad + \left\|\Phi(y) - (T \circ \widehat{\Phi})(y)\right\| \leq \left\|\Phi(x) - \Phi(y)\right\| + \mu,\end{aligned}$$

and

$$\begin{aligned}\left\|\Phi(x) - \Phi(y)\right\| &\leq \left\|\Phi(x) - (T \circ \widehat{\Phi})(x)\right\| + \left\|(T \circ \widehat{\Phi})(x) - (T \circ \widehat{\Phi})(y)\right\| \\ &\quad + \left\|\Phi(y) - (T \circ \widehat{\Phi})(y)\right\| \leq \left\|(T \circ \widehat{\Phi})(x) - (T \circ \widehat{\Phi})(y)\right\| + \mu.\end{aligned}$$

According to Theorem 5.2, the isometry in these equations satisfy $\|(T \circ \widehat{\Phi})(x) - (T \circ \widehat{\Phi})(y)\| = \|\widehat{\Phi}(x) - \widehat{\Phi}(y)\|$. Thus, we get $\|\Phi(x) - \Phi(y)\| - \mu \leq \|\widehat{\Phi}(x) - \widehat{\Phi}(y)\| \leq \|\Phi(x) - \Phi(y)\| + \mu$. □

5.2. A spectral bound for the kernel approximation

In this section, we quantify the approximation quality of the diffusion kernel A from Eq. (2.3) by $\widehat{A}$ from Eq. (4.4). Lemma 5.4 provides a bound for the difference between the associated spectra, while Proposition 5.5 shows the similarity between these operators.

Recall that the eigenvalues of A are the diagonal elements of Σ, $1 = \sigma_1 \geq \ldots \geq \sigma_n > 0$ and the eigenvalues of $\hat{A}$ are the diagonal elements of Λ, $\lambda_1 \geq \ldots \geq \lambda_{|S|} > \lambda_{|S|+1} = \ldots = 0$. For the proofs, we consider a full SVD of $\widehat{A}$, rather than its *s–SVD* from Eq. (4.4). Let Θ be the $n \times n$ orthogonal matrix, whose $n \times s$ leftmost submatrix is $\widehat{\phi}$, and the rest $n \times (n-s)$ constitute orthonormal basis for the orthogonal complement of the subspace spanned by the columns of $\widehat{\phi}$. Additionally, let $\widehat{\Lambda}$ be the diagonal $n \times n$ matrix, whose upper left $s \times s$ block is Λ and the rest are zeros.

Lemma 5.4. *The difference between the spectra of A and $\widehat{A}$ are bounded by $\frac{\mu}{2}\sqrt{n-s}\|Q\|^{1/2}$, i.e., for any $j = 1, \ldots, n$, $|\sigma_j - \lambda_j| \leq \frac{\mu}{2}\sqrt{n-s}\|Q\|^{1/2}$.*

Proof. Due to Lemma 5.3, there is an orthogonal transformation T, for which $\|Q^{-1/2}\widehat{\phi}\Lambda T - Q^{-1/2}\phi\Sigma\| \leq \frac{\mu}{2}\sqrt{n-s}$. Thus, according to Weyl's inequality, for every $j = 1, \ldots, n$

$$\begin{aligned}|\lambda_j - \sigma_j| &\leq \|\widehat{\phi}\Lambda T - \phi\Sigma\| \\ &\leq \left\|Q^{1/2}\right\|\left\|Q^{-1/2}\widehat{\phi}\Lambda T - Q^{-1/2}\phi\Sigma\right\| \\ &\leq \frac{\mu}{2}\sqrt{n-s}\|Q\|^{1/2}. \qquad \square\end{aligned}$$

The last step assumes a worst-case scenario in which the difference between $Q^{-1/2}\phi\Sigma$ and $Q^{-1/2}\widehat{\phi}\Lambda T$ is concentrated in a single coordinate with absolute value of $\frac{\mu}{2}$. The spectrum of A is of great importance, since it indicates the dimensionality of the embedding for which the lost information is negligible. Lemma 5.4

Table 5.1
μIDM computational complexity: m is the size of the ambient space, n is the number of samples, s is the dictionary size.

Operation	Operations
Initialization (done once)	$O(mn^2)$
Membership test	$O(n^2s^2 + ns^3)$
Update	$O(s)$

states that the spectrum of A can be approximated with an error controlled by μ. More specifically, the diagonal matrix Q holds on its diagonal the degrees of the data points (see Eq. (2.2)), and it satisfies $\|Q\| = \max_{x\in M} q(x)$. Thus, μ can be fixed such that the bound from Lemma 5.4 is sufficiently tight and the numerical ranks of these operators are similar.

Proposition 5.5 is a direct consequence of Lemma 5.4. It shows that A and $\widehat{A}$ are almost similar, namely they act almost the same, up to orthogonal change of basis. It shows that $\widehat{A}$ is a rank-s approximation of A, where, as in Lemma 5.4 the error is controlled by μ.

Proposition 5.5. *There exists an orthogonal $n \times n$ matrix P for which $\|P\widehat{A}P^T - A\| \leq \frac{\mu}{2}\sqrt{n-s}\|Q\|^{1/2}$.*

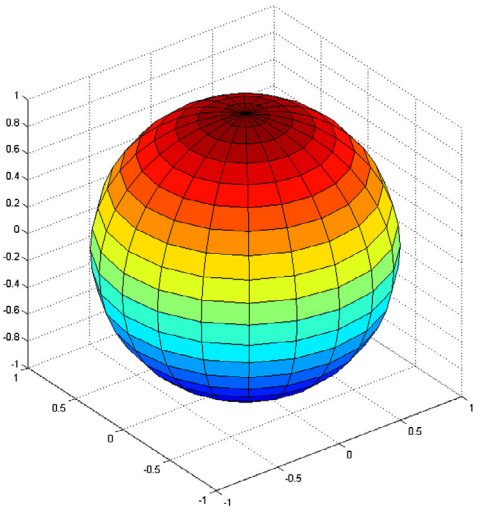
Proof. Obviously, $\widehat{A} = \Theta\widehat{\Lambda}\Theta^T$. Define the $n\times n$ matrix $P \triangleq \phi\Theta^T$. Then, P is orthogonal, and $P\widehat{A}P^T = \phi\widehat{\Lambda}\phi^T$. Thus, due to Lemma 5.4, $\|P\widehat{A}P^T - A\| \leq \frac{\mu}{2}\sqrt{n-s}\|Q\|^{1/2}$. □

5.3. Computational complexity

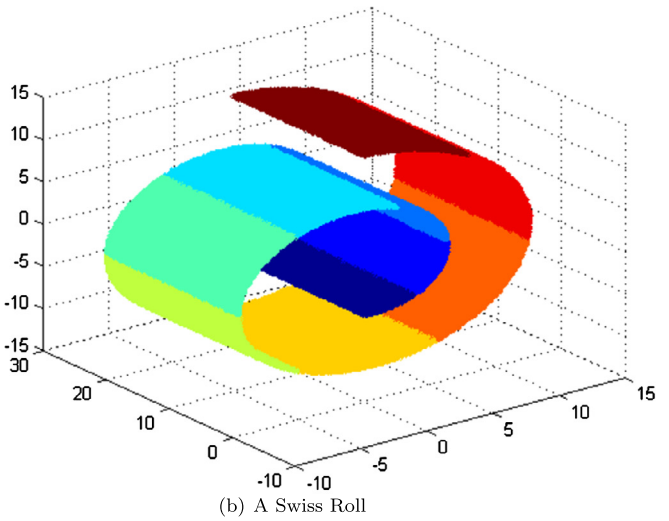
The analysis of the computational complexity is divided between the three main parts of μIDM: 1. Initialization. 2. Membership test, and 3. Update. This section assumes that the μIDM is applied to M of size n and finalize with a dictionary of size s.

1. *Initialization*: μIDM computes the pairwise affinity matrix A, the corresponding degree matrix Q and the first mapping approximation $\widehat{\Phi}$. An accurate computation of the degree requires $O(mn^2)$ operations where m is the dimension of the ambient space. Additionally, the mapping initialization $\widehat{\Phi}$ requires an additional $O(n)$ operations. This step is done once.
2. *Membership test*: At the κ-th iteration, we have $|D_\kappa| = n_\kappa \leq \kappa$. For a new data point $x_{\kappa+1}$ in M, the μIDM computes the matrix C (Eq. (4.8)) that takes $O(n_\kappa^2 n)$ operations. The matrix C is decomposed by the application of an SVD. It takes $O({n_\kappa}^3)$ operations. Furthermore, the new mapping $\widehat{\Phi}'$ is computed according to Eq. (4.7). It takes $O({n_\kappa}^3)$ operations. The MTM computation is based on the inverse $[\widehat{\Phi}_\kappa(S_\kappa)]^{-1}$ (according to Definition 5.1) with additional complexity of $O({n_\kappa}^3)$ operations. Computation of β (Eq. (5.2)) takes $O({n_\kappa}^3)$ operations. Therefore, the total computational complexity of this step is $O(n_\kappa^2 n + {n_\kappa}^3)$ operations per a single iteration. Assuming n iterations the total cost of this step is $O(n^2s^2 + ns^3)$.
3. *Update step*: For each new member in the dictionary, the μIDM updates the relevant index set with a cost of $O(1)$ operations per iteration or a cost $O(s)$ operations for the entire run.

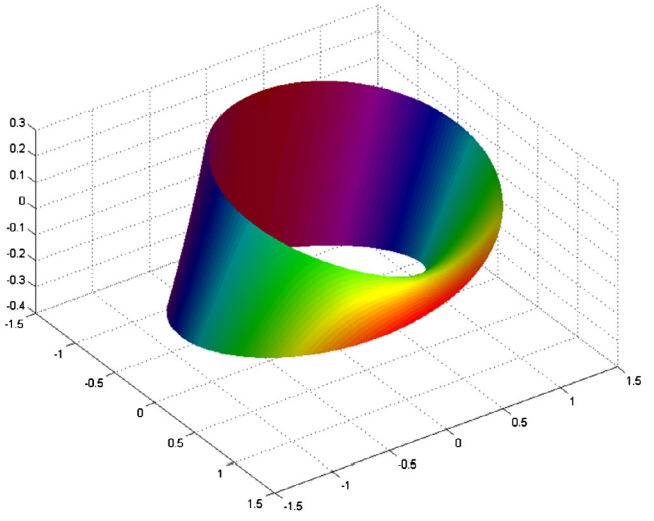
Table 5.1 summarizes the estimated complexity for computing μIDM. The most expensive task is the computation of the affinity matrix and the degree matrix, which takes approximately $O(mn^2)$ operations. Under the assumption that $m \ll n$ and additionally, μ was chosen such that the final dictionary size s is smaller than n such that $s \ll \sqrt{n}$, then the μIDM is more computationally efficient by an order of magnitude in comparison to DM computation.



(a) A Sphere



(b) A Swiss Roll



(c) A Mobius Band

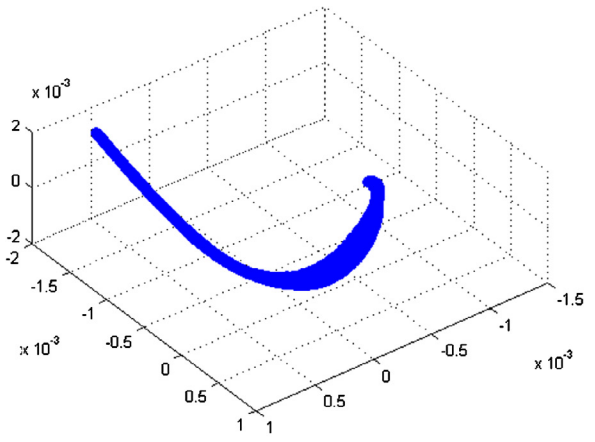
Fig. 6.1. Examined manifolds.

6. Experimental results

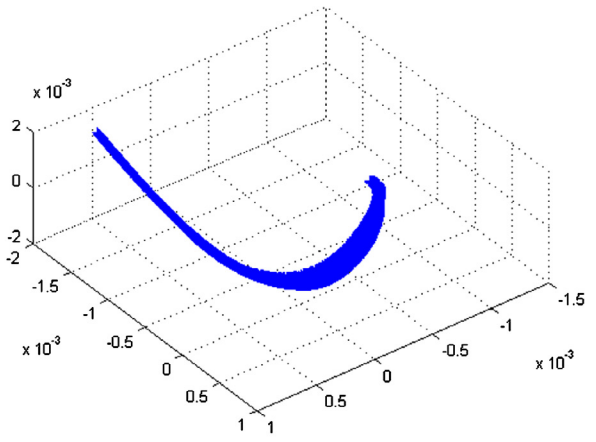
This section displays the μIDM characteristics for three manifolds given in Fig. 6.1. Specifically, these examples validate Theorem 5.1 and Lemma 5.3. Algorithm 5.1 is used in the analysis.

The examined manifolds, which reside in $\mathbb{R}^3$ and illustrated in Fig. 6.1, include the unit sphere S^2 (a), the three dimensional Swiss-roll (b) and the three dimensional Mobius band (c). Each dataset was embedded in a high-dimensional space, then it was uniformly sampled in 10 000 data points. These datasets were embedded in $\mathbb{R}^{17}$ by a random full-rank linear transformation, whose representative matrix is a 17×3 matrix where its entries are uniformly i.i.d. in $[0, 1]$. Its full rank guaranties the preservation of the intrinsic dimensionality of the manifolds.

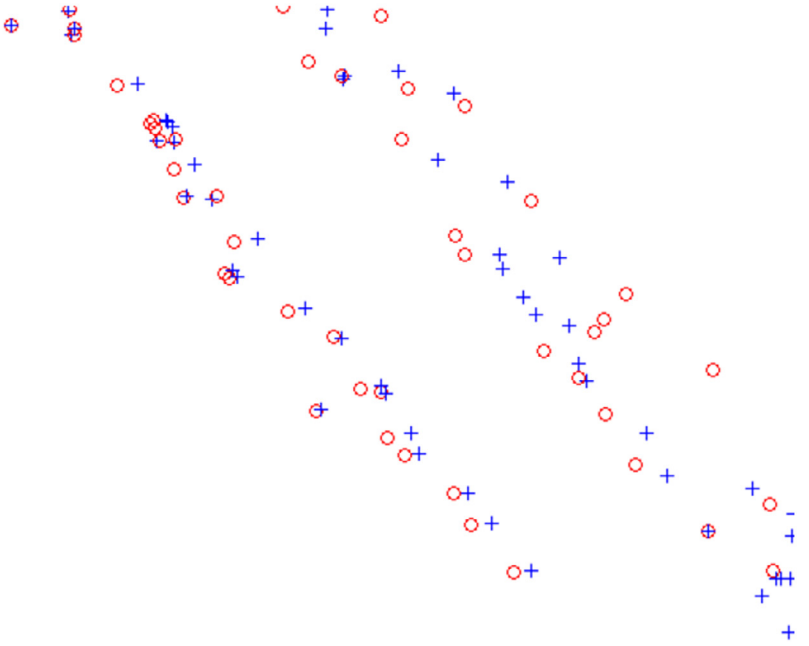
Algorithm 5.1 finds the μIDM of each dataset. Fig. 6.2 compares between the first three coordinates of μIDM and DM embeddings of the Swiss-roll. μIDM completes the scanning of the 10 000 data points with a dictionary of size 236 where $\mu = 1.25 \cdot 10^{-4}$. It is clear from the figure that both maps are similar even though μIDM utilized an SVD of a matrix of size 236×236 instead of SVD of size $10\,000 \times 10\,000$.



(a) The μIDM of a swiss-roll



(b) The DM of a Swiss roll



(c) DM (+) and μIDM (O)

Fig. 6.2. The embedding of a Swiss roll via DM and μIDM.

In order to analyze the approximation errors in terms of pairwise distances and coordinates, the Cumulative Distribution Function (CDF) of each error of μIDM relative to DM are computed. The corresponding CDF is the probability that any approximated coordinate of a data point or approximated distance in the embedded space is less than or equal to a threshold τ. More rigorously, the CDF is defined by

$$F\big(\tau, f(\text{Error})\big) = \Pr[\text{Error} \leq \tau], \tag{6.1}$$

where $f(\text{Error})$ is the distribution function of the respective error.

The CDF describes an interval on which there is a positive probability to find an error and the percentage of non-negligible errors from all the error distributions. The estimated CDFs of the two errors from the Swiss Roll example are presented in Fig. 6.3. In each case, $f(\text{Error})$ is estimated by integrating the corresponding histogram of the relevant error. For the coordinates error calculation, which were caused by the μIDM embedding, the MTM between μIDM and DM is utilized.

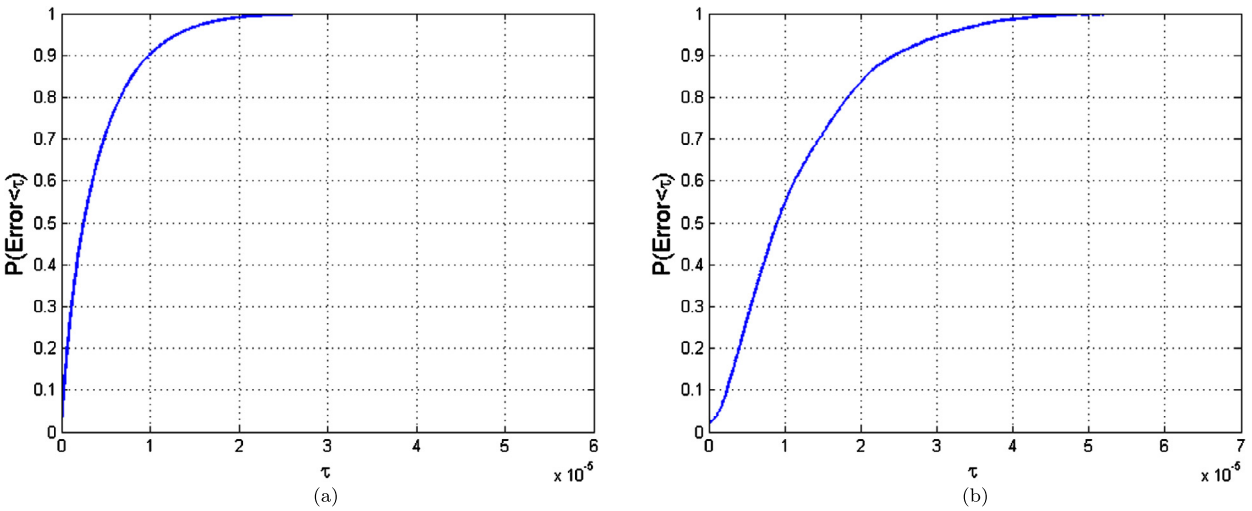


Fig. 6.3. The CDFs of (a) pairwise distance error and (b) coordinates mapping error between μIDM and DM embeddings.

Table 6.1
μIDM characterization summary.

Dataset	ε	n	$\lvert S\rvert$	Max error between μIDM and DM embedding	Max pairwise distance error
Sphere	1	10 000	147	0.29×10^{-5}	0.35×10^{-5}
Swiss roll	70	10 000	236	0.46×10^{-4}	0.60×10^{-4}
Mobius band	1	10 000	85	0.18×10^{-5}	0.37×10^{-5}

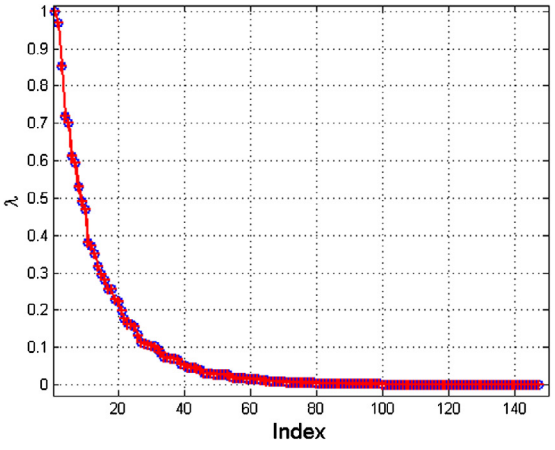
Table 6.2
μIDM: spectral difference, bound and empirical measurements.

Dataset	μ	$\lVert Q\rVert^{\frac{1}{2}}$	$\lvert S\rvert$	Approximation bound (Lemma 5.4)	$\max \lvert\lambda_j - \sigma_j\rvert$
Sphere	7.80×10^{-6}	32.11	147	0.0125	0.2×10^{-3}
Swiss roll	1.25×10^{-4}	27.12	236	0.1675	0.9×10^{-3}
Mobius band	7.80×10^{-6}	40.88	85	0.0159	0.2×10^{-3}

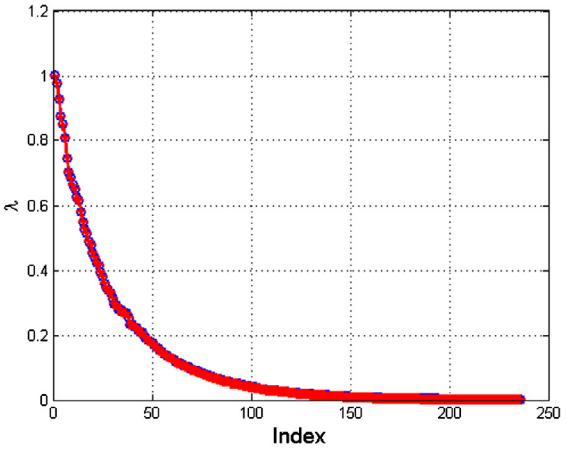
According to the calculated CDFs that are shown in Fig. 6.3, the coordinates errors from the application of μIDM have positive probabilities only on the interval $[0, 0.46 \times 10^{-4}]$ as proved by Lemma 5.3. In addition, the pairwise distance errors from the application of μIDM have positive probabilities only on the interval $[0, 0.60 \times 10^{-4}]$. This error is smaller than the error derived in Theorem 5.1.

Fig. 6.3 shows that in 50% of the cases, the calculated CDF probabilities of both errors are smaller by approximately one order of magnitude than their worst-cases. Table 6.1 summarizes the measured error for the three datasets.

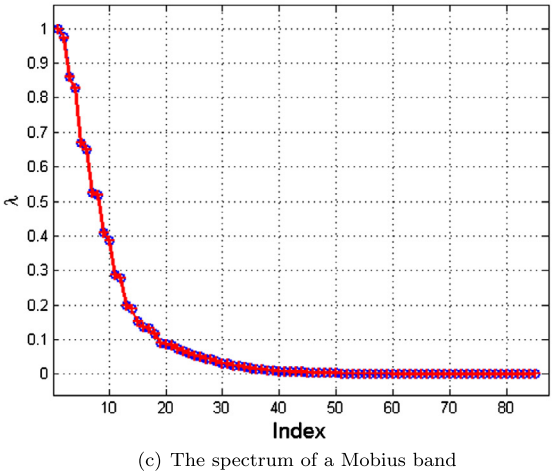
Lemma 5.4 discusses the convergence of the μIDM spectrum to the associated DM spectrum. Fig. 6.4 compares the spectral decays of DM and μIDM for the three datasets. Table 6.2 provides the estimated bound and the measured difference for each dataset. Fig. 6.4 and Table 6.2 suggest that the bound is not tight. The empirical difference is at least one order better than the corresponding bound. Furthermore, for a sufficiently small μ, μIDM has a similar spectral decay as DM. Thus, when DM generates a low dimensional embedding due to its spectral decay, μIDM embedding uses the same number of eigenvectors. Additionally, the empirical difference between the spectra suggests that a diffusion time greater than 1 can also be efficiently approximated by μIDM.



(a) The spectrum of a Sphere



(b) The spectrum of a Swiss Roll



(c) The spectrum of a Mobius band

Fig. 6.4. The eigenvalues of DM (denoted by +) and μIDM (denoted by ∘) for each example.

7. Discussion and conclusions

This paper presents a computationally efficient embedding scheme that approximately preserves the diffusion distances between embedded data points. The presented method scans the entire dataset once and

validates the embedding approximation accuracy for each data point. This validation compares between a dictionary-based embedding and the exact DM embedding, which is efficiently computed over a subset of data points.

The single scan of the dataset uses an iterative approach, and each iteration utilizes several techniques. In each iteration, a newly processed data point is considered for inclusion in the dictionary that was constructed from previously scanned data points and does not include the new data point. First, the Nyström extension is applied to the dictionary in order to approximate the embedding of the newly processed data point. Then, the PDM embedding of this data point, together with the dictionary, is efficiently computed. Finally, an MTM transformation is designed between the Nyström approximated embedded spaces and the exact PDM embedded spaces. This MTM transformation is used to measure the approximation accuracy of the embedding map. The entire computational complexity of this iterative process is lower than computational complexity of DM. The exact number of required operations depends on the dictionary size and on the dimensionality m of the original ambient space.

The proposed method utilizes the exact pairwise affinities between data points in a given dataset. This computation limits the ability to reduce the computational complexity. However, future work will explore how to efficiently approximate this computation and quantify the resulting embedding errors. In order to demonstrate the effectiveness of the proposed method, we analyzed several synthetic datasets. This analysis showed that any choice of an approximation bound μ leads to a mapping that is similar to DM up to a pairwise distance error μ. Furthermore, the spectral properties of μIDM and their relation to the spectral characteristics of DM were explored and proved. This spectral characterization suggests that the proposed method allows to have an effective dimensionally reduction that is similar to DM.

8. Future work and possible implementation optimizations

Future work will focus on optimizing several aspects of the presented dictionary construction algorithm:

Data scanning order: Algorithm 5.1 constructs a μ-Isometric DM, independently of the data scanning order. Yet, the dictionary depends on the scanning order. However, additional knowledge (or assumptions) on the organization of the data and the order in which it is scanned can improve the results. For example, it can yield faster convergence to a steady dictionary or reduce the final size of the constructed dictionary. Future work will utilize methods such as pivoted Cholesky and QR decompositions to guide the data scanning iterations. The effects of such strategies on the performance, dictionary size and approximation errors will be analyzed in order to provide an optimally-ordered data scanning process.

Randomization: This paper uses a deterministic approach for defining and implementing the presented algorithm. Such approach provides provable accuracy thresholds and its analysis does not include probabilistic elements. In fact, the proved results regarding the distance approximation errors in Section 5.1 refer to the worst-case scenario. Future work will utilize randomized methods to improve the performance of the algorithm.

For example, a dominant part of the dictionary construction process is the low-rank Nyström approximation provided by the constructed ONM (see Definition 4.1 and Section 4). An alternative approach for obtaining such an approximation is to use a randomized method such as the methods presented in [18]. Such an implementation will be explored in future works, and the impact of this change on the distance approximation error will be analyzed.

Another utilization of randomness that will be explored is to utilize a divide & conquer strategy similar to the “Shake & Bake” clustering approach from [14,13]. Using such an approach, instead of building a single dictionary, the algorithm will construct several dictionaries (e.g., using subsets of the data or even just different scanning orders) and then fuse them together. The resulting

dictionary can thus be less dependent on the initial presentation of the data and more robust to its variations. Furthermore, a dictionary fusing method can prove useful when additional data is continuously streamed to the system after the initial analysis is done.

Approximated neighborhoods: The initialization of the presented algorithm relies on having access to all the mutual pairwise distances between data-points. This assumption yields a heavy computational toll on the complexity of the algorithm (see Section 5.3), but it is necessary for proving the approximation error bounds in Section 5.1. Future works will consider ways to remove this limiting assumption and allow the construction to be based on partial knowledge of these distances. Then, without the need for exact computation of all the distances, the initialization phase can be based on fast nearest-neighbor search or similar approaches in order to compute the required neighborhoods and distances in an efficient manner.

Acknowledgments

This research was partially supported by the Israel Science Foundation (Grant No. 1041/10), by the Israeli Ministry of Science & Technology Grants Nos. 3-9096 and 3-10898 and by US–Israel Binational Science Foundation (BSF 2012282). The third author was also supported by the Eshkol Fellowship from the Israeli Ministry of Science & Technology. Part of this research is supported by a Fellowship from the University of Jyväskylä.

Appendix A. Proof of the invertibility of $\widehat{\Phi}_\kappa(S_\kappa)$

This appendix is dedicated to the presentation and proof of Lemma A.1. It uses notations that were presented in Section 5. The presented lemma shows (and proves) the invertibility of the matrix $\widehat{\Phi}_\kappa(S_\kappa)$, which consists of the ONM embedding of the data points in the set S_κ as its rows.

Lemma A.1. *The matrix $\widehat{\Phi}_\kappa(S_\kappa)$ in invertible.*

Proof. By Definition 4.1, $\widehat{\Phi}_\kappa(S_\kappa)$ is the upper $s \times s$ submatrix of the $s \times n$ matrix $Q^{-1/2}\widehat{\phi}\Lambda$. Obviously, is suffices to prove that the upper $s \times s$ submatrix of $\widehat{\phi}\Lambda$ is invertible. Due to Eqs. (4.5), (4.8) and (4.7), this submatrix equals to $A_{(S,S)}^{1/2}\psi\Lambda^{1/2}$, where $\psi\Lambda\psi$ is the SVD of $C = A_{(S,S)} + A_{(S,S)}^{-1/2}A_{(S,\bar{S})}A_{(S,\bar{S})}^{T}A_{(S,S)}^{-1/2}$. Since A is strictly positive definite, C is invertible and, as a consequence, so is Λ. □

Appendix B. Proof of Theorem 5.2

This appendix presents the proof of Theorem 5.2, which states that the MTM transformation in Definition 5.1 is an isometry. In order to prove this theorem, we first prove Lemma B.1. The notation used in this section are the same as those used in Definition 5.1 and Theorem 5.2.

Lemma B.1. *The matrix $[T_{\kappa,\ell}]$ of size $n_\kappa \times n_\ell$, which defines the MTM in Definition 5.1, satisfies $[T_{\kappa,\ell}][T_{\kappa,\ell}]^T = I$, where I is the $n_\kappa \times n_\kappa$ identity matrix.*

Proof. By Definition 5.1, we have $[T_{\kappa,\ell}] \triangleq [\widehat{\Phi}_\kappa(S_\kappa)]^{-1}[\widehat{\Phi}_\ell(S_\kappa)]$. Thus,

$$[T_{\kappa,\ell}][T_{\kappa,\ell}]^T = \left[\widehat{\Phi}_\kappa(S_\kappa)\right]^{-1}\left[\widehat{\Phi}_\ell(S_\kappa)\right]\left[\widehat{\Phi}_\ell(S_\kappa)\right]^T\left(\left[\widehat{\Phi}_\kappa(S_\kappa)\right]^{-1}\right)^T. \tag{B.1}$$

The entries in the matrix $[\widehat{\Phi}_\ell(S_\kappa)][\widehat{\Phi}_\ell(S_\kappa)]^T$ of size $n_\kappa \times n_\kappa$ are the inner products between the embeddings of data points in S_κ that were generated by the application of ONM to S_ℓ. Since $S_\kappa \subseteq S_\ell$, Corollary 4.2

is applied to these inner products. They are equal to the inner products generated by DM embedding. In addition, these inner products are also preserved by the application of ONM to S_κ, which are the entries of the matrix $[\widehat{\Phi}_\kappa(S_\kappa)][\widehat{\Phi}_\kappa(S_\kappa)]^T$. Therefore, we can replace $[\widehat{\Phi}_\ell(S_\kappa)][\widehat{\Phi}_\ell(S_\kappa)]^T$ with $[\widehat{\Phi}_\kappa(S_\kappa)][\widehat{\Phi}_\kappa(S_\kappa)]^T$ in Eq. (B.1) to get

$$[T_{\kappa,\ell}][T_{\kappa,\ell}]^T = \left(\left[\widehat{\Phi}_\kappa(S_\kappa)\right]^{-1}\left[\widehat{\Phi}_\kappa(S_\kappa)\right]\right)\left(\left[\widehat{\Phi}_\kappa(S_\kappa)\right]^{-1}\left[\widehat{\Phi}_\kappa(S_\kappa)\right]\right)^T = II^T = I. \qquad \square$$

Proof of Theorem 5.2. Consider two arbitrary data points $u, v \in \mathbb{R}^{n_\kappa}$ and their MTM-based transformed versions $T_{\kappa,\ell}(u), T_{\kappa,\ell}(v) \in \mathbb{R}^{n_\ell}$, respectively. Then, by using Definition 5.1, we can write the inner product of the transformed data points as

$$\left\langle T_{\kappa,\ell}(u), T_{\kappa,\ell}(v)\right\rangle = \left\langle u[T_{\kappa,\ell}], v[T_{\kappa,\ell}]\right\rangle = u[T_{\kappa,\ell}][T_{\kappa,\ell}]^T v^T.$$

Due to Lemma B.1, we get $\langle T_{\kappa,\ell}(u), T_{\kappa,\ell}(v)\rangle = uv^T = \langle u, v\rangle$. Since all the inner products are preserved by the MTM transformation (recall u, v are arbitrary) we also get $\|T_{\kappa,\ell}(u) - T_{\kappa,\ell}(v)\|^2 = \langle T_{\kappa,\ell}(u-v), T_{\kappa,\ell}(u-v)\rangle = \langle u-v, u-v\rangle = \|u-v\|^2$, which proves Theorem 5.2. $\square$

References

[1] D. Achlioptas, F. McSherry, Fast computation of low rank matrix approximations, in: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, STOC '01, ACM, 2001, pp. 611–618.
[2] N. Alon, N. Kahale, A spectral technique for coloring random 3-colorable graphs, SIAM J. Comput. 26 (6) (1997) 1733–1748.
[3] B. Aspvall, J.R. Gilbert, Graph coloring using eigenvalue decomposition, Technical report, Cornell University, 1983.
[4] B. Aspvall, J.R. Gilbert, Graph coloring using eigenvalue decomposition, SIAM J. Algebr. Discrete Methods 5 (4) (1984) 526–538.
[5] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2003) 1373–1396.
[6] A. Bermanis, A. Averbuch, R.R. Coifman, Multiscale data sampling and function extension, Appl. Comput. Harmon. Anal. 34 (1) (2013) 15–29.
[7] A. Bermanis, G. Wolf, A. Averbuch, Cover-based bounds on the numerical rank of gaussian kernels, Appl. Comput. Harmon. Anal. 36 (2) (2014) 302–315.
[8] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, Comput. Netw. ISDN Syst. 30 (1–7) (1998) 107–117.
[9] M.M. Bronstein, A.M. Bronstein, Shape recognition with spectral distances, IEEE Trans. Pattern Anal. Mach. Intell. 33 (5) (2011) 1065–1071.
[10] F. Chung, Spectral Graph Theory, vol. 92, CBMS–AMS, May 1997.
[11] R.R. Coifman, S. Lafon, Diffusion maps, Appl. Comput. Harmon. Anal. 21 (1) (2006) 5–30.
[12] J.K. Cullum, R.A. Willoughby, Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Volume 1, Theory, Classics Appl. Math., vol. 41, SIAM, 2002.
[13] G. David, Anomaly detection and classification via diffusion processes in hyper-networks, PhD thesis, School of Computer Science, Tel Aviv University, March 2009.
[14] G. David, A. Averbuch, Hierarchical data organization, clustering and denoising via localized diffusion folders, Appl. Comput. Harmon. Anal. 33 (1) (2012) 1–23.
[15] D.L. Donoho, C. Grimes, Hessian eigenmaps: new locally linear embedding techniques for high dimensional data, Proc. Natl. Acad. Sci. USA 100 (May 2003) 5591–5596.
[16] P.F. Felzenszwalb, D.P. Huttenlocher, Efficient graph-based image segmentation, Int. J. Comput. Vis. 59 (2) (2004) 167–181.
[17] C. Fowlkes, S. Belongie, F. Chung, J. Malik, Spectral grouping using the Nyström method, IEEE Trans. Pattern Anal. Mach. Intell. 26 (2) (2004) 214–225.
[18] N. Halko, P.-G. Martinsson, J.A. Tropp, Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions, SIAM Rev. 53 (2) (2011) 217–288.
[19] L. Jingen, Y. Yang, M. Shah, Learning semantic visual vocabularies using diffusion distance, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, CVPR 2009, 2009, pp. 461–468.
[20] Y. Keller, R.R. Coifman, S. Lafon, S.W. Zucker, Audio-visual group recognition using diffusion maps, IEEE Trans. Signal Process. 58 (1) (2010) 403–413.
[21] J.B. Kruskal, Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, Psychometrika 29 (1964) 1–27.
[22] S. Lafon, Diffusion maps and geometric harmonics, PhD thesis, Yale University, May 2004.
[23] S. Lafon, Y. Keller, R.R. Coifman, Data fusion and multicue data matching by diffusion maps, IEEE Trans. Pattern Anal. Mach. Intell. 28 (11) (2006) 1784–1797.

[24] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (December 2000) 2323–2326.
[25] X. Rui, S. Damelin, D.C. Wunsch, Applications of diffusion maps in gene expression data-based cancer diagnosis analysis, in: 29th Annual International Conference of the IEEE, Engineering in Medicine and Biology Society, 2007, EMBS 2007, 2007, pp. 4613–4616.
[26] A. Schclar, A. Averbuch, N. Rabin, V. Zheludev, K. Hochman, A diffusion framework for detection of moving vehicles, Digital Signal Process. 20 (1) (2010) 111–122.
[27] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.
[28] G.W. Stewart, J. Sun, Matrix Perturbation Theory, Academic Press, INC, 1990.
[29] R. Talmon, I. Cohen, S. Gannot, Supervised source localization using diffusion kernels, in: 2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA, 2011, pp. 245–248.
[30] R. Talmon, I. Cohen, S. Gannot, Single-channel transient interference suppression with diffusion maps, IEEE Trans. Audio, Speech Language Process. 21 (1–2) (2013) 132–144.
[31] R. Talmon, D. Kushnir, R.R. Coifman, I. Cohen, S. Gannot, Parametrization of linear systems using diffusion kernels, IEEE Trans. Signal Process. 60 (3) (2012) 1159–1173.
[32] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[33] U. von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.
[34] G. Yang, X. Xu, J. Zhang, Manifold alignment via local tangent space alignment, SIAM J. Sci. Comput. 26 (1) (2005) 313–338.
[35] Z. Zhang, H. Zha, Principal manifolds and nonlinear dimension reduction via local tangent space alignment, Technical Report CSE-02-019, Department of Computer Science and Engineering, Pennsylvania State University, 2002.