Product Manifold Learning

Sharon Zhang
Princeton University

Amit Moscovich
Princeton University

Amit Singer
Princeton University

Abstract

We consider dimensionality reduction for data sets with two or more independent degrees of freedom. For example, measurements of deformable shapes with several parts that move independently fall under this characterization. Mathematically, if the space of each continuous independent motion is a manifold, then their combination forms a product manifold. In this paper, we present an algorithm for manifold factorization given a sample of points from the product manifold. Our algorithm is based on spectral graph methods for manifold learning and the separability of the Laplacian operator on product spaces. Recovering the factors of a manifold yields meaningful lower-dimensional representations, allowing one to focus on particular aspects of the data space while ignoring others. We demonstrate the potential use of our method for an important and challenging problem in structural biology: mapping the motions of proteins and other large molecules using cryo-electron microscopy data sets.

1 Introduction

Consider a data-generating process T which maps vectors of latent (unobserved) variables to observations,

$$(\theta_1, \ldots, \theta_m) \xmapsto{T} \mathbf{x} \in \mathbb{R}^D. \tag{1}$$

Our focus is on the case in which the latent variables are low-dimensional and the observations are high-dimensional vectors. As an illustrative example, consider an industrial articulated robot where the latent variables $\theta_1, \ldots, \theta_m$ correspond to the angles of its m rotary joints, and the observed vector $\mathbf{x}$ is a set of measurements recorded by an external monitoring system.

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

When T is deterministic (i.e. without noise), smooth and injective, then the space of observations is a submanifold of $\mathbb{R}^D$ (Lee, 2012, Theorem 4.14). This observation underlies the field of *manifold learning* (Tenenbaum et al., 2000; Belkin and Niyogi, 2003; Coifman and Lafon, 2006; Talmon et al., 2013). A key problem in this field is dimensionality reduction, where a set of high-dimensional observations $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^D$ is mapped to a low-dimensional space, hopefully one whose dimension is not much larger than the dimension of the latent space.

In this work, we explore the idea of manifold learning when the latent space is a *product manifold*. If each latent variable θ_i lies on a manifold $\mathcal{M}_i$ of dimension d_i then the latent space is the Cartesian product

$$\mathcal{M} = \mathcal{M}_1 \times \cdots \times \mathcal{M}_m \ , \tag{2}$$

which is a manifold of dimension $d = d_1 + \cdots + d_m$. We refer to the decomposition (2) as a *manifold factorization* of $\mathcal{M}$ and to each of the manifolds $\mathcal{M}_i$ as *factors*. In this paper, we present a data-dependent algorithm for manifold factorization based on spectral graph-based methods for manifold learning (Belkin and Niyogi, 2003; Coifman and Lafon, 2006). These methods output a set of graph Laplacian eigenvectors which approximate the eigenfunctions of the Laplacian differential operator on the manifold. By mapping the observations to their eigenvector coordinates one can obtain a low-dimensional embedding of the data space (Bates, 2014). On a product manifold, it is well-known that the Laplacian eigenfunctions are nothing but pointwise products of eigenfunctions on the factor manifolds, one eigenfunction from each factor (see Section 2). The eigenfunctions are typically not observable; however, they can be approximated by the eigenvectors of a data-dependent graph Laplacian. All that is left is to find an approximate factorization of the set of Laplacian eigenvectors. We describe our particular approach for this factorization in Section 3.

In traditional methods for dimensionality reduction, each data point is mapped to a point in a low-dimensional vector space. In contrast to this, after running our algorithm, each point is mapped to a low-dimensional product space, where each factor encodes

a different aspect of the data. This opens the door to exciting new methods for data visualization and representation. One can, for instance, use it to visualize observations that come from a 5-dimensional latent space as pairs of points on a 3-dimensional and a 2-dimensional manifold, associate different interpretations to the different factors, and ignore factors that correspond to nuisance variables.

In this paper, we mainly focus on products of two manifolds. However, extensions to more than two manifolds are briefly discussed in Section 5.

1.1 Related Work

One work that is closely related to ours is the spectral method of Singer (2006a) for linear independent component analysis (ICA), as it is also based on the separability of the Laplacian on a product space. This was later extended to a method for non-linear ICA that assumes a product latent space and that the Jacobian of the mapping $(\theta_1, \ldots, \theta_m) \mapsto \mathbf{x}$ is either known or can be estimated in some way (Singer and Coifman, 2008). Recently, Rodolà et al. (2019) used product manifold eigenfunctions for representing maps between shapes.

In the neural-network literature, learning *disentangled representations* has drawn interest in recent years (Reed et al., 2014; Locatello et al., 2019; Tran et al., 2017; Siddharth et al., 2017; Kim and Mnih, 2018; Sorrenson et al., 2020; Khemakhem et al., 2020). Most of the disentanglement literature treats broader classes of data-generating processes without a formal notion of what constitutes a disentangled representation. One notable exception is a paper by Fumero et al. (2021) that assumes an explicit product manifold. However, their method relies on the availability of pairs of inputs for which one of the latent factors is fixed.

2 Theoretical background

In this section, we briefly review basic properties of the Laplacian operator over product manifolds and then make the connection to manifold learning methods based on graph Laplacians.

2.1 The Laplace-Beltrami operator

We now recall some results from the spectral theory of the Laplace-Beltrami operator, which is the natural generalization of the classic Euclidean Laplacian operator to the manifold setting. For a proper introduction, see the book by Grigor'yan (2009) or the lecture notes by Canzani (2013). Let $\mathcal{M}$ be a compact Riemannian manifold with associated gradient $\nabla_{\mathcal{M}}$ and Laplace-Beltrami operator (or simply, Laplacian) $\Delta_{\mathcal{M}}$. Laplacian eigenfunctions are scalar functions on $\mathcal{M}$ that satisfy the Helmholtz equation,

$$\Delta_{\mathcal{M}} f(\mathbf{x}) = -\lambda f(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{M}. \tag{3}$$

In the cases where $\mathcal{M}$ is a manifold with boundary, we will specify Neumann boundary conditions,

$$\nabla_{\mathcal{M}} f(\mathbf{x}) \cdot \nu(\mathbf{x}) = 0, \quad \forall \mathbf{x} \in \partial\mathcal{M} \tag{4}$$

where ν is a vector normal to the boundary. Neumann boundary conditions arise naturally in Laplacian-based manifold learning (Belkin et al., 2012). The following summarizes some known results from the spectral theory of Riemannian manifolds.

Theorem 1. *Let $\mathcal{M}$ be a compact Riemannian manifold, either without boundary or with Neumann boundary conditions. The solutions to the Helmholtz equation (3) satisfy the following:*

(i) The eigenvalues are real, non-negative, and tend to infinity, $0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \cdots \to \infty$.

(ii) There exists a complete orthonormal basis for $L^2(\mathcal{M})$ of real eigenfunctions $\{f_k\}_{k=1}^{\infty}$.

2.2 The Laplacian over product manifolds

We now describe the behavior of the Laplacian operator on a product space. We begin with an example.

Example 1 (2D rectangle)**.** *Consider the rectangular domain $[0, a] \times [0, b] \subset \mathbb{R}^2$. Its eigenfunctions, with Neumann boundary conditions, are*

$$f_{m,n}(x, y) = \cos\left(\frac{m\pi}{a}x\right) \cos\left(\frac{n\pi}{b}y\right), \tag{5}$$

$$m, n \in \{0, 1, 2, \ldots\}. \tag{6}$$

The corresponding eigenvalues are

$$\lambda_{m,n} = \lambda_m + \lambda_n = \pi^2 \left(\frac{m^2}{a^2} + \frac{n^2}{b^2}\right). \tag{7}$$

The eigenfunctions are precisely the products of eigenfunctions on the closed intervals $[0, a]$ and $[0, b]$, which are $\cos(\frac{m\pi}{a}x)$ and $\cos(\frac{n\pi}{b}y)$ and the eigenvalues of the product space are sums of the corresponding eigenvalues on the intervals. See Figure 1.

This example generalizes to any product manifold.

Theorem 2. *Let $\mathcal{M}_1, \mathcal{M}_2$ be compact Riemannian manifolds where the eigenfunctions of $\mathcal{M}_i$ are $\{f_k^{(i)}\}_{k=1}^{\infty}$ and the corresponding eigenvalues are $\{\lambda_k^{(i)}\}_{k=1}^{\infty}$. Let $\pi^{(i)} : \mathcal{M} \to \mathcal{M}_i$ denote the canonical projection of $\mathcal{M}$ onto $\mathcal{M}_i$. Then the eigenfunctions of $\mathcal{M} = \mathcal{M}_1 \times \mathcal{M}_2$ are given by the pointwise products*

$$f_{m,n} = (f_m^{(1)} \circ \pi^{(1)})(f_n^{(2)} \circ \pi^{(2)}). \tag{8}$$

The corresponding eigenvalues are the sums,

$$\lambda_{m,n} = \lambda_m^{(1)} + \lambda_n^{(2)}. \tag{9}$$

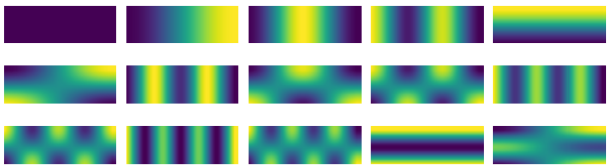

Figure 1: The first 15 Neumann eigenvectors of a rectangular domain, in order of increasing eigenvalue from top to bottom, left to right.

For a proof, see (Canzani, 2013, Section 4.6).

By repeated factorization, Theorem 2 extends to products of more than two manifolds. The eigenfunctions of $\mathcal{M}_1 \times \ldots \times \mathcal{M}_m$ are

$$f_{k_1,\ldots,k_m} = \prod_{i=1}^{m} \left(f_{k_i}^{(i)} \circ \pi^{(i)} \right), \tag{10}$$

where k_i is an index to the eigenfunctions of $\mathcal{M}_i$. The corresponding eigenvalues are

$$\lambda_{k_1,\ldots,k_m} = \sum_{i=1}^{m} \lambda_{k_i}^{(i)}. \tag{11}$$

2.3 Graph Laplacians and diffusion maps

The results of the previous sections hold for the Laplacian operator on a manifold. However, in typical data analysis settings we know neither the geometry of the manifold nor its Laplacian, but only have a sample of observations $\mathbf{x}_1, \ldots \mathbf{x}_n \in \mathcal{M} \subset \mathbb{R}^D$. To estimate the eigenfunctions, we turn to the discrete graph Laplacian. Consider the sample to be the vertex set of a weighted graph $V = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and let $W \in \mathbb{R}^{n \times n}$ be the matrix of edge weights. Gaussian affinities are often used for the weights,

$$W_{ij} = \exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/(2\sigma^2)\right). \tag{12}$$

Let D be the diagonal degree matrix that satisfies $D_{ii} = \sum_j W_{ij}$. The (unweighted) graph Laplacian is

$$L := D - W. \tag{13}$$

Several variants of the graph Laplacian were proposed, including the symmetric-normalized graph Laplacian and the random-walk Laplacian

$$L_{sym} := D^{-1/2} L D^{-1/2} \tag{14}$$

$$L_{rw} := D^{-1} L = I - D^{-1} W. \tag{15}$$

Both L_{sym} and L_{rw} are positive semi-definite and have the same spectrum (Von Luxburg, 2007),

$$0 = \nu_1 \leq \nu_2 \leq \ldots \leq \nu_n. \tag{16}$$

Defining $A = D^{-1/2} W D^{-1/2}$, the limit $A_\sigma^{t/\sigma}$ as $\sigma \to 0$ is the heat operator $H_t = e^{-t\Delta_\mathcal{M}}$ for all $t \in \mathbb{R}$. Thus, the eigenvalues λ_i of the continuous Laplace operator $\Delta_\mathcal{M}$ and the eigenvalues ν_i of L_{rw} and L_{sym} satisfy

$$\lim_{\sigma \to 0} \nu_i^{t/\sigma} = e^{-t\lambda_i}.$$

So to compute the spectrum of $\Delta_\mathcal{M}$, we simply log-transform $\nu_2, \ldots \nu_n$ (Lafon, 2004, Section 2.3).

Denote the eigenvectors (which can be chosen to be real) that correspond to $\nu_1, \ldots, \nu_n$ by $\phi_1, \ldots, \phi_n \in \mathbb{R}^n$. We view them as real functions on the set of input points $\phi_i : \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \to \mathbb{R}$. Viewed in this way, Laplacian eigenvectors define an orthogonal Fourier-like basis of oscillating functions (Lee and Izbicki, 2016). Laplacian eigenvectors can be used for dimensionality reduction by the following map,

$$\mathbf{x}_i \longmapsto [\phi_2(\mathbf{x}_i), \phi_3(\mathbf{x}_i), \ldots, \phi_\ell(\mathbf{x}_i)]. \tag{17}$$

This approach was pioneered by Belkin and Niyogi (2003) under the name *Laplacian eigenmaps* and later analyzed and extended by Coifman and Lafon (2006) under the *diffusion maps* framework.

2.4 Asymptotics of graph Laplacians

A graph Laplacian constructed from an i.i.d. sample of points on a manifold converges to a linear differential operator. For affinities that use the Euclidean distance in combination with a decay kernel (as in Eq. (12)) and when the points are drawn uniformly, the limiting operator is the Laplace-Beltrami operator $\Delta_\mathcal{M}$ (Hein et al., 2005; Giné and Koltchinskii, 2006; Singer, 2006b; Belkin and Niyogi, 2008). In the more general case, when the data has a non-uniform density $p(\mathbf{x})$ then the graph Laplacian converges to a Fokker-Planck operator that has an additional drift term given by the log-density $U(\mathbf{x}) = -2 \log p(\mathbf{x})$ (Nadler et al., 2005; Coifman and Lafon, 2006),

$$\Delta_\mathcal{M} f - \nabla U \cdot \nabla f. \tag{18}$$

In addition to the pointwise convergence, several works have proved the spectral convergence of graph Laplacians. i.e. the convergence of their eigenvalues and eigenvectors (von Luxburg et al., 2008; Rosasco et al., 2010; García Trillos and Slepčev, 2018; García Trillos et al., 2020). The convergence of the eigenvectors is the key to our approach.

Alternative graph constructions, such as the Euclidean k-nearest-neighbor graph also converge to a Fokker-Planck operator (Ting et al., 2010). If a non-Euclidean norm is used to define affinities then the graph Laplacian does not converge to a Fokker-Planck operator, but to some non-intrinsic second-order differential operator (Kileel et al., 2020).

3 Method

As discussed in Section 2.2, eigenfunctions φ_k on the product space $\mathcal{M}_1 \times \mathcal{M}_2$ are pointwise products of an eigenfunction φ_i on $\mathcal{M}_1$ and an eigenfunction φ_j on $\mathcal{M}_2$. Combining this fact with the convergence of graph Laplacians (discussed in Section 2.4), we conclude the following: if we construct a graph Laplacian from a large sample of points on $\mathcal{M}$, then each Laplacian eigenvector φ_k should be approximately equal (up to normalization) to an element-wise product $\varphi_i\varphi_j$ where φ_i and φ_j are vectors that approximate eigenfunctions on $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively. Recall that the multiplicity of the eigenvalue $\lambda = 0$ is the number of connected components. It follows that for manifolds with a single connected component, the only eigenfunctions with $\lambda = 0$ are the constant functions. If either φ_i or φ_j correspond to a constant eigenfunction with eigenvalue $\lambda = 0$ we call φ_k a *factor* eigenvector. Otherwise, we call $\varphi_k \approx \varphi_i\varphi_j$ a *product* eigenvector.

The main idea behind our method is to first differentiate between the factor eigenvectors and the product eigenvectors and then to divide the factor eigenvectors into two sets that correspond to the eigenspaces of $\mathcal{M}_1$ and $\mathcal{M}_2$. The division scheme is based on the observation that if $\varphi_k \approx \varphi_i\varphi_j$ then this suggests that φ_i and φ_j belong to different factor manifolds. Mathematically, our method is based on two assumptions:

Assumption 1. *The mapping of the latent space to the observations* $(\theta_1, \ldots, \theta_m) \mapsto \mathbf{x} \in \mathbb{R}^D$ *is an isometric embedding into* $\mathbb{R}^D$.

Remark 1. As isometric embeddings preserve the Riemannian metric, the spectral properties are not affected by the particular embedding.

Assumption 2. *The latent variables* $\theta_1, \ldots, \theta_m$ *are drawn independently of each other.*

Remark 2. The last assumption may be relaxed by using the diffusion map normalization for the graph Laplacian (Coifman and Lafon, 2006). This graph Laplacian converges to the Laplace-Beltrami operator on the manifold regardless of the sampling density.

3.1 Factorizing product eigenvectors

Suppose we have the first N non-trivial eigenvectors $\varphi_1, \varphi_2, \ldots, \varphi_N$ of L_{rw}, sorted by the corresponding log-transformed eigenvalues $0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$ which approximate the spectrum of Δ. First, for each eigenvector φ_k, we find the pair (φ_i, φ_j) whose element-wise product is closest. This "closeness" is measured by the absolute cosine similarity,

$$\mathrm{S}(\varphi_k, \varphi_i\varphi_j) := \frac{\left|\langle \varphi_k, \varphi_i\varphi_j \rangle\right|}{\|\varphi_k\|\|\varphi_i\varphi_j\|} \in [0, 1]. \tag{19}$$

We take the absolute value of the dot product to account for the sign ambiguity of eigenvectors. For each k, we find the combination (i, j) with the maximal similarity. To avoid computing the element-wise product for all $k(k-1)/2$ combinations, we skip triplets (i, j, k) for which $|\lambda_i + \lambda_j - \lambda_k| > \delta$ for some threshold $\delta > 0$. This *eigenvalue criterion* is based on the results of Theorem 2. We additionally apply an *eigenvector criterion* that filters out triplets whose similarity is less than γ. This is summarized in Algorithm 1.

Algorithm 1: Identification of individual factors

Data: Non-trivial eigenvectors $\{\varphi_1, \ldots, \varphi_N\}$ of L_{rw}, sorted increasingly by their log-transformed eigenvalues $\lambda_1, \ldots, \lambda_N$.
Result: List of triplets (i, j, k) where $\varphi_k \approx \varphi_i\varphi_j$ and the corresponding similarity scores.

```
for k ← 1 ... N do
    maxS ← 0;
    for i, j < k do
        if |λ_i + λ_j − λ_k| < δ and S(φ_k, φ_i φ_j) > maxS then
            maxS ← S(φ_k, φ_i φ_j);
            i_max ← i;
            j_max ← j;
        end
    end
    if maxS > γ then
        add (i_max, j_max, k) to triplets;
    end
end
```

The result of the algorithm is a list of index triplets (i, j, k), which indicate that the best factorization we could find for φ_k is $\varphi_i\varphi_j$. Next, we use the triplets to divide the factor eigenvectors into two sets that correspond to the manifolds $\mathcal{M}_1$ and $\mathcal{M}_2$.

3.2 Assigning the factor eigenvectors to factor manifolds

Let $\mathcal{T}$ be the set of triplets returned by Algorithm 1. We will use $\mathcal{T}$ to split the candidate factor eigenvectors into two separate subsets via an approximate MaxCut algorithm. To do so, we define a weighted graph $H = (V, E, W)$ where V is the set of unique factor eigenvectors in our list of triplets produced by Algorithm 1. The undirected edge set is defined as

$$E = \{(i, j) \mid \exists k \text{ s.t. } (i, j, k) \in \mathcal{T}\} \tag{20}$$

with edge weights given by the highest similarity score,

$$W_{ij} = \max_{(i,j,k)\in\mathcal{T}} \mathrm{S}(\varphi_k, \varphi_i\varphi_j). \tag{21}$$

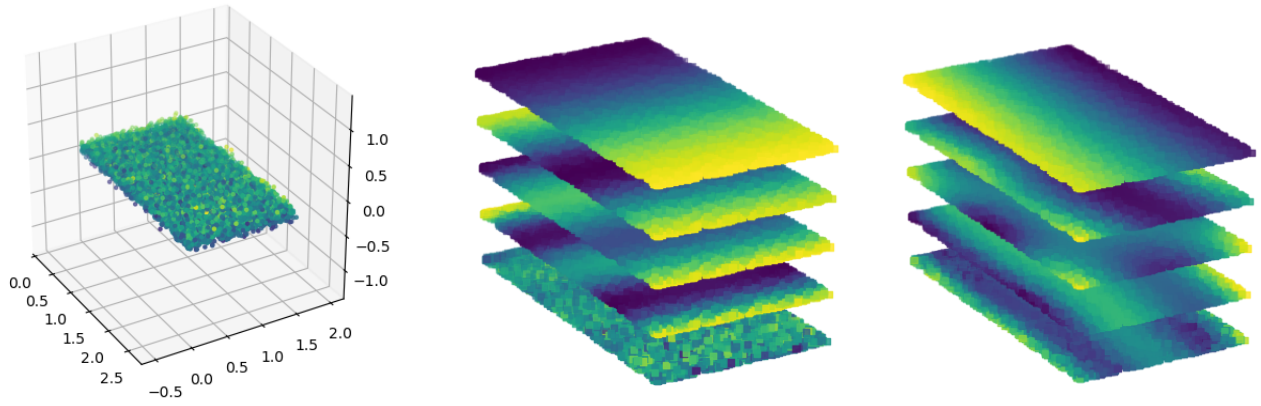

Figure 2: **Rectangle with noise.** The algorithm was run on 10,000 samples with parameters $\delta = 0.5$, $\gamma = 0.85$. The eigenvectors are plotted on the ground truth x and y coordinate plane. **Left.** The original data, with noise in the z-direction indicated by color. **Middle.** First five factor eigenvectors associated with the y axis as determined by the algorithm. **Right.** First five factor eigenvectors associated with the x axis.

The idea here is that a high score suggests that φ_i and φ_j are the true factors of φ_k and thus must belong to different factor manifolds. Using a MAX-CUT SDP solver with H as the input, we sort our eigenvectors into two bins that correspond to $\mathcal{M}_1$ and $\mathcal{M}_2$.

3.3 Implementation details and runtime

The algorithm was implemented in Python, and the MAX-CUT SDP was implemented using CVXPY (Diamond and Boyd, 2016). The first part of the algorithm computes the diffusion map of the data. The eigenvectors are computed using the Scikit-learn randomized SVD implementation (Pedregosa et al., 2011; Martinsson et al., 2011). In the subsequent step, we search through $\binom{N}{3}$ triplets. Taking into account the element-wise vector multiplication at each step, this takes $O(nN^3)$ time. In practice, however, the eigenvalue criterion quickly discards of the vast majority of triplets. The last part of the algorithm iterates through the list of triplets to form an undirected weighted graph and then runs a MAX-CUT SDP solver.

4 Simulations

We tested our algorithm on three different datasets: noisy samples from a 2D rectangle in 3D space, a torus, and a synthetic set of cryo-electron microscopy images.

4.1 2D noisy rectangle

First, we ran our algorithm on points sampled from an axis-aligned 2D rectangle $[0, \sqrt{\pi}+1] \times [0, 1.5] \in \mathbb{R}^2$ embedded in $\mathbb{R}^3$. The data was generated by taking 10,000 random points and adding noise in the z-direction. We used the parameters $\gamma = 0.85$, $\delta = 0.5$. The results, shown in Figure 2, show cosines along the x and y directions, as expected (see Example 1).

Table 1: Running times for each step of the algorithm on the 2D noisy rectangle dataset. The number of eigenvectors varied over $N \in \{50, 100, 200, 400\}$. The diffusion map step includes computing the affinity matrix and the Laplacian eigenvectors. The third column refers to Algorithm 1 described in Section 3.1 and the fourth column refers to the MAX-CUT SDP and graph construction described in Section 3.2.

N	Diffusion map	Alg. 1	Sorting
50	5.59 s	0.454 s	0.022 s
100	6.07 s	3.32 s	0.020 s
200	8.18 s	20.3 s	0.020 s
400	12.3 s	141 s	0.020 s

Table 1 lists running times, averaged over 5 trials, on this dataset. A single core of a 2.9 GHz 2017 Intel Core i7 processor was used for these benchmarks.

4.2 Analysis of threshold parameters

Our algorithm has two threshold parameters: δ for the eigenvalue criterion and γ for the eigenvector criterion. A question one might have is: for a given product eigenvector φ_k, how much of an outlier will the similarity of the true combination of factor eigenvectors $\varphi_i\varphi_j$ be compared to the similarities of all other pairs? To investigate this, for each product eigenvector φ_k we plot the eigenvector criterion $\mathrm{S}(\varphi_k, \varphi_i\varphi_j)$ against the eigenvalue criterion $|\lambda_i+\lambda_j-\lambda_k|$ over all $1 \le i < j < k$, shown in Figure 4. We see that in most cases the similarity score for the chosen triplet is indeed an outlier. For product eigenvectors with lower eigenvalues, the highest similarity stands out more clearly, as this part

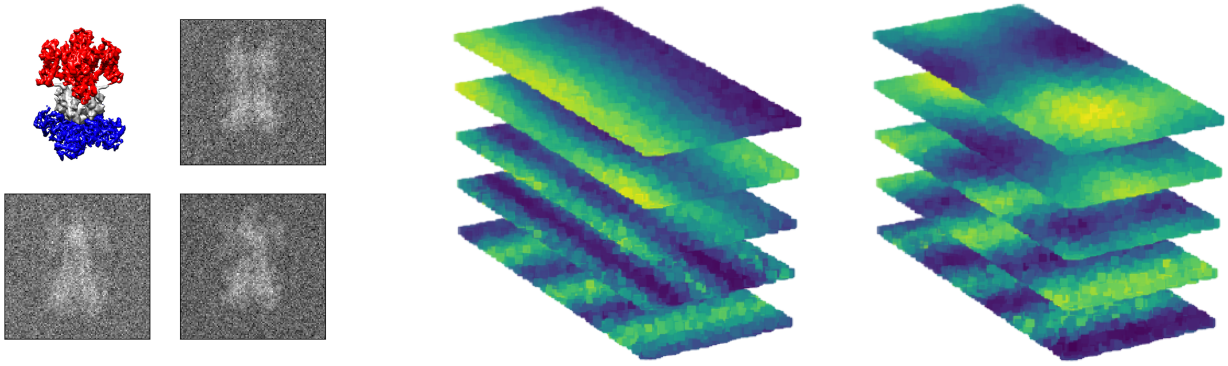

Figure 3: **Two-factor noisy cryo-EM data.** The algorithm was run on 10,000 images with the parameters $\delta = 1.0$, $\gamma = 0.80$. The eigenvectors are plotted on the ground-truth values of the rotation angles $[0, 360]$ and the x-stretch $[-20, 20]$. **Left.** A synthetic cryo-EM image of a potassium ion channel with two independently deformable subunits is shown in the top-left. The red portion rotates uniformly around the z-axis, and the blue portion stretches independently along the x-axis. Three simulated cryo-EM projection images are shown. **Middle.** Factor eigenvectors associated with x-stretch. **Right.** Factor eigenvectors associated with the rotation.

of the spectrum is more stable. As the eigenvalue of the product eigenvector increases, the best pair, which has the highest similarity score, is less well separated. However, this ambiguity tends to be between relatively few candidates. One can also see that the highest scoring triplet has a small eigenvalue criterion. This demonstrates that the eigenvalue criterion can speed up the triplet search without sacrificing performance.

A sensitivity analysis of both parameters is included in Figure 5. Each figure shows the manifold assignments obtained with 100 eigenvectors on the noisy rectangle dataset. The figure demonstrates the stability of the algorithm, since the assignments are consistent across a large range of parameter values.

4.3 Single-particle cryo-EM data

We present an application of manifold factorization for cryo-electron microscopy (cryo-EM). Cryo-EM is a technique used for imaging proteins and other macromolecules that has fostered major advances in structural biology (Kühlbrandt, 2014; Cressey and Callaway, 2017). Recently, it played a key role in unraveling the structural properties of the novel coronavirus SARS-CoV-2 (Zimmer, 2020). In cryo-EM, molecular samples are rapidly frozen in a thin layer of ice, thus capturing them in their native states (Dubochet et al., 1988). Once frozen, a transmission electron microscope is used to capture 2D tomographic projections of their electrostatic potential (Vulović et al., 2013). These 2D images are then used to reconstruct, through a series of computational steps, a 3D reconstruction of the molecule. (Zhou et al., 2020, Fig. 1).

The classical cryo-EM reconstruction problem is posed as follows: given n tomographic projection images $I_1, \ldots, I_n$ of a particular molecule at random (unknown) orientations, how can we recover the three-dimensional structure of the molecule? A core obstacle to this problem is handling the low signal-to-noise ratio present in the micrographs (Singer and Sigworth, 2020; Bendory et al., 2020). This task is made even more difficult if we consider the *heterogeneity* problem, which has been the subject of recent literature (Frank, 2018; Nakane et al., 2018; Andén and Singer, 2018; Sorzano et al., 2019; Lederman et al., 2020; Zhong et al., 2020). Here, due to natural variations, each image I_i captures a different spatial conformation of the same molecule. Several works have applied diffusion maps for the analysis of cryo-EM datasets with continuous heterogeneity (Dashti et al., 2014; Schwander et al., 2014; Moscovich et al., 2020; Zelesko et al., 2020; Kileel et al., 2020; Dashti et al., 2020).

For molecules that exhibit two or more independent continuous motions, the manifold of 3D electrostatic densities is a product space. For our experiments, we use the potassium ion channel model of Moscovich et al. (2020) with two independently deformable parts: a spinning subunit with 360° of rotation, and a subunit that stretches in the xy-plane (see Figure 3). To create the images, we sampled 10,000 conformations of the molecule with uniformly drawn angles and x-axis stretches ranging from $[-20, 20]$. All the images were projected from a single view. This models the approach of Dashti et al. (2020), which computes a diffusion map for each viewing direction separately. Finally, we use PCA with four components to preprocess the raw images and transform them so that they are zero-centered with unit variance.

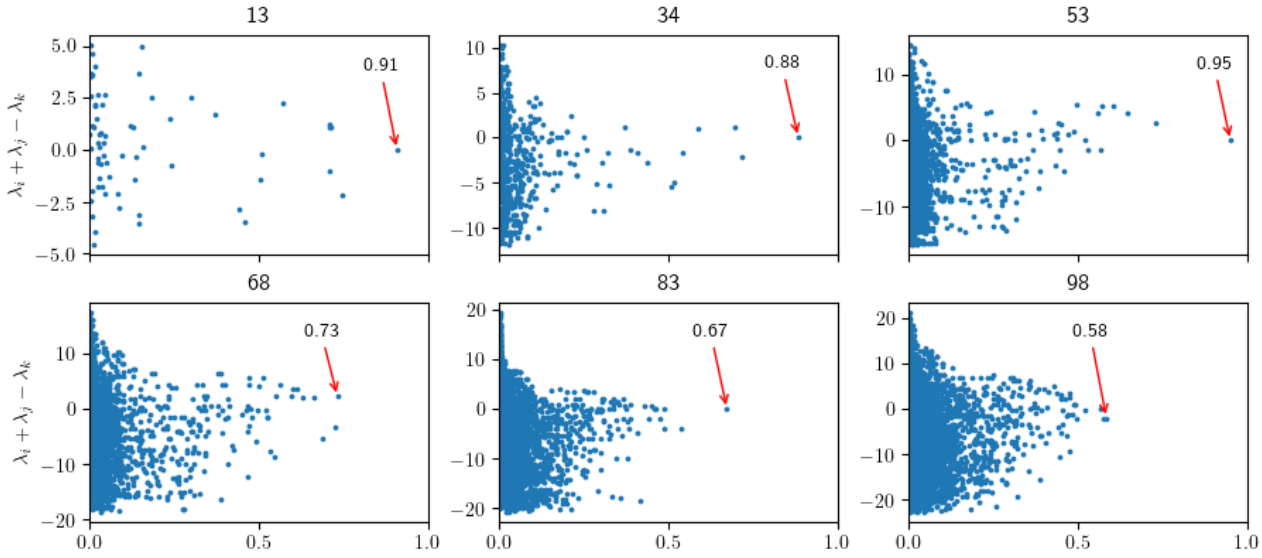

Figure 4: **Analysis of eigenvalue and eigenvector criteria for the rectangle dataset.** We picked six product eigenvectors φ_k at random. For each one, we plotted the eigenvalue criteria error (y axis) and similarity scores (x axis) for all triplets $(\varphi_i, \varphi_j, \varphi_k)$, $1 \le i < j < k$. Numbers at the top of the scatter plots are the index k. The point that corresponds to the eigenvector with maximum similarity is marked by a red arrow.

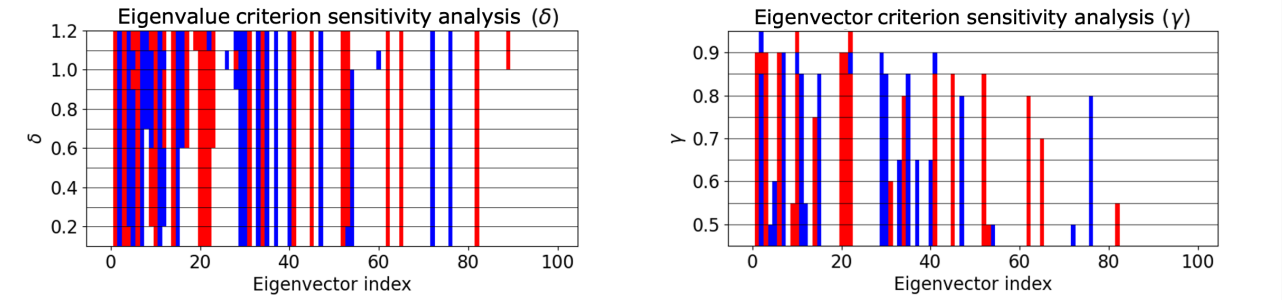

Figure 5: **Sensitivity analysis for δ and γ.** Each column corresponds to a different eigenvector and the colors denote assignment of the eigenvector to either $\mathcal{M}_1$ (red), $\mathcal{M}_2$ (blue) or to $\mathcal{M}_1 \times \mathcal{M}_2$ (white). Hence each row shows all the assignments our algorithm makes for a particular choice of the parameters. The base values of each parameter are $\gamma = 0.75$ (when varying δ) and $\delta = 0.4$ (when varying γ).

We ran three different simulations: first, a dataset of molecules with rotations and stretches in the x direction only and low noise; then, a dataset of molecules with rotations and stretches in the x-direction only and high noise; and finally, a dataset of molecules with rotations and stretches in both the x and y directions and low noise. The parameters $\gamma = 0.80$ and $\delta = 1.0$ were used in all simulations. The results for the second simulation are shown in Figure 3. The eigenvectors corresponding to the x-stretch manifold exhibit the ground truth cosine waves, and the eigenvectors corresponding to the rotation manifold (a circle) exhibit the ground truth sinusoidal waves.

Note that even though the latent space is a product space, Assumption 1 does not strictly hold due to the additive Gaussian noise. Nonetheless, our method works nicely on this dataset.

4.4 Visualizing manifold factors

We can visualize the manifold factors by plotting their diffusion map embeddings. Recall from Equation (17) that given data samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and Laplacian eigenvectors $\varphi_1, \varphi_2, \ldots, \varphi_N$, the embedding map is

$$\mathbf{x}_i \mapsto [\varphi_2(\mathbf{x}_i), \varphi_3(\mathbf{x}_i), \ldots, \varphi_\ell(\mathbf{x}_i)].$$

For our visualizations, we use $\ell = 3$ and plot $\varphi_2(\mathbf{x}_i)$ against $\varphi_3(\mathbf{x}_i)$. In the cases where our algorithm is applied first, we can visualize the diffusion map embedding of each manifold factor. We need not perform

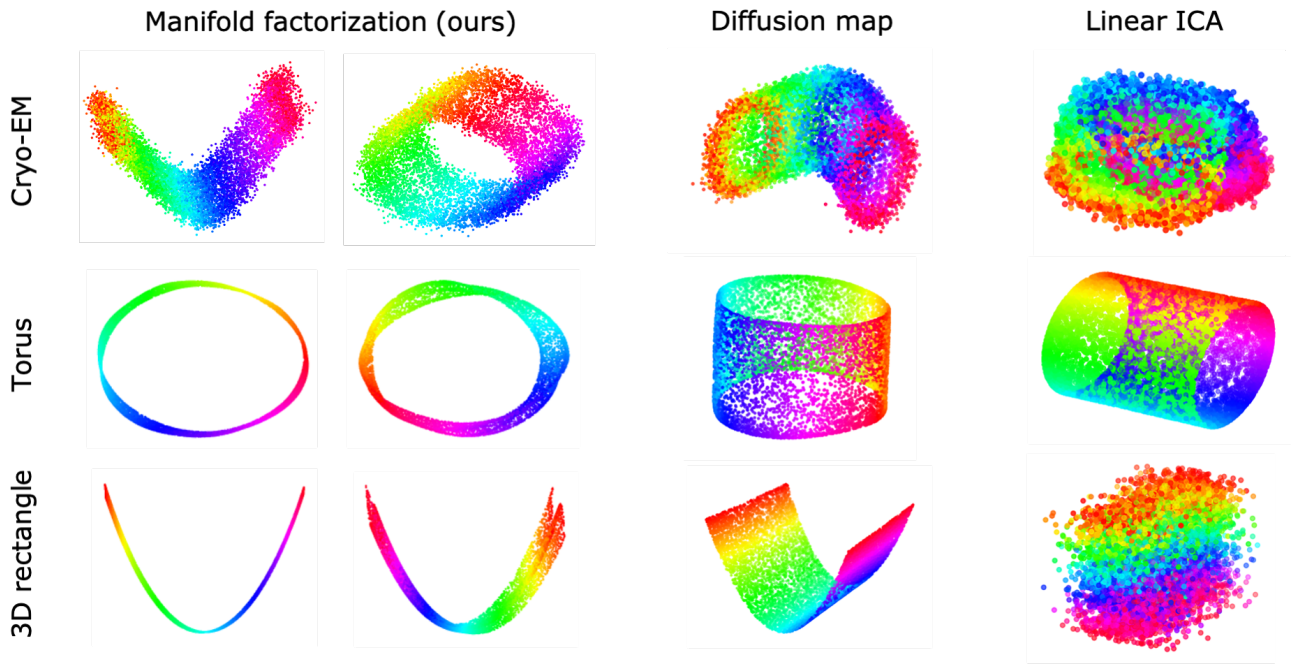

Figure 6: **Visualizing manifold factorizations.** We visualize three different datasets. The left two columns show the diffusion map embeddings of each manifold factor as determined by our algorithm. The middle column shows a standard diffusion map of the points from the product manifold and the rightmost column shows a three-dimensional linear ICA. The colors represent the ground truth data in a single dimension of the parameter space. **Top row.** Cryo-EM images (see Section 4.3). **Middle row.** 10,000 points uniformly drawn from a torus $S^1 \times S^1$ embedded in $\mathbb{R}^4$ where the factor circle radiuses are 2 and $\sqrt{\pi}+1$. **Bottom row.** 10,000 points drawn from a 2D rectangular domain with noise in the z-direction (see Section 4.1).

an additional diffusion map computation since our algorithm has already produced the eigenvectors for each manifold factor $\mathcal{M}_i$. Colors indicate the ground truth coordinate of the point $\mathbf{x}_i$ on the respective manifold factor. The left two columns of Figure 6 show the diffusion map embeddings of manifold factors computed using our algorithm for three different experiments. In all three cases, the embeddings are in agreement with the factors of the data manifold.

While we are not aware of other dimensionality reduction methods that aim to accomplish the same goal as our method, the two most immediately comparable methods are standard diffusion maps, which our algorithm builds upon, and linear independent component analysis techniques (Linear ICA). Though these methods do not directly apply to the specific task of product manifold factorization, we can use them to visualize the entire product manifold (also shown in Figure 6).

Our algorithm can be used to accomplish several objectives. First, we can use representative factor eigenvectors from each manifold factor's eigenspace to compress the data representation. This is a complementary approach to other methods for parsimonious representations of manifolds (Blau and Michaeli, 2017; Dsilva et al., 2018; Meila et al., 2018; Chen and Meilă, 2019). Additionally, by separating the factor eigenvectors according to their respective manifolds, we also disentangle the diffusion map coordinates into sets that correspond to independent latent factors.

5 Hierarchical factorization

Extending the algorithm to $m > 2$ manifolds is possible by using a Max k-Cut SDP (Newman, 2018) in place of the Max-Cut solver and adjusting the triplet search to an $m+1$-tuple search. This approach requires additional evaluation and we leave it for future work. Another possibility is to run the two-factor algorithm in a hierarchical fashion, i.e. apply it again on the product eigenvectors assigned to one of the two manifolds. As an example, we ran the two-factor algorithm twice on 10,000 points drawn uniformly from the box $[0, 1.0+\sqrt{\pi}] \times [0, 1.5] \times [0, 7.0]$. The results of the first iteration are shown in bins #1 and #2 of Figure 7. For the second iteration, we ran our algorithm on the eigenvectors of bin #2, and the resulting factor eigenvectors were sorted into bins #3 and #4. Together, the eigenvectors in bins #1, #3 and #4 form our $m = 3$ factorization.

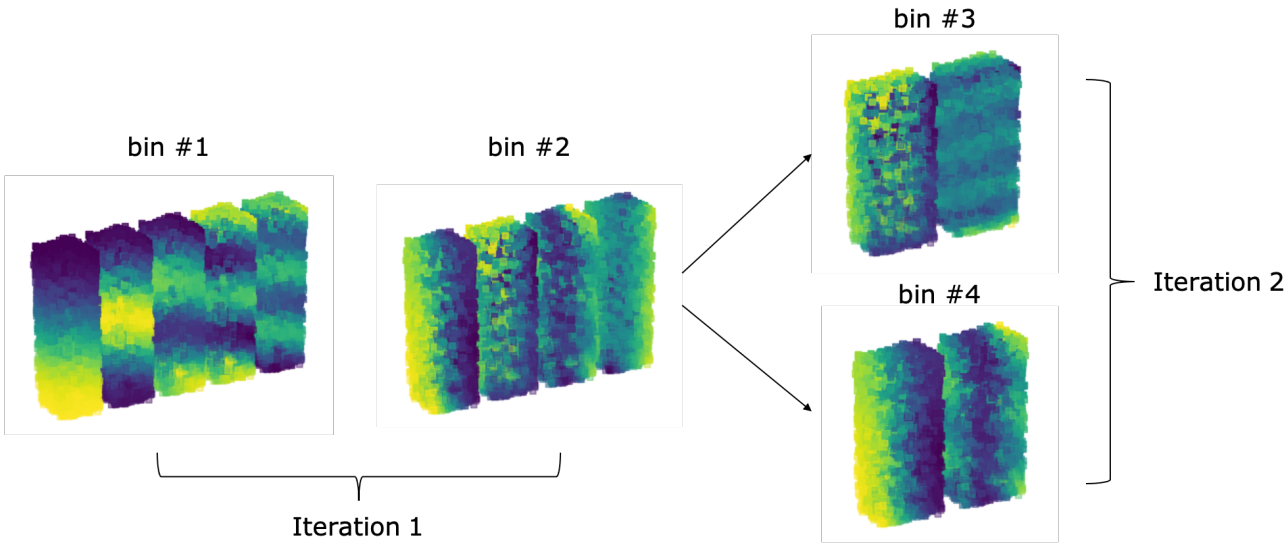

Figure 7: **Hierarchical factorization.** Manifold factor eigenvector bins obtained by running our algorithm hierarchically on points from a 3D box (see Section 5). Bin #1 and bin #2 are obtained from the first iteration. Rerunning the factorization on the eigenvectors in bin #2 resulted in bins #3 and #4.

6 Conclusion

In this paper, we presented a spectral method for nonlinear dimensionality reduction and data representation. Our method is based on manifold factorization and applies to data sets with two or more independent degrees of freedom. We tested our algorithm on several synthetic datasets, including simulated transmission electron-microscope images of a protein with two independently moving subunits.

One direction for future work is the extension of the method to products of more than two manifolds (see Section 5). Another interesting direction is to test our approach on broader classes of spaces, other than product spaces, but for which the Laplacian operator separates nonetheless.

Reproducibility

Code for reproducing the figures in this paper can be found at: https://github.com/sxzhang25/product-manifold-learning

Acknowledgements

This work was supported in part by AFOSR Awards FA9550-17-1-0291 and FA9550-20-1-0266, ARO W911NF-17-1-0512, the Simons Foundation Math+X Investigator Award, NSF BIGDATA Award IIS1837992, NSF Award DMS-2009753, and NIH/NIGMS Award R01GM136780-01. We thank Nicolas Boumal and the anonymous referees for their helpful suggestions.

References

Joakim Andén and Amit Singer. Structural Variability from Noisy Tomographic Projections. *SIAM Journal on Imaging Sciences*, 11(2):1441–1492, 2018. doi:10.1137/17M1153509.

Jonathan Bates. The embedding dimension of Laplacian eigenfunction maps. *Applied and Computational Harmonic Analysis*, 37(3):516–530, 2014. doi:10.1016/j.acha.2014.03.002.

Mikhail Belkin and Partha Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6):1373–1396, 2003. doi:10.1162/089976603321780317.

Mikhail Belkin and Partha Niyogi. Towards a theoretical foundation for Laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74 (8):1289–1308, 2008. doi:10.1016/j.jcss.2007.08.006.

Mikhail Belkin, Qichao Que, Yusu Wang and Xueyuan Zhou. Graph Laplacians on Singular Manifolds: Toward understanding complex spaces: graph Laplacians on manifolds with singularities and boundaries. In *Conference on Learning Theory (COLT)*, volume 23, pages 1–26, 2012.

Tamir Bendory, Alberto Bartesaghi and Amit Singer. Single-Particle Cryo-Electron Microscopy: Mathematical Theory, Computational Challenges, and Opportunities. *IEEE Signal Processing Magazine*, 37(2):58–76, 2020. doi:10.1109/MSP.2019.2957822.

Yochai Blau and Tomer Michaeli. Non-redundant Spectral Dimensionality Reduction. In *Joint Euro-*

pean Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), 2017. doi:10.1007/978-3-319-71249-9_16.

Yaiza Canzani. Analysis on Manifolds via the Laplacian (lecture notes), 2013.

Yu Chia Chen and Marina Meilă. Selecting the independent coordinates of manifolds with large aspect ratios. In *Neural Information Processing Systems (NeurIPS)*, 2019.

Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006. doi:10.1016/j.acha.2006.04.006.

Daniel Cressey and Ewen Callaway. Cryo-electron microscopy wins chemistry Nobel. *Nature*, 550(7675): 167–167, 2017. doi:10.1038/nature.2017.22738.

Ali Dashti et al. Trajectories of the ribosome as a Brownian nanomachine. *Proceedings of the National Academy of Sciences*, 111(49):17492–17497, 2014. doi:10.1073/pnas.1419276111.

Ali Dashti et al. Retrieving functional pathways of biomolecules from single-particle snapshots. *Nature Communications*, 11(1):4734, 2020. doi:10.1038/s41467-020-18403-x.

Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17:1–5, 2016.

Carmeline J. Dsilva, Ronen Talmon, Ronald R. Coifman and Ioannis G. Kevrekidis. Parsimonious representation of nonlinear dynamical systems through manifold learning: A chemotaxis case study. *Applied and Computational Harmonic Analysis*, 44(3): 759–773, 2018. doi:10.1016/j.acha.2015.06.008.

Jacques Dubochet et al. Cryo-electron microscopy of vitrified specimens. *Quarterly Reviews of Biophysics*, 21(2):129–228, 1988. doi:10.1017/S0033583500004297.

Joachim Frank. New Opportunities Created by Single-Particle Cryo-EM: The Mapping of Conformational Space. *Biochemistry*, 57(6):888–888, 2018. doi:10.1021/acs.biochem.8b00064.

Marco Fumero, Luca Cosmo, Simone Melzi and Emanuele Rodolà. Learning disentangled representations via product manifold projection. Technical report, 2021. arXiv:2103.01638

Nicolás García Trillos and Dejan Slepčev. A variational approach to the consistency of spectral clustering. *Applied and Computational Harmonic Analysis*, 45(2):239–281, 2018. doi:10.1016/j.acha.2016.09.003.

Nicolás García Trillos, Moritz Gerlach, Matthias Hein and Dejan Slepčev. Error Estimates for Spectral Convergence of the Graph Laplacian on Random Geometric Graphs Toward the Laplace–Beltrami Operator. *Foundations of Computational Mathematics*, 20(4):827–887, 2020. doi:10.1007/s10208-019-09436-w.

Evarist Giné and Vladimir Koltchinskii. Empirical graph Laplacian approximation of Laplace-Beltrami operators: Large sample results. In *High Dimensional Probability*, volume 51, pages 238–259. 2006. doi:10.1214/074921706000000888.

Alexander Grigor'yan. *Heat Kernel and Analysis on Manifolds*. American Mathematical Society, 2009. ISBN 978-0-8218-9393-7.

Matthias Hein, Jean-Yves Audibert and Ulrike von Luxburg. From Graphs to Manifolds – Weak and Strong Pointwise Consistency of Graph Laplacians. In *International Conference on Computational Learning Theory (COLT)*, pages 470–485, 2005. doi:10.1007/11503415_32.

Ilyes Khemakhem, Diederik P. Kingma and Aapo Hyvärinen. Variational autoencoders and nonlinear ICA: A unifying framework. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Joe Kileel, Amit Moscovich, Nathan Zelesko and Amit Singer. Manifold learning with arbitrary norms. Technical report, 2020. arXiv:2012.14172

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning (ICML)*, 2018.

W. Kühlbrandt. The Resolution Revolution. *Science*, 343(6178):1443–1444, 2014. doi:10.1126/science.1251652.

Stephane S. Lafon. *Diffusion Maps and Geometric Harmonics*. PhD thesis, Yale University, 2004.

Roy R. Lederman, Joakim Andén and Amit Singer. Hyper-molecules: on the representation and recovery of dynamical structures for applications in flexible macro-molecules in cryo-EM. *Inverse Problems*, 36(4):044005, 2020. doi:10.1088/1361-6420/ab5ede.

Ann B. Lee and Rafael Izbicki. A spectral series approach to high-dimensional nonparametric regression. *Electronic Journal of Statistics*, 10(1):423–463, 2016. doi:10.1214/16-EJS1112.

John M. Lee. *Introduction to Smooth Manifolds*, volume 218 of *Graduate Texts in Mathematics*. Springer New York, 2012. ISBN 978-1-4419-9981-8. doi:10.1007/978-1-4419-9982-5.

Francesco Locatello et al. Challenging Common Assumptions in the Unsupervised Learning of Disen-

tangled Representations. In *International Conference on Machine Learning (ICML)*, 2019.

Per Gunnar Martinsson, Vladimir Rokhlin and Mark Tygert. A randomized algorithm for the decomposition of matrices. *Applied and Computational Harmonic Analysis*, 30(1):47–68, 2011. doi:10.1016/j.acha.2010.02.003.

Marina Meila, Samson Koelle and Hanyu Zhang. A regression approach for explaining manifold embedding coordinates. Technical report, 2018. arXiv:1811.11891

Amit Moscovich, Amit Halevi, Joakim Andén and Amit Singer. Cryo-EM reconstruction of continuous heterogeneity by Laplacian spectral volumes. *Inverse Problems*, 36(2):024003, 2020. doi:10.1088/1361-6420/ab4f55.

Boaz Nadler, Stephane Lafon, Ronald R. Coifman and Ioannis G. Kevrekidis. Diffusion Maps, Spectral Clustering and Eigenfunctions of Fokker-Planck operators. In *Advances in Neural Information Processing Systems (NIPS)*, 2005.

Takanori Nakane, Dari Kimanius, Erik Lindahl and Sjors HW Scheres. Characterisation of molecular motions in cryo-EM single-particle data by multi-body refinement in RELION. *eLife*, 7:1–18, 2018. doi:10.7554/eLife.36861.

Alantha Newman. Complex semidefinite programming and max-k-cut. In *Symposium on Simplicity in Algorithms (SOSA)*, volume 61, pages 13:1–13:11, 2018. doi:10.4230/OASIcs.SOSA.2018.13.

Fabian Pedregosa et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Scott Reed, Kihyuk Sohn, Yuting Zhang and Honglak Lee. Learning to disentangle factors of variation with manifold interaction. In *International Conference on Machine Learning (ICML)*, 2014.

Emanuele Rodolà, Zorah Lähner, Alex M. Bronstein, Michael M. Bronstein and Justin Solomon. Functional Maps Representation On Product Manifolds. *Computer Graphics Forum*, 38(1):678–689, 2019. doi:10.1111/cgf.13598.

Lorenzo Rosasco, Mikhail Belkin and Ernesto De Vito. On Learning with Integral Operators. *Journal of Machine Learning Research*, 11(30):905–934, 2010.

Peter Schwander, Russell Fung and Abbas Ourmazd. Conformations of macromolecules and their complexes from heterogeneous datasets. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1647):1–8, 2014. doi:10.1098/rstb.2013.0567.

N Siddharth et al. Learning Disentangled Representations with Semi-Supervised Deep Generative Models. In *Conference on Neural Information Processing Systems (NIPS)*, 2017.

A. Singer. From graph to manifold Laplacian: The convergence rate. *Applied and Computational Harmonic Analysis*, 21(1):128–134, 2006a. doi:10.1016/j.acha.2006.03.004.

A. Singer. Spectral independent component analysis. *Applied and Computational Harmonic Analysis*, 21 (1):135–144, 2006b. doi:10.1016/j.acha.2006.03.003.

Amit Singer and Ronald R. Coifman. Non-linear independent component analysis with diffusion maps. *Applied and Computational Harmonic Analysis*, 25 (2):226–239, 2008. doi:10.1016/j.acha.2007.11.001.

Amit Singer and Fred J. Sigworth. Computational Methods for Single-Particle Electron Cryomicroscopy. *Annual Review of Biomedical Data Science*, 3(1):163–190, 2020. doi:10.1146/annurev-biodatasci-021020-093826.

Peter Sorrenson, Carsten Rother and Ullrich Köthe. Disentanglement By Nonlinear ICA With General Incompressible-Flow Networks (Gin). In *International Conference on Learning Representations (ICLR)*, 2020.

Carlos Oscar S. Sorzano et al. Survey of the analysis of continuous conformational variability of biological macromolecules by electron microscopy. *Acta Crystallographica Section F Structural Biology Communications*, 75(1):19–32, 2019. doi:10.1107/S2053230X18015108.

Ronen Talmon, Israel Cohen, Sharon Gannot and Ronald R. Coifman. Diffusion Maps for Signal Processing: A Deeper Look at Manifold-Learning Techniques Based on Kernels and Graphs. *IEEE Signal Processing Magazine*, 30(4):75–86, 2013. doi:10.1109/MSP.2013.2250353.

Joshua B. Tenenbaum, Vin de Silva and John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000. doi:10.1126/science.290.5500.2319.

Daniel Ting, Ling Huang and Michael Jordan. An Analysis of the Convergence of Graph Laplacians. In *International Conference on Machine Learning (ICML)*, 2010.

Luan Tran, Xi Yin and Xiaoming Liu. Disentangled Representation Learning GAN for Pose-Invariant Face Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1283–1292, 2017. doi:10.1109/CVPR.2017.141.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. doi:10.1007/s11222-007-9033-z.

Ulrike von Luxburg, Mikhail Belkin and Olivier Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, 36(2):555–586, 2008. doi:10.1214/009053607000000640.

Miloš Vulović et al. Image formation modeling in cryo-electron microscopy. *Journal of Structural Biology*, 183(1):19–32, 2013. doi:10.1016/j.jsb.2013.05.008.

Nathan Zelesko, Amit Moscovich, Joe Kileel and Amit Singer. Earthmover-Based Manifold Learning for Analyzing Molecular Conformation Spaces. In *International Symposium on Biomedical Imaging (ISBI)*, 2020. doi:10.1109/ISBI45749.2020.9098723.

Ellen D. Zhong, Tristan Bepler, Joseph H. Davis and Bonnie Berger. Reconstructing continuous distributions of 3D protein structure from cryo-EM images. In *International Conference on Learning Representations (ICLR)*, 2020.

Ye Zhou, Amit Moscovich, Tamir Bendory and Alberto Bartesaghi. Unsupervised particle sorting for high-resolution single-particle cryo-EM. *Inverse Problems*, 36(4):044002, 2020. doi:10.1088/1361-6420/ab5ec8.

Carl Zimmer. The Coronavirus Unveiled, oct 2020, The New York Times. `http://nyti.ms/2GO6PDV`.