Product Manifold Learning with Independent Coordinate Selection

Jesse He [1] **Tristan Brugère** [1] **Gal Mishne** [1]

Abstract

In many dimensionality reduction tasks, we wish to identify the constituent components that explain our observations. For manifold learning, this can be formalized as factoring a Riemannian product manifold. Recovering this factorization, however, may suffer from certain difficulties in practice, especially when data is sparse or noisy, or when one factor is distorted by the other. To address these limitations, we propose identifying non-redundant coordinates on the product manifold before applying product manifold learning to identify which coordinates correspond to different factor manifolds. We demonstrate our approach on both synthetic and real-world data.

1. Introduction

Consider a high-dimensional dataset $X = \{x_1, \ldots, x_N\}$ in $\mathbb{R}^p$ for some large number of features p. The task of *dimensionality reduction* is to map X to a lower-dimensional space $\mathbb{R}^d$ where $d \ll p$. In classical methods (e.g. PCA), this has the added benefit of providing a set of coordinates which reflects the underlying structure of the data. However, in many cases the underlying data-generating process has a nonlinear structure which linear methods like PCA fail to capture. In these cases, one typically turns to *manifold learning*, a class of geometric techniques which aims to recover underlying geometric structure even in nonlinear cases (Tenenbaum et al., 2000; Belkin & Niyogi, 2003; Coifman & Lafon, 2006; Fefferman et al., 2016). However, it is not always obvious how to assign coordinates that reflect the constituent underlying parameters of the data manifold.

To do so, we follow (Zhang et al., 2020) in considering *product manifolds*, which we wish to *factorize* into their constituent latent variables. Product manifolds appear in neural data, where grid cell activation is suggested to have the structure of a torus $T = S^1 \times S^1$ (Gardner et al., 2022), and in cryo-electron microscopy where a subject molecule may contain multiple rigid bodies that move independently from each other (Nakane et al., 2018).

However, this method may suffer in practice, since spectral embedding techniques are prone to *redundant* embedding coordinates. That is, different eigenvectors may correspond to the same direction on the manifold despite being orthogonal in the original $\mathbb{R}^p$ (Chen & Meilă, 2019; Dsilva et al., 2015; Goldberg et al., 2008).

Related work has identified methods for non-redundant spectral embeddings such as (Singer, 2006b) for spectral independent component analysis; (Dsilva et al., 2015), based on local linear regression; and (Koelle et al., 2021), which provides embedding coordinates with domain-specific meaning. The work of (Kohli et al., 2021) pieces together local views of a manifold into a global embedding with "tearing", which enables representing closed manifolds in their intrinsic dimensions, from which one could also analyze the constituent features of a data manifold.

In this paper, we identify how the phenomenon of redundant embedding coordinates impacts the quality of manifold factorization given by the product manifold learning (PML) method of (Zhang et al., 2020), as well as the phenomenon of *warped products*, in which the geometry of one factor is distorted as a function of the other. We then describe a way to resolve this issue by leveraging the method of (Chen & Meilă, 2019) to limit our potential factorization coordinates to independent eigenvectors, which we will refer to as independent coordinate product manifold learning (IC-PML).

We will first discuss the theoretical background of manifold learning in Section 2. We describe our contribution in Sections 3 and 4, first identifying some of the difficulties that may arise in manifold factorization in Section 3 before describing in Section 4 how we augment the factorization algorithm with independent coordinate selection. We provide examples in Section 5 where our method, IC-PML, provides better factorizations than PML. Finally, we offer concluding remarks in Section 6.

[1]Halıcıoğlu Data Science Institute, University of California San Diego, San Diego, USA. Correspondence to: Jesse He <jeh020@ucsd.edu>.

Proceedings of the 2^{nd} Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML) at the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. 2023.

2. Background

We begin by describing spectral embedding methods in manifold learning and the relationship between the graph Laplacian and the Laplace-Beltrami operator on manifolds. We then discuss the method of (Zhang et al., 2020) for product manifold learning and the method of (Chen & Meilă, 2019) for selecting independent spectral embedding coordinates.

2.1. The Laplacian and Spectral Embeddings

Both (Zhang et al., 2020) and (Chen & Meilă, 2019) are built on spectral methods for manifold learning, as introduced by (Belkin & Niyogi, 2003) and (Coifman & Lafon, 2006). Consider a manifold M of dimension d embedded in $\mathbb{R}^p$ for some $p \gg d$, and suppose we have a set X of N data points $x_1, \dots, x_N \in \mathbb{R}^p$ sampled uniformly at random from M. We wish to recover the underlying manifold M from our dataset X by producing a new embedding $(\varphi_1, \dots, \varphi_q) : M \to \mathbb{R}^q$ where $d \leq q \ll p$.

We begin by constructing a weighted graph $G = (X, W)$ on the data points with edge weights given by the radial basis function with bandwidth ε:

$$W_{i,j} = \exp\left(\frac{-\|x_j - x_i\|^2}{\varepsilon}\right). \tag{1}$$

Remark 2.1. In some cases, G is constructed as an r-radius graph or a k-nearest neighbors graph, but we will consider the fully connected case.

Then defining the degree matrix $D_{i,i} = \sum_{j=1}^{N} W_{ij}$, we can define the (symmetric normalized) Laplacian $L_{\text{sym}} = I - D^{-1/2} W D^{-1/2}$, noting that it is positive semidefinite and hence admits an eigendecomposition over $\mathbb{R}$. Then, in the limit as $N \to \infty$ and $\varepsilon \to 0$, it can be shown that L_{sym} converges to the Laplace-Beltrami operator Δ_M on M:

Theorem 2.2 (Belkin & Niyogi, 2003; Coifman & Lafon, 2006; Singer, 2006a)**.** *Suppose $X = \{x_1, \dots, x_n\}$ is drawn i.i.d uniformly at random from a compact manifold $M \subseteq \mathbb{R}^p$, and construct L_{sym} as before. Then as the number of samples $N \to \infty$ and the bandwidth $\varepsilon \to 0$, the eigenvectors of L_{sym} approach the Neumann eigenfunctions of Δ_M.*

We can also define the random walk matrix $P = D^{-1}W$, whose eigenvalues are related to those of the Laplace-Beltrami operator by the following result:

Theorem 2.3 (Coifman & Lafon, 2006)**.** *Let P be defined as above and denote its eigenvalues by μ_k. Then denoting by λ_k the eigenvalues of Δ_M, we have*

$$\mu_k \to \exp\left(-\frac{\varepsilon}{4}\lambda_k\right) \tag{2}$$

as the number of samples $N \to \infty$ and the bandwidth $\varepsilon \to 0$.

We can also symmetrize P by defining a matrix $P_{\text{sym}} = D^{1/2} P D^{-1/2} = D^{-1/2} W D^{-1/2}$, which is similar to P (and hence has the same eigenvalues) and has the same eigenvectors as L_{sym}, allowing us to leverage both Theorems 2.2 and 2.3.

This approximation of Δ_M and its eigenfunctions then allows us to apply the following well-known theorem to obtain an embedding of X which approximates M:

Theorem 2.4. *Denote by f_j the Neumann eigenfunctions of Δ_M and λ_j their corresponding eigenvalues. Then the eigenvalues λ_j satisfy $0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \to \infty$, and The eigenfunctions $\{f_j\}_{j=1}^{\infty}$ form a complete orthonormal basis of $L^2(M)$.*

Thus M can be embedded in $L^2(M)$ by its eigenfunctions, and we can in turn approximate this embedding with the eigenvectors of L_{sym}.

2.2. Product Manifold Learning

To formalize the notion of an underlying data manifold having multiple constituent features, we use the notion of a *Riemannian product manifold*, focusing on the case of a product manifold with two factors. We also discuss the behavior of the Laplace-Beltrami operator on products, justifying the algorithm for decomposing a product manifold into its factors given by (Zhang et al., 2020).

Definition 2.5. The *product* of two Riemannian manifolds (M_1, g_1) and (M_2, g_2) is the Cartesian product $M_1 \times M_2$ endowed with the metric $g_1 \oplus g_2$.

The method is based on Proposition 2.6 (as well as its converse), which states that the eigenfunctions of a Riemannian product manifold can be written as the products of the eigenfunctions on the factor manifolds; for a proof see, e.g. Section 4.6 in (Canzani, 2013).

Proposition 2.6 (Zhang et al., 2020)**.** *Let $f_1 : M_1 \to \mathbb{R}$ and $f_2 : M_2 \to \mathbb{R}$ be twice-differentiable functions such that $\Delta_{M_1} f_1 = \lambda_1 f_1$ and $\Delta_{M_2} f_2 = \lambda_2 f_2$. Take $\pi_i : M \to M_i$ to be the projection of M onto M_i for $i = 1, 2$, and define $g_i = f_i \circ \pi_i$. Then $\Delta_M(g_1 g_2) = (\lambda_1 + \lambda_2) g_1 g_2$.*

Example 2.7. Consider a strip $M = M_1 \times M_2 \subseteq \mathbb{R}^2$ where $M_1 = [0, a]$ and $M_2 = [0, b]$. Then denoting by x the M_1 coordinate and y the M_2 coordinate, the Neumann eigenfunctions of the Laplacian Δ_M are

$$f_{k,\ell}(x, y) = \frac{1}{2} \cos\left(\frac{\pi k}{a} x\right) \cos\left(\frac{\pi \ell}{b} y\right) \tag{3}$$

with corresponding eigenvalues

$$\lambda_{k,\ell} = \pi^2 \left(\frac{k^2}{a^2} + \frac{\ell^2}{b^2}\right). \tag{4}$$

The eigenfunctions are the products of the eigenfunctions $\cos\left(\frac{\pi k}{a} x\right)$ on $M_1 = [0, a]$ and $\cos\left(\frac{\pi \ell}{b} y\right)$ on $M_2 = [0, b]$.

The eigenvalues of M are the sums of the eigenvalues $\lambda_k = \frac{\pi^2 k^2}{a^2}$ on M_1 and $\lambda_\ell = \frac{\pi^2 \ell^2}{b^2}$ on M_2.

Here f_i may be the constant eigenfunction, so successfully factoring the eigenfunctions on M in terms of other eigenfunctions also gives a factorization $M_1 \times M_2$. In the context of data sampled from M, the convergence of the constructed graph Laplacian to the Laplace-Beltrami operator means that we should also be able to find eigenvectors $\varphi_i, \varphi_j, \varphi_k$ satisfying $\varphi_k \approx \varphi_i \varphi_j$ (where $\varphi_i \varphi_j$ is the elementwise product). If one of φ_i or φ_j is the trivial constant eigenvector, then φ_k is a factor; otherwise, φ_k is a product eigenvector. Thus (Zhang et al., 2020) aims first to identify the product and factor eigenvectors $\varphi_k \approx \varphi_i \varphi_j$, and then to assign φ_i to one of M_1 or M_2 and φ_j to the other.

To identify triples $\varphi_k \approx \varphi_i \varphi_j$ of nontrivial eigenvectors, (Zhang et al., 2020) identify for each eigenvector φ_k which eigenvectors φ_i, φ_j best match φ_k as defined by their absolute cosine similarity

$$\mathrm{Sim}(\varphi_k, \varphi_i \varphi_j) = \frac{|\langle \varphi_k, \varphi_i \varphi_j \rangle|}{\|\varphi_k\| \, \|\varphi_i \varphi_j\|}. \tag{5}$$

To avoid considering too many pairs φ_i, φ_j, the method employs two hyperparameters: an *eigenvalue criterion* δ and a *similarity criterion* γ. Following Theorem 2.3, we take the eigenvalues μ_k of P_{sym} and compute $\lambda_k = -4 \log(\mu_k)/\varepsilon^2$. Then following Proposition 2.6, we only compute the similarity for triples (i, j, k) for which the corresponding eigenvalues $\lambda_i, \lambda_j, \lambda_k$ satisfy $|\lambda_i + \lambda_j - \lambda_k| < \delta$, and keep the triples with $\mathrm{Sim}(\varphi_k, \varphi_i \varphi_j) > \gamma$.

Once the triplets are identified, the factor eigenvectors are assigned to the factor manifolds by constructing a *separability matrix* $C_{i,j} = \mathrm{Sim}(\varphi_k, \varphi_i \varphi_j)$, symmetrizing by taking $C + C^\intercal$, and using the MAX-CUT approximation given by (Goemans & Williamson, 1995) to assign each φ_i to one of M_1 or M_2 and φ_j to the other.

2.3. Independent Coordinate Selection

Chen & Meilă (2019) proposed to overcome the problem of redundant eigenvectors and select *independent* coordinates for embedding by examining the volume explained by a set of embedding coordinates. The coordinate selection is based on a criterion called the *(regularized) rank quality*, which we will describe below.

Let $\varphi = (\varphi_1, \ldots, \varphi_p) : M \to \mathbb{R}^p$ be an embedding of M, and at each data point x_i let $U(i)$ be a matrix whose columns form a basis for the tangent space $T_{\varphi(x_i)} \varphi(M)$ at x_i. We wish to identify a subset $S \subseteq \{1, \ldots, p\}$ of size $|S| \ll p$ such that $\varphi_S = (\varphi_k)_{k \in S}$ is also a smooth embedding of M. To do so, consider the projection $U_S(i) := U(i)[S, :]$ and denote its columns by $u_k^S(i)$, and with these define the *normalized projected volume* at x_i.

Definition 2.8. Let $U_S(i)$ and $u_k^S(i)$ be defined as above. Then the *normalized projected volume* is the quantity

$$\mathrm{Vol}_{\text{norm}}(S, i) := \frac{\sqrt{\det(U_S(i)^\intercal U_S(i))}}{\prod_{k \in S} \|u_k^S(i)\|}. \tag{6}$$

As described by (Chen & Meilă, 2019), $\mathrm{Vol}_{\text{norm}}(S, i)$ is the volume spanned by a set of unit vectors (not necessarily orthogonal) in $T_{\varphi_S(x_i)} \varphi_S(M)$, and should be 1 when the embedding coordinates ϕ_k indexed by S are orthogonal and 0 when S fails to be full rank. Then the *rank quality* of S is defined to be the average of $\log \mathrm{Vol}_{\text{norm}}(S, i)$ over each data point x_i, using the logarithm to increasingly penalize values near zero. Finally, because S may contain eigenvectors of high frequency, a regularization parameter ζ is introduced to penalize high frequency eigenvectors, resulting in the final criterion for S.

Definition 2.9. Let λ_k be the k-th eigenvalue of the graph Laplacian L_{sym}, and $\mathrm{Vol}_{\text{norm}}(S, i)$ defined as above for each x_i. The *(regularized) rank quality* of S

$$\mathfrak{L}(S; \zeta) = \frac{1}{N} \sum_{i=1}^{N} \log \mathrm{Vol}_{\text{norm}}(S, i) - \zeta \sum_{k \in S} \lambda_k. \tag{7}$$

We require $1 \in S$, so coordinate selection becomes the subset selection problem

$$S_*(\zeta) = \operatorname*{argmax}_{S, 1 \in S} \mathfrak{L}(S; \zeta). \tag{8}$$

We use the greedy approach given in Appendix D of (Chen & Meilă, 2019), which also gives us an order for S.

3. Challenges in Product Manifold Learning

Having introduced the method of manifold factorization in (Zhang et al., 2020), we identify three scenarios that pose difficulties for this approach, and comment both on how these affect Proposition 2.6 in theory and how they affect the performance of the described method in practice. We provide examples of each scenario and compare the performance of (Zhang et al., 2020) with our augmented method in Section 5.

3.1. Incorrect Factorizations

The first potential difficulty for the method given in Section 2.2 is that a factorization $\varphi_k \approx \varphi_i \varphi_j$ may not necessarily reflect the product structure of M, either because one of the factor manifolds itself has a triple $\varphi_k \approx \varphi_i \varphi_j$ or because one of the factors φ_i, φ_j is itself a product of two other eigenvectors. In the first case, the method will erroneously assign φ_i to one factor manifold and φ_j to the other factor manifold, even though all three eigenvectors belong

to the same factor manifold. In the second case, the method will erroneously assign φ_i to one factor manifold and φ_j to the other, even though one or both of φ_i, φ_j belong to neither of the factor manifolds.

To illustrate the first scenario, consider a cylinder $M = S^1 \times [0, 1]$ and notice that S^1 has a triple of eigenvectors

$$\frac{1}{2}\sin(2\theta) = \cos(\theta)\sin(\theta). \tag{9}$$

To avoid erroneously assigning $\cos(\theta)$ to one factor and $\sin(\theta)$ to another factor, one needs to select the eigenvalue criterion δ to be less than 2. However, under some circumstances tightening the eigenvalue criteron may lead to a faliure to find actual triples, for example when the data is relatively sparse or noisy, or in the situations we will discuss in Sections 3.2 and 3.3.

In the second scenario, it may be that a triple $\varphi_k \approx \varphi_i\varphi_j$ contains a factor, say φ_i, which is also a product $\varphi_i \approx \varphi_p\varphi_q$. In this scenario, it does not make sense to assign φ_i to one of the factors M_1 or M_2, so we check the identified triples and remove any which contains a product as one of its factors.

3.2. Warped Product Manifolds

It may be that the underlying manifold M has the structure of a product manifold but fails to be strictly a *Riemannian* product. For example, some latent confounder may "stretch" one factor as a function of the other, or the manifold may be embedded in a way which reflects the product structure topologically, but not geometrically. In such cases, we may still wish to decompose the manifold into its factors, but the assumption that the manifold is a Riemannian product is too restrictive. Instead, we may describe such a product as a *warped* product, as introduced in (Bishop & O'Neill, 1969).

Definition 3.1. Let (M_1, g_1), (M_2, g_2) be two Riemannian manifolds, and $w : M_1 \to (0, \infty)$ a continuously differentiable function. The *warped product* $M = M_1 \times_w M_2$ is the Cartesian product $M_1 \times M_2$ with the metric $g_1 \oplus w^2 g_2$.

Under these conditions, Proposition 2.6 fails, and instead Δ_M behaves according to Proposition 3.2 and Corollary 3.3.

Proposition 3.2 (Bishop & O'Neill, 1969)**.** *Let (M_1, g_1) and (M_2, g_2) be Riemannian manifolds of dimensions n, m, respectively, and let $M = M_1 \times_w M_2$ with warping function w. Then for any $f \in C^2(M, \mathbb{R})$,*

$$\Delta_M f = \Delta_{M_1} f + \frac{m}{w} g_1(\nabla f, \nabla w) + \frac{1}{w^2}\Delta_{M_2} f. \tag{10}$$

Corollary 3.3 (Marrocos & Gomes, 2018)**.** *If $f_1 : M_1 \to \mathbb{R}$ and $f_2 : M_2 \to \mathbb{R}$ are eigenfunctions of M_1 and M_2 with eigenvalues λ_1, λ_2, and $\varphi_i = f_i \circ \pi_i$ for $i = 1, 2$, then*

$$\Delta_M(\varphi_1\varphi_2) = \left(\lambda_1\varphi_1 + \frac{m}{w} g_1(\nabla\varphi_1, \nabla w) + \frac{\lambda_2}{w^2}\varphi_1\right)\varphi_2. \tag{11}$$

Although Proposition 2.6 no longer holds in this setting, (Zhang et al., 2020) demonstrate that their method is robust to noise, suggesting that as long as the warping function w is not too extreme, the factorization described in Section 2.2 may still produce the constituent factor manifolds of M.

This optimism also has a theoretical basis: for eigenvalues μ_i of Δ_{M_2}, define $L^w_{\mu_i}$ by

$$L^w_{\mu_i} f = \Delta_{M_1} f + \frac{m}{w} g_1(\nabla\varphi_1, \nabla w) + \frac{\mu_i}{w^2} f. \tag{12}$$

Then we have the following results from (Marrocos & Gomes, 2018):

Proposition 3.4. *If φ^{μ_i} is an eigenfunction of $L^w_{\mu_i}$ and ψ is an eigenfunction of Δ_{M_2}, $\varphi^{\mu_i}\psi$ is an eigenfunction of Δ_M.*

Corollary 3.5. *The functions $\{\varphi^{\mu_i}_j\}$ form a basis of $L^2(M_1, w^{2m/n} g_1)$, and the functions $\{\varphi^{\mu_i}\psi\}$ form a basis of $L^2(M)$.*

Proposition 3.4 tells us that factoring the eigenfunctions of M can still recover the eigenfunctions ψ of M_2 and associated eigenfunctions φ^{μ_i} of a warped copy $(M_1, w^{2m/n} g_1)$ of M_1, and Corollary 3.5 tells us that these eigenfunctions still provide embeddings as in the unwarped case. In practice, this means that the factorization method of (Zhang et al., 2020) is still applicable, but we may need to loosen the gateway criteria γ and δ to find appropriate triples.

3.3. Large Aspect Ratios

The final difficulty that we will discuss is when one factor in M is much larger than the other, which we describe as the product having a large *aspect ratio*. For example, a strip $[0, 1] \times [0, t]$ has aspect ratio $1 \times t$. In this case, the larger the aspect ratio of the product manifold, the more of the first several eigenvectors (ordered by their corresponding eigenvalues) will be redundant, impacting the quality of the factor coordinates identified by (Zhang et al., 2020).

Example 3.6. Consider an $n \times 1$ strip $[0, n] \times [0, 1]$ for a positive integer n. Then using $a = n$ and $b = 1$ in Example 2.7, the eigenfunctions are

$$f_{k,\ell}(x, y) = \frac{1}{2}\cos\left(\frac{\pi k}{n}x\right)\cos(\pi\ell y) \tag{13}$$

with corresponding eigenvalues

$$\lambda_{k,\ell} = \pi^2\left(\frac{k^2}{n^2} + \ell^2\right). \tag{14}$$

Then the eigenvalues are ordered

$$0 = \lambda_{0,0} \le \lambda_{1,0} \le \lambda_{2,0} \le \cdots \le \lambda_{n,0} = \lambda_{0,1} \le \cdots \tag{15}$$

Because we need to examine more eigenvectors, each of which has higher frequency, the factorization becomes susceptible to sampling noise. This ultimately results in misclassified eigenvectors or a lack of identified triples.

4. Independent Product Manifold Coordinates

To address the potential problems in Section 3, notice that when the underlying manifold M is a product manifold (even a warped product), the coordinates corresponding to each factor should be orthogonal. Thus by selecting independent coordinates as described in Section 2.3, we can reduce instances of factorizations that fail to reflect the product structure of the manifold.

Algorithm 1 IC-PML

Input: Eigenvectors $\Phi = \{\varphi_k\}_{k=1}^N$ of L_{sym}, eigenvalues $\Lambda = \{\lambda_k\}_{k=1}^N$ of P, intrinsic dimension d, regularization parameter ζ, eigenvalue criterion δ, similarity criterion γ

Output: List of triples $\varphi_k \approx \varphi_i\varphi_j$ and similarity scores

1. $S \leftarrow$ GREEDYINDEIGENSEARCH$(\Phi, \Lambda, d, \zeta)$ ▷ Record the order of S
2. `triplets` $\leftarrow \varnothing$
3. **for** $k = 1, \ldots, N$ **do**
4. $\quad \text{Sim}^* \leftarrow \max_{\substack{i,j \in S \\ |\lambda_i + \lambda_j - \lambda_k| < \delta}} \text{Sim}(\varphi_k, \varphi_i\varphi_j)$
5. $\quad i^*, j^* \leftarrow \operatorname{argmax}_{\substack{i,j \in S \\ |\lambda_i + \lambda_j - \lambda_k| < \delta}} \text{Sim}(\varphi_k, \varphi_i\varphi_j)$
6. $\quad$ **if** $\text{Sim}^* > \gamma$ **then**
7. $\quad\quad$ `triplets` $\leftarrow$ `triplets` $\cup \{(\varphi_k, \varphi_{i^*}, \varphi_{j^*})\}$
8. $\quad$ **end if**
9. **end for**
10. **for** $(\varphi_k, \varphi_i, \varphi_j)$ in `triplets` **do**
11. $\quad$ **if** $\exists\,(\varphi_i, \varphi_p, \varphi_q) \in$ `triplets` or $\exists\,(\varphi_j, \varphi_p, \varphi_q) \in$ `triplets` **then**
12. $\quad\quad$ `triplets` $\leftarrow$ `triplets` $\setminus \{(\varphi_k, \varphi_i, \varphi_j)\}$
13. $\quad$ **end if**
14. **end for**

We describe in Algorithm 1 our modification of the algorithm for identifying individual factors (Algorithm 1, Zhang et al., 2020) using the function GREEDYINDEIGENSEARCH (Algorithm S4, Chen & Meilă, 2019). In contrast to (Chen & Meilă, 2019), we allow S to be comparatively large, since we require several eigenvectors to successfully identify triples as in (Zhang et al., 2020). The first key difference between PML and IC-PML occurs in lines 4 and 5 where we select i, j from S to prevent the situation in Section 3.1 where φ_i, φ_j represent the same direction in M, and hence cannot possibly correspond to different factors. Normally, the eigenvalue criterion would suffice to prevent this, but as we have seen in Section 3.2, we may need to loosen the eigenvalue criterion. We also order our factor embedding coordinates by the greedy ordering based on the regularized rank quality discussed in Section 2.3, which ensures that our embeddings use the most relevant coordinates instead of simply the lowest frequency coordinates as in (Zhang et al., 2020), which orders the factor eigenvectors by their eigenvalues. This is especially important with large aspect ratios as in Section 3.3, where a lower frequency eigenvector from the shorter factor may be erroneously assigned to the longer factor. In this case, the greedy ordering will ensure that we consider the correct eigenvector for the long factor first, even if it is of higher frequency. We also add the step in lines 10-14 where we remove any factor eigenvector which is itself a product of nontrivial eigenvectors, addressing the second scenario described in Section 3.1. We then apply MAX-CUT as in (Zhang et al., 2020) to give the final embedding coordinates for M_1 and M_2.

5. Experiments

In this section we provide examples illustrating some of the scenarios in Section 3, using both synthetic and real data. We demonstrate how our algorithm resolves limitations of (Zhang et al., 2020) and identifies factor manifold coordinates which represent the geometry of the data. We will denote by $\varphi_i^{(1)}$ the eigenvectors assigned to factor M_1 and $\varphi_j^{(2)}$ the eigenvectors assigned to factor M_2, with the order given by each method: for PML i, j are ordered by their corresponding eigenvalues, while for IC-PML they are ordered by the greedy coordinate search described in Section 2.3.

Implementation. We implemented our method in Python using `numpy` (Harris et al., 2020), `SciPy` (Virtanen et al., 2020), and `NetworkX` (Hagberg et al., 2008), along with original code from (Zhang et al., 2020; Chen & Meilă, 2019). Visualizations are generated using `Matplotlib` (Hunter, 2007). For image data, we use `scikit-learn` (Pedregosa et al., 2011) and `sklearn-image` (Van der Walt et al., 2014) for preprocessing and feature extraction. Code is available at `https://github.com/he-jesse/ic-pml`.

5.1. Synthetic Examples

We begin with comparisons of PML to IC-PML on synthetic product manifolds with large aspect ratios, as discussed in Section 3.3: an $n \times 1$ strip for $n = 6, 7, 8$ (Figure 1) and a cylinder with radius $r = 2$ and height $h = 1$ (Figure 2). In both examples, we use the factorization $M = M_1 \times M_2$ to reconstruct M by using the first one or two coordinates from M_1 with the first coordinate of M_2. We can see that independent coordinate selection helps avoid the reconstructed manifold being "collapsed" along its shorter factor.

We also give an example of a warped product in Figure 3. We sample 5000 points uniformly at random from a warped cylinder where the radius r increases with the height z by $r(z) = 1 + 2z$. That is, the radius increases from $r = 1$ to $r = 3$ as z increases from 0 to 1. Because the eigenvalue criterion must be loose, as discussed in Section 3.2,

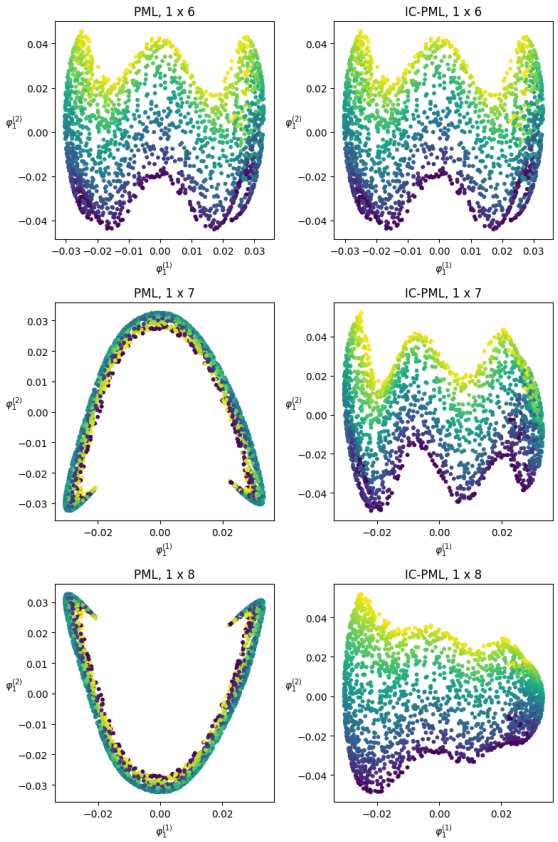


Figure 1. Reconstruction of 1×6, 1×7, and 1×8 rectangles with PML (left) and IC-PML (right). Each point is colored by its ground truth coordinate along the short side of the rectangle. Each example uses 2000 samples with $\delta = 1.0$, $\gamma = 0.6$, and the right uses $\zeta = 0.01$.

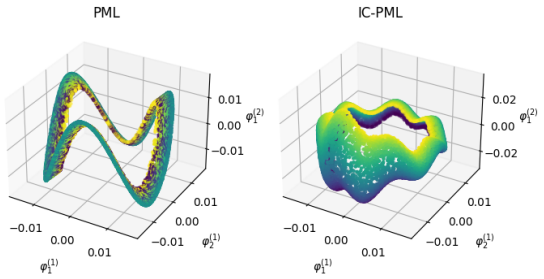


Figure 2. Reconstruction of a cylinder of radius 2 and height 1 with PML (left) and IC-PML (right). using $\delta = 0.2$, $\gamma = 0.6$, and $\zeta = 0.001$. Points are colored by ground truth height.

PML fails to distinguish between products which reflect the manifold geometry and those which are simply results of the S^1 factor as discussed in Section 3.1. Our method, however, successfully produces a circle in one factor, and in the other successfully identifies the dominating eigenvector associated with height.

5.2. Real Data

Rotating Puppets. We use 8100 video frames of two puppets (Yoda and a bulldog) rotating at different rates from (Lederman & Talmon, 2018), as shown in Figure 4. Because the puppets are rotating at different rates, the underlying manifold has the structure of a torus $T = S^1 \times S^1$, where each factor manifold S^1 corresponds to the rotation of a different puppet. The PML factorization does not capture this structure, producing one topologically trivial factor embedding. Our method, however, successfully produces a closed curve when embedding both factor manifolds.

A Partially Plowed Field. We also illustrate the use of factorization on a warped product structure, as discussed in Section 3.2 using an image of a partially plowed field in Figure 5. We embed the pixels of a downsampled segment of the image, where each pixel is represented by its 5×5 surrounding patch flattened to a 25-dimensional vector for the graph construction.

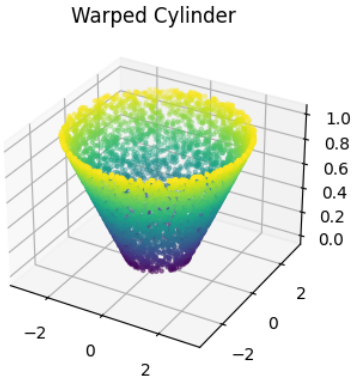
We can see that the underlying manifold is another warped cylinder $[a, b] \times_w S^1$ with $w(t) = r_0 + \varepsilon t$ for some constant r_0. In this case, the periodicity of the furroughs gives rise the the S^1 factor, while the average intensity gives rise to the interval factor. The warping w reflects the increased contrast as the average intensity increases. In this example, both algorithms give the same result, but the eigenvalue criterion must be loosened to $\delta > 6.14$.

6. Conclusions, Limitations, and Future Work

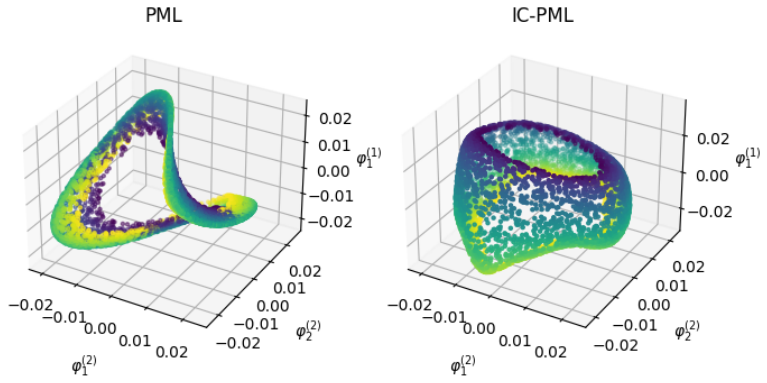
In this paper, we identify three potential difficulties in learning product manifold factorizations: incorrect factorizations, warped products, and large aspect ratios. To address these difficulties, we propose an algorithm which augments a product manifold learning approach with independent coordinate selection and show that this augmented method has the potential to produce correct and higher quality factorizations. In future work, we will optimize our framework in order to provide an implementation which is more efficient in runtime. An important application of product manifold learning is that factorizing the structure of a product manifold into separate factor manifolds lends itself to downstream analysis, such as regression or decoding of observed signals from the factor manifolds. We leave this to future work.

Acknowledgements

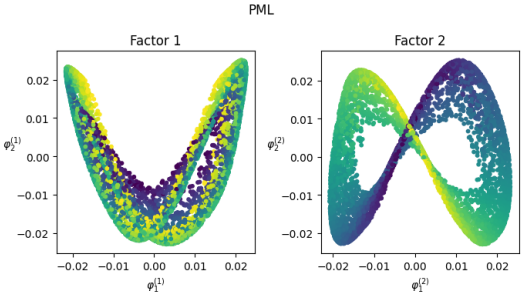
This work is partially supported by NSF-CCF-2217033.



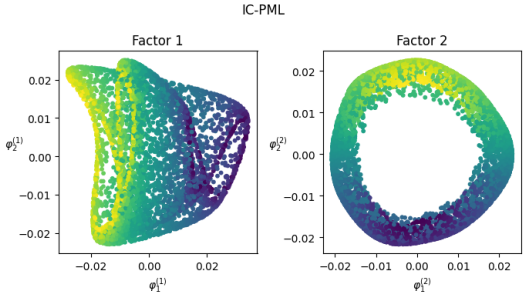
(a) 5000 samples from a warped cylinder colored by height.



(b) Reconstructions of the warped cylinder in (a) using the PML factorization (left) and the IC-PML factorization (right).



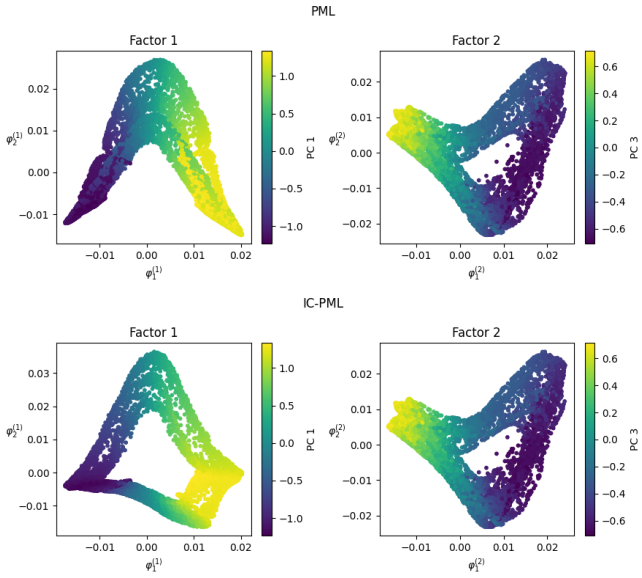
(c) Factors of the warped cylinder in (a) computed by PML. Factor 1 (left) is colored by ground truth height z, while factor 2 (right) is colored by x coordinate.



(d) Factors of the warped cylinder in (a) computed by IC-PML. Factor 1 (left) is colored by ground truth height z, while factor 2 (right) is colored by x coordinate.

Figure 3. A warped cylinder $[0, 1] \times_r S^1$ with $r(z) = 1 + 2z$ (a), as well as its reconstructions (b) and factorizations (c,d) using $\varepsilon = 0.5$, $\delta = 2.0$, $\gamma = 0.6$, and $\zeta = 0.02$.

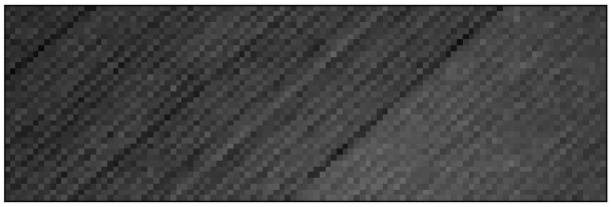
(a) Example of video frames of rotating puppets from (Lederman & Talmon, 2018).



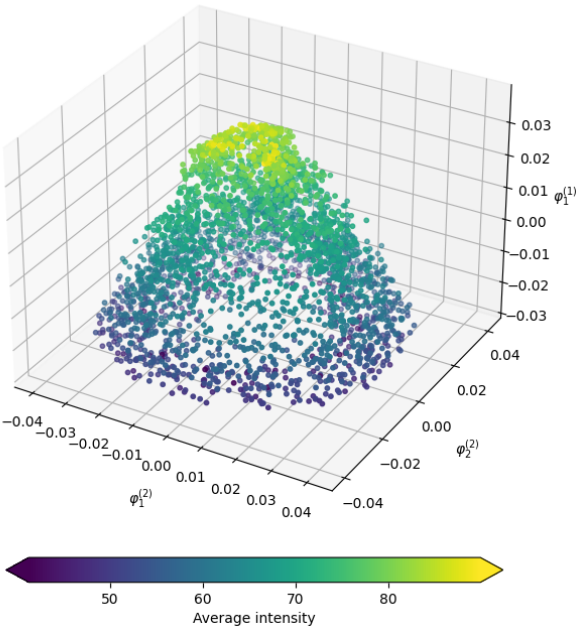
(b) Factorizations of 8100 pictures of rotating figures using PML (top) and IC-PML (bottom) with $\delta = 1.0$, $\gamma = 0.6$, and $\zeta = 0.01$ colored by the first (left) and third (right) principal components.

Figure 4. Video frames of rotating puppets (a) and representations of the factor manifolds (b).

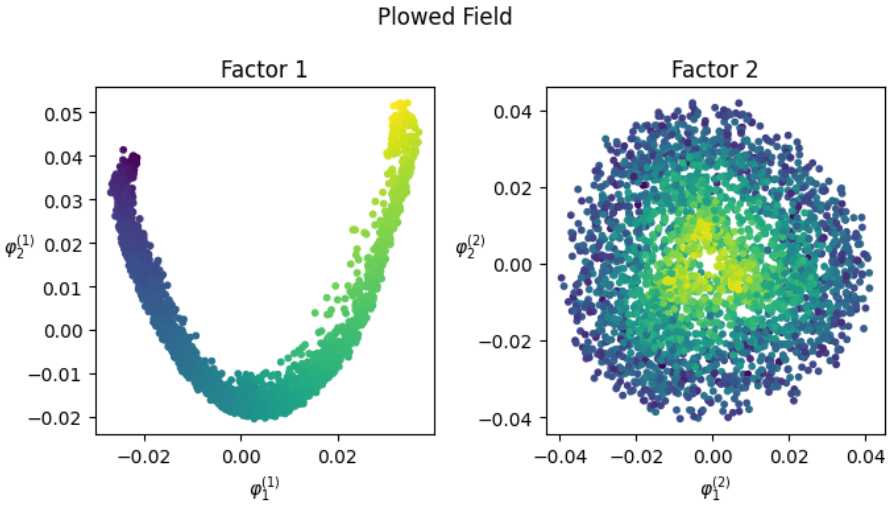
(a) A partially plowed field



(b) A downsampled portion of (a) converted to black and white.



(c) Spectral embedding of 5×5 pixel patches in (b).



(d) Factors of (c). Factor 1 corresponds to height, while Factor 2 is an annulus corresponding to the warped radius along the height to form the truncated cone.

Figure 5. Image of a partially plowed field (a) and spectral embedding of image patches (c) with factorizations (d) computed using a bandwidth of $\varepsilon = 0.1$ and parameters $\delta = 7, \gamma = 0.6$. Points are colored by mean patch intensity.

References

Belkin, M. and Niyogi, P. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6):1373–1396, 06 2003. ISSN 0899-7667. doi: 10.1162/089976603321780317. URL https://doi.org/10.1162/089976603321780317.

Bishop, R. L. and O'Neill, B. Manifolds of negative curvature. *Transactions of the American Mathematical Society*, 145:1–49, 1969. ISSN 00029947. URL http://www.jstor.org/stable/1995057.

Canzani, Y. Analysis on manifolds via the Laplacian (lecture notes), 2013. URL https://canzani.web.unc.edu/wp-content/uploads/sites/12623/2016/08/Laplacian.pdf.

Chen, Y.-C. and Meilă, M. Selecting the independent coordinates of manifolds with large aspect ratios, 2019. URL https://arxiv.org/abs/1907.01651.

Coifman, R. R. and Lafon, S. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2006.04.006. URL https://www.sciencedirect.com/science/article/pii/S1063520306000546. Special Issue: Diffusion Maps and Wavelets.

Dsilva, C. J., Talmon, R., Coifman, R. R., and Kevrekidis, I. G. Parsimonious representation of nonlinear dynamical systems through manifold learning: A chemotaxis case study, 2015.

Fefferman, C., Mitter, S., and Narayanan, H. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29, 2016.

Gardner, R. J., Hermansen, E., Pachitariu, M., Burak, Y., Baas, N. A., Dunn, B. A., Moser, M.-B., and Moser, E. I. Toroidal topology of population activity in grid cells. *Nature*, 602(7895):123–128, Feb 2022. ISSN 1476-4687. doi: 10.1038/s41586-021-04268-7. URL https://doi.org/10.1038/s41586-021-04268-7.

Goemans, M. X. and Williamson, D. P. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, nov 1995. ISSN 0004-5411. doi: 10.1145/227683.227684. URL https://doi.org/10.1145/227683.227684.

Goldberg, Y., Zakai, A., Kushnir, D., and Ritov, Y. Manifold learning: The price of normalization, 2008.

Hagberg, A. A., Schult, D. A., and Swart, P. J. Exploring network structure, dynamics, and function using NetworkX. In Varoquaux, G., Vaught, T., and Millman, J. (eds.), *Proceedings of the 7th Python in Science Conference*, pp. 11 – 15, Pasadena, CA USA, 2008.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

Koelle, S., Zhang, H., Meila, M., and Chen, Y.-C. Manifold coordinates with physical meaning, 2021.

Kohli, D., Cloninger, A., and Mishne, G. LDLE: Low distortion local eigenmaps. *J. Mach. Learn. Res.*, 22(1), jan 2021. ISSN 1532-4435.

Lederman, R. R. and Talmon, R. Learning the geometry of common latent variables using alternating-diffusion. *Applied and Computational Harmonic Analysis*, 44(3):509–536, 2018. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2015.09.002. URL https://www.sciencedirect.com/science/article/pii/S1063520315001190.

Marrocos, M. A. M. and Gomes, J. N. V. Generic spectrum of warped products and G-manifolds. *The Journal of Geometric Analysis*, 29(4):3124–3134, 10 2018. doi: 10.1007/s12220-018-00106-x. URL https://doi.org/10.1007%2Fs12220-018-00106-x.

Nakane, T., Kimanius, D., Lindahl, E., and Scheres, S. H. Characterisation of molecular motions in cryo-EM single-particle data by multi-body refinement in RELION. *eLife*, 7:e36861, jun 2018. ISSN 2050-084X. doi: 10.7554/eLife.36861. URL https://doi.org/10.7554/eLife.36861.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Singer, A. From graph to manifold Laplacian: The convergence rate. *Applied and Computational Harmonic Analysis*, 21(1):128–134, 2006a. ISSN 1063-5203.

doi: https://doi.org/10.1016/j.acha.2006.03.004. URL https://www.sciencedirect.com/science/article/pii/S1063520306000510. Special Issue: Diffusion Maps and Wavelets.

Singer, A. Spectral independent component analysis. *Applied and Computational Harmonic Analysis*, 21(1):135–144, 2006b. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2006.03.003. URL https://www.sciencedirect.com/science/article/pii/S1063520306000509. Special Issue: Diffusion Maps and Wavelets.

Tenenbaum, J. B., de Silva, V., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500): 2319–2323, 2000. doi: 10.1126/science.290.5500.2319. URL https://www.science.org/doi/abs/10.1126/science.290.5500.2319.

Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., and Yu, T. scikit-image: image processing in Python. *PeerJ*, 2:e453, 2014.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Zhang, S., Moscovich, A., and Singer, A. Product manifold learning, 2020. URL https://arxiv.org/abs/2010.09908.